

## APPENDIX

### Test 1 : Random state = 0, Balanced dataset

```
# -*- coding: utf-8 -*-
"""Diabetes Prediction Using Decision Tree and XGBoost Algorithm.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/181ps0JTGijd4jukqk5yeLP_cWeZcACO
-

#Importing Library and Preparing the dataset
"""

# Commented out IPython magic to ensure Python compatibility.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(color_codes=True)
from sklearn.model_selection import train_test_split
# %matplotlib inline

diabetes_df = pd.read_csv('diabetes.csv')
diabetes_df

"""#Exploratory Data Analysis"""

#Column in Dataset
diabetes_df.columns

#Show data types and null value each column
diabetes_df.info()
```

```

#Menampilkan 10 baris pertama
diabetes_df.isnull().head(10)

#Checking if there is null value
diabetes_df.isnull().sum()

#Checking if there is zero value

#replace 0 value with NaN
diabetes_df_copy = diabetes_df.copy(deep = True) #deep = True -> Buat
salinan indeks dan data dalam dataframe
diabetes_df_copy[['Glucose','BloodPressure','SkinThickness','Insulin','BMI'
]]
diabetes_df_copy[['Glucose','BloodPressure','SkinThickness','Insulin','B
MI']].replace(0,np.NaN)

# Showing the Count of NANs
print(diabetes_df_copy.isnull().sum())

"""#Berdasarkan Jurnal A model for early prediction of diabetes
oleh Talha Mahboob Alam et al, yang bersumber dari :
https://reader.elsevier.com/reader/sd/pii/S2352914819300176?token=7D4F21
A1900E1B47FA971E110A1469CE2A8882E14CFBF3A4C3790507841542704D1370C180ADD4
341ED02B627B619B3F&originRegion=eu-west-1&originCreation=20220928035824

>Data cleaning consists of filling the missing values and removing
noisy data. Noisy data contains outliers which are removed to resolve
inconsistencies . In our dataset, glucose, blood Pressure, skin
thickness, insulin, and BMI have some zero values. **Thus, all the zero
values were replaced with the median value of that attribute.**"

Jadi di tahap ini yang saya lakukan adalah mengisi value 0 (missing values)
pada beberapa kolom tersebut dengan nilai median untuk mengatasi nilai
inconsistent

"""

#Fill null value with median
diabetes_df_copy['Glucose'].fillna(diabetes_df_copy['Glucose'].median(),
inplace = True)

```

```

diabetes_df_copy['BloodPressure'].fillna(diabetes_df_copy['BloodPressure'].
    median(), inplace = True)
diabetes_df_copy['SkinThickness'].fillna(diabetes_df_copy['SkinThickness'].
    median(), inplace = True)
diabetes_df_copy['Insulin'].fillna(diabetes_df_copy['Insulin'].median(),
    inplace = True)
diabetes_df_copy['BMI'].fillna(diabetes_df_copy['BMI'].median(), inplace =
    True)
#inplace = True -> untuk menyimpan hasil modifikasi pada dataframe
    diabetes_df_copy

"""#Berdasarkan Jurnal Deteksi Dini Penyakit Diabetes Menggunakan Machine
    Learning dengan Algoritma Logistic Regression

yang          dikutip          dari          :
    https://jurnal.ugm.ac.id/v3/JNTETI/article/view/3586/1646

"Pada tahap ini juga dilakukan pengecekan terhadap data yang tidak
    seimbang. Penanganan terhadap data yang tidak seimbang dilakukan
    menggunakan synthetic minority over-sampling technique (SMOTE)"

Pada tahap ini yang akan saya lakukan adalah oversampling minority value,
    yakni value 1 pada variabel outcome yang berfungsi untuk menyeimbangkan
    data yang nantinya akan digunakan untuk prediksi

#Berdasarkan Jurnal SMOTE: Synthetic Minority Over-sampling Technique
yang          dikutip          dari          :
    https://www.jair.org/index.php/jair/article/view/10302/24590

"However, the nature of the application requires a fairly high rate of
    correct detection in the minority class and allows for a small error
    rate in the majority class inorder to achieve this."

Alasan saya menggunakan teknik Oversampling adalah untuk meningkatkan
    sampel kelas minoritas dan untuk memperkecil tingkat kesalahan di kelas
    mayoritas

"""

#Counting 1 and 0 Value in Outcome column

```

```

sns.countplot(diabetes_df_copy['Outcome']) #membuat bar plot perbandingan
jumlah value
print(diabetes_df_copy.Outcome.value_counts()) #menampilkan jumlah value 0
dan 1

from sklearn.utils import resample
#create two different dataframe of majority and minority class
df_majority = diabetes_df_copy[(diabetes_df_copy['Outcome']==0)] # semua
data yang value outcome nya = 0
df_minority = diabetes_df_copy[(diabetes_df_copy['Outcome']==1)] # semua
data yang value outcome nya = 1
# upsample minority class
df_minority_upsampled = resample(df_minority,
                                n_samples= 500, # to match majority class,
                                menyamakan jumlah value 1 dengan 0
                                random_state=42) # reproducible results,
                                random state 0 is better than 42
                                #Random state =
Mengontrol pengacakan yang diterapkan ke data agar hasil yang didapatkan
tetap sama
# Combine majority class with upsampled minority class
diabetes_df_copy2 = pd.concat([df_minority_upsampled, df_majority])
#menggabungkan Outcome 1 (minority) yang sudah di upsample dengan
Outcome 0 (majority)

#Counting 1 and 0 Value in Outcome column
sns.countplot(diabetes_df_copy2['Outcome'])
print(diabetes_df_copy2.Outcome.value_counts())

"""#Berdasarkan Jurnal Prediksi Risiko Penyakit Diabetes menggunakan
Algoritma Regresi Logistik
oleh Qatrunnada Refa Cahyani,Mochammad JanuarFinandi, Jathu Rianti, Devi
Lestari Arianti, Arya Dwi Pratama Putra yang dapat diakses melalui
https://journal.literasisains.id/index.php/jomlai/article/view/598/470 ,
kita melakukan tahap preprocessing yaitu pengecekan outliers pada
variabel. Outliers dihapus dengan menggunakan Z-Score. Beberapa data
dihapus karena nilainya jauh dari Z-Score. Maka jumlah data yang awalnya
1000 menjadi 941

#Berdasarkan Detection of Spatial Outlier by Using Improved Z-Score Test
yang dapat diakses melalui : https://booksc.org/book/77788899/9f8c40 , "The
zscore test has long been used to detect outliers in data."

```

Fungsi Tes Z-Score adalah untuk mendeteksi apakah terdapat outlier dalam data.

```
#Alasan Menggunakan Z-Scores
```

```
##Berdasarkan Outlier Detection in Multivariate Time Series Data Using a Fusion of K-Medoid, Standardized Euclidean Distance and Z-Score
```

```
yang dapat diakses melalui :  
https://link.springer.com/chapter/10.1007/978-3-030-69143-1\_21
```

```
Z-score technique produced a better outlier detection result of 0.9978 F-measure as compared to inter-quartile of 0.8571 F-measure.
```

Alasan saya menggunakan Z Score adalah karena Teknik Z-score menghasilkan hasil deteksi outlier yang lebih baik

```
#Checking Outliers using Box Plot
```

```
"""
```

```
sns.boxplot(x=diabetes_df_copy2["Pregnancies"])
```

```
sns.boxplot(x=diabetes_df_copy2["Glucose"])
```

```
sns.boxplot(x=diabetes_df_copy2["BloodPressure"])
```

```
sns.boxplot(x=diabetes_df_copy2["SkinThickness"])
```

```
sns.boxplot(x=diabetes_df_copy2["Insulin"])
```

```
sns.boxplot(x=diabetes_df_copy2["BMI"])
```

```
sns.boxplot(x=diabetes_df_copy2["DiabetesPedigreeFunction"])
```

```
sns.boxplot(x=diabetes_df_copy2["Age"])
```

```
""">#Removing Outliers using Z Scores
```

```
#Berdasarkan jurnal Performance analysis of machine learning based optimized feature selection approaches for breast cancer diagnosis
```

yang dikutip dari :  
<https://link.springer.com/article/10.1007/s41870-021-00671-5>

"Data preprocessing is a useful step that helps to remove noise, inconsistencies and redundancy to achieve high quality data which improves the performance. During the data pre-processing, all the missing values are filled by the mean of the corresponding feature. Formatting of the given dataset is ensured to be consistent. All the incorrect data types of the features changed to their required datatype. Data normalization is performed on the given dataset to make its range consistent. We used z-score normalization on all the features to restrict the range of values between 3 to - 3"

"Generally, some of machine learning algorithms does not handle categorical data. Diagnosis feature in this dataset contains two categories M(malignant) and B(benign) is replaced numeric with M by 1 and N by 0."

Menurut jurnal tersebut. ambang batas Z-Score yang dapat diterapkan adalah antara -3 dan 3 untuk 2 kategori diagnosis (True or False)

"""

```
import scipy.stats as stats
z = np.abs(stats.zscore(diabetes_df_copy2))
data_clean = diabetes_df_copy2[(z<3).all(axis = 1)] #print all of rows that
    have z<3 (z score below 3)
data_clean.shape

#z<3 = print semua data yang Z-Scorenya kurang dari 3
#.all(axis=1) = melihat di bagian kolom untuk setiap baris (melihat kolom
    z score setelah di proses dengan print z kurang dari 3)
#np.abs = berfungsi untuk bilangan mutlak (mengubah z-score yang minus
    menjadi positif)

#Cleaned Outliers data using Z Scores
data_clean
```

diabetes\_df\_copy2

```
""">#Print baris yang mengandung outlier""
```

```
#Print rows dalam dataframe diabetes_df_copy2 yang not isin(tidak didalam) dataframe data_clean
```

```
#lambang (~) menandakan NOT
```

```
diabetes_df_copy2[~diabetes_df_copy2.index.isin(data_clean.index)]
```

```
""">#Data Correlation
```

```
Berdasarkan Diabetes Prediction Based on XGBoost Algorithm oleh Mingqi Li et al 2020 bersumber dari https://iopscience.iop.org/article/10.1088/1757-899X/768/7/072093/pdf
```

Setelah data preprocessing, kita harus mempelajari apakah data tersebut diproses dengan baik atau tidak dan bagaimana caranya

banyak korelasi antara data tersebut. Koefisien korelasi dapat digunakan untuk mencerminkan kedekatan

hubungan antar variabel. Koefisien korelasi dihitung dengan metode selisih.

Hal ini juga didasarkan pada dispersi kedua variabel dan rata-ratanya masing-masing. keduanya

perbedaan dikalikan untuk mencerminkan tingkat korelasi antara dua variabel

```
#Berdasarkan Jurnal Prediksi Risiko Penyakit Diabetes menggunakan Algoritma Regresi Logistik
```

Jika nilai korelasi  $> 0$  maka terdapat korelasi positif. Sementara nilai satu variabel meningkat, nilai variabel lainnya juga meningkat. Jika persamaan korelasi  $= 0$  maka tidak ada korelasi. Jika korelasi  $< 0$  maka ada korelasi negatif. Sementara satu variabel meningkat, variabel lainnya menurun. Ketika korelasi diperiksa, ada 2 variabel yang bertindak sebagai korelasi positif terhadap variabel dependen outcome, variabel tersebut adalah glucose. Seiring peningkatan ini, variabel dependen juga meningkat. Dengan demikian, semua variabel digunakan untuk modelling karena korelasinya berdekatan.

```
#Berdasarkan Complex heatmaps reveal patterns and correlations in multidimensional genomic data
```

```
Yang dapat dikutip dari : https://academic.oup.com/bioinformatics/article/32/18/2847/1743594
```

Heatmaps are a fundamental visualization method that is broadly used to unravel patterns hidden in genomic data. They are especially popular for gene expression analysis (Eisen et al., 1998) and methylation profiling (Sturm et al., 2012). With the increasing availability of genomic



datasets, \*\*visualization methods that effectively show relations within multidimensional data are urgently needed.\*\*

Alasan saya menggunakan heatmap adalah karena heatmap correlation dapat digunakan untuk menemukan hubungan potensial antara variabel dan untuk memahami kekuatan hubungan ini dan mendeteksi hubungan linier dan nonlinier

#Fungsi Heatmap Correlation Matrix

Berdasarkan Jurnal Implementasi Seleksi Fitur Klasifikasi Waktu

Kelulusan Mahasiswa Menggunakan Correlation

Matrix With Heatmap

Yang dikutip dari :  
<https://ejurnal.ung.ac.id/index.php/jjee/article/view/14403/4663#>

"atribut yang memiliki warna merah muda adalah atribut yang relevan dengan atribut aoutput dan sebaliknya atribut yang berwarna merah tua adalah atribut yang kurang relevan dengan atribut aoutput."

Atribut berwarna merah tua yang dimaksud oleh penulis adalah atribut yang memiliki korelasi negatif tinggi dengan kolom output

"dari 13 atribut awal yang digunakan terseleksi menjadi 9 atribut yang memiliki relevansi atau berkontribusi terhadap hasil output yaitu : jenis kelamin, kelas, umur, SKS1, IPS1, SKS2, SKS3, SKS4 dan SKS5, sedangkan atribut input yang kurang relevan terdapat 4 atribut yaitu : IPS2, IPS3, IPS4 dan IPS5"

#Alasan menggunakan Correlation Matrix Heatmap pada penelitian ini

Untuk melihat relevansi antar kolom. Disini target kolom saya adalah kolom "Output", dan relevansi antar kolomnya memiliki nilai positif. Maka saya akan menggunakan semua kolom sebagai indikator

"""

```
sns.heatmap(data_clean.corr(), annot=True)
```

```
#.corr() = correlation matrix
```

```
#annot=True = memberikan value korelasi antar kolom dalam bentuk angka
```

```
""">#Machine Learning Model Building
```





gggSp5jjBIFUxi6AO3R+kkjTBBALSzXGCQapiFkEduj9IJWmCCRKkmuMEg1TFLII6dH+QStIEEYRINccJBqmKWQR16P4glaQJJKiQao4TDFIVswjq0P1BKkkTTJAg1RwnGKQqZHUofuDVJImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHOCyJCqmEVQh+4PUkmaYIIIEqeY4wSBVMYugDt0fpJI0wQQJUS1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyConR/kErSBBMkSDXHCQapilkEde+j+IJWkCSZIkGqOEwxSfBmI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnMOMEGVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHXo/iCVpAkmsJBqjhMMUHwzCOorQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALSzXGCQapiFkEduj9IJWmCCRKkmuMEg1TFLII6dH+QStIEEYRINccJBqmKWQR16P4glaQJJKiQao4TDFIVswjq0P1BKkkTTJAg1RwnGKQqZHUofuDVJImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHOCyJCqmEVQh+4PUkmaYIIIEqeY4wSBVMYugDt0fpJI0wQQJUS1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyConR/kErSBBMkSDXHCQapilkEde+j+IJWkCSZIkGqOEwxSfBmI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnMOMEGVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHXo/iCVpAkmsJBqjhMMUHwzCOorQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALSzXGCQapiFkEduj9IJWmCCRKkmuMEg1TFLII6dH+QStIEEYRINccJBqmKWQR16P4glaQJJKiQao4TDFIVswjq0P1BKkkTTJAg1RwnGKQqZHUofuDVJImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHOCyJCqmEVQh+4PUkmaYIIIEqeY4wSBVMYugDt0fpJI0wQQJUS1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyConR/kErSBBMkSDXHCQapilkEde+j+IJWkCSZIkGqOEwxSfBmI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnMOMEGVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHXo/iCVpAkmsJBqjhMMUHwzCOorQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALSzXGCQapiFkEduj9IJWmCCRKkmuMEg1TFLII6dH+QStIEEYRINccJBqmKWQR16P4glaQJJKiQao4TDFIVswjq0P1BKkkTTJAg1RwnGKQqZHUofuDVJImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHOCyJCqmEVQh+4PUkmaYIIIEqeY4wSBVMYugDt0fpJI0wQQJUS1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyConR/kErSBBMkSDXHCQapilkEde+j+IJWkCSZIkGqOEwxSfBmI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnMOMEGVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5Dqev48PfbIXMLD0x8e90TPj/zN3/T4zL3rZvFSBqnmOMEGVTGLoA7dH6S6Hjcz3SjeyiDVHCcYpCpmEdSh+4NUV/fnxfpZ0+Pjb+9pDw+PT8/Pu7vnxX5qdx/Xp7hw0+e4e5ZPP3Gs9AieGGDVHOCyJCqmEVQh+4PUq3lZQP62QL00cPj81kWoZdnxyN8b7cXuo3dIV7fINUCJxikKmYR1KH7g1RL2u1nvFBz488tR075DNa17J7xIgf7/q1j9JnrTD/9YJ//Hceew1SfBmI6tD9Qap1/Xn60Xp291/1+vqqa6367QKXugg/V1uZqpkEdSh+4NUa7v4xeIEXzwnt7ICvNZB+ut++puN+zds/6kMd79V+fUPrf/+GHvcW5CqmEVQh+4PUi1vuJyd97PMnb88oQf/u8sOvN5Bei4vC9qg/n/xR40/9d0vNhj3GKQqZHUofuDVLDgtJyde0HafjKuZTV4xYP0nE6t/9/9qWTzUX+1zrnPIFUxi6AO3R+kuhGD5ey8S9LWE/FDzCa86EF6XieV/y+/Xfz5MX/7/WjuNkhVzCKoQ/cHqW7Hyb8Vfc4PND8/iXM+mhbE6x6k53bCbvbLl1fip1H/9Rx7uN0hVzCKoQ/cHqW7JyW+cnetts897mZ9i1uGVD9Lz+776f7UYPz3c75c6dxykKmYR1KH7g1Q35sQ3zs7xRtb5P93RLeDFD9JL+G43+9V6P0q0c9Q6dx2kKmYR1KH7g1Q350q7mXuZXvHyB+llfFP8v1iSF1jMnMk6ZHUofuDVDfoCrvZ54f0Y8xWFECQXsg3b5v9Vs0fP8x5fgrhzoNUxSyConR/kOomffexDn7revJpLzvPhUo3gRoI0ov55ueSX/mJ4TKLmTNZxyyConR/kOpGXW43+/xI7mXVqIIgVzXvSv8MJX+u94e5+yBVMYugDt0fpLpZ19nN/BhTR6iDIL2kr982++fV7Pjuz1bx3H+QqphFUIfuD1LdsPPvZn6MqU8ohSC9qLN+onmxxcyZrGMWQR26P0h10866m32+c/cyrTFJvin8fynUo8XsnDXPQwSpilkEde+j+INWNO9tu9v1tCT/G1A71EKSDq63zS64mDmTdcwiqEP3B6lu3j12s437dC/TKwoiSC/um7r/Wb0e3+1ZFzNnso5ZBHXo/iDVHfjt3ezz/Z33AqWbQk0E6RV8/bbZT2r2uPLP/OMIjxKkKmYR1KH7g1R34Td3Mz/G1Jeoiic9ht/+RPPCi5kzWccsgjp0f5DqTzvSbrZxN+5l+oC6CNkr+Kbqh6V7vOidvFJ5nCBVMYugDt0fpLobv7Cbfb4LP8bUMUojSK/kFz/RPLqrC5Q+jxSkKmYR1KH7g1R35B93s88XOfcyfUZxBom1/NrbZpfdzJzJOMYR1KH7g1R35cTdbON6tfe3/RhTWyiPIL2ab2r+1A3raDG7SPHzWEGqYhZBHbo/SHVnfrab+XaZTkaFBokVfa7eQydV8nHbXOanEh4sSFXMIqhD9wep7s58N3Mv0wA1EqTX9E3Jf79mXWcxcybrmEVQh+4PUt2h0W62ceMLXZd0m6iSIL2uf3vb7KghLvZCY8XpCpmEdSh+4NUd+nk3ezRt8s0RKEE6ZV9XfFFF/XRWne5DuAbg1TFLII6dH+Q6k6dupsdcs/TdyiVIL22bwr+i/eBr7aYOZN1zCKoQ/cHqe7WD3YzP8bU9yiWIL2+H36iefTXLtkDPGSQqphFUIfuD1LdseFu516mU1AuQbqAH71tdsXFzJmsYxZBHbo/SHXXTt/N/BhTJ6JignQJX79ttrF1HTfIZX864UGDVMUsgjp0f5D



qzp26m/mGmU5ExQTpGr4u908/fhzd/NI/nvCwQapiFkEduj9Idfe+fh/hnbuZTkK9B0kivvl  
J5GORX3kxcybrmEVQh+4PUhVwN9MvolqCdBmnf6J5dMvLf6DPAwepilkEdej+IFUFdzP9Gmo  
lSNdx6iear01xhdrnkYNUxSyConR/kKqEu51+CZUSpCs56W2z6y9mzmQdswjQ0P1Bqg7f/Pr  
NIXczfYk6CdKlfl2a7d82W2AxcybrmEVQh+4PUjUY7GU77mb6AlUSpGv57hPN5yv/7v8rHj1  
IVcwiqEP3B6nu36kfZB640pVKt4AaCdLVTKr+WuXOwwepilkEdej+INWDG75d9sbdTH9BhQT  
pck6v/KvVOo8fpCpmEdSh+4NUd23j6vt49HjaFcvdTJuoJyBd0Ilvm13vw3ueQJCqmEVQh+4  
PUT2xz1em123r1HcT3M20geoI0hWdtJpd8bcqeQZBqmIWQR26P0h1rzbWr/dNy91MP0ZtBOm  
STqjza/7nLjyFIFUxi6A03R+kuk9bH2N+uAS5m+mHqIwgXdQ3b5tdt7x5EkGqYhZBHbo/SHW  
PNI5Hn98acdftj1AXQbqqL8v8yrXNswHsfBmI6tD9Qaq7s3Ep+ssFyN1MP0VBVom6/v622bU  
Lm6cRpCpmEdSh+4NUd23j2/riN2nczTRGTQtpwv62ml3zV8z2eB5BqmIWQR26P0h1V076GPM  
DdzMNURFburLtIr/6YuZM1jGLoA7dH6S6HxvXn1P2KXczjVAPQbq2+Y8s18AZCVIVswjQ0P1  
BqnuXsWCdeulxN9MA1RCkizuu8SVqmeCSpCpmEdSh+4NU9+Ef3xNwN9PJqIUgXdxRha9RyDy  
ZIFUxi6A03R+kugMbe9X4yuNuphNRCUG6tiUXM2eyjlkEdej+INXN29ipfvQrNO5mOg11EKR  
LO6rtBX7FbI+nE6QqZhhUofuDvdFuN3+1+dTdbJkrm66BIgjSlS26mDmTdcwiqEP3B6lu2cY  
q9W/vaLmb6VuUQJAubNXFzJmsYxZBHbo/SHW7Ntaof7/ouJvpGxRAkK7r49vKK30Yz1MKUHW  
zCOrQ/UGqW3W+/wtNG/e8yd2sEy9/kC5r3cXmmaxjFkEduj9IdZN+/WPMj9zN9He8+EG6qoU  
XM2eyjlkEdej+INUNOsvHmB+5m+lveOmdDFeFC3mxcuVZBamKWQR16P4g1c3ZWJrOccFxN9M  
2XvggXdPSi5kzWccsgjp0f5Dqtpz5Y8wP3M20hZc9Sfd01C3LFSrPK0hVzCKoQ/cHqW7Jxq5  
01l+ccTfTz7zoQbqgj4vZWr9itsczC1IVswjQ0P1BqpuX8XbZ+VeiU3ezBS97OhNe8iBdz8f  
aXbFCeWpBqmIWQR26P0h1I7ZWPu8VbW1D25yN2vBCx6kqzkq3CXLk+cWpCpmEdSh+4NUt2B  
rPbrYpebk3cwpNTvwagfpWo6Lds3S5MkFqYpZBHxO/iDV8jY3o4u+BXDqR5ruZhv4rYN0Jcc  
Fu+r7uTy9IFUxi6A03R+kWtvz4+YbVhfegE5fzfxQ8/7xSgfpf15fnp8eG+Fh4fHx6fnP1u  
19udzzyxbkzy/IFUxi6A03R+kWtffP0a89KXm9A80d3zj7L7xMgfp+bswJzUhpdc7W1F269  
v5AfWrUeeYJCqmEVQh+4PUq1p4yf/Axe/lgeNnt5cDm7Y7zIQXous58KPlu5FhmKQapiFkE  
duj9ItZyXn/y/uxat/Z7Z3styTuhHSPpHvMJBeibDnwmOrf3hOk8ySFXMIqhD9wepVvKXj2M  
2XPSdgJ9eHR+e3M7uEK9ukJ7Hvy1mq//SI08zSFXMIqhD9wep1nD6ThYXWnu+/l1jlew++eXZ  
veGWD9Dz+ZTNb/yN1nmiQqphFUIfuD1Jdz58Xz0/TjeyDh8enp+fn3R1xn79k/9Q+/udw+/j  
tmfIIulm8okF6Hj/ezFZ/u2yP5xqkKmYR1KH7g1TX8Y+/PrPt398k+Ndfzt7NTVw19Te8ikF  
6Hj8ryFv5L1B4ukGqYhZBHbo/SHUdbma6VbyKQXom84q8of8wmGccpCpmEdSh+4NUkiaYIEF  
6NpMfYm7s1xp51kGqYhZBHbo/SCVpggkSpGd02n+EcoP/qQnPEhVzCKoQ/cHqSRNMEGC9Ly  
+/lBzt5Td4gfkPP0gVTGLoA7dH6SSNMEECdLz+7P7v+By+F8L7/4/M930/8k8/h1BqmIWQR2  
6P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHxO/iCvPAkmSJBqjhMMUHWzCO  
rQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVC  
H7g9SSZpgggSp5jjBIFUxi6A03R+kkjTBBALszXGCQapiFkEduj9IJWmCCRkkmUEg1TFLII  
6dH+QStIEEYRINccJBqmKWQR16P4glaQJJKiQao4TDFIVswjQ0P1BKkkTTJAg1RwnGKQqZhH  
UofuDvJImmCBBqjLOMEhVzCKoQ/cHqSRNMEGCvHocYJCqmEVQh+4PUkmaYIIEqeY4wSBVMYu  
gdT0fpJI0wQQJUS1xgkGqYhZBHbo/SCVpggkSpJrjBINuXSyConR/kErSBBMkSDXHCQapil  
kEdej+IJWkCSZIkGqOEwXsfBmI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVc  
wIQEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnMOMegVTGLoA7dH6SSNMEECVLCncYJBqmI  
WQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHxO/iCvPAkmSJBqjhMMUHW  
zCOrQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKq  
YRVCH7g9SSZpgggSp5jjBIFUxi6A03R+kkjTBBALszXGCQapiFkEduj9IJWmCCRkkmUEg1T  
FLII6dH+QStIEEYRINccJBqmKWQR16P4glaQJJKiQao4TDFIVswjQ0P1BKkkTTJAg1RwnGKQ  
qZhHUofuDvJImmCBBqjLOMEhVzCKoQ/cHqSRNMEGCvHocYJCqmEVQh+4PUkmaYIIEqeY4wSB  
VMYugDt0fpJI0wQQJUS1xgkGqYhZBHbo/SCVpggkSpJrjBINuXSyConR/kErSBBMkSDXHCQa  
pilkEdej+IJWkCSZIkGqOEwXsfBmI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjB  
IVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnMOMegVTGLoA7dH6SSNMEECVLCncYJ  
BqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHxO/iCvPAkmSJBqjhM  
MUHWzCOrQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5x  
gkKqYRVCH7g9SSZpgggSp5jjBIFUxi6A03R+kkjTBBALszXGCQapiFkEduj9IJWmCCRkkmUE  
Eg1TFLII6dH+QStIEEYRINccJBqmKWQR16P4glaQJJKiQao4TDFIVswjQ0P1BKkkTTJAg1Rw  
nGKQqZhHUofuDvJImmCBBqjLOMEhVzCKoQ/cHqSRNMEGCvHocYJCqmEVQh+4PUkmaYIIEqeY  
4wSBVMYugDt0fpJI0wQQJUS1xgkGqYhZBHbo/SCVpggkSpJrjBINuXSyConR/kErSBBMkSDX  
HCQapilkEdej+IJWkCSZIkGqOEwXsfBmI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao  
5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnMOMegVTGLoA7dH6SSNMEECV  
LCncYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHxO/iCvPAkmSJB

```
qjhMMUhwZCoRQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhd9wepJE0wQYJ
Uc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALSzXGCQapiFkEduj9IJWmCCRK
kmuMEg1TFLII6dH+QStIEEYRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkkTTJA
g1RwnGKQqZhHUofuDVIImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHOcYJCqmEVQh+4PUkmaYII
EqeY4wSBVMYugDt0fpJI0wQQJUS1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyConR/kErSBBM
kSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpg
gQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JImmCCBKnMOMegVTGLoA7dH6SSNME
ECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHXo/iCvPAk
mSJBqjhMMUhwZCoRQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhd9wepJE0
wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALSzXGCQapiFkEduj9IJWm
CCRKkmuMEg1TFLII6dH+QStIEEYRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkk
TTJAg1RwnGKQqZhHUofuDVIImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHOcYJCqmEVQh+4PUkma
aYIIEqeY4wSBVMYugDt0fpJI0wQQJUS1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyConR/kEr
SBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9kqT1MK1VzCKoQ/dLktbDpFYxi6A
O3S9JWg+TWsUsgjp0vyRpPUxqFbMI6tD9kqT1MK1VzCKQJP0Eq4TLhPSr7ChJkqRVuJlJkiS
tws1MkiRpFW5mkiRjQ3AzkyRjWoWbmsRJ0irczCRJklbhZiZJkrQKNzNjKqRVuJlJkiStws1
MkiRpFW5mkiRjQ3AzkyRjWoWbmsRJ0irczCRJklbhZiZJkrQKNzNjKqRVuJlJkiStws1MkiR
pFW5mkiRjQ3AzkyRjWoWbmsRJ0irczCRJklbhZiZJkrQKNzNjKqRVuJlJkiStws1MkiRpFW5
mkiRjQ3AzkyRjWoWbmsRJ0irczCRJklbhZiZJkrQKNzNjKqRVuJlJkiStws1MkiRpFW5mkiR
Jq3AzkyRjWoWbmsRJ0irczCRJklbhZiZJkrQKNzNjKqRVuJlJkiStws1MkiRpDf/73/8BFRW
dG/8VnMAAAAASUVORK5CYII=)
```

```
"""
```

```
X = data_clean.drop('Outcome', axis=1) #menggunakan semua atribut kecuali
class (Outcome)
```

```
y = data_clean['Outcome'] #hanya menggunakan atribut class (Outcome)
```

```
"""#Metode 90 10
```

```
#Berdasarkan Jurnal Analysis and Prediction Of Pima Indian Diabetes Dataset
Using SDKNN Classifier Technique
```

```
yang          dikutip          dari          :
https://iopscience.iop.org/article/10.1088/1757-899X/1070/1/012059/meta
```

```
"Our concept is implemented on Pima Indian Diabetes Dataset (PIDD). The
analysis on Pima Indian Diabetes Dataset (PIDD) is carried out by
splitting dataset in to 90% training data and 10% testing data. We have
found that, in our proposed technique, average classification accuracy
gives result 83.2%, a great improvement as compared to other
conventional technique."
```

```
Untuk tahap train test split, saya akan membagi data train menjadi 90% dan
data test menjadi 10% karena dapat menghasilkan akurasi yang lebih baik
```

```
#Berdasarkan A Novel Diabetes Healthcare Disease Prediction Framework Using
Machine Learning Techniques
```





pELAM4CjbXbSaZAQBEIzKBAEQhmbWIExcAjoLNdZu5AqB1tk5fUAGAJ7A1eSn+kmzRZwYAEI  
VkBGAQhWQGABCFZAYAEIVkBGAQhWQGABCFZAYAEIVkBGAQhWQGABCFZAYAEIVkBGAQhWQGAB  
CFZAYAEIVkBGAQhWQGABCFZAYAEIVkBGAQhWQGABCFZAYAEIVkBGAQhWQGABCFZAYAEIVkBGA  
AQhWQGABCFZAYAEIVkBGAQhWQGABCFZAYAEIVkBGAQhWQGABCFZAYAEIVkBGAQhWQGABCFZA  
YAEIVkBGAQhWQGABCFZAYAEIVkBGAQhWQGABCFZAYAEIVkBGAQhWQGABCFZAY81Wx80u12T8  
az/BiA55LM4E0tJmWYuctGxpmNc+mGk8kiP39ws1/T2+K/2+lFnPcEcOQkM3hTvbOb5XI+vx  
4NBrmkMricL5fLq2F+WBpeFSXFfnPnhYHR5Xcxzc9bLBQc3/DQqP8Rg9C3OewI4cpIZvL1eb3  
h1czO/bISzf//ZH25yaRHcbq7Oho8IQIvJ+NV7sopPUYTHm408CcBzSGZwKL2zmlWHWGd6uu  
8w5WJycjpnseyxvVKz7+d/8iQAx0MygwMaXq07zm7Pv2x1s9m4f37bGV0//gDmYnIxzZMAHB  
PJDA6qd/bjjmw2G1fdZY8/VriYfCkSHQBHSDKDA2selFxls+ow5h06yxaz8UnZ0wbAUZLM4P  
CGV5vZLB3GfHx3WRHnTvqnaSSL0u15f3ekjcvSug5CVg3PsUjt5Yhkjb66NEDjaI+T4tnZ5m  
HW2qLMgXsG8qhWUhdXM1XLGt+xJIjFWH0c0JKWYV98LZdycOuLazLRdS5/vPrUtWogjtr8+j  
KNcVEpFr+5wjxzf390uJF+TWbCysH81gP+9F4r1leFq8H/Vh5xicjckPx11nLpe7Gqt1sV/z  
XkNdVyKS2mzwxiaf4MUOwOvr7sSBSlyZeLn38aBzl/jS8+FLucOq8NPN/s1ZccFGv/kQ+i9o  
blaxC359/XvQez8ZeLP53dQ6blWv7crst/F/N9KAdhK8wv875u+ks3BNEZq48Dkswgit7Zt7  
pXa/c6zWfQnd0UVt1pnWnnUzqFrYhelZhk5YPPf3sG2uj982+eWitHYyWttMfltayKr79+f  
fTzU09CFvv7FMu//OfI5o8h7H6eN8kM4iiHOpi1d/04t1s0+jTnj107+Pndf/ZjgcHqv6HKv  
4NPm92/NX1t3/n6X94EmP18c5JzhBDGupid1dfWu9cjbJ906an3/p3SCf3mN6B3qqAVvx1  
h9vHuSGUSQzvAaXV8Ne40Bzg6QzSr5As4vX359+vFj9xx+OBRj9dEckhkcXBq8bLaaJOog2S  
wNcdE//fnh23x5c1MmxfWEHFhZN9M1KvC+SWZwYLm7rDmm7KGyWRERY/HQrtfL1Yn78Fzjaj  
C7huYoYbPm03cOH2asPlpEMOND2uouWz1ENqsPFO29OgCe6mp9TXA5JMt8uWxW92H9dPnMnZ  
dClp9prFXqgWtRkrWzQ5kvVPz3N0K9JeP0/fTuvfT2p3nE87n/MwLteXg9vi2dP+VvpKZ1  
d2+40cWmnlq5UUj4tH67A4Pe0b15YXIZnB4eTusnr0sA1vn83mf6udjFEteGHNO5DtUda89S  
B6j2WsPt4byQwOo2xxp1tjXt113HAjm2037B/g9ufvqq+gaOs/5sXrIjH6CTauhFtMGkd/6i  
h3B6NjsDb8uicGVWa/it/B00d0NVYf741kBm+tOoO1X41Gcd+mfNGMPmU4G88WD9jw52HJ0k  
vKE2D6Fx+qfoLF7Ht9/vS+xn2971y9ML3ypo6HqfBL52MdJbflDjUpd7br9zX7/XO9ksbbXd  
xRTjuUaymie2614LZCx5BN1YfRy/1xdIi+Yuv5VLeynZTvLJzL8nGiTnb1qfY3K28fWWee1  
Qdcld/MGlnWfPVzTUHo8t8rKZ4bSqrF5VK0jybdU7FmZSr2PdpHvIxCCx/j7Vceq/ds8QKZZ  
158L1U9y6hsqpkD1rY/K47v2Z3rOiu9T9u5VtDuX6V7kUlpMJwid/Ouv5VKAX8hbkFouvd9ugE  
klD48zdyWjwgPDUT77bFC2M1b325fMCMXRTADexM4RzfIQ94vfvv8u+SwCY/URNWQGWbupT2  
acXpRXmaRgtv/Ur5dmrD6Oh2QGwFupu83SZSNvFsyM1ccxkcvAeDvDPMTE9LQcze/po2U8hr  
H6OCaSQGBvaD08y5P7sIzVx7uWrwSgnfIXX8ulAI+RtyC1XPPa+aLix1/JuLoIMqsvklxdZ1  
nYt9Tt11Wv3ChtXG+5ubT1ZZjX6/mbxc3lbJQ/TH51LZfSYipB6+RffY2XAjxG3oLUculDpS  
zzlCEm3uFYfflvtVxKi3WLF7k60A7dbjdPJSOa8ATP25LMxt2LD/N046TWs01mi/PMaHhTi8  
nF9I1Gy4DjI5kB8JbecBgzoEKOZraOnnPg+Z6+JSkhff37bXl1aLgKbTJb9Jk8Krk2yJ1uy  
fjNO7/5Mt5561uyATHSJ9Z62ifAc/3iC3JbNw9bYwcNrrWX9Zkm8wWfWYAvKbh13raIjQ6hV  
gg99Jn1jraZ8Dz2ZK8FH9JtugzAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAw  
CIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMg  
MAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQj  
IDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiE  
iYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjID

```
J7aHvwbkhemEMfwatlxdnv+ZWsXMBv3z287o+vHH8BcTC6mefIVvc1aYB9tD94PyQwi6Z39uC
ObzcZVd9nGuWcPsp8KRLda3ubtca+2h68J5IZxNI8qLnKZtVhzCd0lylm45Oyp+11vc1aYD
9tD94XyQzCGV5tZrN0GPPx3WVFnDvnp07rncnteX//SBuL2WR8sh684+RkPJntnkqzKgbKM6
TnF5NjGtPjoWuB16DtwTu0pGXyF1/LpUSzvhggGV3n8serT12rBuLYksbrGKTxN+qHaeatud
O7GeW55tfVEjfe071r4R1KX/daLn2wzfpdatanjep/Z+UvKl196H/TVi0samxzWJpBHnFmy7
yYKc+Qnp9fXqYVP3QtT5YXV8ultJhK0Dr511/LpcSzhvjgebuAezJTuf/bKV7tFFc7xLSEzf
nKmSSzVktf91oufYT/UcGrpx9Sn+6te0fQ9kiLXcultJijmRBU7+xbvRPZvU7zJaRrCkbfk
9dG37K0Wx6kVc6/3tbvIe/8/SgMvy63q3CU2wME7OrrHWDyx/PG015Nu6f/vn842Y19185dG
C11tvz/uoeg+nU/sh11zxXb3j/W4PXJZ1BUOXeYnViytk9mv8jqz6Wk+LWytyGuVzTA2vr
g3Tj/rffz8IU/CE9UB//b8e/NGZKWyeu42Gx5H24PjJ1BS01CsNH1fHXI56Wz2eK/NDdTne
fwlPIlB/0P1e5zetrvnhtxLL2N3tnZo8fvgE11x3CamP7ajGYpmH16Xg3T9uBISWYQUloac3
R9Nez11gOcvWw2S90ED9N4DymeVfksF8Bz1H1Tq/6r5CWCmbYHx0oyg2jS4GWD1SAZr5bNku
2+iv3KG6+vLl13Lu7DVWTrwdL2Pn6tYtD6imU77em4w0/bgaElmEEvuLumuOKfuq2Wyzr6Jptn
mHwN7ZVRHPyvEHcsH+e3vC4+wc0Vz8/nlbn06fhjybtgdHRzKDQLa6y1ZeIZutToO+Y4GLyU
XnnyodzsarXVRvWOSz1cgDtz9/i2Y82+YRzRTMPn981rn/TdoeHBvJDMLI3WV7Bwp4+WY2im
bFAvvdka4/4vZ5KR//u/6cNjmv0NveJV7OeAl1N1m5RHNfwtm2h4cK8kMQ1jMxuU1Y+ms/1
y0ZSOB9R/fmK93MuWNZtKLN8YFuD0/rW+r1032T8u7QTUPJ23301SnTO/Zge6sBR6gjlHT07
J18szRMjJtD45VvkKF1shffc2Xcjjz9bDk9w8wXsyX56oMykE18nP3aYy1njTXsf1cViv6z1
DK6y3Xlx7nIdI33+t9a+Edyt9zLZc+3boC3Xs15R6rV+Y6V/6g8tT+611p1NCyhm9V2Oqljc
K71/JclWJXciktphK0Tv7113IpB7K/Yb6za7pnd/OQvcM6+63SVUP57Hrxe+4ped0q31AKkP
Vse288eP9aeGeqr3ollz5Hbge8Npjt/jyav4k7fjqR2h55ebVcSoupBK2Tf/21XArwGHkLUs
ulz5Kyz+ODWeGo2x7VIldyKS3Wlf716ka7dLvdPJWoAMATvMKWZDbuXnyYNweMaQXbZLa4Ag
CAwyvH133J0TLgWEImABzcSw9jBkfl0czW0XMOPN8Lb0kWk5P+32/1jSxbxTaZLfrMAHh7s/
FJkU100jj8i8mX886L3ZAJjps+s9bRPgOe771bkmPo5ZXRdSv7y0q2yWzRZwbAmxt+rW9QmY
agaGssg136zFpH+wx4PluS1+IvyRZ9ZgAAUUhMAABRSYGAAFFIZgAAUUhMAABRSYGAAFFIZg
AAUUhMAABRSYGAAFFIZgAAUUhMAABRSYGAAFFIZgAAUUhMAABRSYGAAFFIZgAAUUhMAABRSYG
YAAFFIZgAAUUhMAABRSYGAAFF018tlnqQdutlungIgdTl9JkBAEQhmQEARCZGAQBETZkBAE
ThCgAAnmJ1OZH9CLWgyQwAIAPhMwEAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAop
DMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAK
KQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAA
CikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzA
AAopDMAACikMwAAAGLodP4fSJaSB2q93ToAAAAASUVORK5CYII=)
```

```
"""
```

```
#test size 10% and train size 90%
#Random state = Mengontrol pengacakan yang diterapkan ke data sebelum
menerapkan pemisahan agar hasil yang didapatkan tetap sama
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.1,
    random_state=0) #split test 10% dan train sisa dari test yaitu 90%
```

```
"""#Decision Tree"""
```

```
from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(random_state=0) #Mengontrol keacakan
    estimator agar hasil yang didapatkan selalu tetap
dtree.fit(X_train, y_train)
```



```

from sklearn.metrics import confusion_matrix

"""#Berdasarkan Jurnal Machine learning models for non-invasive glucose
measurement: towards diabetes management in smart healthcare
yang          dikutip          dari          :
https://link.springer.com/article/10.1007/s12553-022-00690-7

**Precision:** Precision is measurement of how many positive predictions
are made for actual positive values.

**Recall/true positive rate/sensitivity:** Recall is measurement of correct
positive predictions from all positive predictions made

**F-1 Score:** F-1 Measure is a combination of precision and recall

#Penjelasan
**Precision:** Terdapat 76% pasien yang **sebenarnya** terkena diabetes
dibandingkan dengan keseluruhan pasien yang **diprediksi** terkena
diabetes.

**Recall : ** Terdapat 87% pasien yang **diprediksi** terkena diabetes
dibandingkan dengan keseluruhan pasien yang **sebenarnya** terkena
diabetes.

**F-1 Score : ** jika F1-Score punya skor yang baik mengindikasikan bahwa
model klasifikasi punya precision dan recall yang baik

#Berdasarkan jurnal The advantages of the Matthews correlation coefficient
(MCC) over F1 score and accuracy in binary classification evaluation

"Checking only F1, one would read a good value (0.66 in
the [0, 1] interval)."

nilai F1-Score bisa diilang bagus jika value F1 berada diantara 0,66 hingga
1

#Alasan menggunakan Precision Recall
yang dikutip dari : Jurnal The relationship between Recall and Precision
https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1097-4571(199
401)45:1%3C12::AID-ASI2%3E3.0.CO;2-L

```

"Recall and Precision and, in particular, Recall-Precision plots, have been used for many years to characterize document retrieval performance."

Alasan saya menggunakan Recall dan precision adalah untuk melihat kinerja model

#Berdasarkan Jurnal Diagnose Diabetic Mellitus Illness Based on IoT Smart Architecture

yang dikutip dari : <https://www.hindawi.com/journals/wcmc/2022/7268571/>

"for the classification considering the evaluative measures like accuracy, precision, recall, and F1-score on PIDD and claimed to have achieved comparatively enhanced results on binary classifications"

Alasan kedua saya menggunakan precision recall adalah karena precision recall lebih cocok digunakan untuk binary classifications

#Berdasarkan buku Educating Engineers for Future Industrial Revolutions halaman 331

yang dikutip dari :  
[https://www.google.co.id/books/edition/Educating\\_Engineers\\_for\\_Future\\_Industry/PQ8jEAAAQBAJ?hl=en&gbpv=1&dq=balanced+data+evaluation+metric&pg=PA331&printsec=frontcover](https://www.google.co.id/books/edition/Educating_Engineers_for_Future_Industry/PQ8jEAAAQBAJ?hl=en&gbpv=1&dq=balanced+data+evaluation+metric&pg=PA331&printsec=frontcover)

"For each algorithm we used an original and balanced data set to improve evaluation metrics as precision and recall."

Untuk Balanced dataset menggunakan precision dan recall berdasarkan buku tersebut

"""

```
y_pred = dtree.predict(X_test)
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score
print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))
```

""#Berdasarkan Jurnal Diabetes Mellitus Prediction using Classification Techniques

Yang dikutip dari :  
[https://www.researchgate.net/profile/Anny-Leema/publication/344072667\\_Diabetes\\_Mellitus\\_Prediction\\_using\\_Classification\\_Techniques/links/5f50f16da6fdcc9879c511e9/Diabetes-Mellitus-Prediction-using-Classification-Techniques.pdf](https://www.researchgate.net/profile/Anny-Leema/publication/344072667_Diabetes_Mellitus_Prediction_using_Classification_Techniques/links/5f50f16da6fdcc9879c511e9/Diabetes-Mellitus-Prediction-using-Classification-Techniques.pdf)

"Confusion matrix is used to visualize the performance of the algorithms which cross tabulates the observed and predicted classes with associated statistics, evaluation metrics like sensitivity, specificity, precision and accuracy are used to evaluate the performance of the method. Factors like True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) are used"

Alasan saya menggunakan confusion matrix sebagai visualisasi data adalah untuk memvisualisasikan performa dari sebuah algoritma klasifikasi

![format confusion matrix.png] (data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAOoAAACNCAIAAA A2B3H7AAAAAXNSR0IArs4c6QAAAArnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7DACdvqG QAABCoSURBVHhe7Z1teetGEOUvhWAIBXmwhGawhWWWDMIgCIxgCSyBMAiH3Hd12sc1pZlJ4i m1JG+9P+ap/lCp+/RRS7PW2v/7u2lOS9u3OTFt3+bEtH2bE/Nu3/81zRkYf124sO+ImiJa0n LavvNoSctp+86jJS2n7TuPlrSctu88TiTp9+/fX15efvnl11E+Koez77dv3zj1HUa/EzJ/8F LsK1+/fh2droF9f/vtN7qN81FJI3wv7DV07Mt1/9dff6nIMJ6fmxX/+eefT09Pis/ILpL+8c cfUUNAWxS+b1+gw14e+PekEe5v3yRrkv719dXOPh27SKq7WdrQ/P777yO6Qdu3gLX02Bfpf/ 31Vxbmy5cvdCDmE2R9P37EK4GYJzkq2b+5M47auXD2EU3kln2FdmIpg4xs1aNHZV+aLCCHjN plQ5H+6fA5xBHCCezLI4SURUS0o5UijqTGfkl1FegJLpccPlmGXLXwXSdf25aF2RIsy2A5ZEE QPulYm2hc9EY1uxOwdzoZf0VOHk4r+nE5Nc0iSnsC+oCVJ1zo19iu4iPSskCoBx9P0j7fOLE C8I5qI70WR0bZY2W6WMvZftC+VUUMrb08DDqa/s80hzgXeC61hLxjGLftaaEHNvftqGRJsFe o2E847oolIK2uIydZff7Eg2F6NrhqX47CviSjmitzIrp8ApxxRAuf077rDLuwi6TSKioQVc KXmi63ozu7L9CTGnqSS18elFmte5EG8F7YfWScYTxuX2LUV71gqUY0EYyxooms7Ruhns1Y4i RVk30FPXk80DOD+ic1JwubRnhu+7KRKEZlirKvvsbFpwUknqyyYBgmjmoi0umVfqwRJ1WhfKl 0PPgof2/2Ap51tDh6hOJZ9+YrAMHicSnsndzrq07cEuqEmvuQoSc+aSXR6Kg/1+PjWwM7NLp JKw2iyiP2H8/RPB2gLNEk0PyfQTWLySbliLQRJ2DhgvVJbw91HtHAG+2IyxmB8WcuaIhorQd GRSu3B01tFi9GUWK0s0mSJDWcf0Sx+aBRYb42YG1kAI6IVAZKiTxSfo9DWNXju//IAHEgN9X SQ12fCeUe0cCD7fj5a0nLavvNoSctp+86jJS2n7TuPlrSctu88WtJy2r7zaEnLafvOoyUtp+ 07j5a0nLbvPFRScn4i+35dXpja60ducBZJ7wsVf2i8O9vaN/6AN/6kEaSC2UIOvrU1Cp/Rvk lD8fT0xExHjw+xFiqunU66xXp9AEYyooVi+wIe0ssfKLL+mThNMARVvLy8HER1USVpQq/ayL JwM0q1Pg7effB62I4kab19hd7qwMF6h8agy0bSIDpn/Bnsi4BktozsmuhMTckLNCQh20Zr9D hJ0q3s+/z8LD+lN/d+mDdIglh6swNJ9OKY4BDtMRfTMBYi/QZIrjV7Xf7jgmHE1OulM1p1+x Oc7scJlvnGIh10yaUXrB6BbCMqBQHJHGWHq7xS2F8xm6oarUJdKlHofQOGh0Ex1Kjboqsw4 VqNBgXby1oFWQe0cKG9uVTr4cqFswWFDNV66KebsKX8j0Q0CSbAodgX+pRlphW1ZOHbsqmol yuoq8lFYEMHM4nMa00kZCiLhvneQSfvrZUIrO1AibiMWMazYVYqvq5AkFkKVRpo/5JKIjJkV 2nU2cvr3z9n5yciODOglYRwxkXgx5RBbas3GAFmY+nhGpxn0OXuFm6SUUFRR+JBQjnYaubtB Mc4lZgSIg7CotlFQpyar2BFeloxqbiI8SzF6J5yYi+3lCPIrMgjt6SNXXxo4CnTE9rlyQitt qQhFXRecBrfWtBC4njhItBj6iCuONKQXkOXSwNlWuQFa0J3G2tF1CpZQPXEFT1IINbIeWxQV WfiEb/MOQZUSmaFyPEHAR8IoVuI2rypQh6rJKYLAGdidXZJKHcX6yFjRsB9d5N6LYmDuZxSD iihYtBj6iCaF/fcbAOulgaaqIoEdYDgTrZyKs341SSnITK5mGvVY6tgpzSnW4ehg5UXMtGaT

Uvjz+iJj9ogWYXJ4t6MrFqQEeNwjLs2KoMfI7y25OYNgJ2AS9N6rYFcZxwMegRVRDtC8wQ36  
Aas7U0nFF3Q007Hv3pySH0IZXXg3ptNipG3dcqplUB6c5ZyOBdQQfGGy6k4sdIZ69C84oOM2  
qKg9fsvEEOiCjxU9CpeRrYYG1YQdBQ68F003WglYRwxkXgx5RBCm+gAWRLEqDQamJW4W14P  
Ck10hSRt3XKqdVEZWR3aPooMq4wcf1+zDrs5egeV0doZ67ovgYiBpdq/GQKE6MISVfCwtcD1  
TKwaPq7oJWEccJF4Me0cPIqXFiQuJaGhWBSVLJBW2NqKTIMoBa1aRDJAo1tFJkhcDJcaE6oy  
Y11P/I+AYdqLy6GIyZzQM4ta38COQcUSlMn8y3nKFWbbcsAQ7zjohc0gcQh6Lqk1DooCZU5V  
PipP2bDO5m7ixofSQf0UK9fWUpwRxG7Rt6YB2FXYLoS0+mKrEE81d9RFqwgMSShkOIvTxaOR  
XjMKKI6J5EFyWPOelMa3L8hyHbiIpgIsue3o1TMyiMt2h1RtF26KM5ApZFCirXQukRi3o6ke  
pNkFYTnddnP+bqglZB5hEt1Nv3cVDNjhTSMcl3fI4j6afhBPblul871UtZX3VPRNu3nBPY17  
sP9z7uero38Y1x0358Ctq+5ZzAvnp48FMam/Hp911xHEK/DUNSI9r309CS1tP2nUdLWk7bdx  
4taT1t33m0pOW0fefRkpbT9p1HS1pO23ceLWk5bd95tKT1tH3n0ZKW0/adR0taTtt3Hi1pOW  
3febSk5br959GS1tP2nUdLWk7bdx4taT1t33m0pOW0fefRkpbT9p1HS1pO23ceLWk5bd95tK  
T1tH3n0ZKW0/adR0tazj37Ns3xGX5duLDviJoiWtJy2r7zaEnLafvOoyUtp+07j5a0nM9g3+  
/fv78sf/BilI9K27ecA9mXM97i/q/yxb76heCjffT2GuG31W+xTox+JyQN/r2wy6z8i7xHef  
nlkvrV8qN8Azocfxn2GiH2RUOUVDEqrD8XoPiMJE13tq/2iWhfkf6Swpq27x3SxZ8Ufn19tb  
NPR5L0oPYV2on9Zxrib/1N9qVJ3dhaOGTULkul3xOcDp9DHOGOMIykMMLqb3mg/5e3vxXCJ8  
j6fvyIVwKxRa768x//Fc4+ooXD2Zeh2hG9/Skr7nfIrYddbvxRvkiJrPpDRqyKs+FXhNbhPK  
I/p1PTHHaRde1SGNBEAiILVzitFJGRGvs1FekJKNhdWt3WswkSXoI+yZG22JluxmVabL/on  
2pxOWKwbusPQ1oTX9nm0Ocy44wjGRfkPLpjkSN/QouYuWosNbiHx/wtoDzjmjhvZaa5iARLS  
4mW3+xiLlc4HTbumpfjkJckrgV1Dkr12ACnHFEU8Iwbtk3KgbUXLWv1E6wGavbTDjviBaOZV  
+I8uFLGZcL/c7uC/Skhp6k0mOZMqt1L3YfgEgKi/9q33WGxUiSvhdSwxw9o1Qz2aMNYmt3M  
m+gp48HuiZQf0x/WhbSMWt2UXSNVcVTnoKam7Z11gLYSaLKRjGiBYObV/LB0nuaf8qXQ8+Ch  
/b/aDvK4rn4BHUC8P49/b1HyBDNyupr3HxaQHvtn3Hjy2iySL2H85D03ryFKFvDOyyFP2cQD  
etRFwVehKThCUBHnyvnmU7OPuI9kMKr+cucdJ3WbohF77kKG0Q+F5iSnA6UM9a3NpxtiZJuq  
d9OWNkvTUiImoCWuNgAlRjGfgcxyxHoa9r0Dd+m+ZAAqing7w+E847op2IQoEV1jUfFVQJKI  
nIVGoPjiZaEWJk1Jar/gVtOWI2nH1EC3va99PTkpbT9p1HS1pO23ceLWk5bd95tKT1tH3n0Z  
KW0/adR0taTtt3Hi1pOW3febSk5br959GS1vMT2ffr8g7axJ8fgrNIel+o+HP43dnWvvGnkf  
GHtyAvzBzy6EWzUfim9k0aiqenJ2Y6enyItVBx7XTSLdbrAzCSES0U2xfwkF7vQJH1awY0wS  
hU8/LychCVRZwkCb29JmtabSrV+jh6X2cUDkaStN6+Qi/K4GC98GHQZSNpEJ0z/gz2RUAYW0  
Z2TXSmpuSdJJKQbaM1epwk6Vb2fx5+lp/Sy5A/zBukQSy9FYVkestJcIj2mIgz2GA7RfGmk15  
q9Lv8Xw2Bi6vWGFK26/Qm/YJWKZNA1195ZewSyjagUBCRz1FEausbv2fEzU6Gq1SbQpR6FYq  
8hoIPgWGqUTZ11uFCNBuPirQWtgswjWtjQvznzqpVLFgtmCYqZqXdTTTTfhSvgeCmmRT4BDsSz  
3KEtOqevLQTDlU1MtV9LWkIpCBw/kkppUmElLUZem8j+Cz14JKZLZwWEQ8ZkyjuRBLVT9XII  
gsRst91D8JBTE5sut06uzliO+qk5MTEdxZ0CriOOFi0COqWJaVG6wg8/GUUC3uc+gSN0s3qe  
ij6COxAOE8bHWTdoJD3AoMCXFHYbGsT0FOrTewQhzF2FR8hHj2QjQvGdHXG+prZBbE0Vuypi  
5+FPCU6WmtklDEVhuSsCo6D3itby1oIXGccDHoEVUQd1wpKM+hi6Whcg2yOjWBU631AiqlbO  
AaYqsMZHarPdw2qOoT0egfhjwjKkXzYoSYg4BPpNBtRE2+FEGPVRKTJaAzstqbJTT7i7WwcS  
Og3rsJ3dbEwTwOCUe0cDHoEVUQ7es7DtZBF0tDTRQ1wnogkGZOjOhWnEqSk1DZPOylyrFVkf  
O6083D0IGKa9korebl8UFU5Act0OziZFFPJLYN6KhrWIYdW5WBz1F+exLTrsAu4KVJ3bYgjh  
MuBj2iCqJ9gRniG1RjtpaGM+oOaHzXoz89OYQ+pPJ6UK/NRswO+1rltCog3TtkLgbwr6MB4w4  
VU/Bjp7FVoXtFhRk1x8JqdN0hBB2S0+EmolHwtLLA07CBo6LUAut1a0CriOOFi0COqINKXsC  
CSRWkwKDVxq7AWHJ70EknKqPta5bQqgjOieXQdVbK3+Lh+H2Z99hI0r6sj1HNXFB8DUaNRNR  
4SxYkxP0RrYYHrgUo5eFTdXdAq4jjhYtAjehg5NU5MSFxFloyIwSSq5oK0R1RRZB1CrmnSIRK  
GGVoqsEDg5L1Rn1KSG+h8Z36AD1VcXgzGzeQCntpUfgZwjKoxpk/mWM9Sg7ZY1wGHeEZFL+g  
DiUFR9Egod1ISqfEqctH+Twd3MnQWtguQjWqi3rywlmMOofUMPrKOWBB96clUJZzG/qqPSA  
sWgljScAixl0crp2IcRhQR3ZPoguUj51pTY7/MGQbURFMZJnQO3FqBoXxFq3MKNOOfTRHwL  
JIQeVaKD1iUU8HUqkJ0mqi8/rs1Fxd0CrIPKKFevs+DqrZkUI6JvmOz3Ek/TScwL5c92unc  
nrq+6JaPuWcwL7cvfh3sddT/cmPjFu2o9PQdu3nBPYVw8PfkpjMz7dvuiOI+mnIU16Rpt+G1  
rSctq+82hJy2n7zqMLLaftO4+WtJy27za0nLavvNoSctp+86jJS2n7TuPlrSctu88WtJy2r  
7zaEnLafvOoyUtp+07j5a0nLbvPFRSctq+82hJy2n7zqMLLaftO4+WtJy27za0nLavvNoSc  
tp+86jJS2n7TuPlrSctu88WtJy2r7zaEnLuWffpjk+w68LvT00J6bt25yYtm9zYtq+zWn5++  
//A3Xnpu/lu9g3AAAAAE1FTkSuQmCC)

#Beberapa definisi dari bagian confusion matrix

True positive: 41 kasus di mana mesin memperkirakan jika pasien memiliki diabetes, dan mereka memang memiliki diabetes.

False-positive: 13 kasus dimana mesin memperkirakan jika pasien memiliki diabetes, padahal sebenarnya tidak memiliki diabetes

False-negative: 6 kasus dimana mesin memperkirakan jika pasien tidak memiliki diabetes, padahal sebenarnya memiliki diabetes

True Negative: 35 kasus di mana mesin memperkirakan jika pasien tidak memiliki diabetes, dan mereka memang memiliki diabetes.

```
"""
```

```
cm = confusion_matrix(y_test, y_pred) #memasukkan y_test dan y_pred
plt.figure(figsize=(5,5)) #ukuran plotting heatmap
#settingan heatmap,
#data=cm -> menggunakan variabel cm,
#linewidths=.5 -> Lebar garis yang akan membagi setiap sel,
#annot=True -> menampilkan value setiap sel,
#cmap = 'Blues' -> color space = Blue
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label') #Mendefinisikan sumbu y untuk Actual label
plt.xlabel('Predicted label') #Mendefinisikan sumbu x untuk predicted label
all_sample_title = 'Accuracy Score: {0}'.format(dtreescore(X_test,
y_test)) #melakukan kalkulasi score akurasi, {0} berfungsi untuk
menampilkan angka akurasi
plt.title(all_sample_title, size = 15) #setting ukuran tulisan "Accuracy
Score: 0.8"
```

```
"""#XGBoost"""
```

```
from xgboost import XGBClassifier
```

```
xgb_model = XGBClassifier()
```

```
xgb_model.fit(X_train, y_train)
```

```
y_pred = xgb_model.predict(X_test)
```

```
from sklearn.metrics import accuracy_score, f1_score, precision_score,
recall_score
```

```
print('F-1 Score : ',(f1_score(y_test, y_pred)))
```



```

print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
y_test)) # menampilkan score akurasi
plt.title(all_sample_title, size = 15) #format title untuk accuracy score

"""#Metode 80 20

#Berdasarkan Jurnal Improving the Classification Accuracy using Recursive
Feature Elimination with Cross-Validation
yang dikutip dari :
http://www.puneetmisra.com/admin/uploads/journals/5f136d202b8ba1.1864411
7.pdf

"In
this study, we have used the train_test_split() method of
the Scikit-Learn library of python. Through this function, we divide the
dataset into a different ratio. However, the 80/20 (train/test) rule is
mostly used in the studies."

Tujuan penggunaan train test split menjadi 80 dan 20 adalah karena rasio
tersebut paling sering digunakan dalam penelitian

"""

#test size 20% and train size 80%
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2,
random_state=0)

"""#Decision Tree"""

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(random_state=0)

```

```

dtree.fit(X_train, y_train)

y_pred = dtree.predict(X_test)
from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score
print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score: {0}'.format(dtree.score(X_test,
    y_test))
plt.title(all_sample_title, size = 15)

""">#XGBoost""

from xgboost import XGBClassifier

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)
from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score
print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

```



```

all_sample_title = 'Accuracy Score: {}'.format(xgb_model.score(X_test,
y_test))
plt.title(all_sample_title, size = 15)

"""#Metode 70 30

#Berdasarkan Jurnal A Comparative Analysis for Diabetic Prediction Based on
Machine Learning Techniques

Yang dikutip dari : https://www.iasj.net/iasj/download/a371daadb33b96fd
Disimpulkan bahwa Train Test split dengan rasio 70 dan 30 dapat
menghasilkan akurasi yang lebih besar dibandingkan dengan metode K-Fold
Cross Validation
"""

#test size 30% and train size 70%
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,
random_state=0)

"""#Decision Tree"""

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(random_state=0)
dtree.fit(X_train, y_train)

y_pred = dtree.predict(X_test)
from sklearn.metrics import accuracy_score, f1_score, precision_score,
recall_score
print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

```

```

all_sample_title = 'Accuracy Score: {0}'.format(dtree.score(X_test,
    y_test))
plt.title(all_sample_title, size = 15)

""""#XGBoost""""

from xgboost import XGBClassifier

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)
from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score
print('F-1 Score : ', (f1_score(y_test, y_pred)))
print('Precision Score : ', (precision_score(y_test, y_pred)))
print('Recall Score : ', (recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
    y_test))
plt.title(all_sample_title, size = 15)

""""#Metode 60 40""""

#test size 40% and train size 60%
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.4,
    random_state=0)

""""#Decision Tree""""

from sklearn.tree import DecisionTreeClassifier

```

```

dtree = DecisionTreeClassifier(random_state=0)
dtree.fit(X_train, y_train)

y_pred = dtree.predict(X_test)
from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score
print('F-1 Score : ', (f1_score(y_test, y_pred)))
print('Precision Score : ', (precision_score(y_test, y_pred)))
print('Recall Score : ', (recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score: {0}'.format(dtree.score(X_test,
    y_test))
plt.title(all_sample_title, size = 15)

"""\#XGBoost"""\

from xgboost import XGBClassifier

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)
from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score
print('F-1 Score : ', (f1_score(y_test, y_pred)))
print('Precision Score : ', (precision_score(y_test, y_pred)))
print('Recall Score : ', (recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

```

```
all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
y_test))
```

```
plt.title(all_sample_title, size = 15)
```

```
""">#Random State = 0, Balanced Dataset
```

```
![hasill.png] (data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAANC AAADYCAIAAAA
GfVInAAAAAXNSR0IARs4c6QAAAArnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQ
AADrySURBVHhe7Z1RYus2sgXnJzvIz6xhdpK9vOVkP7OzeWUduHMMEHQFAPAsdX34NpsASJQ
bgG8SK//6X5IKSZIKc8hTNkmsJElm8c8p+68kSZIKSUZQTtbq1C1RMohU2iLNJENyXUQpu4h
U2iLNJENyXUQpu4hU2iLNJENyXUQpu4hU2iLNJENyXqW+ZRn5//7v/8qF8ccff/znP/8pF5d
pjfbXX39xq1ys5bzSP//8k8Zb+vxgm77//e9/y3Uvf//9N+PwtVYPg2FLlHwrrbJ/aMGOqnb
Wmp6+pbRIkTn42Vg0p5i0Blh05fobFhL5gafslhyxPOX1T1nYPdK23u4i4XB935kHr1ei5IN
hYapKq+J/SnmwlvQCvBIvEH8xiJdMkVn4DV50p7KpUf2QuK1U3SoXQ2GFxCL5FX+Xhd1Ttg/
mviv8D0ga8g7HTCq25HdBsSmoiJ/yK4mHVqcsWUNL6Wvr2ftHd81H4Vvc3FOWZcNpIKra7cq
Zd8r6IUg5v/SU7d5rmDtDdZyy+k7500ziITPj27Mt/idSnbKPCqVv8jb4Fjf31OUQ1crhrN0
eddUppyws4cYu+yvgIJG1AhrxqmoBjSSskIK9TNvIXbJXXkabyH3QFhirRCaqNhiBO2dtLfaG
ZMgViDEQsYm0TcMnUFCiOmUYzXQou9QJCBmKcW/MfDWIQJWmj78VJdbQsUZJsil+o/FgFffW
1x1YA0VgNivZFHVvdiWHL9QZ1jwoP1FGU1G0T45JbPCueK0qL5CPxArBodFlo51WsAq1WEXd
jGUSsSo01wLqKmDagJG2Axrqly2hJHIUEJfjqNeIZOTjkrilxjrQgCfRRuPEdIDYtwyZVbd
J4K7mApqm8t4GiDU12mVnULLP0tw1uAYBH+er9fctviopaKbuGv8YmpUoSb6rS4UnovxiFSi
jmNJV9frKjdgXu6r3IXbLmMtYgLHWEKkaVaup6pt8IJRbiYjLnxM2PkoQysVt/ChTQaVGxku
TelXed3bw6o82Abfict4ayTh+NKAWXisP3v0KjFOiE1QbDYHb493ibYmjmcAGfaMBLx/TOZ4
aeTKGz+kyBvdxeIS/D/L1uOPxw9CsREmyKX6h8isXBovHPk41yjKaHSzq82gVVGWwsJebQzB8
dkDmzBJL3xgvDotEbnxZDRbl3gwZxUHLesqGZynS7zLjUXk/j9z51IaYjmLLevyNa/PTY6ci
bptOaGhlidGrxlQy3sHHiTacmFserfGPUd8kEaqiKdyh8isXNyhCFZtvF1GKimMVVJV5Hq2
CqowZtno9QUsR7T1OPhbKoETE5c/RGx91Vp2CKnqvPwrX23BXRJLGXHpxc6kRPuGUBS5jUry
SXjgyPgWP5U3TaU2NxxjFOxNpf4h18HNRvEvgbxuNb4x9CsREnSKH6VX7n4+YO4x5RiNPNVUFx
mebQKqjLmcBfetvDQeP9t3+QDiZoEi4ZufIy2rW8q1Z9CacZJya3d9aDyVuztxhKi/d1TFhi
ztfaOFqS6K74C45ToBHoHLdTA3wq4dFFMIR5BPqbDy0dHH1btgQY01kCakZE6xTTWpY+jWHm
guxRp/HglYwUpoVmJkqRR/Cq5cmH/kgIiKkjFvgqIo01VmefRKqjKWC8Tb8iDaMbXaMZzdVd
9oePRydsQNQkWDdr4VKMiShBYFSX7vcsr114f10EUdNyK9ROZwGm6BB0DehYjg/LAQ0v0r3/
9+9//LthPvMca6goMUqJ7uBwnJiiYTpUp7W6TVEBT0DqPwZkRX21AMpohM3SR11ttH7f7X2g
cz1SDgMd31dGmRmH48Uf5a1aFSo/31hU7dSqn2st9t3Du1WiPkhsOMKfpe58jfyxoK+m5GO
hBkpEXP589sa3Lfpt5tfxXKWvTJpJkuSNeb1TlGm1foYVZKqfQH8jeZa0SDNJkrwxL3fKgv9
TI3iDv8gCEylR8pM0kyTJG+Nb3Kucsm9JKm2RZpIkeWPy1F1EKm2RZpIkeWPy1F1EKm2RZpI
keWPy1F1EKm2RZpIkeWPy1F1EKm2RZpIkeWPy1F3Er1P6H/t4nalksSVJ8sbkKbuIVNoizSR
J8sbkKbuIF1T698//od6zyGJLkuSNyVN2ES+oVB/9Wi6erXzbkiRvTJ6yi5ikVB93TsBX0Oe
S61P+4dbkC/909fi/hQTcjY9N56s+4ZKvCgTxxrW39Py24DmOWKEmS503wLc6i3PhGM0Op/89
G/FL/3xIC/VWVy/jvmNRGDeLvstEx/mqrMzUOVAKd3zqtaa/8EBiwrEmSJG+Hb3EW5cY3mk1
KdUAq1hHo/5dAnZr6G6qjY9L/ibGPIzhZdcOyYHVrLFMHT5IkeS6+xVmUG99oJik9c8ry19A
4TZ2Tp+z21limDp4kSfJcfIuzKDe+0UxSeuaU5avOy4qH/i6rf8g8g0lmkiRjXGhf4izKjW8
0k5SeOWWjwOVbvQiUAMulaneME5ZoCUoBv0721FMMpMksfIK+BZnUW58o5mhlDOSYYWOSeg
xTkTO0XJt/00Tt3Tp4+jvrCR16QetMqDMKIYPmCRJ8jr4FmdRbnyjSaUt0kySJG9MnrKLSKU
t0kySJG9M85RNkiRjkuQ65WStTtkSJYNIpS3STB/prZtU10d66809WZQ2R5NKW6SZPtJbN6m
uj/TWh3uzKG20JpW2SDN9pLduU10f6a0P92ZR2hXNkm2RZvpIb92kuj7SWx/uzak0OZpU2mK
xGX1Gh6g+0Cp+S7hcN6CZf0aHiN9R1id+LODue47FvVvzPONNDaDqu94b8LgSzUezC+IX0OG
uN3f+9/enzYi39xboN/Uf8hYgp1qqT/dm0TNsvjeptMVKM9q2tGFpvcVBy9LVx2PpEzyUrIh
db7t0I0OwZgG3XnISPC4+Pow4Nv0z3kKOPv8k/DzFGyxTx4yiwIDpyxV0eFMMb+9NMC+eWH3
S3BlvAQ1CFLyCN4vW2vwEummLlWZyn75oWBh+2VhKQzqkXejiSxd8xfpKnspKb/LjP5Fovme
80cZPGuTEt+Ap3mCZOuZeohv4kYoz3lwasft/e2/AvHhcTF0c8RZQZtx10a/gzaKFNj+EVNp
ipRmWpZ+ysQgJuKUKVJcV1S1b7YDV5TxWetOkYodCjs6Ph7wJGuhb8CxvsFJdwNSibB71hu2
o2w/xxrPQUi6+Oe8NY5QrDcL5i3iz6BlV+N6k0hYrzBAm/YCMRUh+9/QFXs+7QHXXsph9Clq
9OoSmsriimLXmhRw/bs97E3RR92d5g8XqBHJA8UPeMEP7cvEZ3jRHtPAVYvrnvamLN3grBxy
```



9owrfmlTaYqUZrTS+6jLOy4PVuyV6iWet3vUVhSUEipxy/aA38APjRXa9ZTBx5hjxSW/c4m2  
FMp/gTUtMlcbUeK7ik96ijTd4EW8WrbL5OaTSFovN+LYFWpAPnRbaAsrFx5yyzAtLmmy4esg  
b0D5Omo86Zzmdm3nUG415Z5oRf4K3yg+/JzxhgoUKfYGL+LNouVV+Pak0hbPMsMC49Ha9Fm  
NLGDlgZXslxWx5oWWq8YBH3Yqi73FpDRf7fgPeaOxb2rP8gaLlQFzh3LxoDdBA43wCd6Y7O5  
pesYbSd6zgiP2RbxZ9KSN741JpS2eZYZlGxuf1pxiYOnGj8NbWMBVT9BcRnuC7cqfwUpv1d8  
DYrM774182A6e4g3WlxxTw1W5eLDeBFUXbd7eWxyroqPeRDXOK3izaHkVvj2ptMV6M1qr/sM  
ysOR0DLd8fGVu2Z6y3oWRfT+dx0pv+ntAHJPhyuMdb9xy27SXoqd4g5XqAHtBM+frDTDjbT7  
BGxOMGvM5PuSNlq/mzaLlG9/bk0pbrDTDQuNxsPtPiliBuluub3DpC1UNhK/SGJmVXFkT4V  
lWgKT1QQhtj9x7C3MBGyUagPrvQGPK9ESmCOUC+PYG0J0F9yY+ARvmiBUx+GxNwdLVfLp3ix  
aa/MTSKUt0kwf6a2bVNdHeuvDvVmUNkeTslukmT7SWzepro/01od7syhtjiaVtkgzfaS3blJ  
dH+mtD/dmUdocTspkwb6SG/dpLo+01sf7s2iJEmSJELGUE7W6pQtUTKIVNoizfSR3rpJdX2  
ktz7cm0VpczSptEwa6SO9dZPq+khvfbg3i9LmaFJpizTTR3rrJtX1kd76cG8Wpc3RpNIWaaa  
P9NZNqusjvfxh3iw6Z/M/e5/LDN2fXKXPlznovv1wuw78U2wq5n0gCIOX6ALMffddZH41Q8w  
8RHx2TPUhi638Fr4L/nE8+gBC0V38j8KzSrSKK96uOx8IjyvREnbneKZmfKfyTytr5WfD40o  
0H9aXJhiUG+bzYLt2RaH9YMy+IMsov141YrPVgq2Hwn2OvC27j1ela1z3h12/TuqukFvuX4  
Xrpt5CP+OE8R3vJXfopqPxdffoQKW61Ncyy/yFurTSs/m5XqWnO8WzOUWXSktBvRv5RwzBt  
z9DMFOTf15iuH26PHiR0+OBhzNu7NotM2d6eqlm2Zh6Ft+W1FSPaXc975/NKW/CerQX5q71  
u5iG8XP12x2ps5bdwC6JsqJnnDWV/4u8Xek7g5XqducYGUEbbpWLB2KPApZ8vHMrv4Blz6q  
2OOTIGHn/qYL87g8ZNKsMQ2vMBbg3i07b3D1lgeTKH7K6uR2yK36iuVigeJZPxlm2Ga3hopl  
HQWMIzKWB41a+glJnudKyVTaM0+o711/k7aLz4axUd9LP8dyput0GrfwkFpdcgB/9bMFkiZU  
EViKv5D92CNYm+d1bQYy5AN6kRMT1z0ds8qI03p6ymqemoTbCf6AoKfs5jnG4jMnrrjY1Zb6  
ORNvdyKsN1NQtsRt9A+C4CqsBhYpe3fXC8SBvHG38G9+CziXqgmdJst5EycANx/dim1QsIUy  
EOKZTTTnswe6UwS/DgC417SS0L9ESVGO8OS8ZdQWtvINS3eLrtmwEKH6afje/yNuVvjNYqe7  
MHI9rptVxsTRY6S1gmrHW0Mg7xAHBLb8MlFHj1rrW+p2Be7PotE1t5VRSuf6GuWn+EFuWCo4  
McfSKpALF5BkBFwTokCkC7oYdksqDbil5G6M0k+hbk31oppYBb6sRyvVtzJggd/VQRgY1t4N  
sOX6Nu0R3FVa8D6CLjFzFfHeTwPvHazMRvfZ2ytFMj4txyN/u/+NBDfQTA9qo8Xn8uWuQkO2  
3rJUPNGUFu82Ye7SZzS/yBlf6DmexuuM5HtcMt+gLSfREKz8VHleihWAVtjttta6Erzg5dbtH  
utN2RfMwFuDeLTtvUtLdvTElp/qowh8Ykd2vOraFyW0D0Ukc91/1yybAEdIzBNaDiXWJAH+d  
qKMEIjt6Kr+X6GzVucbfBAcwiCgt4tL8zt/Ye2E2CT40GMU41ZSF7oDKNoIJB41m7Dz2GYUu  
0Ciar+iEoqrutvMBGTN/VObsdJ/FbvIkrfyzwn3XHM9MnGKjOxVYrr9p5Sex2JmQvXLfXA  
Gd+2xVLc7WzXmbHjPEhGXPx+xqerZfptj/ruTbG1S2tY1f40sYnejlZrKtZvicdrifXANqHi  
XGNBhqHhnyZip3r6XY5f4xjek04V8XTuauLObhJ8ai7K8yBvNND0sR1BaWHE94KgY8HTt0R  
LYMrxksw6DLTyARletcKFSFe5mA9PL9ESur3Blb4zWKnueI7na4a+MY7Tys9gcckB6+ugKni  
fu3OnQXUAHY85A/dm0Wmb2nyrqVI6kWSG2ymRpDjKhUEX0lZlRyaSDKXrtLP7cxlQNv3w0IC  
Kd4kbnRhKMIJvpuLR4j5+jQO2NSHn8YY+32A3CT41b+N5ja8pe0zQmjLjMBqU60foNtOBTwc  
I9PRWvoWrE9h7qB6u81u8jXI+kGUPop7jQzVDvcU4Tis/g2XeAhRBufjJdg3uQrPKz8GYk3B  
vFp22qTlyWmG/JhMbrhrErJgwjZWMNkoS8JU8d4ndTiTRGmYJik9LYuXdvGZUvMttvPpbVZ2  
yDOI6LVpEE+P5AHHr3EAL1NVCEiVfO7xwkyZzG6SgNH0qvSLAWj621PWvypq6w/VpQLQ4+5  
K2IWOJVOCM43vOFOOuJXfhZl6A9pDubjdDUvz+EXehjgfyEp1rTkSg2I4rhmW4a6cVn4Si0s  
OsLerhTfhVrlog5/tptQacx7uzaJzNpkALbdUc9AWLKImaFNS301t6IK7qkhdcosGcRmDRAa  
qDCoZQTFstfpgQ4SKNc/zwx/ULT06esND6BniR5Bg001IEv20+9zidfetyZk/JCUSWUgmsU  
jItAE4xLCg2CovtPlqBKtQu8Pmn5Qsj/z1aUIdYrVKyCjW1PhQSVahWYHHd7UAM70nQ2PK9E  
SNEGIOR7UDLga+fb1fDxKL4AnlmgJ7DBVvcSettlvSarx7r4XbMdcAG9SIuLy53Kbn8Dbk+1  
e8F1sfaS3blJdH+mtD/dmUdoczXsr5UfL7U+XJ81i6yO9dZPq+khvfbg3i9LmaN5Safxj/yv  
/5OotzSwgvXWT6vpIb324N4uSJEmSJBlBOvmrU7ZEySBSaYs000d66ybV9ZHe+nBvFqXN0aT  
SFmumj/TWTarrI7314d4sSpujSaUt0kwf6a2bVNdHeuvDvVmUNkeTslukmT7SWzepro/01od  
7syhtjqZb6R+b/y3BqF9F//NGudig/364+/dzzsNtSRSKU0om/ZfZDz5eI/q6mfhvrYfxSnY  
yPKtEq9j1FpDU3XL9k40+3Fr8QQgt15zElbm3PmMhPptiWb0BjyvRfNCiCQblxr06FC/rzaK  
1VfgJXFHKhk53bf2KF+9KU11cbKgLeWT68CzcRpIlyTJWXEE+2vPacdbGe/qSX7OAX8GbkI2  
WNGj1ja0w7q6BJ5ZoPrep98999yBhkOjFCMsODJ5VoslQUf5TLBN0hx57HTov682itQv4E7i  
i1E9ZoFD88rezuNh4XCxgVp1Wo8vU91cuDPJ+kNBX174dAh1bK38sr+ANdMTubmpBq6/42jU  
PT5rhrFR3Ze6UXFVdgjHjhGDMZfaWeavOP+YYHo59ilf2ZtHaBfwJXFFKxdA9TgLqW8v1t70  
42DgaY2ViEhQH2N4mgaSfsvj3b0pAm93uw3kRb7dj4s5udez8zAhjWanuyty5xavCwU+B1eV  
Ul1j2oAodh4NineGVvFj3J5htzRW11y1JkCUnNUWdcAnUTd8FXb7SBkvpZr+rF18iOCjUm8Lh  
b7y+ispWkTZS18g/R16sbyWQKvHZMNthNCjmm74L8xGWA1ZA21VfwpqLYrS7n2DnF47W6gNZ  
7zuDK3FVd0ht9uatGFEOrdmew0lvABF3RsU/xyt4seobN9+aKUHWWCkLFobKjaIjhluoLSor

GiuMQ9fKKpPojnrVg5S5DVxOWR5N7C/AmEoKNSMgT/AQ9CrRKRt8trsbEr4m0/hXjJqvRIF  
/U4IwuQCeXqJVbLlJQky5uuu0nAPJVq9JLFZ3fe5adFpljOb1qVtVHU5istfBfGNPEwc+K17  
Qm0XPsPneXFGqvSzwWiGmesrF7SmOWhKoziq4q/3RazHwJm1iJwVVOYEOnuhYvcxJNNRKeE+  
9ufxUsHq5VS1sISfBtvvugJPGBUq0iq23aufa/c1DbPsGt4PmnU/ZIXNnAWpxVc5Vk7vOh7P  
Ym5C9cnHjwOeWV/Nm0TNsvjdXl1B5vXqohUOUUpDpSMg9Chb5ydLHWAQWRUherIXT91432qwf1  
lzsMIJVoCc4kTlBfe3ePI756yDq9dtUERQsrFfF7B28md69g511VmNivVMbUhc2cQLS5JVhL  
IL5vO4pIDJrs1duBzy6t5s2i5zbfni1IVxO4OTp35wUazOPMcklGXDN397AQuaexFqdEoZRr  
fmnrBfrTtnqhTVHxQ7zjTa7IKpa3kx8V/I8XsGbgqjMXZ93nWOykjmb7UtOYUdcKbkYx8e  
k8KpVPI913gJmB+Xi9Pp1Xs2bRcttvj1XlJ4/Zakbf5DKSMnoHrUVpyyVF0kCDagKV1Eqjbb  
sC2pTff3rn7LAS8a+xttu9zshH298vDCD1IsbjOMTR9TuN2ssL+KNyUZM4B6CY+dcVpnZrFQ  
3Z07UZKw+iHG0ABcUm1hccoC9anbHPite0JtFy22+Pd1KqSr6Cq8YKNmfecqoZ038o0FJfSd  
jWG6RiQaqQh2rQu09Q7FWGf0cIKqXvAtdSrQKvSf4ci2pPcmVEyarW8LdikcN9MGDSrQKzQ7  
Cm4iSU2GIqpkagCe12QXvelpodnBm7sRq5utrW1FRdVqea+BxJVocU3NjgSYOVYH9Cm8WLV/  
Ab08qbZFM+khv3aS6PtJbH+7NorQ5mlTaIs30kd66SXV9pLc+3JtFaXM0qbRFmukjvXWT6vp  
Ib324N4uSJEmsJBlBOVmrU7ZEySBSaYs000d66ybV9ZHe+nBvFqXN0aTSFmnmj/TWTarrI73  
14d4sSpujSaUt0kwf6a2bVNdHeuvDvVmUNkeTslukmT7SWzepro/01od7syhtjmaGUv8shWD  
4RyIwZnzaAIP/ceITuh+C8Uu0ivhEjurzJY5/n120+gK3dn+JfhK8Q41W4Z9kU1I3DpwItKh  
BcJyfbzIHiv1vXmwHH8fRch5r/1mfrrCA3bn7p3kcz53u2/V4t1ZnwONKRFz+fMYCfnvmKWX  
kOAW191lonRAeMFuMzLDFlqlujYMwSLYH313S0XH2x3fXW6hs75hufskwtZkcQJXfgU5D3JK4  
0Tiu/gJXqvqxtvDH3SFJlrTV167rjnCDyzGXZQft0bxCutB3tzr21Hu/W6iR4VomIy59rbX4  
I85QycpQgUElRiEPw8Q92hG5WFhvLz9+fecUlT45X3UFFcdsWfzqqSxepDwu/DB3/URy1wn  
QpkQ3aKNxWvkFrFS3641dXhnaQ+t9dvsC+bBhflnVPd1bVSG04Va52FCtxz01Ogn3ZtHaBfw  
JzFNalRqlM7Z6fPzffspWS0sbnLY8FiRxxG456CuqVT0bnl6iJTD32OVRAQqOnWyh/W6DVn4  
GK9XtenM4ObZJsdTxxpWE6nIqL+UNaLObF9V6pCXty8W5Wh2Fe7Nooc0PYZ5SR05So6q4pIB  
0Kagtkt4M1BKIEFV2wquTy+j4Bqcsj4ul5StNX9Uglrdz0Fcgzb3NhqeXaAkcBjwRCUzct79  
jJxU02FXUyk9ipbpd8FuMmg5f4XTYjBh3gQeuFsuNtyW4z9F9WitDsS9WBR2AX8C85QycrA  
tRwpRlCRXG1C7xBSfWnqSlhQigeovypdYeaDXrz51Nd+wpJVcrbRq+SdhvtWqng1PL9Eqte/  
5HM/4dBhBxVbRyk9isbqtN4E38nCwprZ9yXj7Z50WC2h5E0w8Cm8XOnrfr2t1IO7NouUL+O2  
Zp5SRKUc115VkvQBDkWmZGmxQcUHccvra3yq+68+ZaFysjsdphlTdo77Vqt6NrxAiVbFLV  
V+cPt+Aw0QrkdWv1J8J41WsKut4Ca4Vbr4xt3885ZY+97Sad7Xp8qFYHwrNKRFz+fMYCfnv  
mKwvKhQmqSD8eWlpVnYHO0XJhKM8Bo3GoSOV9zDc4ZR0evbu7kdw9Zz1t3+2qnsbib0wt5ks  
N7M605VNQUbu9Wv15rFR3xhv5XW+7fXVUKAk0WDad1/GmbapcNKBL1cs5rtWxuDelftr8EOY  
pZeQ4ErTk4oCsFqRQsqRP1k/5Jr1Lq1FyK2Y/i67fW+Lurmgh7PSW1USkiHfWYFPQRVFITm  
t/DyWqTvjdFc2rbeDvp5HHeYVz2b35Wdw7I0pnzkqb8uxucBbt2bgL2/RKpufwzyljOyBFAV  
EpvVjoBYkST8sSaqs1as6qn18WnrHITB+iRbCQ1sTYfqy1KLv97ao3/OUBaYcs6Meqpk+Ax  
oQJmVC60Vn8dKdcfegHprlU2rb8RatsvsvYI3Y1AM2rvKxU/oEr2cm7U6Fvdm0TM2vvdnh1K  
qjWGDkr09S+ikLbc31ACos5L6T1KvuoxAx63gWZEFW6MMWKL56Idi0M8TQeShOmLJaK22+oI  
2u2DNrseDSrQKzQ5i8zpw4s2AlrtbXis/Fd6tREu4GfoiZuort1VvQm2gshSLEYELNR8eV6I  
laIIQc49ZB2GPOJrtrseDWp0NDy0Rcflzuc1PIJW2SDN9pLduU10f6a0P92ZR2hxNKM2RZvp  
Ib92kuj7SWx/uzAK00ZpU2iLN9JHeuk11faS3PtybRUMSJEmSjKcCrNUPW6JkEKm0RZrPI71  
1k+r6SG99uDeL0uZoUmmLNNNHeusmlfWR3vpwbxalzdGk0hZppo/01k2q6y099eHeLeEqbo0m  
lLdJMH+mtm1TXR3rrw71Z1DZH063UPxrC8V9mV5u7v5+ujkDLv/76a82HJ9yF9ynRKuITOao  
PBIj8we+tx++2Q3T334Jf9ikBPKtEq7jrrcoHd/3QUXfL9WSWPUjs+nmoZio/XoQrVzGPK9E  
SWnUVn+nR8nbg527fGfC4EhGXP5+XGN+ei0opuKrU4vLP7/9T7HHd+AjqsN9HsCblGgJzB0  
iztUpxo9ibX+tg7b6Lgj6KuDu3W/EKF7EWYvvhPiR7WiwHPXqWn501szWD5kYkL4r1b2CN4L  
I73o78HO37yTcm0VrF/AncFEptULFl1sNlMtx0agB9Veub/Xnl09kcbHxuDhBWXWYip9qNe5  
uXjTbnr5VhvFjR5jKK3iDVj448KMj5KcWJ7FS3a6fkzWz68eXrdZ1uzjP070B+djoyMepGRz  
4udt3Ej/eofz5M5sM4aJSP2UpvuqAVDFFAW151nZ2hsXF5iZZaaDAj1X5rCQDy5L87q2AcTT  
mbHiNEi1h1xu08i3cdz6X7XTOSnVn/LRq5q4ftoLdjpN4urfqlKwut7ifr/sOxB9k0arHfw4  
XlXrNETx6yqK9aLPbTN2F3y2pn8czlxSu8noN3k2XUDAPQccSLYGFp1dlpr6MSYZVCdkepcq  
osTtxsLE1PINK8AatfIvvg0x15KtQgwSfNYZP7s1c9fPSeED2b7DPFrrFFeKgVu02a5TUfl  
5q09Y3JtFC21+CBeVUH+MEFTFoYrha7luoMIVMYJXG0UzhUhsA3KLWPX61fPnvYUiv1814xH  
Kn4deJVqFdi7/W4LPESRK89qlJZyML+ypvII30cpvcT+SHJcnRxxjCYnXHf1o1c+yHPBnwTg  
bhleiJWY9kfH5ahnurtOtn/N9h8ODSkRc/lxu8xO4qJT6iPVGUBWHKoaov5fqQqD+1p4ipP90  
KyHhx+6lDEEepnuvES56HXiVaBTKZC8/drrrAb+3CTLfe7vYaCC9ZolXseonNwfos3qHa9uz/  
ZDGSxumM/LWln/LBIScZ6nM3TvVVOtGYPasb9PNp3IO7NouUL+O25qJT6iAOMoqmKQxXD13J



9e1xAY/AjgUvyGpCR/Zbglp+yPj5BrGoCb9YHA5ZoCbXwvD9z331/n2MLG1TekIbYcjGfF/F  
2xqeo/LzIrjebYz8HNXPsd23uluoonu5NEPEQEGtx9pfdT0XcU/iCLVj3+c7iolFrRobiLCo  
v5XoDi307vJXh63ZkMv7CvsIJovqrwu3j+gjnYQo8LkTtvn+YOYZmLpw9MbSs4RW8nfEptn7  
UWEUV14pns+xBx36Oa+akH2o1xp/NK3jzPAK3e1eF+3m07yjcm0WrbH4OF5VeP2VpECPoUiu  
crxEXdxcddNYjDho7R1xtD/Jgpot15GLBES/AXZrL+8sDL0KBctMGaz5RxoFzcJCCwXezjRbw  
d+xQ3PTt+CKI9gbeZykp1LT83JXdq5q4f6jAaLOBFvCnWHnW80Co/D/UdiHuza00C/gS61VI  
o9A1212K51/7XovRiHB+KuNz7+YhYySpEoWE94wu+pg6U1CP09bqCXhViBerHFHAtIppFG3D  
P71+0vgtj4UElWoVmB75zQcn+zMflsR/aKOkVNRseV6IlaIIQfg6cELvGrR8CZWBnppQU8sUR  
L0BzBhUDY879XRLNjP7t9Z8PjSkRc/lxu8xNIps3STB/prZtU10d66809WZQ2R5NKW6SZPtJ  
bN6muj/TWh3uzKG20JpW2SDN9pLduU10f6a0P92ZRkiRjkiQjKcDrdcqWKB1EKm2RZvpIb92  
kuj7SWx/uzAK00ZpU2iLN9JHeukl1faS3PtyBRWlZnKnm0RZrpI711k+r6SG99uDeL0uZoUmm  
LNNNHeusmlfWR3vpwbxZNSBm/Diz06/+e9A8EiN/FBvJ/f//fs6tBRPU7y68J71miC/yx6v9  
dupIhZs7jpsXKjZteZba/zB5Em+3nVwDfnWXVYDuUaBW7fvzD0g4+Tcc+K2D30wBiXZfrySx  
7kNj15p/x0vqEhJbbk86Hw+NKNJ+L69QVVc3ovvjI4B1KRFz+nGYzCsurixHtMvFDRp4Ris  
wThcNEenuLqsRXhBeskS9qG5ef6aPct3MeagWPx1RGUuNwOPdn2aQr3xVhELJGGQ2K73BTU/  
thylHknXaKk4aRzNe25f/U9bvSnVf1vbqKqas/c2diJbbk85nsMwbc7yyTgEzJTK0hUKMsAb  
3ZtE0m9Wi0rQVC25t64ZmYVMj+PeAuMpcZ/i34bpsXqm1IH81182cp1JHUUXZeAmR365S+np  
10qYqVLRAsGw80hvs+mExKgPbtRyQD/P0DUVayxppqJSvV7XqrNivacKtcfNnye9L5DJY9K6p  
FIOF80gW6V4Ydym/ZIhXuzakZNPk/46uqCLxodOtAEGhlehuV2nGvh5ixV/KGJeqCWWtPZ5z  
1u9JULpq5Akqj/IhDLMWp+nTI+LGqqovulB+ZGZXTYrG3u34wsE2CRJWLn5e37W7pfidWqrv  
rDWizmw9ablV5SSwuuQA/59cpUFS8KkQvZ33VuTeLJttEE4/AF0uupG4ov6sm4C5t/ExlnOq  
FLYHK5m5Ed+UrcMkttQF/ykUYrURdUEX6G2tkoGciHjnbVKx61LTDa+YIa8GfFO0FQp3FW3  
AL+muBrqsvq3H0L5Ea+E1fWoo4k2YEX1VQoUmG8UPrbrkq7rw1cecymJvx35a0AuUR4Q3kA  
D81WUFvNZ+ay7dQX44W652NDqeDDgJFZ6C5jmQ+sUKC2+qrS2bRht2SIV7s2i+TYpLF97gsn  
zaDkSqCQj1F6L06mU0UyHCvj3Xx8q/TNu92cslfG4H1EdwmJiYCEkCdWYbWSgAQufcc0SdI  
g2kA00+NiHPK3+18N5E0N41tGGzU+jz93JbyqawQyvEzrWy+lsWhVMJp4JGdUTov131p+mDV  
5iPjWvGwgCkYcQx3xu6o7riuEhIQtlbfHzifB40q0kEfXqaniq3YkOp7pOxD3ZtFkm5o84Ku  
kbjB5krFlB+5Fm517p84iw9ftkuZrKx9BuXGDCo7HjYKnlOhxeHlfh0zEX49bflfsJoG+4Zw  
GM7yNa57La4efCCoYJJ61+9BjGLZEa2HKVaUpw/sQ1NRPVC2BmmEptLjS2fACJVRfSrs8mzi1  
s10tvqCtvL4eMU+VVbNV3ZBKL1R17201WtNu28pNY7E3IXrm4ceyzj5W5PWWgXCzBvVk02ab  
GZ/4EvnGjg8x2u3cv81sVFhkZr5ZuNG71iRmZGPyoiMeNgvFL9Djxhk5UHnfjzYPdJCAH9Po  
0PQ+Y4RE0kCi+RxGUFkbsmwQdC56+JVoIrlp9i7mMl8dGdXcLr13Vj7MrAiW8pURL00OH/LY  
AVB714ru0CKolGVVUrmeyUt2xNy2xcnHlrlto5WewuOTg+jqlse9sQJe7vcbi3iyaaZMzxh6  
EJlAseDRUledetN1XhRXjINS7qzGPa+V1CZQ7GX0//PgZBYOX6EG2daaXj9LzfdvWFLC029H  
zLsdjgtZ6ZhxGg3L9CN1mrsBky75Q1QP8BvU50qeVn8FKbyf9MPdo43hftKtONAgjK98acwb  
LHnTsDRXnD8iW21Z+Bsu8Bbd12r9OgWKR/GAMysUS/CUtmmBTHQky/jh5BG/mXtTAq5NbZOK  
bQawTF0hGx908T4mzgUCDxF7p3+CL8PQSPQjvXBkd3pABtUNx16ePGTK7SQJG03zDs2a6PWV  
lmK/E6usP1aUC0OM8cx46lmghzDcmIshEqaAi4i28MI3LxU8wcNBxLiU93fVDqbTmHu1VW15  
C0YUgKna2K9W1vBH7fFFRFaTTcnvgfAbr1+qVdQr42W5KdDnuNRz3ZtEEem0yMYUWIK9c3qpo  
r2RuxqVX5AJtqiEp282/IS9byHA8xZiQZTRldDqFvNL0GtOQor0NORFXtJmNqKCWpKSSD0Sw  
eEYEMxyXo6A0Y6mCbOICHsrQK3rwqDKF5gd+Ny/BZFzTsShfAm5RoFZo+xBwpv5La/IxFx1V  
E5WzLRnmv8NnwuBItQROEEOLrSIQ9YjVruT1wPhueWki1XFynUPlhgyo3bvTtVx3wrBIRlz+  
X2/we315p94LPYusjvXWT6vpIb324N4vS5mjeWyl/vTv4G94xWWx9pLduU10f6a0P92ZR2hz  
NWYqNfwJz5Z9cvaWZBaS3blJdH+mtD/dmUZIkSZIkIyigna3XKligZRCptkWB6SG/dpLo+01s  
f7s2itDmaVNoizfSR3rpJdX2ktz7cm0VpzcSptEwa6SO9dZPq+khvfbg3i9LmaFJpizTTR3r  
rJtX1kd76cG8Wpc3RdCv94/t/mBOM+1X0P2+Uiw3674e7fz/nPDylREtgyjeL/1BumOqDWbd  
+5/1M37HwrBit4Yy3g8rc9XYw51SWPUjs+nEhry9HaLU503cGPK5ES2jVvXwuBx5KasNuvUE  
rPxWeVSLi8udym5/AFaVs3HTXclJ8cDr+O1YwGw79FGTJhUmWtD5+iDa8Uuuw3F2ZJ/u05UW  
8EXgsD1u23g7GnM1KdTc9tR/mHknMUD+KnVabM30n8XRvQBB53oeyUVyxu06hlZ+Ke7Nooc0  
P4YpSNi06s7p0SZH55W9nZbFVa5IVq42ev09W5Hc3L5ptT9CTfYfzCt6A14iY/O4ulvJWohs  
+5mxWqtV144sXD7vv02pzpu8knu4NyEflkI8T16HBbi218rNxbxYttPkhXFFKZda9Vhe15aX  
2231isXEcyipK/WjU5uXbmdDPN9Wtk32Hw1NKtJzwpjh2QFSAYmfXW4WPOZuV6u76YXXvSnN  
abc70HcjTvWlXkQnVZdCqtzN1OAOeWCLi8udTF/C7ckVpdcpSfHFJLVJ/XILOXd0F/ykv2kB  
J/axj9eJrZFTBGhN43K33F6r4SNImylf5h+jrdR29tmL5CcOaelwGyqhxiDrZdzg8pURrcW+  
g4kQC+XBSsevNqcaczUp1x34OpAWtNm6juXp3rhk11IM3KLNyXUKd+twEu7Noict4DfmilI  
/ZVVY2pIoOGK4tfqC0qGx4jhEvTQjqb6Ko471IGVuA5dt1kct+wswppJCzQjIEzwEvUq0F14



1XGkukgAufBdJcDnn+46Cp5RoLe5NkOf1zhyT7s3ZjjmVxepafqgZ8hDLc0urzZm+w+FxJVr  
C1hsZn6/K6eQ6dVr5Sbg3i560gN+YK0q1cQdeZ8RUXrm4PcVRS4LdeuKuzobdmvMkzdrSqPo  
JdMBEx+plTqKh1sPb+vrUfAOpOwAhMdlH+w6BB5VoLZU3UObkxN1bsB1zKovVHfvhFOHW8Q8  
ZrTZn+g7k6d6oHHeodXdcObv1Bq38DnybRU9awG/MFAU6ZXEliZqLW1FFK6GujIPQoW+cnVq  
uEB1VsDpyN/IQ71MN7i9zHkYo0UJ4ZyZVLjBwSnd3LhrsTvZM3yG8iDcuY74UwIFVsfV2/L2  
YwUp1Z/yQv1sZrTZn+o7i6d4oFX8HGtx9JdrsrtNwfgb+khYttPkhXFGqYUJIK9cG9eelQrM  
48wKSUA8Off3sBC5prAFV0BqNEqfXrckXUdy/95T1PVuvioTdrbCCZ1vVJ/sO4RW8VQWgmlH  
cYuutGnMBy9Sd9EPNVE62tNqc6TuKV/DmecqGclLcYnedQis/A/dm0Sqbn8MvpedPWUrHH6Q  
SVDK6Kw1xylJtkSTQgKpsFaLiaMPCVptqMfyiU5ZX3fXJy3CrXLThOxI2gpN9R/Ei3sjEDxZ  
89yPeZddb63sxj5Xq7vrBye0aLU503cgr+AtYm0+x5WzW2/Qyk/CvVn0jAX83nQrpdroK6r  
KKNmfeUqwZ038o0FJfSdjWG6RiQaqYB2rQu09o6PUM1RtiTYveRe6lGgVvHksWhFzYSI19Q3  
JrRof40HfqfDEEq1i601o+uB347LlTbTgnApvUqIlaO4QM2UF1dTGCRk1a7U56DsbnlIiJWi  
OUFUIs1ae4ikp89aqt+M6nApPLBFx+XO5zU8g1bZIM32kt25SXR/prQ/3Z1HaHE0qbZFm+kh  
v3aS6PtJbH+7NorQ5mlTaIs30kd66SXV9pLc+3JtFSZIkSZKMoJys1S1bomQQqbRFmukjvXW  
T6vpIb324N4vS5mhSaYs000d66ybV9ZHe+nBvFqXN0aTSFmmmj/TWTarrI7314d4sSpujSaU  
t0kwf6a2bVNdHeuvDvVmUNkczQ6n/fnow/BeuGVMfQwEM/sfojzdi/BIt4U/71A5RbtgndRx  
8vkTrd9vP9B0LzyrRKvyTTERq0KfT8hY17J8wMBseV6I17HqDM3Nv1VvrzKnuwBLN56Cuzny  
+xMF65Fb1MRez4TVKRFz+fMYCfnvmKWXkOAW1bgcuPEaL8RmWmBrVrVEwZonm89//tdXHcs  
11hvz0jTlyW2tw3LX7cm+Y1npDRAVrghivi2ffbeGCTaM511B+1KdV/WNt7gzNxbdxUbcmf  
M2SzzdlxXd/e31jfg4RJ8tAXwxBIRlz/XVUGHME8pI/syo8KgXIzAx6e+xw4OK4ut2suY11Y  
gez8X+d1p0swXvzjZdziLFymPi7kzR+10LZ8Vu96AMWME+i7b+1aq2/UGd+d+UFetMWezzNt  
BXbVqKtjwJLc9rNKEe7No7QL+BOYpZwQqqVzkKfsIzIWfdgmpqchC5a10y2Fxxq9unew7HJ5  
SoiUwx9jNmTIodsJnxa43iSoXm8upLHsQ7Ho7M3daturqzPdiBiu9OV5Xu7XkHHgTX2dsnrJ  
vyTy1jBzLTCVIYelSUHMkvRmoJUTBqRyFVYGX0fGdTLnmG9NkgrxGLMXtyhTKqLFvc2f6Doe  
nlGgJ/AWCJzJZJhhzd9xnRcvb8W44j5Xqdr2dmbt07dbV3e/FJHhoiRbCHL2udmvJofAmGK1  
VqJNwbxY9aen7Y+YpZeRgW3YsZ1UYX2mgf9hCkamlJ21JgRkoLv1KDMTKA73e5pR1Upo4yEP  
Y0y7mK7PCFT3adxQ8pUSr0P7V2qHcZwv3RvtPOGVh6+3M3I/r6vh7MQmeWKKfOrKa8k59gZ  
Ie6I3i5608b0x85QyMoVIwLqtqkeF6FB8SpYWG1SUEOVLrPGB7m9zyjIRX3uVq7vTREVoeBT  
vEHhQiVYhY60JVj5bhDe++jhyeGaE6yxWt/V2cu4HdbUdcwE8rkQL0UzLxU98DTrH6zFP2bd  
lnLJGVqlp1XnZcWRu60nnaLkwlKdwNQ6vqryP+TanLLM7WGM8EjBKRQMa7K7wM32HsNgbumJ  
e1EB179inE960FSoJ5JfNaKW6XW8dc6dBjHP8vZjHSm/iuK6QsLsGHfcmGHCZMeHeLFpu8+2  
Zp5SRo9S0XC1NXVaLWShZ/XjoJ+snnLLMKCZVwRzPLEKahaLgZN8hrPRW1YRKSL48Fnh3nx  
MunNL8WYwqTvw9tDcva7ufi/msexBwa2smnXlbtSLewvIbJNTcW8WLBf59sxTysheiBQWmda  
PulrMJP2wJKmlq17VUe3j09I7DoHxS7QQZsGUy4XBy5yZIIpk0jnZdxSLvTG1KCTqodqnWj4  
rKm8xjsrvzAhDWKmu5e383Ld1dfy9mMfikoODutpdg87Wm0DXMmPCvVn0ji3vvZmhlAXGsEH  
J3p4ldFKWixtqANRfSX0nKVldrQdJvVcsyO/WbjcMWKJVokVazvoLATD1kvqGpBpHG/DlfdB  
3KjyxRKvQNGFrb7ttrBOWNxFFRbOSmg+PK9ESNEGoLO3OPZod15Vuwdb8PHhciZawrauDWiJ  
z15t+oAme81OdRcsX8NuTslukmT7SWzepro/01od7syhtjiaVtkgzfaS3blJdH+mtD/dmUdo  
ctSptkWB6SG/dpLo+01sf7s2iJEmSJE1GUE7W6pQtUTKIVNoizfSR3rpJdX2ktz7cm0VpczS  
ptEWA6S09dZPq+khvfbg3i9LmaFJpizTTR3rrJtX1kd76cG8Wpc3RpNIWaaaP9NZNqusjvfX  
h3ixKm6PpVuofDeH4L2Wrfj9u+y7qCLT866+/lv1S9jG8T4mW8Of3//IVKdFskzoOP1+i9Xv  
xZ/qOhWeVaBUxR5+4/6Z/qwIvOh+Ov8ACur0Jurc+emLlBz8Br1qi+ZypmeqDKSoOvNFRI5T  
ryfiDLFPbhZ/ARaVUTFVScRnleHetV10+8Jrlyr6bYyzWIX7Yswi0/bU2/d2FfbLwWFZ6A0S  
FKwLNF5i7Au1c2yK87nw4K9XdtPV4A5LcghjBkbTdw5PgcSWazEHNEHgcPp0Db5Iw8tFAE0t  
EXP5cvoA/gYtKKYvd/V2oqnYXqlADKqxc39a2Xz6R1cVWKKWJajGT94Vhfncd0mx7EpzsO5z  
Fi5THxdyZo6qxsKgb7a530fkMVqrr9hbcjPwdo5ShYPfWJZ5a9UM7PrcZetNR+xB10m4N4v  
WLuBP4KJNSqAoDoqsOiApSsavStN5Vnm4YnFhlWZrLz4+dz+FMKiJV/dotl30Dy1REvWcmT  
KoNihzW7eedT5Dfaqu+5te1oAmwDGGH17ax6LSy6ImlF816fYettmluDeLHrexveuXFTqtUV  
QbUbaofharvegF212m6m78Ls19fn45pLKV16vwbvp8qDiD6BjIdbCTGPJaUZhVUK2074yaux  
L/Uzf4fCUEi2BbZ0nMlkm6PXGUaleP1s6nM9gpbr3m5Hw4+zAUsaiq/Vrams9BZ4zCaZn+K  
m7Z+OSFNHvopyYz7+LlueYf09uag0TjJrBUBaoQ4WqlCBit3djarlQcqt1ICqThX0V8+f/1a  
DWH3VjEcof561WgtrLd4W58jSJTmtYsLf77TvKhhKiVahHcp3Lgcbx7seXHE+kMXqLnq7HRY  
/+kZ7gtawM1jsTXjNiGoFwU3bP21UYKHuzAijcG8WPWnje2MuKuUw86VVbUa+6d+F7jS09pQ  
aVatbAZ1tgeqhbBFH0eq4TL3keepVoLfhZgajm4j9J7MJMw9ujfYfAg0q0ChlRtFDMrDVCuXi  
SN+BZJvRCRW9fZ8XPxaiVCxSh35rNYm+iqhk49hlU3lit3t73tNm4N4uetPG9MReVUH9xgFE  
fvXFot4q1B1wGNAY/SrkkRwEzeXvKVqvXxyeIU5bAm/XBgCVaCHM5eHOfywsabL3Bmb5DWOW

NXTEvaqayR8FQVOWiwXXnolip7ro3ungvYt6/wtf+PHhQiVaxrRkuD3w63PK71SnLyEznrvw  
huDeLnrHxvTcXlVIfccpuUcUcrDSKqSrHOEf5uh2ZjL+wVyRBVLnyiru5PkIHLNdMxLCzDE  
02wo/2XcIK73xredxMd/q+47JMwfkdeejWKZuidfMtOS8q7egqpljnxWVNzXWJhaXimfjD7J  
ouc2356LS66csDWIEXWqF8zVioKa5qwZRo3SMvt4Yqh8no915GLBEC+GlmWO5MHgZ/4G3BQa  
2Mz3ZdxSLvfk3miLxGBQDWNbFwkXnAlmp7ro3ukSvCnq1bs1gccnBtmZaPrd8Wft513URuP+  
puDeLnrHxvTfdSnUKBtulSOmUe+1/LUovxvGhiMu9n4+IyqNLsX0P6xkv0JK6UVKP0NfrCvw  
4Ui0//ZgCrkWQVONoA+75o09UeGKJVqFpQtjz2hNhhtglP+R8Njy0REvQNOFRb77iYHftV1a  
nwjuUaAnbmhGyAX7XLw+80UbJZUcs8LgSEZc/19v8BFJpizTTR3rrJtX1kd76cG8Wpc3RpNI  
WaaaP9NZNqusjvfXh3ixKm6NJpS3STB/prZtU10d66809WZQkSZIkyQjKyeqnbJIKSZIkY81  
TNkmSJEImkadskiRJKswiT9kkSZIkmcP//vf/z6tTVzVmuAUAASUVORK5CYII=)

#Random State = 42, Balanced Dataset

! [hasil2.png] (data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAANyAAADXCIAIAAAA  
Y6YvMAAAAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQ  
AADvrSURBVHhe7Z29Tt3K9/7D/zKi6AgRriFFBCKocih/RQpIleorhToKTUqaRkLBoqLKocg  
FAMUpSERxriGJogjlnVh/8DOsLPy2vb1ts7dZn2KzZnlm7Hm8ZsZv2EtXV1cPgiAigiDomv+  
X/gZBEARB0CnpLHZpaUnpIAiCIAhmws4P/5li44pxt4SkVYQyQRCMGD/ExYXiIAiCioiFmGK  
DIAiCoBdiig2CIAiCXuh9illaWjo+Pk4Jx+rq6s7OTkrMTFVtHz58YFFKzCubm5uovKSdpqh  
N2R8/fqR0W75+/Uo9/KZ0EHRKVdhP1WG7inb6mtZeJOUIGlb0O8Vqcj060lLSoBd9//49Jbr  
g27dv+/v7KXED8+vu7m5KzDEnJydfvzB4Pfk0WLgQPDt7e2UmI21tTW2gd+UDoJOoc8yCBB  
jueCfamTY2tqiyMrKSkq3hb6mDdDa//nnn2xbrh4/fjz7/B3cZ/qdYt+9e/f69evT09NcmNK  
7iN2U6BqmGU3tb9+++ff/+vZyLFCFNvshrDiMPokBLTw0FJnLYGw/C///2vdGq8kz67sbFRejs  
5t7eXrGbY4BMEoscpLsGas9U3b95gf/78Wc4B6Oo07m5htkvWUHAYtBAN/cE44Ag4Wbep8vd  
K1Uo5Zp3qFHkcg0/QIT1OsZ8+feIslgD9+++Dw8Pk7eCdN/jBrsNaTds/B0anGTag1/HjBj  
MSUwSGCQJdB1CfrBqdY/W/JrPZPsV3RX+VFJbBwqpbhqhgNmieOyMR4toprXUsikpSLJGXVd  
YX1+XAipOwevcN7dmhVUIJ3m0L+ZBunAcKPzomPwq2v29W+sgymC279TW2T1WbUo3RgVfcmU  
3vEiyiHVpvThzg09w39EtBz064vv37wzZsnXpMnejkaWvX7/O2boR8v79e/mZm80mD8h5vd0  
PHuhGDihpObHtVoouOqmgNsOc5rckdu5OzCyowobdPTHmgPYphVISRWRCcxVW0DN1N/nAWw  
1jFzIaE6tS21X5aoEfd3XuW8W8SunIJuKq/56yJasIliJLgWesPCzXiCPbeJX0etJrtm+syT  
6p6I0jElaB7S+XhqVLdebcMWdewhkhCzs9KfrUY/4g5TI6rcYFYSpeXxcEqzy+2EdfOhbHoN  
FtjpfG06be1Shel2VH3zxWaCeZDUgN8pgePXNttabMsmUIOyloGnt+bUNw0/HtC61LTKfT2  
swm8P4mt19fvXQbZkBUeh+IXCLyUcBB5+m9IIS8tW06mbo16QC2N1MQ/Z/KoNPE26QDBufGD  
0daH48PBwd3f3+qpKBp6DgwMtKsKQbU8dn56ePnv2DOPy81IeoTsiv379UnLc5J68oO2rq6u  
6PPv582dbqktn7969U29vj5qPX/+nAjQmFXpt2/fkpWxvLyskSgIBkNXZc/OzhgukmsoiP/  
ceQBdUrdvr0e3sts0QSB6mWlJOA3fhg4qqwJxf3+fmVXBSv/Z2tpKC27f1YS//vorWwoHDkz  
b7abRq1evmEplCxbR89G2xf/VsHeYlXMTZz3//vtvsjKaTMxB0BU72bMXRptUDx91Bau+uLh  
ICUc2t10fpG5vb+dGqiAQvUyxBJweJDaYBv7++++qx+04G7NLovbvrUy0x076+rqSTM8k/ex  
bd/ntUZ0Fhba/ePFCNg1HIoQyD9ipZG7+y/H7929+NVuzC3Sg8/Pnt36RqHg+WnwYZG9vj2M  
g8x8eHk77zwxBMCN2RmiJrIzB4AB3d3fXP2NFx+FFXYkNyx1xjmDwCTpDE5sZM+LHa39Pwt/  
MwG8RqVuquQAFuyxjizByHrsfoyrwZkZS66JmkB90a0f83//9X7Ju+72tqmaBSpI1ieKdHmE  
NFDQn50n5ssbK8E3QUYtVrovJZMBp2RDT5MIvbc1ZrMd7cpWAtydKR55kbfceH/wW3opVofD  
za4uinVj12ao6e+kt3qoQ9ZXYgCP8ulScX8tvFfreFNxbiIFkXV3NxcfsOB7M/V9a0bNw3K2  
k80woEwTbiPFDXI//F9sQZtPcdU48T58+TYkgCIIGWEzufeor1D+YJki0e4QmCIAiCuWlLhS  
PkpC0ilAmCIIR44e4uz+LDYIgiCIJRElNsEARBEPRCTLFBEARB0AsxxQZBEARBL8QUGYT0jrq  
UCIIGCGYmniJui5C0ilAmCIIRE08U31P8dwWCIAiCvokp9h7x6tWrZAVBEAT9E1psgnF8fLy  
UvQJLX9JfZ2cHWpzQhy3KNvgr9IUQj0/fv+sJVixVPZzU8ru5uUkGfmUIbBbZ0iAIgmBaYop  
dJgXZr6r7LMhBwch2GdnZ7r0r+vAZKD2vV7yrK+v48HWx0kw/vvvP/uwIMmTtkxPm0dP  
TU3mA5MrKcOUy1fHro11BEATBVMQUu0hsbW3pa1xMfvw+efKE3y9fvug71foUHxz+/JmpMTs  
FXdJUyppyqRcLqsc8ZMcvaZ8U462Va1Yd7NdE2/0xvEARBYMQU0J+/vxph7AUU82R15eXyQq  
CIAhmIKbYEBk8vJz7PuBUPhr0iN8fP34oQRBELQjptgR8uLfi9PTU/v/nK9fv/qbqSQhJcp  
YW1t7/Pjx8+fPUzp7K0WygIAiguboQqIZQVf0Ien79+++znXaNbqYKb79+/ZqcX758SekHD/7  
++28VZ5GSvp7v37+zyG7EWma7swvydEXnFQZBEMwPfohLL6FYihfude1IwUoEwTbiPFDXfw  
oDoIgiCIJe+HMWq3QOBEEQBLNgZ7F/pti4dtctIwKVoUw7QrfWhHTtCN3a4XWLC8VBEARB0As



xxQZBEARBL8QUGrBEAS9EFNsMHL8R4dyr6yyrwmldAVkW1ldTYkb9KkiGos3ErXuTY20U0  
ZIFd29LqpdQZapQUNdPOa594PM3rdDH03bCrDMDTJddW71+0qw4ygK0LSKOZU5vv376zuy5c  
v2HpBh161AXoFB4Ze0yFnDhWHx48fJ1cGVZkHg6TsXqnaYJ5gdfama2xpCE10M3H0khPT505  
0g8Gko0UWYEDzprW00E02jF43QbtYo16eYzTrZSCDCQXzoFuyBpbyPhCSVjGkMnR032Pprv7  
tV3KcdeNSKOL7Lfju6rtxrwypm/TxhyNqbxPdy00nGcSxXXAnusFg0tH2ZGWgj6RoopsXDdv  
rP3rdgHaxOmumaKKbQZix1IszD7rFheJgzOQ+h7CxsXF2doZxcXFBd5UT8Df/cMKPHz8Y+/Q  
lQcAgmbsEveg8fPiQX/8NRH0coolua2tr+rqisAt390E32p6srL3fVn2TFE1086L9/v2bCUO  
e+6AbbG9vM0HmvgNwVj9+/fr12bNnKZEXJ7rFFBuMHM2pOeiofKQDRkMZS0tLxTuvHk08ueK  
MickaBbSOIZ5Rj5Hrw4cPr1690uTRTjeNffdBn8/nz5//97//yZ5KN6aBvb09fbAZ7oNudqP  
0+q6puxHbXDcUy03Pc6JbTLHBmGFu4NDVovDPnz911HB1dWXD+D7DEM8JxPr6OnaT7w2X6qa  
Thvv5SX+mhxvcvXqREntdOKB5/Pjx6ekpU0hy3QPOz89pNQZqfPnyxX8orAqv2870zsnJiex  
5I6bYYMwWuL9//56zsezgeOng4MB/pC+oQbPjP//8s7u72/prhhS0s7F7BeoxAerOoZrw9u1  
bJo/X2UexJk4zowG56Ji0HXttbY3mHx4eatFEipeI54qYYoORozEL90jEmzdV+N3Y2NAUIji  
7rb847NHdHSuuS0+6eTkmOKtgduQY5fv37xyaaLifSjeKvHv3zqaZe6Kb8FeJYdp40yUE2fd  
KN7G8vCyjiW6fPn2yY2gOBwlXjOPj4znRLabY4L7w6tUrzmg14j99+vT09FR+oB+yNCUmQQ1  
MP/Yo0OXlJaNhi/OVeUaX1tUoftht3+wXk+a6qQb/+M990M1ALrRKiVbXhjJ//fWxjNhrx1S  
ae2ZCU2kt3Tgc0TE02BPFHBrOi27aMjOCrghJqxheGZ2/vq74f7uJT/NTMJfBF6Fm6pfdK0P  
qxqkAq5M+YFp5u0Y3FnmlyS+J7kQ3GFI6QL2iMs3jDVG57kPutFAizHfxql0I+e86ZasIaW  
8J4SkVQypDF201QGjXnI56H5amtIZJH0vVQbhu6jVTDDOrp5hXckaBBqrBoKNfaJeN1PGYJR  
UHHheN2BlyRoE2ggp4ajXDUG0FLxi4j7opgZCbi6s182DSjnnneuWvrmzFB8t6pqqTtIpQph2  
hW2tCunaEbu3wusW92CAIgiDohZhigyAIgqAXYooNgiAIgl5I14yX7tObRIIGCIKGP+xe7J8  
pNm5rd0tIwUo047QrTUhXTtCt3Z43eJCcRAEQRD0QkyxQRAEQdALMcUGQRAEQS9MN8Xu7Ow  
slfH169eUY0ooWF+cNTZ/P3sVWksp9pmz+YS235+vbffQHmmp3574Yg7by10RgVvKfVL1Fo6G  
07b7h9o71IjX6UK19EHSUzB4zRGapRJR8ZQeF+OMt6sMMYail5f611HhP7VbFN37ND2ytvTG  
V7bRNvX7LWdl7zjqhuaRV6A12xZeELTqzKzMVfo9j2B7HMD+bVPr+0qxosVnIvff4AOZBN7q  
/OYuvbvaU6mMvZbRKhmfI6a5Vmy1mNMZaJULOGueDO9dtBPHWwYXib9++0bwmH9G9Q/QJsxz  
6POHcsre3h7AEGQdoyRVMD5FpX+fY2Niwr7Lv7u6+e/dONv0ZtWV7qsqyR+b5E5WdUNV2+/b  
ly5cvNe4XqdJnbW2NQYfxLqXHyOwx8/HjRzp+SmRw9sbsgrP40fvRMNZ46+ZerL663PwjugO  
jDxulxG3mdpalU9GdJOynt5/kDFrA2HR+fP4SN5+i1FGLfWot980so7QsMB/reU51V64WndK  
2+350eXnJoULK3OY+6FPFjDFzfHzMXJISn+zs7DBPaDQYK6ONT+xcdoOLAJqOKH1QEckLMX  
2xxr++ltyuQ93UA9JlQItpYid9VMtyAZ/cJdcmZM8dk3AKi81V6Fg79r00wbbinxmy0P+5Kq  
GbMlqBeuSyNoSOQ2vsO2Lo102BKEh2NacXJNNPShtMvikKaCkRGsI+ZM1CIoxtpyNtLgi6Xe  
imm9xaJSWBeWUGt7fK/Ogm1HqNOr1IcB8jA3AkNLNEjPk0SJ+TSKcqpBfIf8ADLmuscZbspp  
Lqf2NHCl9Aw3Dz1KwwUuq4cG2UuaUIRs/NaAjBnJIJgyWmjo45QctkjOrI2WtYlmWcsimnAZ  
bqxpSOqvTGshSrZSaQc5iJUXqN2MiVhXNsG17ALnWScTrb6kT2H7bbBqizS422bJpdVYP/mz  
5Hx2UQbsMyKPMzfHrHQYJ4neZbxrkGuUplvV4ufpmHnQT1um8hqWU6kOFVXr2BNUqrEfoHTP  
qYjKsuMZJW1RTc+fMiW4LHW/J8q56NJT7EV/QBvws1UweMuMsbaSiR0MbOhYVzMS5Lqj1+kG  
QJNViFMNRdilWoYf1qipBDR5tFb8pfYMYVzExQw20wnoUsGq/zSzyS0WpE3zTyGD15JospB4  
oRs3IQSW2rtKV1k01yRoKGqv4wZCHtpsN6pk+uoxi2RwoUFSyD+ZBNw9hwCJiJqUrKOpDQUi  
JQRhYunYxg5LW3chgEuViVZ20NFY7Z650QxAWLVy8dfZ/saenp+iysrLy8+fPXau3trbs3nU  
Nb968Qd+1DN0q8/z+/TtZN7C6JtVoi67mp03P0CMGbbSkv0aWvReOjo40Dg4kBBBe5LxBDKX  
3G2a8CXF8fMyKzs/PWZc8NRW+e/eOzSND8yc47pDNzc29vt0iU7tmj+8/ffrUWgqX15f8Fm/  
Y15bNMdbnnia2/eTkhD6YEtWMVZ8qWscMvX59fV1dnv5F18cojoSPHj1K1rgYa7x1M8Xq/5A  
QiN/15eV///03c/8B58Rn4SSu9CXUSof4//77L1kZ9ihBh7AZ/BYjm72bW3tPaNWSQmgyp+Pz  
5c7b8eguL4pQ6G0LB7e1tjqD98xTSQXNPjrw1NY4KP378+OntJ46fkncuoWkMVTYqMYrpsSY  
96GR7+devXxz8yjaqyubguIQJOyXGQsO2r66uThzxr61PFbPEDFNI6vDuLJZa1eEgNad8Geq  
eo2HE8dbBFEuzOeYiJjTavnjxAnXsH3GyXThDwkU2H8EyybnYKnNLvJ4NKwzEYjUyEmFPT0  
PTFuY41Pi5gDif//7n63dnH3w6tUr+2cSQXdik3Z3d7X21y9feoVREjvKnbJ1rk9ZHRqXHla  
DZ1N/GMFkQ5qsE9mUmGOQjmMjHfzBxcUFUST7/c0/6tBAc3+W1dNWUNhbE8mj4au2j5Wfar  
oI2bISU+03kflhK7s0TDmeMuOma6nNBn1sKdVKgezXcqRoTtbAqXk1KmYkNPu/AFLiRuTVRF  
cLWmVmAdyHvaQj7zcJoHfKrDbHil9+1E0vyJ/gyS5GtwSIE+yypkGVg7/Y7tcrv2+LbXap00R  
GH5xSUH6wbLYKM9RAS4LpIKiqKHITqCpZQ6HtBzXfsNb5puWyKQOYs1TnAWBlyRoKtRGs7b6  
L5dqOR9lq9PGDALSLnxawrmQNgloHTWIGj483QZ6ck6SK+5Ghb1hdsgZBDQRr+wjiLX1zZyk  
+WtQ1o5eUw+p2/6gXwda00K01IV07Qrd2eN06e9wpuFccHx/ft8dYgiAIpiWm2GAKfvz4wQE  
anJ+fz/mDTkEQBHdOOp910FQ6CIIgCIJZsAvFf6bYuObeLSFpFaFMO0K31oR07Qjd2uFliwv  
FQRAEQdALMcUGQRAEQS/EFBsEQRAEvRBT7Nyxurq6dJuu3iS1mZESBfs0sF6PMiZosmQ00oI

HDz58+CCPvQmriO2OnDJNyi40VbrV6Omp0s38/b0f7c6paSOLavqgQUHVkNI3EHVNi8oVbo  
tdj+9yAj6IpZJNURmfQuEtnF978sLkMGGxr693B9+fLFvzjGbDaJRBI9emUYh14TY1U1Kds  
586BbjZ6eKt3IbPkxlGcAhpSuqo1oxWaALS1FiiFgSju0qL54t7C6ZPXptWplujXpa3PVT8H  
rlqwhpbwnzCIpIUJxwkVJosQnF50hgy3Xqehy1v3YDFvqu6LBUj/SkceSE8v2wTzoVqOnUa+  
bh/5yr7vrjyG1q28joVITLZohqmtBD8MEem5gH3FDX97X6eKsv2xNet7hQvABSbGzww/ycXzC  
R3AvBDw8Pnzx5gqGrRrb06dOnxS97XFxcJCvDPnjSpOyiU6Vbld9TpRs2Y9/5+bn80MeXsu6  
cWdq4s7PDNFD6XtLj4+OXL1+mxBgp1W0E/TSm2AVAn8p5+PAhv6urq7m7C3bbxt+ksTyQXF1  
Zu8mhUvyah9wrqE6wFzkBtXkneeyGnPwLAZuNAivZV8DolnRp+Q1NAX6bGzwny44Gr5uny+  
lusHe3t7BwQHhRFmieuqcvZd0trduIYswBHE+rZ0FakC1i+skd34yMUt1G0E9jip13CDgij2N  
bxjJGNIJpd3dXlyDockyQz549U/Lbt2+aLw1TTi/kJMjkVNmsyus6+WUPzfvNOjz++30EIGX  
JT54vX76wRuqUk6XkfpFuHYuw8WclFgC63KtXr1KiAS9evOD348ePSPz+Ovc+UKVblb9Gt62  
trffv3xNojKftviEx/7Ruo74jyexy3W+zzmUHzyG5VrmM1rrNez+13Skj6IpZJP3HfeYPmNj  
SAndjX6QcNygnt2B8LBudzhYwszjndlnnz83hFgjiza0SVvB3MY0RFUND1urgwZgs6WVUNT  
tqSG/oSINy3YOa0nWshjdPFV+KNVNqFTO2TesLlmdUNNGjpuhJW6TiysNATSDYT2OX11VvA/  
uXLdF7KfAipIV92LnGYsGtK+T6za66KE8gpxNroRw+ktH5WR0qeJ/J3KV/PXXX/w2qXlu4TT  
dX9XUDRvZoCpf4jVPhErKZt2G42t+G5YdBzndjCq/KNUNOC3DppT8o/z/k67a+OjRIx1HR0f  
qqnBwchB6eoaqC1Fjols3EFTTmGIXGIVLrrPJOfFqycnJCEHIQbHufySv4+zsLFk3DB+dHXJ  
xcaGnxgr9kl+T7tevX0ghuxQORDgo0ceFpi270OR0M6r8ObxuHKIxpfiZ404eP+mVWdqoKSF  
3IEunUlcVdharIBwNVbqNoZ9qz5kRdMUSkto1opR2PL59bVZBkxJZUr++uJxAwDlfnwxJ4Y  
q1IUUfs22PESt8uioUHKgtzEN8Rs8GGxqTk+2XKOVG1WqtmCpv+IEzct2yJzoJqr8nqJuJKU  
bmIYDMKR09W0kmfN46HG2FKPYuXyGAZgH3Zr3tWK8NS/bLV63ZA0p5T2htaRECWWFTXIieW/  
7iaHkdZMFGLrxmnVsgipZVAIak4Vyu896ure4+8W5zZyIhRj11Cw5dZjPSaCmixI5jShsVr  
kKS3bK6wrWUNRpVupn81ropsWQWnNpChqkjiUIaid4NmQU2y4x/bZsJWHL8CUXffdIPF6qf  
A6pJldZW+ubMUHy3qmpC0ilCmHaFba0K6doRu7fc6xb3YIAiCiOifmGKDIAiCoBfS+Swntko  
HQRAEQTALdqH4zXqbl9y7JSStIpRpR+jWmpCuHaFb07xucaE4CIIGChohptggCIIg6IWYYoM  
gCIKGF2KKXSQ+uE/UGaUvGZ4F6rRKK1L56uqg7MWFJkir4+Pj5Mr4mn2/T9TISPHcm2Z/uC/  
92Rvaxkep3a7oXNvRGwal+MiZo2FmOpFDpgLpvV2X1/nx+q2tgkZmYp2y9XGWYEXdGfPnR  
sb37BIP16yvcr1UBtVr9ejJJ7J9nsUGeyBoHtV3P0eh3/CpiJutmLY3KvmzFNJNEWL45hRck  
ahCrdJrad/CYXeXz81OyLXh1Suqo2VsVSEK02bAtiaH6B4DNSFb/VLWxSczMURYPvG7JG1L  
Ke0J/k1Kz72OEEaREF/j6c0NkJwwZbIxrftvptlVzia1LDL+Q4MuVK53dEf86Bbk7YznCXrZ14  
xu2pf9M1g0k1sYy6WSqHTgc/G9pvy1M1S2X0zZMiVtrFhzMxStg+8bnGH0BgZfxcXycrWHzP  
Z29vb3t5eK1zJnIi+42HQb5M1Iqp0a9J2/0Wmy8tLhjbZNftiNmZexuPj45cvX6bEdeh8fn6  
eEg8eLC8vJ2tElLaxoZ6z1O2bmGIXm83NteLm6OgopTPs9oP/Sh055bR7PP6GWZP7QwtKVb/  
a39/nGFPhtjPe36L3JmtENByPatpOgDHwvX37NqUb17nQzNjGcQJY8UN1HA7qo5NI+vPnTy/  
paKhqYxM9ZynbOzqZNSPoiV4kzfZbonjJiAmDwMLglwy6fvL3338rp3eSkwkGQ1fz+MUGbPm  
BUmST3RXUn6z+UXtNJRpOUvoYueYXya7t1V/co1Rxf/TEHOpW03YWUQQSfprU2ROsKfK9M7G  
NNbEEVhAjl100XA2rK9sFguoliG5vHzCx1O4cVJQs7/RlWYvtAf5JSS6ZABq9c19Ns4SHC5Ew  
5CijywOYY7NfMsZDTpLQ5NNOaXASRczobnYtTAXufrEGYqFuTtqMbZYkxJZvsiz5gXcnqn/o  
21sQSK1kfJCBz2ahHc8ZgogGrS9YglLaxYczMURZzWFey417sQnN2dnZ6euqVb19eXuZ65v7  
+Ps60+DbHx8dLS0vn5+cKzbGytraWtMhCf29vT37Ps2fPkjUNOzs77IKUGB31ujVs+8nJcEn  
aSjTbF4t06zYeHR2tr6/rxs3BwQfDg+Nr9m9Rm5ub1L0ysqI6R31bp6qNTfScpWzv2Op1BF3  
Rn6TUbKdcOgelg18dtck25MxdIdG0qoLeBuwxncUaxTMDg0XW/CLX5x2FgkKJ2fDMD+6TdV  
2yhalrdkXfXAN0pW2EU+Thvuyue6JMVhzb1tRkzZWxcwsZXvCb0CcxS42W1tbRA8Hv/rHao7  
ambH9Q5Sng3I+f/48uTKnDJ3g/vfff0qOGE4IOOvipCqLHSadkk3Q1QN7vBY9x/oUT1G3qdp  
eqm3NvhgNHbarkzP6r52BXVxcMFvIHg0T21ij5yx1h0AzrR1BV/QhKwCp2U5LJG+2LqFDuZT  
IUAYGcPPrxsncrKQZOicWrMv81FWRTqDCZPWPDMkhd9Z1fqCZyZuBxw54dYBskMRpshi5Gnq  
CFSWrf6p0q2k7tnTzUeqVqapzAFhpsnqmpo21sQTYFm8GuuWcKqXfzP3B6pI1CGogWBtr9PT  
ZQNmgSdm+YaXJurpK39xhncpRdEVIWkUo047QrTUHXTtCt3Z43eJCCRAEQRD0QkyxQRAEQdA  
L6XyWE1ulgyAIGiCYBbtQ/GeKjWvu3RKSvHkKtCN0a01I147QrR1et7hQHARBEAS9EFNSEAR  
BEPRCTLFBEARB0Asxxc4denVwEXs1EyiP319agwoCOT98+DDWNxBNXL7u5zUULPIvw8qBYio  
IObVJpgV11Y6DGt2AiKqRrlQf8ifXDfKpJHrdeGppSjtq4q2+znFQ1caGfa2mL9do3jcxcc4  
dW1tbV9nLmHLvDeqLs3Fqe3s7Jaoh4KyGvb293d3dtOCegVxIIR0YwuyrCeQ3uXfu5Hj+/Lk  
KouT6+rof9T59+qRFsL+/n7wjoko3gac+oor6UOTVq1fJ1b185+/RvQgQanQjScidnZ1pafI  
6quKtfl+Mg5o2TuxrNX15ouZ9E1PswmCxdXJyYu8Gq4KYI+DevHmjJEXotLLvG6enp4zssjc  
2Nn7+/ClbX+GoGeWPj4+/ffsmW+JfXfwoibztpS6zQFTpJj5+/FgTUaX6/P79295sDIhp9Y+  
JKt0Y63XcbEGVoybe6vffOKhqY50+VtWXJ2o+ADHFLgD0PWI1JZrx8OFdfhkH1QQ67crKskr



cJ+hj5+fnKfHgwfLycrIm4ecDoJ5kzV/F2t7e5uh42v2yQNTORkC+fPkyJcoolSf3MYDDw8M  
nT56kxIio0mlnZ4c5oPQkTnTEW+sYXiCq2jhlX5uo+RBkz8/Xp88yggq6YUUVICjoMv2Rj20nC  
hslh+U7oMO8koZvMnwX5pct1+eztJe7G7NsM6P37lmQoKJmsQ/sm+bcCm0lLflLkEPhJSohVa  
bVtJBshTr7Ik50Y22K81vlXQT9aHOhrJ3wpDS1eqGIHLyK+SvwcdbfQz3R5Pt7Iqae003Jpa  
MrCv/CaoWmneFX1eyhlz9PWFSGSW0aE4ozgyjEaT2wCkWtsBpUVklClhXJj1MVKjQVzdc1swP  
MLMs12CqrbKxC/uZQKl1lDoW5W0qbnmUVuW5vNNwRnTANupkOGB0lq9KHmltETmsGlq6om3q  
iSVdUNUcx3mpiuD/uXdfPxL5GQV92Ws07xOuWLO8K0mFGSznJ/Fhme6SYGG0eipPZ8hNnhLI  
WGXiK0amVYthoqPV6cgnBEYiVrKHQYQhr9ccK4rptNuh4xYIGChT17IN50I1IsKi j4U2kK9V  
HNade/wwsXVE3FDAbfP8qxWcWxToH4M51y1Hf13J9eVrNO8TrFvdiF4AmT9YsOX5k2CN5+/v  
7CtXpnr7xw3rnp/cAxaNHj/j9/fu3ksbl5WvUvL3j+XwN2Nzc3NvbW11ZYWuV1L850zs7Z2d  
nKVFgrM89lep2dHS0vr6uMds4ODg9PcX4WvPY0V9yL+6ujrWJwOaxJv6VxXfEgTs56LTpI2  
z9LV6zfsjptgFYGtra+J4RFwayvzvV/9qEeDhAFDPCzC61T6RyHCZrBv0zJSHMC1mm2doMht  
svYsZYtrt50iFj12j//n5+dOnT1NiLFTpdnJykoLmncXmmPKUdTn4uJiY2MjJcZ1W4owGG  
uOqBRGLTFeJs9huefhm2cqq8117xXYoodLcQoh80yiTPOofRMPL/Yx8fHWkSXZqn+vceOHDn  
fZQATHiOD6ePhj/0BpqliPqEJbDBHx0oyuDMryG6CrgTYo540NtdjJWP9HLOIzKibUarP4eH  
hixcvUmJcVomGANQo6yxkeJ/dd8xRGM9d7Yt5pkkbp+1rDTXvHR2Qmhf0RwTJdc/AKN48IG7  
SsupboZSiH18Vdlp2exV2b0MXk4Wq9R5/CyS5MpJrGtqVmgVtKtBvk+2t68B0xly2r7OQLP6  
GdJX+fcDqkjuUain43Qza7v2WrV4flpbW1itsSbIGQW2HXEtJyp/rTcPWFw8iuSr2RU+wumQ  
NghoI1saaWMJj2ar6MprQ3jesL11XV+mb00vx0aKuCUmrCGXaEbq1JqRrR+jwDq9bXCgOgiA  
Igl6IKTYIgiAieiGdz3Jiq3QQBEEQBLNgF4r/TLFxbz1bQtiQqpl2hG6tCenaEbq1w+sWF4q  
DIAiCoBdiig2CIAiCXogpNgiCIAh6occpdmdnZ8mhd3N4p71gCDY3N5M383/9+rWY3yh9feU  
oWV1d1Qtfgtb40BJpQSavPPYKmBxVZX/8+JHsk97Qu9CU6jNV24ne0t5a5R8HpbqhlZyQe0d  
YERv3UrpBrI6A1vFW08eBau8y2K4yzOgWe+mGfZ0D3t/+AAKQwXv0lpPcW4fs5URK5mqYQ9j  
IZLVFbzaZ/5ZOy+zKNI do8a+18q8WwvB28eUvNWVtpyhWfXj3x5C6QSZPiT7N266uapUYVf7  
+GFK6a9UKutFkc6JbTaeWOLkMWZU1+6Jv71w3mBhvNf0UmyJgnmFgjcncq+0s7Kysrihi9HRC  
4DNnd3fUfe+EQAxG9Z39/36QpQp0ISrX+JHh25vCYem9vj6iipSM+T+qb379/2xtf4eLiwlK  
x9PTU7I2NjeKnEarKEngWrvrQEiUuHBO1+kzV9o8fP2pYzFhLHwdVcXVyciLj5cuXGhWLCmb  
GYIg4uc9hTYzVEDa63mr6+NraGpMc86uSd0Lv92JtRtQFz/X1dR9eKEjs3v5sIM3bt29TooB  
9kKER5vDaC52N2FJU6SN0QQtyLw0/PDX88uSjBMay8/Nz2bC8vJysG6rK+v4M1J0scVgqt/O  
207W5M1LCUEufDaW6MQwqCZeX15yipcRtGIiYD9TrPRNjdQS0jreapj4X6GTWjJ7QZS5CU7  
z5WeWTekyNCX7SWEsOiUyTPrcBYFSv1bKL5D0xzh+LTNCbclqBZ1QG60tldPwhym2zUWnbF1  
yUTNNB5Sxfs508ZcNvFZ+LPBjiuDKrndWg/5kzUsbKRvGhKxJbQIvyKhh1xZD1JM1fzWDXk  
bE32q2k4oqgi/Xrcqf98MKV29bjViqv/mepyYkLY7xG9D38wSbwZLi3GFZ8hgA69bsgaQEnU  
gJW6g5aya2Eppd/UclN/PHCKnF9nYPbL901K/7T/t0WxhLx3eKm+HFZcglhCQIAo1dcdgqJyA  
CSdnWTJxksDxg2bQ6qwd/tvw6g3RTBtt15FHm5vj1Dgmb6mUEPGxMk11fLcToe9Vw0DnD61a  
vt03bzW8hJ6r8ftOwdFW60Wr8YN3KoxHJML2NTSP1Q5hjckahNbxZpT2Uyq8Q92S1beUGpc  
BCZIRg5bj9FOs8KJo8vDCaYaQh18fr1oRv1V+M9Kcjd46PGtJ1vSw8T6YaIjfpBb5paLUCZQ  
1zclg9Xi/wXrZbNPHjBxUYusqXWk9VJusYaHJuUiTh+3BSK4KimXfxIIdMrxu9fpUtZ0osrD  
xIVf1LH4CBpavXjYazCDVS+gb6o8+vzmmRV19nT8yVbk0arhpS4gYEHZLYwOs20P/Frq+vszL  
61e7urn94Z2Njg9/Pnz8r2RDdAH/37h2/v379ynyJhw8f8nt5eVn1X1tbQ262Z2mOH38/Ojo  
60DjQ0+dA0JyentqD/qVP/Jc6m3N8fMyKzs/PfEJQUYHKs31kYFc+e/YseeCbNnV1ddXfEtv  
c3Nzb28NDZCopf5FiWUH8nJ2dpcToqNenpu1Er/oXECeELgYaVv1TsbEwMa50Tk6YCVKiGv/  
EycQ6R0B9G5v0tap+ercMMcUiFkevGPv7+8SWPe4FeqaJeXfaGcLHLKNCrriis8pPiLMXme/  
p53P4X6cadNhCQ9OeHYgQQ0W5Sp0NoeD29jb7yD9noUjloERJjw5TPn78+OnTp9zzCHPLxcW  
FjucETWaiT1GM0Z+k7CK5soli4fBi3vpzV9TrU9929S9hZ6vETJU/FRsFdeOKmcdyGE+fPi0  
OWerXTepcaGaJN600n949ingzOoeacxcbs9X+WZ2dNvlsdD/1QFAGf12FRXjsOie2XUPAaQV  
L/ayF7i0nhiqxDM91zg5rT9aUsM05xYATpEKkGapb76uv5U6MahN7TWD1VL81tkrrMtTKut  
XqqQM0Oq8pzKUTNaA0F5riMBjoYIUZHcpLiW/qQfokMvQBwPrVqXPVG1naamvF6eGFK6iXF  
FF6tqu5cFw3RuHqvdMg+6YYNsQIm3Yj8VVGW1DYPXLV19SEmrqfZYy1M6Iydc8mYgVqnF0Hx  
gJG/hVnnyOj9z9VpTs0roGQntKtNmWFV4sivGU6wVn1KndY0JMWpJssD1s1WYyyUtiRo3jW  
oqt28QlXJGgq2PBcYQu0CvzSXLJblmgiTsVdYUubKGQq0DU6Cm7dhFkVladEKVvyfYtmQNQib  
MNdzGumlyFaIFj5cCW9n8CABYwr3SrXm81fZxBiiVEu3GqxawrmRdXaVv7izFR4u6ZvSS7uz  
sFP+BrwkRb00I3VoT0rUjdGuH122gx52CkXF8fLwoDzoFQRDCFTHFBlPw4+aV3Ofn54vyoFM  
QBMfdkc5nGTSVDoIgCIJgFuxC8Z8pNq65d0tIWkUo047QrTUHXTtCt3Z43eJCCRAEQRD0Qky  
xQRAEQdALMcUGQRAEQS/EFdt3rK6uLt2mq3cpb2akRAE9LXzc6Yfu5wpklJ4p7aSeqPCHDX+  
8dPZkNYzvLbuC9qYW3pAW3FDU04MsWgom78Q6x0FVXBFF81fFTE1cleo5Mmr6I4tqx16o0Wd



i2X65yjAj6IpZJNUrmfQuEtLdvtKlB4YPNr3kxV4ZJpDUVMXIVuzHo+KWGXJvH8u996onhtS  
NJuv1XiL36pxSPXPk3mEE9XX2ypDS0ShrF4bFFYb5q2KmJq6Keg7APOiGCGwG2NJSSvVpWLZ  
zWGOyBvvSTtCara0t4uPUfWknmAp008sj9YEMa0ntixQbGxs/f/6UXEtjx48a8gQn+laV3m9  
1cXGh5Gj4/fu3/79nGmhaVenp4ZSi+GaSmjrHRFVc7e7u6uNgwPyxt7cn26iJq1I9R0aVbmt  
ra0xUjIFK1lK1T5OyfRNT7AKgz0cwQikZTMXOzg59rPiuR+aJ8/Pz1HjwYH15OVm3YeB7+fJ  
lSmTk3rlBPckaEbkP4BweHj558kR2lZ4e5o/t7e2lpSV/XFhT55gojSvmAH5NgadPn/ovyYi  
auCrVc2Q07I+lzM+McUuADqg0ydvV1dXc3d0GPKU9PcbLA8kV1bW71KoFL/moTY8qhMI1qz  
0NdTmneSxm2ryzy1sMGMZxyjaWkgLsm6prxnSHBTWdxVzUJxun5sbiJBiJmuMoA+Ro0+J1ej  
pYQLm7IFzNcZNCzCPr3NklMYV56PFQzHETFYFF1cT9RwBTfpjFXOtD1sGZgRdMYuk/16sbid  
w3oBtvTTLdc3r16/t/hZLdUNCoZZzqqxs6pShFcmTVZxu/+hmm98A6pRTKBSGfoypoFSy+kc  
NVGMBW0oKNE79yWEFMUqzmZIDMKRUh1SyGKvXs4iCR9Hi8XUOWMDSfEMKj3VJkCzqX6VUxVW  
Vn1x57oZOE9OUr1aVi2Q7xuyRpYyvvALJjQIDN858QmEFMiW4tHOTFKOYFL1W9L1A9E7yeZ  
7uEIfQ7OsfCxtTENU1TDkhjYJa0Mbi9Qin8cgs7UUNUp7aWnBnhhSN49Uk12vZy1IVvwSX+c  
ADCydwufjKqeb+lqNAj5zjlI9e+LOdTOat5NffzQzX7QqvW1wonl+sB1Y9V6ILTcojyDnx6hO  
sra0Rc+vr60sV/wCQq+Svv/7it0nNc86jR4+S1f0Dyd7e3srKCropKb9xdHQkieDg4OD09BT  
jq/tXCQq7OztLiZFSf0XX611F8TmUEV81htK4evr0qSYPCX15yW+VAVvXNdbnnib2x4bMmz4  
xxS4w6qJ+0Ac51YdrODk5IZQ54tP9j+r1FDt51YgWz2hoyx0c0BA8TJk2QzCbFh8/kUTCzML  
tviyi0ZkXUZOpuli40NN2okrPZJVxfn6euledq3NMVMWVwsa66q9fv4go2Tknx1VRzXHQPd8  
2ZN70iS12saGjCqaVEtnxL784t7e3bRwsnqfS1eXc398v7erv3r1jJLWCRPz77FrwxSHQRgO  
tIRwmqyEMMY8fPyPp4M+M6jsJuigxB4Bpf7crDMAg8PX7x4kRLVelahN5nYcYnI1TKmauI  
KoeQnVHZ3d9+8eSO/Z2Jcleo5Amsbj8Y86qODdOCrmgtKaGW7Z1rGM6SNyN5b/t9LPp7h81  
147RqWYTHMujkTDeHhPJ7j+5teI+/W5zby1lQJF1D4Ye55MqQEYsCyCUFBtSn11ZMq0A7WFG  
yhoI9XpQCSvUkqcw+TorKVNXXZK2xJsgZBbYdcSylyECG5nG5VcVWvZ6+wxmQNgtOIXjd/gR1  
Iyo89Md6qyvYN60rW1VX65s5sFlSoa0LSKkKZdoRurQnp2hG6tcPrFheKgyAIGqAXYooNgia  
Igl5I570c2CodBEEQBMEs2IXiP1NsXHPvlpC0ilCmHaFba0K6doRu7fC6xYXiIAiCIOiFmGK  
DIAiCoBdiig2CIAiCXogpdpH44D5RZ9irdrQCOvWwGaDy1dVV2YsLTZBwemL8TX7fp+okZH  
iuTem/nBf+su9wHJm1OrWp01e2Nz7iar20xZiWnuaCamLba3QTVKilKT0uqnQTP9W2fpWy/XGW  
YEXRFf5JSs71VB4Nkh+98oTarX6+beVz9Y92UGeyBoHtV3P0qpd/3dfUJupmL47JvabHNJF  
E/mU9/cGKkjUIVbpNbDv5TS7y+Pip2Re9Mqr0tN2aj6H2wsS21+gGKpVz9s086AYT2z5L2T7  
wuiVrSCnvCf1JSS0+hoiebgPI11/s6rMzZLaxAfjtp12WZFHDId53YMiVyu20/pgh3Zz0nUE  
tWTFHKZX7Yu+GVI61mUq0UZ6EEaTtlfpBponVNWQ3Llu0Ktts5TtA69bXCgOxsFzUWYmvy  
HYvb29ra3t5cqrSjVYG9pF4yVyRoRVbo1abv/Ssz15SVDnuyafTEm0OT8/DwLHjxYX17mt0n  
bq3SDnZ0dDvL29/dTeoyU6gZN2j5L2b6JKXax2dzcpK8eHR2ldIbd1rBbqkBO0e22or/x0/r  
rjPNP1ThOx+MYU+cTpfvdmjNXH8/qiobzX03bCTAGvrdrv36Z04zoXGg7d9IFImv/z509rfvO  
253Sj1Onp6cbGRuqrI70RW6pbw7bPURz3dJrRtAV/Uma7bdE8RoIEwadGYNfMuj6CYdyyum  
d5GSCwdBVKX6xAVt+oBTZZHcF9Serf9ReU4mGk5Q+Rq75RVAPUuI21BrsMtQc61btDhZRCBx  
+mtTZE6woWYOgE1aFm83bXtRNma0sdlU0dg7rStYgFHVr3vZzYnYO60oWdvoZrJT3gf4kpWZ  
NgXTCXNBotvAQYXKmHAUUiGBzDPZopljIaVLaHJppTS6CyFWds3NxmAdjzkUIE3Vr0nZ0oyw  
xpmSTfdeHrCtZg0C7NKf6Bk7Vdq8bkekzq8008tCkqFvzts9Stn08bnGheIE50zs7PT31V4M  
vLy9zk8H+/j70tPg2x8fHS0tL5+fnCs2xsra21rTIQt8+++x59uxZsqZhZ2eHXZASo6Net4Z  
tPzk5YzhLiWb7YtHZ3NyKXSSrK2qj3YKZqu053TyPHj1K1rio0s1T1fZzyvZNTLELDCHFodn  
u7u7Xm39PJiYdGUBiq3cfSCS29vbHFmP+xxKj559yD2wIzjOmpZ+Kkc2TMz+EZwXUtRtgra  
vrq4WR7eafbhQ6P6ftffo6KjYHxu23XSb+GzUCKjSrUnbZyk7ADHFLjZ01NevX6+vr+uf2T1  
S5uDXH8TRn+V8/vx5cmVOGTrB/e//5QcMZYvc9bFyUfKO0w6JZugKwc2SqJnriePhqJuU7W  
9VNuaafbHoMILT1+wM9eLigt1UtjYdq8bv/Rx67NU/t49bDwOqnRr0vZzyg5Bdt3i+uRaRtA  
VfUiaC5HkzdY1OCv1SVAGIAqT68ZJ/C1phm5XCNZ1fsqqSCdQYbL6x26A0bTkyvA3xnmh8mb  
goYvKz11CJ4nTZDFyNfQEK0pW/1TpVtN2bOnmo9QrU1XnALDSZA2CmgkWSDVtt2xVug1NGOC  
250ofVpesQVADwXQTPw3PZVMGafK2b1hdsq6u0jd30LaSEXRFsfFKNO0K01IV07Qrd2eN3  
iQnEQBEEQ9EJMSUEQBHQc+181hNbpYMGCIIGMAW7UPxniolr7t0Sk1YRyrQjdGtNSNe00K0  
dXre4UBweQRAEvRBTbBAEQRD0QkyxQRAEQdALMcXOHXp1cBF7TQkoj703sQoVBHJ++PBhrG8  
gqmfz5it+Rlrgvvqnl+mUgnTKA34XNCm76FgfbfC09IFURVa05IBpLXyQ7DkplM+rbXhVvAo8  
Wpfs4mEU3oHgxA6VU58TRsi+uMswIumJGSR8/fpx7w4s17X01X6q/vwa+BhXRY4nuHLYkWF1  
Dk/37dFDMXv6CPnrhi17hVPXKoeJ7dqBh2W4ZUjdAKNMkw9prTpQpfe1XjeZCouWcvTKkdLT  
LmoaRe6nQxLaXxhuoYKng/bEoutnLs3IZqMQ8LK0fMDuEdSURvrSzQnj7+k9OTiykquCQjaB  
88+aNkhSh68q+V/z+/du/b/3i4uLVqlcYOqTVp5tXVlboiu/evbvOcRuyFT/C07DsonN6eiq  
tYGNj4+fPn7Lt/bovX77UwJejSnPj48ePI47Gkt1EfdtL4w1+/PihI+Zv37411+iYRbel7Ct

GzKYpfcPu7q71Tfqpvcd4SGKKXQCOj4+nvcb78OFDfolLJYEZmvkgJe4NepG6cXh4+OTJEWz  
GfXlE6Uc5gd65vb29dPuKaMOyiw5j+vn5eUo8eLC8vMyvd6HLy0uGrZRvVGkuCGbm5pQYI6W  
6iYlTl4032Mk+zjPuj2LNolspOhS2aKSf6gs8AxNT7ALgI68hDIUC9B0cHJTehMBzfXciwy9  
Nrtv3Qkja/Qz1fLtlgl95FgJaypbbJNFkXmRQ4+iYiYT+7zUZ5Zyag+Ge+GEXoxunFDprN3A  
SljlnkZzmiEap3Bw8Mqp0a9L20nijIHMDJ3bqdCD/yJhFt1I4FEBG1Ljhdrrt9dXiwW/z3Ad  
mlDQXHIzpaUGGLhRPvLXwj/tyjtWgskoyDbMi+XGqQhZhs0hOsDyArbLK1uI2JKWSNSyMXLa  
lvo0godSuUrzg05btCtaSrKHQSWrxBhhtxw8+MErxmoOJh1Gstj/Y1GQNQqluU7Xdx5sCzIo  
Xa+6PxdKNpT4Dftn41KQD9FPwuiVrYCnvAzNKSnd42MpFhu+BE6E4ms0/UUjwaZGBx0ennzY  
wbJTUej22kc2hVLKGxQ4ORK4tE2cLWmq6TVu2E1hrsoZCilU1kIBhkZ9Bi3jNyWkRi5g+3vp  
mYomKurVou8VbbqrfwbNvFks3lvoMod3UZ4fXLS4ULwClT0DkSJeQMn5k2FXc/f19Be6nT5/  
4LXliIvdwvaNHUj/j9/fu3ksbl5WUuyhfl/1DuiiWszY9ICJJ7kx6F8Hth2rKLYObmJu1CMBw  
x+08QJycnDGEpUuz086Ojo/X1dYXowchB6ekpBnm0dDSU6tai7VW9Xn1zfHSLm6GHJFIiG7v  
49SPAMMQUuwBsbW1NjIzrwf4GZf7333+1CPBwDKj7EIX6uQ1VfJ8F0DNTHrp3MdtCCHFxsBG  
xkRK30bMk/iHYUs7Pz+m0KXFDw7ILB6HCjrbRnJGudL8TSzUjfk5zpuQUo06MZGT3Zat0a9F  
2i7fS5+mGnyp6pUPdDGWz+fjXr1/UIHtIYoodLcSof2KCY0A9E88vtr0wgZNDlurfe+xMhfn  
dwrHYjYlaTlwsG/iHgOaZw8PDFy9epISDg+KzszP7R5QqJFeubzcsu4iW69nRdnbOZMnoJts  
o1cRTpfmIaaJbE7y2/NIZraNR+fuyB7kXmq50y4FQqpMhbnd31/6JcVB0gGBG0BWTJdW9FqN  
488Afi2En720oRT2+Kuy07PYqdL8H/EUVVes9lg2SKY05pqFdqVn4Unj7gW7MgJdF4FRmywN  
e55qyvcIakzUUaiaYen5wz8UeHi9yUXMPZwuWdg7blqxByOS5prSNUbZbtqp4E+TRIT8T+4b  
VJWsQ1ECYSjfwIX4MZNSciJvcvUPq0vW1VX65g7H4zKCrghJqwhl2hg6tSaka0folg6vW1w  
oDoIgCIJeiCk2CIIgCHohnc9yYqt0EARBEASzYBeK41J7EARBEPRCXCgOgiAIgl6IKTYIgiA  
IeuDBg/8Pb2K6vKjg600AAAAASUVORK5CYII=)

#Random State = 0, Imbalanced Dataset

! [hasil3.png] (data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAANC AAADZCAIAAAD  
NIYGCAAAAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQ  
AADr9SURBVHhe7Z29bhtJ9/7x/zJWaIXA17ABsgkcsA5/AYFN5Ogr4RjhhNAJFrEtOSJiCbg  
A28EGBjnYawCEkMvt8P+4n/KhXN3V09NvnmfTzA+dbqquuuZU1X9Mu5a+fXr1z3HcRzQb  
g/4W/juM4juP0jc+yjuM4jjMU4Y7xysqK0o7jOI7jdCF+Fpt7lvUHtP3ikuZwZRzHmTDJEod  
3jb3HcRxnKHyWdRzHcZyh8FnWcRzHcYZi8F12ZWXlw4cPIRGxurq6u7sbEp3J1XZwCMckFh  
UNjc3UalMO31Qm7Lfvn0L6bZ8/vyZevgMacfplVzYz9Vh+4p2+pr2XibkcJy2DDvLan599+6  
dkgYd6evXryHRB1++fDk8PAYJa5hi9/b2QmKBOTk5+fTpEwafvyJajB0Ivr29HRLdWftb4xj  
4DGnH6RX6LIMAMZYE/1wjw9bWfKuePnwY0m2hr+kAtPd//vmnOJZfjx496j6FO3ecyWfZ169  
fv3jx4vt0N1lUOhjhGxJ9w0y2f3Vq1dv3ryRcx1h9g1WYxh0GCBCYN44L/GLV2cc/ve//1X  
OjrfSZzc2NipPKPf394PVDBt8HMcYcJZlvOaa9eXL19gfP36UcwT6upi7XZjwgjUWnAktxaW  
/Mw04CQ7WTXL+Qcnt1NPWuS6UpzH4OP0y4Cz7/v17rmWJ0b//vv4+Dh4M4RnInFYI017eBM  
/rcFJBjz4deaIwbTEPIFBkliXIEQHq1bPa82vKU12vKpBIR6g1FGBWqoHSCgtiifQePRJpp  
pLbVsSgqS7FF3F9bX16WailPwKvf1Y1phlchJHn0XiyCdMw0UfnRMPHxt8XNc6yDKYHbcqa2  
zx1i1Id0YFRTBVTz5Ism9qX94kwGH8e5ekUFmNEXX79+ZdSWrXuYyUNHtr548SKx9VDkzZs  
38jM9m00ekPPquO/d00MdUNJyYttjFd19UkEdhjnNb0ns5KlMF1RhQ/RoKsaaA9imFUhJFZE  
IbFVbQM2UP84D2Goa+ZHRnNqX2q7KVQnE9Vzlv7Ep5yCbCqu+ushW7Ac5zq6FHjCws96gTy  
yCV1Fb9xzzY47u6J3LirDmKR1QOtr7FHZkt6UlHXuIIRBsApCivF2hxCEkCjqtzAVRkP54tA  
kXuWPR3aIo9/yGGyy3cW14bTpRxWq4+X8EBfvAvUEqwhJQIMRq8ex2dFiWzaBGpS1DBy8Nae  
+afjxgPalpFUe18Mu4uNBfO2uvv4cZAUW45SCXyJ8QiKCwMNVsxphadlqOnVz1AuSMFYXiYF  
bvGsDT5Mu4EyBJDCGumN8fHy8t7d3dXulAM/R0ZE21WHUtt8hn56ePnyBOPy81IeoacJP37  
8UHLAjd/Eo02rq6u6T/vx40fbqntor1+/Vodvju5uPX361AjQsFXPly9fglXw4MEDDUaOMxq  
6PXt2dsZwEVXjQfwnJwF0ST3KvRrdqp7XOI4xyCxlGkEN3RqmYvFw8NDJlffk11oa2srbLj  
5hBL+/PPPYE0d+jBttwdIOzs7zKayBZvo/Gjb4p9t+HaYmJO5s55//03WAVN5mbH6Yvd4nc  
YRptcv0XqC3Z9cXEREhHF2HZ1nrq9vZ2MVI5jDDLLEnP6abHBTPD333/nfoDHNZnd27F/e2W  
uJXzX19eVZiYmGU/A9ZDffrmzPNd2Z8+eyabhSIRQ5gG7oEymwISfp3/yqQmbr0DnOt+/f+c  
TicpXpeXfhuzv73MaZP7j4+N5/8PBcTpiJ4VczsoYdc5x9/b24p9c0XH4VffiwJKTzgzKMPk6  
faG4zoyPxxB0/n4gfbOC3oNTj1SRGwe7P2CaMxGPPZpQErs9Ial/UDPkDHvOI//u//vwWTX9  
sq6ouUEmwZ1f+6iOsgYLmJJ6Qr2isjLgJONgXynVXmQw4LRtimlz4pS05y/XEnqQSiO2Z0pE



nWM6dJw5+C2/Fq1D4xQOLop1YjbPlOnvl495ciMaV2IAj4n2pOJ+W3yqMe5NzZyEGglWwECv  
fcVaY/L9a2bN03K6ki4wr4zjOhEmGuAH/X7YhTKjJDU88jx8/DgnHcRzHWVpuf5aNf6onSLb  
4RY/jOI7jLBoLccd4krikOVwZx3EmTDLE3f61rOM4juNMFZ91HcdxHGcofJZ1HMDxnKHwWdZ  
xHMDxhsJnWSegl9iFhOM4jtMH/hvjoXBjC7gyjuNMGp+N8d0lXn7AcRzHGQGfZe8QOzs7wXI  
cx3FGwWfZJePDhw8rxeux+ITd3V1sLb0JRZYrtIKs0FoiGF+/ftWiv2xVPVza8rm5uUkGpMU  
IbDbZVsdxHKcFPssuE0yNWj2Qye9XscDI0dER9tnZmR4D6IYwk6it70ue9fV1PNhaxGTjv//  
+s1UISZ6cnDCVnp6eygMkHx58yCYmZvxa4ctxHMeZF59114mtrS0t3cX8x+dfF/3F56dPn7S  
0tDatg48fPzI7FheiK5pNmVa1SVg9tvARE62tQca1LzOrFvrVXNt8WV/HcRwnxmfZCfL9+3d  
b8FLMNU1eXl4Gy3EcX+mGz7IT5MGDB8lignNx//59Pr99+6ak4ziO0xqfZSfIs2fPTk9P7Z9  
2Pn/+HD9YJQkhUcXa2tqjR4+ePn0a0sULK4LlOI7jzIXuKJrh9MUQkr5586b40q7Qg1UR2Y9  
evCDnp0+fQvrevb//lvF2aRkXM/Xr1/ZZA91LbM95QV5+qL3Ch3HcRaHZIgrL6hY8dfx9I1  
LmsOVcRxnwiRDnN8xdhzHcZyh+H0tq7TjOI7jzOF2Ir2V/z7J+E69fXNIcrk7XLfWuHTtcN3  
akejmd4Wdx3EcZyh8lnUcx3GcofBZ1nEcX3GGwmdZZ+LEyxMLL7SydYdCOgPZVLdXQ+IaLWo  
EU11KIIdYtaWMT3ZQBkrKT102tM9AqbGigW6x58uqYyetmaIWxuXQzECfpgguh268CM5y+cE1  
zjKnM169f2d2nT5+w9e4OvYUD9HYODL3BQ84EFYdHjx4FVwFVmqeDpOxByR3kQLA7exs2tjS  
EJrjqZOhr/ielzK7rBaNLRIgswopnSCLroJhsmr5ugXexR79UxmuhmkMGEggXRLSRGVvMu4JL  
mGFMZ+mfcaemx8bux5ATryZVQJO66EPfYuCcPyp16SZ/4jETtbaIbeeKZBnHsK7gV3WA06Wh  
7sArQR1I00S0WDTvWf/K6Aelid9ZM0UQ3gzBjayzOgujmd4ydKZOsmrCxsXF2doZxcXFBj5U  
T8DdfX+Hbt28Mflp2EDBIJveil50//viDz3jBRK0h0US3tbU1LcUo7A7eXdcNtgeraO+XL18  
kRRPdYtF+/vzJnCHPXdAntre3mSOT1cOa99PPnz8/efIkJAoWRzefZ22Jo2k1gb4aD2rAgCh  
jZWW1/BQ2RnNPUpXhMViTgNYxyjPwMXgdHBzs7Oxo/minm4a/u6BbzMePH//3v//Jnks3ZoL  
9/X0t8Ax3QTd7aHr1BDV6KNtcNxrLZujF0c1nWWfKMD1wAmt9+Pv37zJq+PXr1/XkuwyjPjC  
R6+vr2E3WJ67UTZcOcy1vPBmYIZ49exYSeRLdOKd590jR6ekps0hw3QHOz89pNQZqfPr0KV5  
SLEes2+7u7snJiewFxDZz8owvr9584ZrsuIUeeXo6Che0c+pQRpkP//8s7e313rpQwraNdm  
dAvWYA5ILqSa8evWK+eNFsXzWzJlmiAXHZO2Y6+trdH84+NjbZpJ+V7xouGzrDNxNGyBfkn  
x8uVLPjC2NjSLCK5x6+8Sx+hJjxXXPSg9yJwSXFswQXKa8vXrV85ONOLPpRtFXr9+bTPNHdF  
NxLeLYd54040E2XdKN/HgwQMZTXR7//69nUZzRki4Ynz48GFxdPNZ1rkr7OzscF2rQf/x48e  
np6fYAl2RrSEXC2pgBrJfBl1eXjIgrhrqWWR0j12N4hPd/il+ddJcN9UQ/xroLuhmIBdahUS  
reEOZP//8U8bkdwM2TX4/odm0iW6ckeg0Guw3fapwdLpBuOjgznL5wSXOMr4yuY19k/g9v5k/  
8KZzhiItQM/XLHpQxdeOCgN1JHzCtYrtGNzbfApNfEt2KbjCmdIB6ZWWaxxugTJznLuhGay3  
G4jbOpRs5F1C3kHtztTuCS5pjTGxodewOGPiCK4IeqK0hXUAY7qjKIOJeaJXtK4nRYNhXsEa  
BxqqBYMOfqNfN1DEYKJUHxtcN2F2wRoE2QkhE1OuGINoKsWLiLuimBkIyHdbrFoNKiXMRdAs  
L9Kz4Ckd945LmcGXa4bqlxqVrh+vWjkQ3fy7rOI7jOEPhs6zjOI7jDIXPso7jOI4zFOH28cp  
des+I4ziO4wxH/Fz29yZrT7n7xSXN4cq0w3VrjUvXDtetHYlufsfYcRzHcYbCZ1nHcRzHGQq  
fZR3HcRxnKOabZXd3dleq+Pz5c8gxJxSsL84em7/GPYf2UomtibaY0Pa7sy7HQGxuboYv+5q  
w4WZg5JadQX91SKI0558MNboBkWmLgFZSqe23b9+Cy3WromOsTgObZUK6EE2emuE61x+b1B2  
cXwVmzEQvOI3fVqXXX5VfCbY4cLT2V1W0w716jVoVS9C64XmkubQK+7KbxfBdror0xzCNY5  
VJI0DNXmzcRnCW/Jz2Pbit5x/UBZEN4UlxEqWqdTWgpmt1DCObsC+gjUwXXTrGktDMJpuQpN  
LMuKR1CitrbFERq4/Nik7BiluIZF4a6g8XPUZtWcB4Wg5bN18GXHsDnfMzSXNwXGOPBiNQ3d  
lmpNIx9dtocummb0uFj/Xk2P/oCyIbuKqF+VbXal1t4qE5w/W+hNGk66JbTdlKPUdgzJDTzJK  
cTNDweNJfk2QOFhQ09cgjhRuWHYJEt5BormbupADnaG3oQk2U90tzSStBZ+1JPbdyGjsCHZX  
pApLG51scZgngV4aH6o1c/6hGWcvlcs6ifp+NFNboE4GvpAYmNuSbl7dYuaK1YEYU7dKZZK  
pUX0tESHXh5uUHYhEt95+/YRANOBbsWRu7ulLcK2s2LPGDx8+kFQp0FaK2IOHzQLZYLfsIbg  
KJ3nsuUwLB516/KniOmDbUbx3y0P+4BqMjx8/7u/vYzDFHh0dyWnECTvzhrJTtGshIdjWnKT  
Jph5UNhnipCmgpERbcDhIDtsWmNSylOqKFm8xXfXcXbAqJaxA55w/W5Eh0a8JmBUW8Auv0aKG  
bMW+sLjt0n9PT042NDY0nEDYU82J956rpjzPLjstVVDvPOQvHTebytazubbIVaL0cZJMT20q  
ZU4Zs/NTA6QYGE7au3jDYaic40OUHbZKzqCnkIxD1z0E25TTsGwrpok5rIFu1U2oG0cuVlKk  
/jJlYcZ2C2fEAcuGRVtbeSidw/HbYNESHXW6yZdPurB78xfbfOiiDvjIgjzI3J97vmHCosYx  
G3OSYuPlgDc/5Q3owFkq3Jl0ActriUTIw61I10W3eWN1IEbtjCayLwsJbKmkYc38ypb0tVx  
/bFJ2INhRsApCivHwOEOvjB61gTZjxJAZZ2VsxS1HjlgscRWSkdyXriSpFoOCVrkqlF2JVRj  
DflWVoIYYHRWFIX2NMueYmaEGWmHBAew6PmY2xVtFpRPippHB6kmaLKQeqBubkUALtq/KndZ  
DtcEaf5ocx08MrSirgYciIRH13pw/pAdjoXS76kwlflRjpbaxgCNwK9J10a2yrKjUcyBG0y3  
pU/G8oP511COnpj/OLDsQ7CtYBb3dMeZ6nzY8fPjw+/fvSRbsbW3hDPnyvHz5Em10u6B8B/L  
nz5/BuobdNa12XnSHIRx6wZcvX/BwbPrmjCL7ILx79+7o6EhSAPtFXrv1UXkPpOONEd26Pz8  
/Z1/y1FT4+vVrDo8MfE1PnjwJ3sWm/vZdZSseP35sasDl5SWf1JDzKzkxutz2FGVtd3d3z87  
OQmKidNGtRaxOjPv37wfr3r21tbUw4BZDrp6jxdT0x511x6GfWVaPctSGBw8e/Pvvv4X7Nzg

1V9WALibH+vp65Sj/33//BauAaoPVH/p6CHQ1DSb1Z08DoV1LCqEY+vjxY7H96gjL41Q6G0L  
B7e1tzvsODw+D61oHhWwCscv5+Nu3b9+/f88pVPAuNhcXFxsGyFRgtOL8jNCmsmRcKPHz+  
4jMDI+SdJvW5NSLQ90DhgnlB0TZguurWI1VWVHM2UygiVBwizDsFMecJr0x1zZkdA4bsZMNOJ  
zRR/Sxc0NPDRMSWWylnGbjLLaXnkxIjvDLAVf7H96njkvLq9cn2DBcP8uhUgPwUtjyqUXUL  
RX3rHhibE19/6kkKiSPJJBtu70WuoP4waOBjTwdAhxW23A6bJeCqdGNSmQ6UsGUDNj5ustFE  
3orLxTpWUAAdrTBEqoWCwRoT2WkMSaHKuIUgkuSSR1ZDzD8pC6XbVi0r9qEyiLbpBSBQRNVX  
puujWLlaHYEzdaJcpgxHHCXAKyBISJcissuhW7o/1ZYcg0S0kGqqJEOQsk7RKQ7Aw4dR+Iac  
GdMFWKaUkm8hgSavEPJB40JEaZENySBAffWgigZC+OWfEO7Kccfn1hdWQJ1jzoMohGYyC99o  
ft8UOu9JpIqMPTikpD1g224UzaqAlwXQVfUWuQ1UFayx4MjVcKNSK4EnzmwKJM3P+YedfQV  
rLMq6QdyRwWIAW51z2saxJBLlh4N9BWsU2uknymVzeo4AewzWKNBwNdNGP2t7ebzFGQtV7o8  
1ZYeGnQarICzQs+IrHPXN5CXD3d2N7zA3x4OtHa5baly6drhu7Uh089UCnDZ8+PdHLvwKw3E  
cPyM+yzpz8O36xRfn5+fL8rsnx3GcWyRc2DJuKu04juM4Thfi08a/Z1m//94vLmkOV6Ydr1t  
rXLp2uG7tSHTz08a04ziOMxQ+yzq04zjOUPgs6zi04zhD4bPswrG6urpyE73AsjubBSFRQr8  
ftnX0psTnaF2/WMyCp6ZL2WWnddvjDmlr81w3kwt7vB51cBXk/FMip0/ztpMzGeKaaD44vwr  
McPqii6R6YZPeDi07/CqZ5WX8YMu9MafJm3S6102XZdGNuLVWJWfycrvxdYORpWunG+j9d7I  
5ZnuNUc4/NIugW/O26+layVC5CPHm17KLztbWFNEtr8njzAUns5Uv0Mj5Y7qUXxa6tP3k5ET  
G8+fP4zclum4hkWFvb+/169eymV1sDZmcf0rk9Gne9rdv3zKnhkTBosSbJlsznL7oIml8LQv  
EFsnRTmCHZuRg4xyFPYLpKXL+mC5le4fdbWsuemk7kUz0hsQt6QbsMVjD01o3vXc3JKJkzj8  
Co+0IKvP3nYija3MstQTXAsTbyGREJ3udJE0mWV1gqZkvCgCUWVbIQ4vywPBVZS1+ycqxad  
5FMGqE9hdUfoKgyjlJI+Fr/xz0a5Ua6SbBLHGQs4f06Vs7yyRbkKjXkgU3IpuMKZ0rXVjU3x  
3Xf2RUj1/SA/JrevWs014VITPeBisrHMEEt1CYkw17whdJI1nWQWWQoeAw4Yi1xWEDp112yQ  
ah6Y5Vva2DX/akTxFxWGwZdfY8QFQp5x2TDWY8wFpYI1LmqIjjwm54/pUrYv1ks3AoytYKE  
YU1+2d9hXsEYk18acPzej5PwhPSS3r1vDtms0kxHPskZc5wgkuv1z2cWF8FpZWV1fX8fQg64  
vX75gE3nKAEdHR9vb2/ofHcF3dnaGc29v7927d8pAEa2co7JyAgU/f/68tbVFEJBcW1tTIAo  
94daqymxi9RjUHmTObVyclzh4sMx06iLi4uQvibnj+1Sdtlp13aijtBiyCNmyj9cd92m3fY  
uzKsP0fX8+fOQyHC7mvssu7jYKRSTZHDdRL+HUH5BziY/kiLmGP6Yv5mbK3/dn1Ty559/8jm  
Bn1/lfgrR5CcSXcou063bzt1h71TMdUt4/PixzmLF5eUln5za5vxKThLTP0nbuaLQUAZcPJy  
enmJwCRE2X30L8eaz7BKjaEviSU6FYw0mf8zKnN8rlwchB8EbocvimA107PPzc/ptSETk/DF  
dyi47Xdq+urp6//79kIhw3ULiGs59+btu/OPHD7onRs4/YUyfJm3XUCBxYqjvGKhhzi/Hms+x  
yQ1RxHhcSxcrqfOLc3t62S8/y1Sprk+fh4WF1j339+jWnkFaQs8X4NvSovuw5e6X88d0Kbv  
suG7taNF2epn+U4XOu7e39/Lly3r/JEn06aXttxxvOgUww+mL1pLgd9iYao03IHhv+j13C97  
o8X48d8pp1bIj2XQqV/8UFb5Yw+Bnnj0symRHORMKBKs4YmPOT7OnB/wLDVpWHZQ2F2whqe  
LbvEJ2SLoBuwxWAPTRTfBVmWjSHAV5PyDwu6CNTD1svHZdpKxboKc5qyvc1DYY7AKwgI9K77  
CUd+4pD1cmXa4bq1x6drhurUj0c3vGDu04zjOUPgs6zi04zhDES5sucJV2nEcX3GcLsR3jH/  
Psn7/vV9c0hyuTDtct9a4d01w3dqR6OZ3jB3HcRxnKHwYdRzHcZyh8FnWcRzHcYbCZ9114uD  
gYKVE5YuIu0Cd9s5FK19dXZW9pGxubkooI2wooHVkCIk8CBJn+/btW6ir6o2pE4MYUEtDunh  
3mDyQCz+EVQa9dsdoUnYaJLrFDbf3siXUxyqUv4vpUdnGJv0016dhHx+QXwVmOH0xnKTUrJc  
xgd6z0+ObTajN6tcrVx5VrVzWBeoM1vB8/frVvGaET8XKuGYXba14g0yC3lceZzNNJFH8Spr  
hYefBGgs1vBwAM+ONIGoh1RB/BSO/hUeMLF1ZNzwWPyhQ2afIk4tVyH0Xg3LrukGTfprL07y  
P9wt7DFaBX8suN69evSIoy2/2bw0xEaz8W46XiJ8/f25tbYXEvXsXFxc7Ozuy19bWaCzdT8k  
atA5gSBTVrLVVkrSq4CRXMeN6i9Ci4cmSUFyT1S9voot7IpPPhw8fMt2+fv262DK77ATI6ab  
FK+H58+eaThJqYjVX55TItbFJP83lad7HB8VnWwfKJO8HPz4+/uuvv0KiGcypyeqV8VAIDA3  
Bmha7u7sMTzqNiNnf39eSxrnbnsk5x+Ni8TJl1l12AlTqFq9ndXl5qVtQCTWxmvsupsSE2+i  
z7HKzubnJEGZrtgt7JBYvaWePfOwRRfyg6JafW4wC7UWZudbvYzI4Pz9Phr8yt7WilnDQ8NP  
T042NjRAf0UMyXkGuD5gnOL1gZAzem9i0mtCk7FJTto5sgCIkoXejXEMfqzDonwLTb6LPsUrK  
3t6dYpB8ybMXTAJ3z7OwMJyMd2fTbEYzR5cRjNMu5s7Ojh2efPn3CscuKpgs8S24hrx9+7b  
+5BRnJ59OnMaXjr+++8/Pr9//054ALadh2noz54gbI60jsph8+zZM6RTs14qeOZZedGt2  
AE4v19XUaTicNrgxxrNbXOQ0m3kZrlQynL4aTlJo103JB8Hfmab/xoljndiPkKGF2JFNHmz  
VDxRnL7L7gvqDNS40hJOMkLgGARMNDZQxTdChM1vv4tQwpm663AyJ6yApq4csFioxSRxWqpQ  
rOwQcQ7AGpoluBBL0+IdOZeJYbfhdMDMI6FZ00xm/YMrlyfmHI9HNR2WXGK5ZuQaNbwtz3ZD  
EE5di8cVEDFe0XA2fn58Tzce1XeJbcA159+4dVx66Z8D1B1JjxJdfXJr0+LuzReb+/fvBukn  
up0xc3If4K0YcrcKdMPmfQUGlbiCnJ8woIVFFFazmvospMbE2+iy7xNAPoEnb29uzoZ/oZDK  
QbShkk+dkJLe3t7nmOTPDcpcXFxsGyERDMYDcNEEV3L2s1hTm6YJ+aatpeI+CdLRrmxnKL  
VP5PWT1qS34uJmWWWkYa6MYnWTCRJRdasc6mZdht9111uGL+YALjk0qNW5gBok5NHQXI+ffo  
0uAqnDF3m6qHItdk+Ptzbwl7Q/QObPNAzGSCWHWKGuLI44WL0TelnsRZySpbh0p9rffsPlpi  
ZZZeUXnRLyRvJncvOxNuoU3UznL4YQTIk8oK32JfQc8SQKFAGiG9SyUNYK2mGHocI9mX++JF



Jd6gWGOBLOUHM819cpLyY5czI4U5TRYDjzYNCjsK11jQZDWQYJAnfuCatBqPJLI8yaPHmrJ  
Dwx6DNQp13eKem9NNVMYq1OscAXYXrFGobGOTfprLk/MPDfsKVkFYoIezThlOX7ikOVyZdrh  
urXHp2uG6tSPRze8YO47jOM5Q+CzrOI7jOEMRLmy5wLXacRzHcZwuxHeMf8+yfv+9X1zSHK5  
MOly31rh07XDD2pHo5neMHcdxHGcofJZ1HMDxnKHwWdZxHMDxhsJn2YVDrxcuYy9GAeWZuaS  
JCGI5Dw40JvZ+onlBQKmhJGooCTVKsilkuvkVoKecM7+FZSfRDRq2fXV1tbyyCk6V1SuQJkk  
uZpq0vV7b8ncxJXK65fxGri8TfsF7TdgwMr8KzHD6oQokjx49St4RY017Q8qn6xVjK0lrUJH  
RXn1SD0cSrLHQK2CSN1hZEeVqxEy+BfHmzRt778zML6IvFkS3Jm3HySawnIKq9Fof1Zy8HGo  
42FewRqEyZpq0vUbbyu9iaBZBN8j5jccq+jGKxyDiTaByORLeQGfNnu0BHSYmbmtjSKBZ3wgr  
lIM5Cuoi/OHmLjBxsGp4SMZMxjgwaAROQsXI0jMWPR8ZBwQTdoHnb2RRvpVQ8SVB2td1jTok  
qY6Zh23Pa5r6Lobl13SDnN3J92ZQUOvr6ZFEt5AYU807QkdJ41mW4EgmSAKI+pmwirmtbtm  
EKYmtGesrQW31zAQKcrQQ6y/xQ6KUH15x9mJU6jZX25MakqlFZZPAHogxpaPJ7C5pWpO212i  
bKDka8fEMTaVukPPnyPVL/OMEG3C0wSrw57JLwPn5ebAa8/DhQ6bYo60j1aoHPPFzjnhrcN1  
8+EHSnhXpaya49XsKvPAsLB3x6erqksaEDhrChROUqbIeHh3QSDZGmycXFBUnZxsSeed069h  
2hrmJCVWmMmZgZttz2kLDGF5qerr1/DWU+zKJHKPwBS2157Ps4qI5EjCCax6ITi6CMBQUufv  
wAg6PTuuYiXd2duQnD6fPONnEHhXQOPk8Pj7GD4QpwXp2dqZse3t7C/4bfI3q9/37dx0/dvn  
3Oajy9OnTypXI1C1fvXqFMmgSn5FMmya6zYtWc3v79q2SWnVxelTGTJe2D/FdLCC5vjZXH8z  
1Zc5gbKC7Beybk+H0RUdJOXGz+70YmhQNAo76+Qzpwih+9U1f5//777/Ld1TwLekND1rpxj  
2PEP7jbgDbA6lgjU8OgUoiZvtMuIMNdsB6ZbUKU2SOodgEXSbq+1EVbXUkMRPQ+W7w76CNS4  
WMzCz7T1tc99FSA8JOWrWuMS6xeT8Ri6i8I+jmEh082vZJeDJkyfBynN1zXuN7jLZ7Vwuaok  
wjPfv3/P55cuXwn0DzpsDVXD//n0+f/78qaTBOXgyblJ52LYMqF0xXLJzaR4Stdi38PjxY+k  
pdf2iM+6pYrplbDsXGSFuipFof39f/qkS99yZbW+obTmGp0duxKsfCXN9mQvcW7xdDD7LLgF  
bW1szQyR03wJ1/vfff7UJ8HAaqJvGBFwyp4rT09NgXfPHH38E6xp6eDnbIqORSw03TExOROi  
3Dbvf+fm5nvfofPtdufrx4wfayp4MOD36ajsDIqdrBHZITxSLmZhc23Pa1sfwJKnUDXJ+qOn  
LFxcXGxsBIxEraFw2w+mLjpi+6vafPDopthqU1I1f3XGSDW/evGGrMth1KgWtbJwZOLD4cta  
yNYcKgZUKHKEdMIbdccIwG8iGCCFRAGxi11JQdUq3moI9sji6NWw72ZQzgVK503sDMbJ0Iok  
ZUD/2nLa572JoFkc3yPkBQWJNyBaH5ci3iyHRLSRuRc1p01pSgomyRjk+iKGWlf9Y1FLUE1e  
FHbbd3IVFJ0WC67ra2BMHcXAVBnc8tCvVBQYMa21ItZQmJLYGthiB211na2GmnOdfmFfwRq  
Lsm6isu0kbSaIIwcUw6ZnHIRjwE6DNTC5mKlp007TDXJx1fsuBoXdBwtgZuoGsR/wSDdtZeh  
qiOudB44hWAVhgZ4VX+GoblzSHK5MO1y31rh07XDD2pHo5s91HcdxHGcofJZ1HMDxnKEIF7Z  
c4SrtOI7jOE4X4jvGv2dZv//eLy5pDlemHa5ba1y6drhu7Uh08zvGjmuM4jjMUPss6jmu4z1D  
4LOs4juM4QzHgLLu7u7sSocVbYme8nMvm5mbwFv7Pnz+X8xuTXJKiktXV1cVfXW7xQUZFDuE  
UXMwr7OSE5PV1Ri5Pk7LLTtwlRb0/pkvZzaemjRaH9StZkY1KQuImNh6G9IToJWYLYLWPPfQW  
f/iowol/sFTDxe0ze3FxiAsgQe/QuD3vFiSquxd6YoOfIb21lrAQQarLXrvyeK3dF66KzMXfxe  
YrbgiisxJvFWKnMvTpOwQjKkbbYzfUmQv0Mn5Y7qUHYjRpKtpI3FiscfxxNkM8rMJk1XFP/J  
owB6DNTC9xIwksGwkzR6zn0KiW0Gmp2YSHAoj2YJN5faTLtFLAnbi6U7N19e07pJySDrhQI3  
gmgTdlZmLOFQIKiTFIKjkgXJmilyeJmWHYEzdkpBDN2mY88d0KTSQo01X0/Z41MNFHVQMon4  
yHGkMVOiOySLoJo+ojxn0AZPutvopJPsKiUGPAF2oH4GwMeLga1ONcKAIi/NIsvpScxF/N33  
BEQarFbRa/ZB6ODw5p0FHZeYFGU1AglBxGEMglZ0JuTxNybFyLrFoGHcbY2cP6ZL2b64Lem  
sjQQJtptygESzX/GKsvTEclT3jccOu6JdTEDD0RVXMj+Zj9FBLdQmJoNwK8u0Cj5NxE/vrOxlb  
yIFNIFlonBywPJBjX+rVTPoEkm5QH4r10hNqC1QpiQgejo5XTkCbCjrns1K3WUjNNB5TBrwx  
8KSABYq0sD8RjiiuDKsnXWg/5gzUKSMEeOXIOU94TKUzIZenSdkeGVk3g2bGIWHk/DFdyvb  
IrUgXt1EdxwY6NsXJBErF4qhfJz1xHMBcl9EiZpBIPZHPch4Kjt1PIdeTEJZQk3HZhmYDRdh  
1HG2KP6H8CrKYREey2UwTb630m+IafIuN1d9NR6zydlhxCWINAQmCH1t9L+cERCAP25qJkwy  
WByybdmf14C+2X2WQbspgXx151Lk58X7HQYKUv2JahB+smWVyeZqU7Rf2FaxxQb04/IycP6Z  
L2R65FeniNqp7qgcBfpLWiRII1DhWldnKYpcjeSBuXbeYmpgxcTASccbvp5DoFhKJt3c0NAN  
KBVcBiuAsr9tV1F2LpQCN9UUv8/Azy2eTRM5vRthQUP5uusNegjU/HLzFDdCQ+PDYFG8V1U6  
grG1Obqsn9hvs18M2fcxIoBLbV+VO66HaYI0FjVUUXSFh0Bw2KZxy5PI0KdsX4+smfF5IROT  
8MV3K9sitSJe0USOPwdawoQRBSFRzC5xZnXScdrjR8EakXl1jBkFsmIqHuBictGWcfqgJbiP  
9v+z6+jo7Q4K9vb3Pnz8H7717WsL+48ePSjbky5cvfL5+/ZrPHz9+FL7AH3/8wef15WXOv7a  
2huIcz8rN/+tYKN69e3d0dKQfoAoxdXp6ar9Er/xJesffqX/48IEdnZ+fsy95aipEeQ6PDHY  
VT548Cd5FZXNzc39//+HDh0SgkvIbJycndOCQyJDL06TsUsNXvLq6inohfU30H9017LJTbiM  
jTzHkXkGSmJR/Xu7fvx+sKdiZhgtnZ4D4xJDJQb5w+aC2+2nY8yyjGs6jzS8PKSpOzs78s0  
rV6/4Z0qdd5KIJWNiSIorEHN+FCfQmFL5Shbwv1EVH+qNQjOfnYsQamW5Kp0Noed29jbfEV9  
QcBUV8s15iZIx01N5+/bt+/fvt7a2gnchoWn0OhuY6JAKZcfQgWcOXrk8TcoulXcXFzoVTsj  
5Y7qUXXZq2sjJPd2necd5/PhxeSirnG8mQIuY0Xgu7FqWMSpsvuY2+6kOzozeoebkrmOx29+  
7s4unObtKSSxQhvhin0147IYnNpOubJxWsNLPXuwmJ4Yqse/G6uwOew/WnHDMiWLAeVIhUmC

zNW6+bphUOjGoTe01ndVS/JY5VphPbJWNd6qkDNDuYk9zKBisUaClFhI02WyDjpedCbk8Tcr  
2xci6CdSZAiJj+WO6lO2X8aXLtZEjYVNIzCCikqCio5kHo8dhqp7F0a1hzMRCxYzZTyHRLSS  
GUJNWUa0wgUK6II4VDeiGBWLiN1BNGUTwln4XELyRn+nB6jQntcmjZC+0q02HATlx5NckJ9i  
qbJVOaxqS41ST5QHLZrswQwpbEjT1G1TVbqykqmcNhY4f7BtHxUAqnSvqUbZcnpqyg8LugjU  
Wf0OmWEyln80LnXOVHZqRpSu3Ec9VuFQ9GsRpmelQyibi/kUeOeORYWjYXbBGYa6Y4djKTvq  
jOW+rnwJ7DFZBWBKbnxVc46pvJS7q7uxvfYw6OB1s7XLfWuHTtcN3akejmqwU4bfjw4cPi/+7  
JcRzn1vFZ1pmDb9++cZoG5+fnC/67J8dxnEUgXNgybirtOI7jOE4X4jvGv2dZv//eLy5pDle  
mHa5baly6drhu7Uh08zvGjuM4jjMUPss6juM4z1d4LOs4juM4Q+Gz7MKxurq6cpO+3re8WRA  
SJft74Q8fPoT0VKDJKtEIGwpQu0YT4+DgIM5mv7WG5I2p04PwU0uVpL1KQvLav4RkBePiC/s  
i8V5IdKuPw4SkrJHE4SRJ2t6wr9mwmQRVzj8qvwMcPqii6R6YZPe/CK78pUoS8qYwYaG8dt  
24pfI2Ot4Zmqrn/LE2ZJ3k1GVkoPCjoI1Fmq4NRbwmA60Pd4UU6PtyG/hERxJsEahUrdCHCa  
UyxraNDNce+TwdQNL1vQ1NDFZMOz1WDn/0CS6+bxSorO1tUV8xGvyOM35+fnN/H+9FxcXtli  
F1khBWyVrePv2rXq44HJfq0KB3n5FtUpOCeKNAY6GW2PFycmJjOfPn2tYLJPTLmuRyb/MpFK  
3mjiMyWkukjicGJVtb9jXGB5Nz42Nje/fv8vO+UfGZ9klQITR0FGVdJqTLM1xfHz8119/hUQ  
z6OdMJYFrkLyOg6EhWNNit1g6JnmJZrwUzOXlZfyq2Cbs7+9vb2+vzLrVvNRU6tYwDivLinI  
cTozKtjfsa/jPz89D4t69Bw8eyMj5R8Zn2SVAp2BaInd1dfXg4EBPGvSuguhUMn5gY3kguIq  
yZJatUnyaRw/M7MkH42BR+gpbH1B08thDjvmXAg4bBeZaMoz20kvLq2g1PH78OfhTgYZzHcd  
pnb5lCBuuQUyU0cqVzWEM5RqXuZnhzwJvSszUDXJxWFO2YRwuL010E5V9jbm3LWOKtoyWfPY  
5/9hc3Ta+jUc+k6eLpPFzWT3i4iwp2071lilxXvHjxwp73sJUKhkaxxKmynsqlThnYkt1FxeOy  
hO4HxAVBnfhtQ2TBaPOqgVLDGhUONn40JhJW21UglGZXZTMkRGFM3BYI1DTtuPn48YGFWSSF  
ttbYKKKXRCCLCvYA1MvW6iMg6hpcw5MKKSDsFC6Vbf11C1slTOPyiJbiGreJ3udJFUMWFeyx1  
2PLGFHNcoJ0bl+MVWhWnlGBc7yRYHtCIVQxOtFUwOpiGqanw4Wp03xND9cj2Qb8FaihqV2ST  
4OIypG19r3DQFZKIeguBkU0iXKKTnJm5I2iJ42jGadE10q4xDyJVtEocDsWi6BasKqUqpJFv  
OPyiJbn7HeHGxIKv8KQR8Kx5uKY8gp5z1rK2t0VfX19dXMj9wTyr5888/+WxS88LS4nbxu3f  
vJBECHR2dnp5iUE/YXNxlPzs7C41Jc//+/WBFnJycMHKfXpZchTWdyro1j0MrOzMOp0dZt/q  
+trm5ub+/j6qMgUrW+0fGZ9klRn016W9yXl5eKpmDIZLI49RYzy2CN6Ic002GhoXl4uJCPyJ  
rjiQSDglhz8YQjXliqTWp4fhjx5zkJedV5cYyVROwE04Pz+f3vPsmbrVxGGubH0cTon63er  
7GqU484hPSkjjw+MfHZ9nlhl7HeW5IFGd8fOLc3t62kC1frTIxy314eEhmOWNev35N0FtBAvT  
NnL8mXTSOj4+fPXsWEp3ReYn9ABKhkgFi2WEQJzAsALggKAeAXmDSbrjvUnaRmalbTRw20Xy  
q1LR9T19j9n306BFF1OQ8hhORGv8toFMkM5y+aC1pfBeOyAveguC96Y9Dj35+ElzXTQvWTXg  
sg06N8SgJyh97iPjEowcnIjnImVAkWGPbkauZMzXG6PgFSfmxy5lpoz1jbcW8CrSDHQvRLCy  
0FAAQj/tJq/GYRjXaxvEzjmIGewzWKJR1E5VxSLbYmSsr4jgcAQ4jWKNQbntNX80OpdBWSPQ  
J3qpOPRzSLlgFYyGeFV/hqG9c0hyuTDtct9a4d01w3dqR6OZ3jB3HcRxnKHyWdRzHcZyHcBe  
2XOEq7TiO4zhOF+I7xr9nWb//3i8uaQ5Xph2uW2tcuna4bulIdPM7xo7jOI4zFD7LO07jOM5  
Q+CzrOI7jOEPhs+wyrcRtZ2fYC1P6gjr1vhWg8tXVvdLlii3SZ4QNBbSuydtNESTO9i1aFnC  
qb5St1w2IjUq/QBZthTheEVxOvf5pwuT0qddNdCm77CRtnBmHkMvTpOwY/Coww+mL4SS15uR  
dPD2+SYfarH69eOVR32tZUGewhudrsarJSN8x+Y69imjms2H0JqM4m2kiihKyUFhR8Eanhr  
dQILUB0Z1TFJEoaUa410MypjSQU6fJrp1Kds7t64bnpo4FLk8TcoORKJbSIys511gOEmp2WZ  
ZICj77Xtx/YyV/VYOYwZbMv/RrmRkp+PN7HuIAJYtqSH5OoZjQXTTUlggS1ZC8UQ1wBnHENX  
2H1o5xpQup08T3bqUHYJb121m/4VcniZ1ByLRLSTGVPOOMJyk1EzEhITPsvNAW+jMIVEwc5a  
1Z9Jd0SGXjTrjr2M4FkS3mYoBGThainVgPtiWUDXJMBzsKFjDk9Mn54/J5cn5h2YRdIsp998  
yuTxNvzFops/111uNjc3CZ13796FdIE9+rLHq2CPKDDkiZ+cmXPC0F6Uya2fvcm3b9/Oz89  
nLh0zvRXcYmLdeOT09HRjYpPETeZB1+HhIYOLptXd6KEssTqxxYsScvo00a1L2WwNsrub9N9  
cnhZ9v0d8111K9vb2FivEdcNZPA0QTGdnZzgZ0cim35gwiSonTqJZzp2dHV2EcVWBk0AsKpg  
sFxcXNDkkmvH27VsmjJCoAtGePn06cxpeamLd/vvvPz6/f/9O2AB25fmZhrNXr14RWkdHRwo  
trfiGpFc5GiyBvIzk9GmiW5eyy06TNjbpv7k8Lfp+n1irZDh9Mzyk1KzZkQuF5B6L7sLFvCj  
WucMIOUr8c72GHdnkwbZboFO6Y1x5y6jkmPpXuFcvO3THuXZwaFkG35K6vgqesagzSWTgl8Tm  
aeuwrWAOT06eJbl3KDsSt6xbSBWSY2epcniZleyTrza911lxuWbkGjW8Lc32QTAZciuUuGri  
i5Wr4/Pyc+Auu6dLil1tG7d+/W19d1z4ALMqTGiK/4d3d3+QpCYqLU63b//v1g5Xny5EmwisW  
6Q1wWI5GtsD1Vcvo00a1L2WwN3MYm/TeXp0Xf7xefZzCY4oaTvr29PRv6iU4mA9mGQvbbzed  
hJLe3t7m2qL8jOhkuLi42NjZCohknJydhQoiuZe3mMcc3zB+32HXHIdHt8ePHnJmlsVQvAqd  
x5efWnKCg59bWVkhPhZw+TXTrUnbZmdnGjv0316df3+8ZDSJmOH0xnKTU/Cb6USsTAB6mWyU  
f3byNzFY+cYI8gJOYt1K6PzPt08a0giaHRARaxXLlsf1WoE/yFVRW3i8LolssBUasQxlCi/w  
hcQ0N6T2o6hlTupw+TXTrUnYIFkE3keu/Mbk8Tcr2S6JbSNxKB542Q0hK5FGtEbzFvoRmnpA  
oUAYg1ILr2k1YK2mGplvBvszf75hIhcEaC2SxDmzQ8dQ6Yf0Qu5w5HgJMFgOPNg0KOWrWWFT  
qBjjv8HgoJKnM1NJwiJUxv50RjgY7DdYoVoO09bqJucoODbsL1ijk2lgZh2SLnblYzfkHhWM



LVkFYoGfFVzjqG5c0hyvTDtetNS5d0ly3diS6+XNZx3EcxxkKn2Udx3EcZyjChS1XuEo7juM  
4jtOF+I7x71nW77/3i0uaw5Vph+vWGpeuHa5bOxLd/I6x4ziO4wyFz7KO4ziOMxQ+yzqO4zj  
OUPgsu3Do9cJl4hXELmderJhDBYGcBwcHyQvM7gib10v+GWFDwerqapMVTlAvzoaSoa4G38K  
yQ+CpPSEdLa2oxZ3K5PTBDt6VlclHY6JbfRzGVMZkLF08FEyPcrzNbPvM/pj037H5VWCG0xc  
dJX306FHYRiFL2htSP12/E7GSuAYVGfklYzk4kmAND0203zQUvwgGu1Cx4k1PCXpLVJzNXoa  
Fwmyq/yL6gh0FayzU8OTNXyT1ah5trXyRU6U+5DcN8SfVDSrI0pV1w50Lw5iamLS+PCYcSbB  
GoTLeyGbbK+PNULUzu3mPJLqFxmHq3gU6Skrc1MSWemMSTDHKQHIFdBF/cfIWGTPYEomYHpJ  
Zgb43s/shHVi2pAaaM85L78bUDTQ20fCQLkDPeBCK4eUxMadPHH6Kz5AYnjh3ldmTWAXlOIw  
pxyTFa/IPx63rBjPbPrM/XvXeqP+OQKKb3zFeAj58+DDv7bU//viDT1sxGw4PDye2jkcTkVX  
Vj4+P//rrr5BoBuI/f/48JAqS1WSYZoI1LXaLZXOSJZsuLi6CVVC51EpOnzj8Li8vGQpDYlp  
U6tYxDvf397e3t1cmfZu9UjeY2fb6/ljuv+Pjs+wSch5+HqzGMKJx+nZ0dFT5oCJ+zhFvDa6  
bdZ9IHhwcyK9At8dy+JVnKaCl8y4zSXsRPxkiyzDZBGsq0PDT09ONjQ190RA2FDDI5h3rE33  
4I1D11atXIT0hanQzWsQhcw+XRLpzUPlgctmp0a1F2y3eGvbfwSmuaEe9M3BH6ChpckYW320  
D3XBLbkOV+SdaY8dqUFklmYnZkfw4VSGbsNkkJ1gewFZZZWtxI4tSwRqXytt0nD5DSJSQCDI  
qs6GY5RmaMXVT5FjTsNX8ODZA2RQPlZT1IUkRiINqanHdsAYmp1tMZRzGXEVkJibVedVPR2C  
hdGvS9iTezMbISToEiW4hMZqad4eOkjIGxSGSDGRzdTaKk9nyE230c20y8MRRGI+eGDYoaL8  
xdpDNoVSwxSxOD2JoddzwGFptCud66SSNctClQ0jCjIckBuoVyG1FTMpaXA0N+wrWwNTToZpC  
hHiCXRuhmpwRCsdx/B2KhdIOZbU8qmdl/ByLRze8YLWfPnjwJVp5wn6XgW4Hdzj08PCReMd6  
/f8/nly9fCvcNvn//HqyC+/fv8/nz5081jcvLyyRYy89RFpMwt+nevXu3vr4uSY+Ojk5PTzH  
iG+y7u7tnZ2chMWKUD2Jtbs1898Vosr+/L3+ZGn1OTk4YEENiusS6iRZxmNBkNFh2yrqJ+rY  
n8Taz/46Gz7JLwNbW1sxuGYa9AmX+999/tQnwcDbH1ItNJ0/mVEEUBusa/YQqhugvZ1sKLi4  
uNjY2QqIZzARB0Ohc2J7xcBJDn+8yXC4ylT9rShrLoIYmyW9PjJn6EIE5wXR5malbizhMOD8  
/n97vAJrEG9S0vRxx9f13ThyWnSxMh/ZjAcKXs7mdnR1sPrHtfQJEJ1tfvnyJbf+4zVUvcVm  
OcmKUS5D4/7ttFwv08fHxs2fPQqIZuk9gEwwiJAPeSSMXTQDY18sF65ubPwnmsDrBgaykL7  
JTH0Ufrcy5A3KTN06xuGd1Q1q2r7o/VFTvR1OX7SWVM8kjMqHE2Fb/rEopagnrgo7bLu5C3v  
OQZHguq429sSPQ4KrILjmoV2pLnyqegla3DownbHLmRHEenLH+Ivct9As7CtZY0GQ10L59eyg  
bh5PAKYly+sTj5jiKGewxWKNQ1k1UxiHzZfKzK/GD8LumW03b8dTHWwyesvLDwTEEqyAs0MO  
ZqQynL1zSHK5M0ly31rh07XDd2pHo5neMHcdxHGcofJZ1HMdxnKEIF7Zc4SrtOI7jOE4X4jv  
GftvdcRzHcYbC7xg7juM4z1d4L0s4juM4Q+GzrOM4juMMw717/x8u91Uzyf2Q4wAAAABJRu5  
ErkJggg==)

#Random State = 42, Imbalanced Dataset

! [hasil4.png] (data:image/png;base64,iVBORw0KGgoAAAANSUheUgAAANCADYCAIAAAA  
GfVInAAAAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQ  
AADrzSURBVHhe7Z2/Tt3K14YP32UcRudRwJwQJCCIqH8FSngVKmOFOqj0KSKCuONEHVvkiI  
XABSNIbFFroEgFKHcRr4nfoeVYewx3t622dt7PcVmfLM2PPuNX9sb+ylnz9//uE4juM4Tg/  
8X/jrOI7jOE7XhHPZpaUlpR3HcRzHmYb4IvHvWdYvHXeLS5rDlXEcZ8QkQ5xfMYcx3GcvvB  
Z1nEcX3H6wmdZx3EcX+mL3mfZpaW1jx8/hkTE8vLy9vZ2SExNrra9vt02hcSssrGxgUp12um  
D2pS9vLwM6bZ8+fKfFevgMacfplFzYT9Rhu4p2+pr2XibkcJy29DvLan490jps0qAjjfv2LSS  
640LiYn9/PyRuYIrd2dkJiRnm+Pj48+fPGHz+jGgxdid41tZWSEzH6uoqx8BnSDtOp9BnGQS  
IsST4JxoZNjc3KfLo0aOQbgt9TQegvX/48KE4lp+PHz+efgp3Fpx+Z9k3b968evXq50QkiVQ  
6GOEBel3DTKPZ/fXr12/fvpVzHmH2DVZjGHQYIEJicliX+MmrMwz//PNP5ex4L312fX29ckG  
5u7sbrGbY4OM4Ro+zLOM156z//vsv9qdPn+QcgK505u4XJrxgDQUrobk49XfGAYvgYN0m5++  
V3E5Ztk50ojyOwcfplh5n2ffv33MuS4w+f/788PAweDOEeyA32C1Ju3kT363BSQY8+LVyxGB  
aYp7AIEmsyxDygl1Wr+7Xm15Qm097RfRGfUOqoQC3VDSQUMFuUV9B4tIlmWkstm5KCJHvU1Yw  
1tUUpoOIU/JX75jatsErkJI++ilmQzhkHCj86Jp+K9vg+rnUQZTA77tTW2W0s2pBujAqK4Cr  
ufJfKE/vsfnEmg4/j/HpEBZjRFd++fWPUlq1rftM1NR7a+evUqsXVT503bt/IzPZtNHpDz13H  
/8Ydu6oCS1hPbbqvo6pMK6jDMaX5LYid3ZaZBFTZEt6ZirDmAbVqBlFQRicBwtQXUTPnjPIC  
tppEfGc2pfantqlyVQFzPr9w3m/iUU5BNxVV/PWQLluPcRjCCT1j4WS+QRzahq+iNe67ZcWd  
X9E5EZRIttA5ofY09KlvSm5KyzgJCGASrICQs7/QQghASRf0WpoJINU8cmsSr/PHIDnH0Wx6  
DTba7uDacNv2oQnW8nB/i4tNAPcFqQDLQYMTqcWx2tNiWtaAGZS0DB2/NqW8afjygfSlplcf  
1sIv4eBBfu6uvPwfZguU4peAXCr+QicDw8NusRlhatpp03Rz1giSM1cViyBbv2sDTpAs44yY  
JjL6uGB8eHu7s7Py6vFKA5+DgQJvKMGrb75BPTk6ePn2KcX19LY/Q3ZHv378rOW6SH2LQ9uX

lZV2n/fTpk23VNbQ3b96owzdHV7eePXtGBGjYqufi4iJYBQ8fPtRg5DiDocuzp6enDBfBNRT  
Ef7IIoEvqVu6v0a3qfo3jGL3MssScRnBDS8tcL07v7z05Kl17pQpubm2HD7TuU8NdfVr7NC  
HabvdQHR58iWzqWzBjjo/2rb4Zxu+HSbmZ06s57///gtWQZO52XG6Yrv4HQBRPtfvkbqCXZ+  
fn4dERDG2/VqnbmltJSOV4xi9zLLEnH5abDATPH/+PPcDPM7J7NqO/dsrcy3hu7a2piQzNML  
4Aq6H/PbLnTmFtr948UI2DUcihDIP2AllMgUm/Pjxg09N2HwFWutCXV3xiUTls9Lyb0N2d3d  
ZBpn/8PBw0v9wcJwpsUUhp7MyBoM17s70TvyTKzoOn+pKHfYi6BzB4ON0ieY2M6YkHrLj+xp  
xjQ38FpS6vZrEKNj1GduEkXjs3oySwPkZSe2LmkF+0G0e8b///S9Yt/2xraqmgUqCdRfluz7  
CGihoTuIJ+YrGyoiboIWLva6rymTAadkQ0+TCL23Jwa4n9iSVQGzfkR15guUsPHHW3grVoX  
CLx5YFO3Eapwt19krb/fmQjSuxAYcEe9Lxfm0/FZ3JuchYUYCFbBTLz5j1Vh8v9qZc/ccb+  
SzzKujOM4IyYZ4nr8f9mGMKEmFzzxrKyshITjOI7jzC33P8vGP9UTJFv8osdxHMdxZo2ZuGI  
8SlzSHK6M4zgjJhni7v9c1nEcX3HGis+yjuM4jtMXPss6juM4T1/4LOs4juM4feGzrBPQQ+x  
CwnEcX+kC/41xX7ikOVWzX3FGjP/GeHGJXz/gOI7jDIDPsgvEy5cvG+U4juMMgs+yc8bHjx+  
Xisdj8Qnb29vYevUmFFl+oTfICr1LBOPbt2966S9bVQ+ntnxubGyGqU8ZaptNttVxHMDpgc+  
y8wRto94eyOT3s3jByMHBAfbp6aluA+iCMJOovd+XPGtra3iw9RoTjK9fv9pbCEkeHx8z1Z6  
cnMgDJB89esQmJmb8esOX4ziOMyk+y84Tm5ubenUX8x+ft5484fPz5896tbXewwefPnlidix  
ORJc0mzKtapOweuzFR0y09g4yzn2ZWFwiX821zV/r6zi048T4LDtCrq6u7IWXyJqP8vr6Oli  
O4zjOdPgs00IePnyYvExwIh48eMDn5eWlko7jOE5rfJYdIS9evDg50bF/2vny5Ut8Y5UkhEQ  
Vq6urjx8/fvbsWUgXD6wIluM4jjMRuqJohtMVfUj69u3b4kv7hW6sith+9eoVOT9//hzSf/z  
x/PlzFWeTknE93759Y5Pd1LXMdpcX5OmKzit0HMeZHZIHljyYskfx9M1LmkOV8ZxnBGTDHF  
+xdhxHMdx+uL3uazSjuM4juNMQ3wu+3uW9Yt43eKS5nB12uG6tcala4fr1o5EN79i7DiO4zh  
94bOs4ziO4/SFz7K04ziO0xc+yzojJ349UfJAK3vvUEhnINvy8nJI3KXCXGsfYX6UQ65a0sYl  
uygBJ2dHrptYZaBU2NNA1jx5dMzodTP0hrGJdDMQJ+mqM6HbzwIznK5sWXXMqcy3b9/Y3ef  
Pn7H17A49hQP0dA4MPcFDzGQVh8ePHwdXAVWZB4Ok7F7JHWRPsDt7Gja2NIQMupk4ev6J6XM  
vusFg0tEiCzCg+dIKWugmG0avm6Bd7FHP1Tga6GaQwYSCGdEtJAZWcxFwSXXMqQz9M+609Nj  
42VhygvXkSigSd12Ie2zck3t1SN2kT7wiUXub6EaeeKZBHPsK7kU3GEw62h6sAvSRFE10i0X  
DjvUfvW5Au9idNVM00c0gzNgaiZMjuvkVY2fMJG9NWF9fPz09xTg/P6fHygn4m79f4fLykuF  
Prx0EDJLJteh5588//+QzfmGi3iHRRLfV1VW9i1HYFbxFOI22B6to78XFhaRoolss2o8fP5g  
z5FkE3WBra4s5Mnl7WPN++uXLl6dPn4ZEwezo5rOsM3I0rSbQV+NBDRgQZSwtLZXvwsZo7km  
KMywGaxTQkZ5Bj4Gr729vZcvX2r+aKebhr9F0C3m06dP//zzj+yJdGm2N3d1QueYRF0s5u  
mv+6grJdlm+uGYskMPTu6+SzrjBmmBxw1oevrq5k1PDz50/ryYsMozynEWtra9hN3k9cqZt  
OHSz6vffoYIZ48eJFSORJdGNN8/jx45OTE2aR4FoAzs7OaDUGanz+/Dl+pViOwlft7e3j42P  
ZM4jPss6YYXx+/Yt52TFEnnp4OAgfQofU4MmyA8fPuzs7LR+9SEF7ZxsoUA95oDkRkoJr1+  
/Zv54Vbw+686ZzJqGfx2TtmOvrq7S/MPDQ226k/K14lnDZ1ln5GjYAv2S4t9//+VzfX1ds4j  
gHLf+KngM7vRYcV2D0o3MMcG5BRMky5Rv376xOtGIP5FuFHnz5o3NNAuim4gVf8Ok8aYLCbI  
XSjfx8OFDGu10e//+vS2jWRESrhgfP36cHd18lnUWhZcvX3Jeq0F/ZWX15OREfqArsjUk7oI  
amIHs10HX19cMiC3OWmYZXWNXo/hEt/+KX5001001xL8GWgTdDORCq5BoFW8o89dff8kyVw7  
MpsnvJzSbntGNFYmW0WC/MWZ1OE066eDMcLrCJc0xvDI6i32V+T+803/iT8EkQ1yEmqlfdq8  
MqRsnB0xO+oBpFds1urEpVpv8kuhedIMhpQPUKYvTPN4AZeI8i6AbDbQYi9s4kW7knEHdQmJ  
INRcElzTHkMrQ69gdMPAFVwQ9UftDuoBk3FGVQcS91GqmJwdXz7CvYA0CjVUDwYY/Ua+bKWM  
wUCoPDK8bsltgDQJthJCiQnCNQbQVYsXEIuimBkIyHdbrFoNKiXmWdAsv6FnyNxx1jUuaw5V  
ph+vWGpeuHa5b0xLd/L6s4ziO4/SFz7K04ziO0xc+yzojJ349UfJAK3vvUEhnINvy8nJI3KXCXGsfYX6UQ65a0sYl  
s71nW73J3i0uaw5Vph+vWGpeuHa5b0xLd/Iqx4ziO4/SFz7K04ziO0xc+yzojJ349UfJAK3vvUEhnINvy8nJI3KXCXGsfYX6UQ65a0sYl  
vby9V8eXL15BjQihYX5w9Nn+Mew7tpRJ7J9psQtsX570cPbGxsRG+7BvChuJx9vLURCBfgfL  
E76WJI8oeRz4ycrrR3pD065bL06Tsv006tSONg+0N6Xxfu1Mfm71CemB+FphxJ3rAafy0Kj3  
+qvXIsNmBo7WnqnKcdqi/HoNW9SC0TmguaQ494q78FLF5Z3plmkO4xrGKpPG3bzaH1DzUTRT  
BEVJgKFqo05z1Rxx3x4zoZu217TndcnmalO2DwaRz3dqR061hX6vRhxrW5Ar2RKJbSDRXUwc  
dKwJqW38z1pRwtBy2bL4z+9pglmdZjnPgTjUM0yvTnEQ6vm4L3VjYeMaNIY/1Jw9fB4bFE1D  
DYM2ZBd2Sjs8h1XtQLk+Tsj0xmHSuWztyujXpazX6aLzStx2S5Di7uS+rFzU3f+/uwOgtSCF  
xG703eAa5LN4CLWHfv38vpzmp8ZvXgBDVWyd1Wcm2Ji/YMLgCn52dhcTNay/jWLq+VqZLh8S  
IyOlGV5JHoE+wInJ5mpSdd1y3duR0a9LXavTz3t5m9axR9B7p7NdpNIAfG66b566SB9fSk1  
r/PjxI0m72q6tFLF7YBsFssEur0NwFU7y2OX7FjcydfTtXxxAtqN475Zn+lvFd/Lp06fd3V0  
MFmIHBwdyGrHCdmu57JQtQWgItjUnabKpB5VNHjhpCigp0WYcDpLDVr89Pz8vj1YWhwZfgV5  
gTtmrq6tkTYaTOXhmF2pdEetWJn6Lao5cniZ15xfXrR2Vuk3U16QP3Zml8/r6usYo0NZ7QKe  
0ZtyJzsGtK3TQtU22AuOXnGSTE9tKmVOGbpZUoOsGulIqg612KQ+n/KBNchZ1hGxa7BRZqiG  
bcho22oZ0UaclkK3aKTWDnOVKyTQfxp1YcV0nseMB5MIjray91U7g+O2waYgOu9xky6bdWT3  
4i+2/dVAGfWVAHmVuTrzfIEFQTca4aZA0KoacbCp/3RZ4ct29wr6CNSyxbjGippCoIZenSdk



OuRfpXLd2lHWjyb96WoO+FutDJRSxJPadg3ZXS9gFYRE4q1Bo3k5emgAfrZqVIohM87KFko  
Ijw7IURaUIqo/cbjIEmqxaCgVa4KZVdiFcaWx1UlqCFGR8VnSN+gzDnuzFADrbDgAHYdHzO  
b4q2i0g1x08hg9SRNF1IPCFaSZiRQie2rcqf1UG2whoUmW/zQdpKyGwZyVHF0GSrF1ji/gRR  
sQreQ7pNZ0C2mUpCEXJ4mZTvkXqRz3dqR061JX4v1Sfq4BrfKmjSn0a2zK8acm9MkTvOvrq6  
S4XtzcXNnyJfn33//RQKd2n8pXYH88eNHsG5gd02qnRRdNgyHXnBxcYGHY0u+oSJ7LxwdHR0  
cHEgKYL/Ia9czyxc2odLZHF26Pzs7Y1/y1FT45s0bDo8MfE1Pnz4N3tkmuQy1srJiLYXr62s  
+yxf3NjY2dnd38evrjq+li+PjY+IwJMZI5eU72N7ePj09DYkMuTxNys47rls7crrBnX2tXp8  
HDx4Ea3C6mWVpHp+6j/jw4cP//vuvcp8Gp+aqGjSWaThbW1urHOW/fv0arAL9GqVb9AXzZSt  
p8AUne+8J7VpSCM0Hnz59Krb/OsKyOJXOhlBwa2uL87n4ZwLSQdNPwurqKuvKd+/evX//Pvn  
pwcxyfn6+vr4eEje/trBv+fv375yUyzaQhcWNduU6WPpW/kgJQuMcO3DeJbmJvb4/V1SiKry5  
Pk7IjwHVR6VurK1fK+ujlXQyKt6PgBrHzbgTjfxabvWF3blUBnsdJaxm8xyWh45MeKzeLb  
a9UlzMpqDnBjmlYU++SlOeVSh7EqK+u64YqWBNyRuLoqSwfZuzhrqD6MGDSZ0MHRICdvtgGk  
ynkonBrXpUc1LB1Dz4yZrU/yNqGy8UyVlgHZ3pwiVUDBYA0J7rSGC5ksKNT/ZKihl0WL5Y5C  
r7OyJ2dENQqKIirJ0uTw5f98ML53r1o6ybkZNX8vpg2FFMOI8vZLoFhIn1eSgyVkm0UVDsLB  
Gkie4bpwa0AVbkQC/khrxLWmVmAcSD18PNciG8lcvHxVoIoGQvj1nxDuynHHzdYQ1kCdYk6D  
KIQma4L3xx22xw650msjog1NKyGOWzXZhhhoSTADBFWVRW4CVQVRKdhyNTzBWhc3jWSCWRn  
AnHGyMADwO6CNRR13eJ4EKYAtjLn8tSU7Rv2FaxBcn3aUdatpq/hqddNkEfOeETtG3YXrIL  
wgp4lf8NR14xe0u3t7Xb/iOb1g7XrTUuXTtct3YkunX26ydnofj48ePTOfndk+M4zj3is6w  
zAZc3D744Ozub1989OY7j3CPhxJZxU2nHcRzHcaYhvmL8e5b16+/d4pLmcGXa4bq1xqVrh+v  
WjkQ3v2LsOI7jOH3hs6zjOI7j9IXPso7jOI7TFz7LzhzLy8tLt9EDLkdnoyAkSuj3w/YevfG  
xt7cXN9b8hphw23s64i/BfutNZQfxjkaaFpOzFUQ4tGmkL5NpW5Gfd15p1K3iWiM0SeuMHL  
k4JioLg2itpfjqj4OB+JngR1OV0wjqr7YpCcryU4eiTLX3Euw6dfjJiPJ+O1duYd4TQ/hj0  
+5vHNuz703Jn40VH9MbxuuWcMSUwToUxON7izbB8MLF21bg1jppqWPHOtSkNKd++6NW97ZVz  
9isJMHPZKopufy846m5ubxEf8Th6nBe/evdPoJn78+BH/v+/5+fnLly9DIgLZzb++vq53QHG  
6b6++0NOvKK7kmOAcovLBI8QhYxli1rz/o1I3aFJ23qnUrWHM5PQ5Pj6W8ffff2s6GR+5eGv  
S9pxuuTgcGJ915wC9pKL87j+nIYxxdNGQKNA7eYzDw8Mnt56ERARD9+zsLCRu3gGVPI6DPME  
af7u7u1tbW0uly3Tb29ss++ofrlmpGzQpO+9U6tYwZir1iV8jc319HT/ad0xU6taw7bm4ysX  
h0OiU1gynK6aRNL5iDDoJU5K4sVDTRSdtBeKsyp6LOByDqyhLZtkqxad5qa2PXciK1412pUV  
O8rAvbZJ/ItkVag3HrDbyGutkqDkhcrt9ETSfPCZUAqqaaL0yvG58KpCs7QqAyuiKqdStYdk  
+GHJflb0lVmbMnfrUBGFPII+hP+p1q217jW5N+m8fxMcAIZF4nemZrtJ41iU+SUN0Dmxcocj  
1C0KHZLJtEiWqsBOnysq2mNOO5CkqDrOsAjc+AOqUUYgbBn6MiaBUSAZBLZVROZvSBNOWDFs  
54Nw0bEoOwMC6GQoAfeMKGGtyjTJ13ZqX7Rz2FawBiXWLYcVMvT748YB17QFgd8EakLJu9W2  
v162+//YEewxWQUgkXmd6ppFUcWPEsYUDt2whxw3KiVHu28BwXWI5jiF2ks2iFhSpGJporWB  
yMA1RVcOAKna0tKiys9EKrScq0VaOWdomVDp7YkjDEpBOXzSfcZMVqJXq1XVrXrZz7ks60y0  
mFzNN9CGAcBippHtmpnTLtb1eNzY1cTgAiW5+X3Z2sUDJ/VRENzCUR5AzuYtWyerqKiG7tra  
W+4F7Uslff/3FZ5OaZ42joyM1Ew4ODk5OTjC+RP9Hgb28vBzf/onZ2NjY3d11K9oqKb9AutP  
T05AYNbn3Lz148CBYt6nXTeTKjomybs1jplKf4+NjZouQGC+5n0E1aXusW5M4HACfZecYzQ3  
xnAFyX19fK5mDkCXyWDMY9+z7QVvRHksUM3zhZop7Fw2/unt+fm5flxWhlUFs7J1WiZskrI  
B0RgL51GTFpydna2srGDwyeIvWW81IuR0a1J2ZJhuoj5mGurDonD0C5REN6Oy7TndcnE4PD7  
LzjfMHJyrhUSxUuYT59bWlOvd+WyViVnO/f19MssZ8+bNGwLXChKgb6Mff4yJw8PDFy9ehMR  
t6KisnVklK818rGtWoHWJ/XAUoZJOPib0oBITtfgkYcWwEKccGDndmpQde7FucGfMNNENqXO  
U5NqY8+d0q+m/Q6N1vh1OV7SWNL4qVQEb0Hw3vbHoRPFgwyuG6dVyyY81kGnd3iUBOWPPUR  
t4tHND5Ec5J1QJfJdwnGqsQYtSjzA4cXOoom/MGesrZhUgXawo2D1T/xd11uHFNqkwBAK63U  
T1WX7ht0Fq2dyutXEDHYSUVkfzRlimEgz2GOWeianW03b8dTrJuSEOHpfsLtgFYQX9Cz5G46  
6xiXN4cq0w3VrjUvXDtetHYlufSXYcRzHcfrCZ1nHcRzH6YtwYssZrtKO4ziO40xDfMX49yz  
r19+7xSXN4cq0w3VrjUvXDtetHYlufSXYcRzHcfrCZ1nHcRzH6QufZR3HcRynL3yWnSf29va  
WsthDT7qCOvWcGqDy5eVl2fPLly9fpBXECuX8Maz58iTLnH98oIxaGtIFREXNU2HZpCJG2LD  
Aul1eXioJNW3PxeQix9udba+JN6iP1SH4WWCG0xX9SURNyXNhOnwiDLVZ/VSL/bjrd11QZ7C  
GIqfPnbqhgZ0yhsP+fpNorZy/V4bX7VvVy0zsmT25h+1QKn5xCvkt573oBuwrWINQqZs11a1  
yba+MyUXW7c6218QbNkXAPMPAHoNVEBKJ15me/iSlZiIvJIrem8T11MT10+e7rRwGDjZ6Wv1  
tWZDzx3Colqtzvt3298rAumnIyy1EaHKulaaMQB/T+V50gyG1q9QtiTQyWBeLycXkwuoGd7a  
9Jt7Er0gdSjGR6BYSQ6q5IPQnKTXHXDRn2XroYOWR6MbBVZDzG/RetoZElmz5+2aYvrjF0JQ  
dm+q3xhA/Uvi+dIMhpWuiDJrEXdigIICkCuwusm4t2m7xZ1TW3CvJQfp92f1mY2ODkDo6Ogr  
pguX1Zd2c2IteaWe3LuwWRXwH6J7vW/TJ/v4+gc6grveLb3T1/Mb5+TmbQuKGy8vLnD9Yo4D  
mnJycrK+vh/i4faOrOcQY0ajXt7luMZVvdquMyUXWbdK2x/E20/gS05fs7OwoFoknumX8Nii  
C7PT0FCezL9n0uigmUeXESTTL+fLlSy2oWSHiHouVktT1Xr9+TTMPDg6smTm/A1+/fuXz6uq

K8ADsduswrknCLCQWgCa6EWnPnj2rfHvdwsbkuOPNZ9m5RLMjq7xkWUe3ZB7FzwSsNeDZ2R1  
OJlGWySQ11+ollxcXF/RnZtz4DbUjRu+hpB+G9A05/yLz/ft34kcxAX8+fCCEWpw/HR4ePnn  
yJCQWgCa6MQ1YhhyLFpPjjjefZecYZlmJxfiy8PX1dXIHgsDFGTbfhvmVyZhpmlk5uMb006d  
Pg3WbSv/KykqsjGRkmZLzKzLKHjx4EKxJYHkXX75z3WB7e5tuGxK1WEwusm4TtT2Jt9nBZ9k  
5hnh0bezs2NX1loh0513Zhki2WRiS3Nra+vz5853L6jHBkoJ+GxIR1X5d0zNtWW5zeoGR848  
JjW5JzEw6fnEqtr6+HhKuW/F/n8ydDWW0mFxx3SZqexJvM4TOeMxwuqI/SalZV4yFws5+vP7  
48eP4dJatfOIEeQCnVogqxSc2M662xvWTMy7YCdQfrGGhmVIjIecHdJCYkovPen+vDKwbmlg  
gYcQhB3hsaw4iJ1HmXnSDIaXL6YZhNpCtpvlJTFJwkXvr2PZyvAmKq4bBSHQLiSHVXBD6kJS  
Ao1ojeIt9Cc2UIVGgDEAIBteNk7BW0gxNt4J9mb/biZYKg9U/CKImQDxs5fyAJ+6TJoKtQkT  
03x/sK1hdGQ5qow15oMHOsHEN09YNWSqHtuF1A3YXREE062atNizqsCVUTUzCYuomKtt08s5  
4y8Vq37CvYBWEF/Qs+RuOusYlzeHKtMN1a41L1w7XrR2Jbn5f1nEcx3H6wmdZx3Ecx+mLcGL  
LgA7SjuM4juNMQ3zF+Pcs69ffu8UlzeHKtMN1a41L1w7XrR2Jbn7F2HEcx3H6wmdZx3Ecx+k  
Ln2Udx3Ecpy98lp059HjhMvHb2ZTHHjyWQwWbNht7e8kDzBYHmh+EuC0jLc8v3/n2j5o81KZ  
qQ3pc5HRDEdn1qcclbpZ2eSLGBOVuiFFcN0gf5maeKMX3xmu80tNP4X6vpaLyVmIN591Z47  
Nzc2fxaOakue/hM1Fd93a2gqJPISX1bC7u7uzsxM2LB7v37+XDmDPbVaXTp40k1CThyULm/S  
SQQjecVGpG3H1zz//4EEW4rByos3pRuhSXBUIyPyiizFR1o3Gvnz5MrhuH1SkzDH1MUk14+7  
FlfEGd/a1XEzOSLz5LDs3WNgdHx/HD2Orh05KwP37779KUsSeUrZoIEX1K3dWV1fpe5WDnZH  
LQ4/VMuji4iK4Rkelbjj5fP36NZ+PHj16+/btmzdvi23y012cnJir/9cx1+/urqSPSYqdfv  
x44deNynOM69BrY/Jd+/ejbgX5/rpnX2tJiZnJN58lp0DWJoRaiHRjD///JNPuqWSwCTd8GU  
gI4PzeJa3rIUUnlbcG7e1thsJ4uT0+KnVL3nha+SqVGHguz870QuKPPx4+fBisEVGpm94tY7R  
4DSqDwN9//x0SYyTXT+/sazUxOSPx5rPsHBASHK0YUFn9HRwcELVa68XgwS/ircF1+x4Gyb2  
9PfkVu8s3tzrm4oof/ZPzA1a4dLl07s0gAmtklsYSACKGcZHTbaJpNYGR1JgkbIq6Tix0/jE  
y7ow32k4PmmjJi+AMAslUPTIqdWvY13iXOSvXrsPADKcrppSUUCu+nwBhFDYU6Ipx/IaKSUJ  
37FgNKqskMzE7kh+nKmQTNpvkBMsd2CqrbOxC/uZQK1jDUikay2QIiQxJHqkqfQD7zho6YRZ  
0i2MDJIXioUyim2AYHUWx416kq4w3QIH6X1PwzQTHGFK6e9etSV+rj81ZiLeQSLz09EwpKZN  
Z3LWSgSzXgSuhOJktPwFH5GmTgScOxHSMWxQ0H5j7CCbQ61gDQ5Hm7SdVscNryTJQw3xsiP  
plf0xI7o1MRBLkZDoJrRKqy/YOewuWMNSjjeQAIfrRaIbAWY9nQrLkvbHvevWsK/VxOQsxJt  
fMz4DKn8UkBCupxRcFtj13P39fcXZ+/fv+az8EUHyu4AHdx7w+ePHDYWN6+vrpJNTedg2DzR  
RclKk1biJddMvdATJ3dld+ZuwsbFB/kePHqnsiP8prZTjrcXl4q0jo7W1NXXtB40Dk5MTjPh  
Gz/jI9dNcx8vF5IzEm8+yc8Dm5uad3VIRJpT5v//+0ybAw/JQty7o5JW/taP3BusG/YQqhg  
vZ5sjs7OV1ZWQqItlT/5mWjcnDsqqdsufpYS/3S2HhQjeGygZPKY61hqQlm38/Pz9fX1kGj  
G8fFx6NjRuey479GabpP2tTgmZyfefJYdLYRU/CMCVsH6UTuf2PYvZzys1X/82NrPc566c/  
laKZvP3780F4SVv6+YzZrk6cfnqgBcazhLJbFFvd+xxqlbpxOnZ6eMgGEdAMIJ4LHzjOYbxg  
QZY+SSt0ODw9fvHgREk4VsW4T9bUkjmco3rREMSppitaS6t6DUB4JQdiFbfnbopSinrgq7LD  
t9i7svpGuKgtVG3ssGwRXQXBNQrtS7Yhv2CRaxa0D0x1bpwuQyWPWY2NleoV9Bat/crqZPw4  
ngbOJbsE17A9S2F2weqYm3oCt5VaTs4lugjoXUdearE3JKCQpamIStAnuUbfwgh5WATKernB  
Jc7gy7XDdWuPStcN1a0eim18xdhzHcZy+8FnWcRzHcfoinNhyhqu04zi04zjTEF8x/j3L+vX  
3bnfJc7gy7XDdWuPStcN1a0eim18xdhzHcZy+8FnWcRzHcfrCZ1nHcRzH6YseZ9nt7e21CD3  
RI3bKIzY2NoK38H/58qWc37ivx1EOz/Ly8ly8XW6WiUNLhA3RK/ziUCxDt1zIWXyG9BhJ21i  
jZxmit1K6nH8ETB9vkGh+eXmpJIz4CcamD80Prhtq+qBAFpWfuHhzzXvkZ4EZ3WKPMvkcvt3  
m7e3XLAAYZs+r4t1GyTOJ7LkeSiY1zCacZLdaomeazH5LJ2V6ZzPdTMRPhIkfvoOwirEkWGL  
ssTL1B8cmH4fslldUW5jjZ51VDwn3T0+i6c/avRpEm+grUlCWVJjI9Uq2TeD6QYIZVph2Ph  
f0wdjyk+Mgoaadw77ClZBSCTeDkmCRpLJfmyyrQbZcrMsYcee6an/ClswvaQc0sCdahimV6Y  
5ixQE1cIGfxx1+MtBaPBFJOGhmKzs2P0xpG5Q2cacnpVQFhLpIOfvj8GkmzLeKjVPFCYDxUO  
iz4YMOFz1LaWBiQhFF8wGDPKW43Cipt4tiW6935d99Ogr7Sd6dOVzbW1NkSQ4iydpD3Q2Vld  
Xa95r3/m7xsoXK06dy8vLi4sLvVdOb6xzw1B+VvuTJ08wzs/P5RGVr/6ogYChz8/XW/8mpbK  
NOT3L0LX//vvvkIjI+cfBlPFWqXnyliOmimCNC9p1dnYwEn/88fDhw2AlgBlka2trqxjz3B  
N3cc7ZiHfPxeLEEx2dnaW15dZX8RvepGsuY6aQ++WiYPLr4n1+4r/YQyHj6BJJvsrY33ee3  
+Np8+fdLiA+k4PdkNYqVolI/smMtO2Vrf6HaR6YAY+JXhS4FsiLWYPBANka4MS1JcyVmGg+S  
wLfxadz1KES3r6+ttQ4QNI6JjGxM9Yyh0106mHmJ5R8mk8dY8rvRWuJHBcMdAxwiDbldXVzV  
nWwVY13DKqLNVjerivqbVFJ3SmtEftB9C4gYWbuwaLUK60M0vjusXys/WkL4huXRANrtcEG+  
t9FO/rkWwCWex8dclrKTO6bHK22HFJUH8PUSC4McmsJSz0gmIQFK2NRMnGSwPwDdbtzurBX2z  
/1UG6KYN9Zerr5ubE+x0SDtVklFxxqESgYrFEJiBaHhzJbWex4a38MqVuTNsZ6J1hBjLhgzt8  
3Q0pnTBpvtTSnr1mGARhYNxSrbDXgrPQnxMPXRH28WxLdQiLxdo4aD+gYXAUih7Pc81hTiRX  
3Z00S8vBpMwGyYjm/GWFDQR99nr0Ea3I4+Lgv0ZD48NgUbxWVTqCsaU4Gqyf2G+yXwzZ9zEi  
gEttX5U7rodpgDQtNjinnkWDE0ZJae003QLc4s0QboPeyl2D1T5M2JnoazLawiUMu5x+AIaU



zJo23hpoHaxA4gGANghSrFKfogo0ChtCyka15H+8W9hWsgoH+X3ZtbY2d0f6dnZ346uL6+jq  
fnz59UrIhFxcXfL5584bP79+/F77An3/+yef19XXOv7q6yrff8SxV/V58Rjg60jo4ONBVIyD  
yTk507NJH5TWQKS+MfPz4kR2dnZ0pyqGmQpTn8MjAV/n06dPgnW3KlzeJBPUBIKnr8y3o/Fc  
CM0i5jWU9DaJX/QvsXgz5c/5QbFxmH29LzRmvTk9PQ2J0bGxsoAmKSZ/4vtVExCNSV318Soa  
YZdFLa4r9/X1WE7qrKnTxnal30kmCeojVLPeS4grQnP/4+BjRmfJ1G0CbZgeNO4oMoZnPliI  
EYlmuSmdDKLi1tcV3FP/sQgME6xIly7Rseffu3fv3750fZsWS5+fnWtKVYfCi0c0bUvkzing  
8HQF3trFGT/UvYeesxEzOH4qNixbxVq85IxXzx8jCzKDVrLpsYcGCjKtsSeFUoXzfetI+3jE  
KejM6h5rtMpEodvt7d5pCIM6GIiBbGT5EV4zZhMcuC2DbpQCcVrDSz17o4XJiqBLr81bn9LD  
3YE0Ix5woBhwhFSIFNlvj5utCXKUTg9rUXtNZLcVvmWOFdZ1KZeOdKikDtLvY0xwKBmtAaK8  
1JIaDsSDJgWISzaDh5sHoMGxqGFi3+jbm9EyIK4nJ+Xti+JBrF285zTFi/cnWRPzpGVI3ZLG  
209gkPEgmnkoYvhAnJG5o0se7JdEtJPP0e1GoVlhMhHRBEjFBW2CiJH5DU4IrvKU758Eb+Zk  
erE5zamoBJTuhXW06DMiJI78mOWFRVem0iEpTjVZHRBstgspzLalQVOvQXtjpvVBWsoOPI  
kMEyrJJAAP2Wmgcom4vaSR874m+oV9hWsocilSawnkK3stJBLyPl7gmML1iC0jjfAVk7TPO6  
DAo829Q37CtYgqHUQC5Lrg9jKZtpCrEyN5n3DTonVEF7Qs+RvOOqa0Uu6vb2d/GNFQzzY2uG  
6tcala4fr1o5Et4F+/eSMjI8fp87L754cx3HuEZ9lnQm4vHnwxdnZ2bz87slxHOceCSe2jJt  
KO47jOI4zDfEV49+zrF9/7xaXNIcr0w7XrTUuXTtct3YkuvkVY8dxHMfpc591HcdxHKcvfJZ  
1HMDxnL7wWxbmWL55W5+x3dHzljcKQqKEfj880+/+6xxklJ5Kfole9pc81i6Bb6SsW1y8qy9  
opqDJoXk3ha03703t5cKpRtvR62Yk8QbN256Utd/2A5XIOVbKuomcP2aasj3ys8AMpyumkVQ  
PbNJTtmQnj5KZa4YPNj0+Jn7KGh6T9NWRV7kHsNnjY8r6D/b8HYPDCfb/oE/8xJzyw4wkaWV  
Ylms7vG4wpHRQjjfRPO2VZS1JDWz161Cyb2ZEt5w/ZpqqynZPoFhKJ15meaSSNZ1lgzIqT887  
AwaZulgxwsZiaSkOiCvS3aUNQ5N4f29YryTj+tvQqWfSERBZRo+296AZDS1cZb9Ck7ZV1k1J  
kGOVDPX065fwx05Ttg0S3kEi8zvRMIyn9iuKEiJJ0KpLJwDe/DBxs5TKyAbXrh61yDSRpBdh  
3NAAD6xbDqUDcUhQjGhm8ElnKJNrei24wpHS/YqVKLiZtz5WN4bsY5Syba3vOH5PLk/P3TaK  
b35eda66urvjUK3KX15f39vZ0j0F3aOyWQ3yTzPJAcBVlySxbpfg0j24a2V2f+FaQvR9QTvL  
YTTv5ZxYO+OTkZH19XUcLYcMntOXs7ExvYGzO/v4+nYfbjiHPBBwrSBS/KhVJUazJG+vK2o5  
et5p4u7Ptd8aqUX6z27yTa3sTTaYpOxC/ptp7XSaPlWkkjc91dc1NKzL6JzYUuX7B+QSZZbN  
V10bUkxOnysrWiQiGdiRPUXE4Y2bX2PEBUKecQtkw8GNMBKWC1T9qoBoL2PHaFj8eMLkqKrb  
E1StiISM1+oYdBWtY+IotxsDExmJjAVaXaDqkbsK9g9Ux9vI1c2xuWtQwDwDEE2dybW+iyTR  
le4J9BasgJBKvMz3TSKR4MOJxCjue2EKOG5QTo3L8YqtiirrKfx06yWXQCe2QThiZaK5gcTEN  
U1TBweLF6ElarB4O+h5NNIV2CDDX9E6FaiNCCIXWLQUBTDJXs26fhNbIIMuS0HUW3GEy6JvE  
G1W1vUjBOMAD3rlsTTaYp2x0Jbn7FeHaxgLi4uAiu21wW/yOhPIKcctazurrK8Le2traU+Y+  
CpJK//vqLzyY1zzgPHjwIVsTx8TG9MSQmZ9zvJkouF8dHs1s4ODg4OTkBiM821qmRttFeKd  
TZbxBk7aXy9JVT09PQ2LU5HTL+WOmKdsTPsvOMRr7kjFOzuvrayVzMPwXK7OmZqy0264x5f5  
sQ+0csbKywmI1WR+UG8JE0roTnp2dje8+mXF+fr6+vh4SN2Ej7Fy2/h5tTttR6tYw3irbX1+  
WTsrcPI99sAm5tjFRc5qyw+Cz7HzDSMe5RUgUq10+cW5tbV141c9WmZj13N/fJ70cMW/evCF  
ArSBnMG+Li8ZzBxMADbSG707ulhuiB3E0+T1PmWnKzGWHh4cvXrwIicnJ6TNW3aaJt5qyWgf  
buybJk0we806u7U30nKbsQGhZaobTfa01ja+wESXBWxC8t/269SXie2bBde00atmExzLodAS  
PkqD8sYfoTDy6ySGSg7wTigRrKEwiNQti/pYcPxp5pAiw11EeQxBnrMGnbp4HdBWsoaK1JUYa  
2x1s5PCVz2t6XbsAegzUI5XiraTueWMZYwFLLYwymHvsK1iCU2y4q/STrdRM5f6+wu2AVhBf  
0LPkbjrrGJc3hyrTddWuNS9c0160diW5+xdhxHMDx+sJnWcdxHMfpi3Biyxmu0o7jOI7jTEN  
8xfj3LOvX37vFJc3hyrTddWuNS9c0160diW5+xdhxHMDx+sJnWcdxHMfpc591HcdxHKcvfJa  
dJ+L32Rn2cJOuoE575iKVLy8vy553aIsUC+1Iz5on8ebyNck7Dsq6AVERv2kxwV6MaIQNi60  
btZYH6rttUvYyeg3lWHXLxUwsWs0TrwhI5UmEnY14+1lghtMV/UlKzfYcEwySHT4Lhtqsfj1  
0Jn6vRSdQZ7CGQo9wShpCG+3ZMWz9fPPYrJhcnizl02dGdLPHGJkCCZT6EL2Eh/yW8150A/Y  
VrEGo1A2a9NPKspZulxyfbrmYwW8BQ9vLkgryWDYMDV8wI/EWEoOpuTj0Jyk1WxgBkZcLvnB  
E9ddEdmsGDjYNW+UBDqflurg3xuTyNCnbOTOim6DJuVYnYxn62AB6L7rBkNLLdKPh8URSSWX  
ZpBQZkC4kemYw3XIxygDYAHlyx4PfvKKSyH/FuItJAZTc3HoT1JqruZ7L1FNNB2ruSHlv  
ZgXN5mpTtg2H2Y1TqZtRvjSF+NFbel24wHQ5ZXByGBDPHAM5sjHoGXf/Xh1StxiLmRjm0Vz  
DyW8zK3mUbXbize/LzjcbGxuE49HRUUGX2C2KveiVdnbbw26nxTc8zDkyLi8vt0501tfXQzt  
vbvacn5/TM2UbyV2fXJ4mZeednG6TQowRjXrd2ILrtr+/z4DLBIAI1Tdlm2u+Mt43LUIcMwb  
Os70z169fh/Rtdnd39QZPs11dXSnB7MSbz7Jzyc70jvohsUjXXY3eokWAnp6e4mT2JZtes8U  
kqpw46clyvz50hZ90AnQooJR8fXrVz7peDQtsMe6nuiWrnrjpcPMQmIBqNFN0wYTAN2NKAH  
c3ZpoTqlnz57F/X181GOGrcna2hqiMbgf1202NzcZyhjumG5ZzQTvzOCz7Fy2ZGVWRi0oxM  
yj+JnAtY6jgUgTiZRBZ/mWr2o8uLigj7PjBu/oXZkfp/+HR2s43348AEpRnb+1Add6XZ4ePj  
kyZOQWACA6Kb3njKXhPQNTcoy/ViGsVKOGZrMqPX8+XMGn50h1KGUTiFyM/E94rPshMM5K1E  
VXxa+vr507usQoDjD5tsQr0zGTMNEZ3CnNqCPhshYwVmJwy2JkotUuTxNyo4M020iWN7F1/5

cN+Pp06fBylMuyykDXT4kRkoSMzHHx8c6cyjDST9nsZRixFOSz9mJN5915xgihgXvzs40oSk  
PPZN5V7ah7posiklubW19/vx53Etj9bSk7eima26mG6cRnF7INnJ5mpSdd3K6BasZnK6tr6+  
HhOsWwdKWnCFxQ31ZFtPMzZN+BXNHEjMjTMDlxQeKMeiZ/+joSGPgDMWbznjMcLqiP0mpWVe  
MhUKH6VZJlnvx6Sxb+cQJ8gBORfJUik9sZlxtjesnZ1ywE6g/WINAE0wQDGsahvySgk/5Y3J  
5mpTtnBnRTEcXrTmInESZe9ENhpSuXjegu5EnJG5TE6txPWQbZchBOWYMDNxEihlmxCqbN9  
vvIXE8GqOnj4kJWio1gjeY19CM2VIFCGDEIjBdeOkrypphqZbwb7M3+1ES4XBGgp6mhoSD1V  
gDbQVBpCMO3N1Hsj5+4N9BWsoKXNTgGXyYIUd64YscdIYXjdg8EahLJuNFYeQAE5BZ5YqHJ  
ZU8xIaugP9hWsQSjHdCKoyVCvm/JAUsMsxFt4Qc+Sv+GoalzSHK5MO1y31rh07XDd2pHo5vd  
lHcdxHKcvfJZ1HMDxnL4IJ7ac4SrtOI7jOM40xFeMf8+yfv29W1zSHK5MO1y31rh07XDd2pH  
o5leMHcdxHKcvfJZ1HMDxnL7wWdZxHMdx+sJn2Z1DjxcuE78tS3ns4WE5VBDIube3lzy8bdf  
AQKkR0gkLy8v1b5tBOPUC+wp15+hVTXRDLiUN+Ssh6irlzfnHRDneKmOpkQsMaYk3Nnr55S  
auCJa5X1Lpey5Pc8175GeBGU5XTcnp4+i9xMKS9nSY+qeZxDWoyGAPGkuHIWnWUOhxRckTrOx  
xPMnDYhKSbwGozYqwtDsHY9XAoQZrKMq64dFTOUX5YT0xKl7OkPP3x8DS1XUT5Vgqu1nWktT  
AlsEeYzSYbjVx9TZ6YmKu7TV5mmjeOYluITGYmovDlJLSr2rigzDKBzxBmI3pItoi5P3yMD  
BpmErJyad0/pnGWSMO7+IZZTOIdEzs6BbEnKMBmV9DMpCwd6cvz+G1C4Xb5WxlFBZNilFBnv  
4Yt8MpltnXHEMtjWeTWNyeZpo3geJbiExmJqLw5SSxrMsgRKP7ED0xIFVJtFVZ4GBg40uV9k  
zxZ1bOVpI9Df4asY35I16ZQRRWqMM8VmetXP+XhlSukK2iqbhLEKp7pJSrmwMmo815AyLKw1  
0ckKSFdv5mmjeB3YAwu/LzgFnZ2fBasyjr48YyA4ODpaqbmbE9yrircF1+wYGSbvnoXuQy8v  
LSuJXnplFb8VaX1/XAUPY0Ay9PppBjW5fvqmDdHwlr1+/DukR0UQ3mp97FSjFUUavHovJ+Ud  
DjW71sQTNy7X81rwxecfV+fk5cslvIFSwCmry3Kn5MPgs07tojgSM4JoEIozzBoyltTUqsda  
kiPFoccdM/PLlS/nJwzIQJ5vYo4ISJ5+Hh4f4gdCna5yeniRbzs70x48fi9IzytevX/m8urr  
S8WNP9KMbdXXmUZRBk3hFgj7IiBNBgmtENNGN0c2CJ+Hdu3eV7y30+UddjW41sSSaaE6pZ8+  
ejXiZAJvXNS13aj4MPsvOLna9FyO4JmRzc9OKs5pTk03u7rK4U/wx5F1cXGBwVvr8+XP1XjY  
xPROUTMwUVvX+E+gRqYXKmK2Vfrxxbn2UPy/ft3jtNGdtrF6UKyFm4Cyiaj/T+kb5bJiIYGM77  
UaEET3Vh7PXnyJCQiUOPvv/80iYicf0w00a0cS6JJWaYfyfzBWcen1DTnNh8Fn2Tng6dOnwcr  
DtGfQM8Eu59ItmQkw3r9/z6em1QRW0MEqePDgAZ8/fvxQ0ri+vk7uG81Xn1e721H5LRwfH2u  
1MW7KurHeyl0uPjo60uUTYK3GVIFB/pw/FBsjuXhr0qPLZbe3t09PT0NipCRxtbKyorFLMP7  
wmURdkzzQRPOe8F12DuCUtHI4iwmTXoEy//fff9oEefjKaV1MECdzqmDUC9Ynf/75Z7BuoOe  
Xs80y6oFquHGnmJVw1155Pww9p5m8Z5M7de00YH19PSRuw8ojBGJxHUXLMk4mcv5QbBQ0jLf  
KWKovy6KZeaJd6M4RSVwpPGwpuk+kSPbaJIHcv13CBT0ZjhdMaWknCERKYFRQr+j023USrS  
4sxqU/FD8qJ1Ps+Ht27dsVQY7T6WglY0zAwcWn85atuZQYbAGgS00A8agvbIFHttaAwpUthR  
/k+KdMFO6EQbETEjkiSuJyfl7Ykjp6nWDXcXBrixGXA/Zmog/PQOHHJTjioZLE41R1Q2/M0+  
N5n2Q6BYSw6s5elpLSkBQ1ihHDOEstuX/V4ds1BNXhr223d6F9V4FqFC1sSfu5MFVEFyT0K7  
UNNADdbRkX+LWgemMrR6rpYyIdaaS4M3r3wfsLlhDUakboIwkiiFb2Yk+ZSfk/D3BsQVrEMq  
65WIJ8MRS1MuSXx5jsKhjX8EahMq4AlMgPq8gGWcu56nRvG/YY7AKwgt61vwNR13jkuZwZdr  
hurXGpWuH69aORDe/L+s4juM4feGzrOM4juP0RTix5QxXacdXHMdxpiG+YuyX3R3HcRynL/y  
KseM4juP0hc+yjuM4jtMXPss6juM4Tj/88cf/AxD47hTCKfacAAAAAE1FTkSuQmCC)

Hasil evaluation metric bisa dilihat disini :  
[https://docs.google.com/document/d/1IZJ\\_LHIhJSF\\_-ZleAh3TzgvfVPMctrnW/edit?usp=sharing&oid=110045745365014605980&rtpof=true&sd=true](https://docs.google.com/document/d/1IZJ_LHIhJSF_-ZleAh3TzgvfVPMctrnW/edit?usp=sharing&oid=110045745365014605980&rtpof=true&sd=true)

## Test 2 : Random state = 42, Balanced dataset

```
# Commented out IPython magic to ensure Python compatibility.
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(color_codes=True)
```



```

from sklearn.model_selection import train_test_split

# %matplotlib inline

diabetes_df = pd.read_csv('diabetes.csv')

diabetes_df

"""#Exploratory Data Analysis"""

#Column in Dataset
diabetes_df.columns

#Show data types and null value each column
diabetes_df.info()

#Menampilkan 10 baris pertama
diabetes_df.isnull().head(10)

#Checking if there is null value
diabetes_df.isnull().sum()

#Checking if there is zero value

#replace 0 value with NaN
diabetes_df_copy = diabetes_df.copy(deep = True) #deep = True -> Buat
salinan indeks dan data dalam dataframe

diabetes_df_copy[['Glucose','BloodPressure','SkinThickness','Insulin','
BMI']] =
diabetes_df_copy[['Glucose','BloodPressure','SkinThickness','Insulin
','BMI']].replace(0,np.NaN)

```

```

# Showing the Count of NaNs

print(diabetes_df_copy.isnull().sum())

"""#Berdasarkan Jurnal A model for early prediction of diabetes
oleh Talha Mahboob Alam et al, yang bersumber dari :
https://reader.elsevier.com/reader/sd/pii/S2352914819300176?token=7D
4F21A1900E1B47FA971E110A1469CE2A8882E14CFBF3A4C3790507841542704D1370
C180ADD4341ED02B627B619B3F&originRegion=eu-west-1&originCreation=202
20928035824

>Data cleaning consists of filling the missing values and removing
noisy data. Noisy data contains outliers which are removed to resolve
inconsistencies . In our dataset, glucose, blood Pressure, skin
thickness, insulin, and BMI have some zero values. **Thus, all the zero
values were replaced with the median value of that attribute.**"

Jadi di tahap ini yang saya lakukan adalah mengisi value 0 (missing
values) pada beberapa kolom tersebut dengan nilai median untuk
mengatasi nilai inconsistent

"""

#Fill null value with median
diabetes_df_copy['Glucose'].fillna(diabetes_df_copy['Glucose'].median()
, inplace = True)

diabetes_df_copy['BloodPressure'].fillna(diabetes_df_copy['BloodPressur
e'].median(), inplace = True)

diabetes_df_copy['SkinThickness'].fillna(diabetes_df_copy['SkinThicknes
s'].median(), inplace = True)

diabetes_df_copy['Insulin'].fillna(diabetes_df_copy['Insulin'].median()
, inplace = True)

```

```

diabetes_df_copy['BMI'].fillna(diabetes_df_copy['BMI'].median(),
    inplace = True)

#inplace = True -> untuk menyimpan hasil modifikasi pada dataframe
    diabetes_df_copy

"""#Berdasarkan Jurnal Deteksi Dini Penyakit Diabetes Menggunakan
    Machine Learning dengan Algoritma Logistic Regression

yang          dikutip          dari          :
    https://jurnal.ugm.ac.id/v3/JNTETI/article/view/3586/1646

"Pada tahap ini juga dilakukan pengecekan terhadap data yang
    tidak seimbang. Penanganan terhadap data yang tidak seimbang
    dilakukan menggunakan synthetic minority over-sampling
    technique (SMOTE)"

Pada tahap ini yang akan saya lakukan adalah oversampling minority
    value, yakni value 1 pada variabel outcome yang berfungsi untuk
    menyeimbangkan data yang nantinya akan digunakan untuk prediksi

#Berdasarkan Jurnal SMOTE: Synthetic Minority Over-sampling Technique
yang          dikutip          dari          :
    https://www.jair.org/index.php/jair/article/view/10302/24590

"However, the nature of the application requires a fairly high rate of
    correct detection in the minority class and allows for a small error
    rate in the majority class inorder to achieve this."

Alasan saya menggunakan teknik Oversampling adalah untuk meningkatkan
    sampel kelas minoritas dan untuk memperkecil tingkat kesalahan di
    kelas mayoritas

```

```
#Berdasarkan PENERAPAN SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE  
(SMOTE) TERHADAP DATA TIDAK SEIMBANG PADA PEMBUATAN MODEL KOMPOSISI  
JAMU
```

```
yang dikutip dari :  
https://journal.ipb.ac.id/index.php/xplore/article/view/12424/9491
```

```
"the model with SMOTE is more accurate  
than model without SMOTE because has higher AUC value."
```

```
Keunggulan dari menerapkan metode Oversampling minority adalah bisa  
meningkatkan nilai AUC
```

```
"""
```

```
#Counting 1 and 0 Value in Outcome column  
sns.countplot(diabetes_df_copy['Outcome']) #membuat bar plot  
perbandingan jumlah value  
print(diabetes_df_copy.Outcome.value_counts()) #menampilkan jumlah  
value 0 dan 1  
  
from sklearn.utils import resample  
#create two different dataframe of majority and minority class  
df_majority = diabetes_df_copy[(diabetes_df_copy['Outcome']==0)] #  
semua data yang value outcome nya = 0  
df_minority = diabetes_df_copy[(diabetes_df_copy['Outcome']==1)] #  
semua data yang value outcome nya = 1  
  
# upsample minority class  
df_minority_upsampled = resample(df_minority,  
                                n_samples= 500, # to match majority  
                                class, menyamakan jumlah value 1 dengan 0  
                                random_state=42) # reproducible  
results, random state 0 is better than 42
```



```

#Random state =
Mengontrol pengacakan yang diterapkan ke data agar hasil yang
didapatkan tetap sama

# Combine majority class with upsampled minority class
diabetes_df_copy2 = pd.concat([df_minority_upsampled, df_majority])
#menggabungkan Outcome 1 (minority) yang sudah di upsample dengan
Outcome 0 (majority)

#Counting 1 and 0 Value in Outcome column
sns.countplot(diabetes_df_copy2['Outcome'])
print(diabetes_df_copy2.Outcome.value_counts())

"""#Berdasarkan Jurnal Prediksi Risiko Penyakit Diabetes menggunakan
Algoritma Regresi Logistik
oleh Qatrunnada Refa Cahyani, Mochammad Januar Finandi, Jathu Rianti,
Devi Lestari Arianti, Arya Dwi Pratama Putra yang dapat diakses
melalui
https://journal.literasisains.id/index.php/jomlai/article/view/598/470
, kita melakukan tahap preprocessing yaitu pengecekan outliers
pada variabel. Outliers dihapus dengan menggunakan Z-Score.
Beberapa data dihapus karena nilainya jauh dari Z-Score. Maka jumlah
data yang awalnya 1000 menjadi 938

#Berdasarkan Detection of Spatial Outlier by Using Improved Z-Score
Test
yang dapat diakses melalui : https://booksc.org/book/77788899/9f8c40 ,
"The zscore test has long been used to detect outliers in data."

Fungsi Tes Z-Score adalah untuk mendeteksi apakah terdapat outlier
dalam data.

#Alasan Menggunakan Z-Scores

```

```
##Berdasarkan Outlier Detection in Multivariate Time Series Data Using  
a Fusion of K-Medoid, Standardized Euclidean Distance and Z-Score  
yang dapat diakses melalui :  
https://link.springer.com/chapter/10.1007/978-3-030-69143-1\_21
```

Z-score technique produced a better outlier detection result of 0.9978 F-measure as compared to inter-quartile of 0.8571 F-measure.

Alasan saya menggunakan Z Score adalah karena Teknik Z-score menghasilkan hasil deteksi outlier yang lebih baik

```
#Checking Outliers using Box Plot
```

```
"""
```

```
sns.boxplot(x=diabetes_df_copy2["Pregnancies"])
```

```
sns.boxplot(x=diabetes_df_copy2["Glucose"])
```

```
sns.boxplot(x=diabetes_df_copy2["BloodPressure"])
```

```
sns.boxplot(x=diabetes_df_copy2["SkinThickness"])
```

```
sns.boxplot(x=diabetes_df_copy2["Insulin"])
```

```
sns.boxplot(x=diabetes_df_copy2["BMI"])
```

```
sns.boxplot(x=diabetes_df_copy2["DiabetesPedigreeFunction"])
```

```
sns.boxplot(x=diabetes_df_copy2["Age"])
```

```

"""#Removing Outliers using Z Scores

#Berdasarkan jurnal Robust statistics for outlier detection
yang dikutip dari :
https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.2

>Data preprocessing is a useful step that helps to remove
noise, inconsistencies and redundancy to achieve high
quality data which improves the performance. During the
data pre-processing, all the missing values are filled by the
mean of the corresponding feature. Formatting of the given
dataset is ensured to be consistent. All the incorrect data
types of the features changed to their required datatype.
Data normalization is performed on the given dataset to
make its range consistent. We used z-score normalization
on all the features to restrict the range of values between 3
to - 3"

"Generally, some of machine learning algorithms does
not handle categorical data. Diagnosis feature in this
dataset contains two categories M(malignant) and B(benign) is replaced
numeric with M by 1 and N by 0."

Menurut jurnal tersebut. ambang batas Z-Score yang dapat diterapkan
adalah antara -3 dan 3 untuk 2 kategori diagnosis (True or False)

"""

import scipy.stats as stats
z = np.abs(stats.zscore(diabetes_df_copy))

```

```

data_clean = diabetes_df_copy[(z<3).all(axis = 1)] #print all of rows
    that have z<3 (z score below 3)

data_clean.shape

#z<3 = print semua data yang Z-Scorenya kurang dari 3

#.all(axis=1) = melihat di bagian kolom untuk setiap baris (melihat
    kolom z score setelah di proses dengan print z kurang dari 3)

#Cleaned Outliers data using Z Scores
data_clean

""">#Data Correlation
Berdasarkan Diabetes Prediction Based on XGBoost Algorithm oleh Mingqi
Li et al 2020 bersumber dari
https://iopscience.iop.org/article/10.1088/1757-899X/768/7/072093/pdf

Setelah data preprocessing, kita harus mempelajari apakah data tersebut
diproses dengan baik atau tidak dan bagaimana caranya
banyak korelasi antara data tersebut. Koefisien korelasi dapat
digunakan untuk mencerminkan kedekatan
hubungan antar variabel. Koefisien korelasi dihitung dengan metode
selisih.
Hal ini juga didasarkan pada dispersi kedua variabel dan rata-ratanya
masing-masing. keduanya
perbedaan dikalikan untuk mencerminkan tingkat korelasi antara dua
variabel

#Berdasarkan Jurnal Prediksi Risiko Penyakit Diabetes menggunakan
Algoritma Regresi Logistik

Jika nilai korelasi > 0 maka terdapat korelasi positif. Sementara
nilai satu variabel meningkat, nilai variabel lainnya juga
meningkat. Jika persamaan korelasi = 0 maka tidak ada korelasi.

```



Jika korelasi  $< 0$  maka ada korelasi negatif. Sementara satu variabel meningkat, variabel lainnya menurun. Ketika korelasi diperiksa, ada 2 variabel yang bertindak sebagai korelasi positif terhadap variabel dependen outcome, variabel tersebut adalah glucose. Seiring peningkatan ini, variabel dependen juga meningkat. Dengan demikian, semua variabel digunakan untuk modelling karena korelasinya berdekatan.

#Berdasarkan Complex heatmaps reveal patterns and correlations in multidimensional genomic data

Yang dapat dikutip dari :  
<https://academic.oup.com/bioinformatics/article/32/18/2847/1743594>

Heatmaps are a fundamental visualization method that is broadly used to unravel patterns hidden in genomic data. They are especially popular for gene expression analysis (Eisen et al., 1998) and methylation profiling (Sturm et al., 2012). With the increasing availability of genomic datasets, \*\*visualization methods that effectively show relations within multidimensional data are urgently needed.\*\*

Alasan saya menggunakan heatmap adalah karena heatmap correlation dapat digunakan untuk menemukan hubungan potensial antara variabel dan untuk memahami kekuatan hubungan ini dan mendeteksi hubungan linier dan nonlinier

#Fungsi Heatmap Correlation Matrix

Berdasarkan Jurnal Implementasi Seleksi Fitur Klasifikasi Waktu

Kelulusan Mahasiswa Menggunakan Correlation

Matrix With Heatmap

Yang dikutip dari :  
<https://ejournal.ung.ac.id/index.php/jjee/article/view/14403/4663#>

"atribut yang memiliki warna merah muda adalah atribut yang relevan dengan atribut aoutput dan sebaliknya atribut yang berwarna merah tua adalah atribut yang kurang relevan dengan atribut aoutput."

Atribut berwarna merah tua yang dimaksud oleh penulis adalah atribut yang memiliki korelasi negatif tinggi dengan kolom output

"dari 13 atribut awal yang digunakan terseleksi menjadi 9 atribut yang memiliki relevansi atau berkontribusi terhadap hasil output yaitu : jenis kelamin, kelas, umur, SKS1, IPS1, SKS2, SKS3, SKS4 dan SKS5, sedangkan atribut input yang kurang relevan terdapat 4 atribut yaitu : IPS2, IPS3, IPS4 dan IPS5"

#Alasan menggunakan Correlation Matrix Heatmap pada penelitian ini

Untuk melihat relevansi antar kolom. Disini target kolom saya adalah kolom "Output", dan relevansi antar kolomnya memiliki nilai positif. Maka saya akan menggunakan semua kolom sebagai indikator

"""

```
sns.heatmap(data_clean.corr(), annot=True)
```

```
#.corr() = correlation matrix
```

```
#annot=True = memberikan value korelasi antar kolom dalam bentuk angka
```

```
"""#Machine Learning Model Building
```

```
axis=1 = Remove column
```

Dataset X = menggunakan semua kolom untuk kriteria dan ciri ciri pada tubuh pasien

Dataset y = sebagai hasil apakah pasien memiliki diabetes atau tidak



ECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgj0f5BK0gQTJEg1xwkGqYpZBHXo/iC  
VpAkmSJBqjhMMUhWzCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqh  
D9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALszXGCQap  
iFkEXWl+StDBGtir58heh4yVJy2Nwq4+vFRHaXZK0PAa3+vjaF6HdJUUnLY3Crj699Fzp  
ekrQwRrYq+fLXoe+DVJImmCBBqjLOMEhVzCKoQ/cHqSRNMEGCVHOCyJCqmEVQh+4PUkm  
aYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUXSyCONR  
/kErSBBMkSDXHCQapilKedej+IJWkCSZIkGqOEwXSFbMI6tD9QSpJE0yQINUCJxikKmY  
R1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnmOME  
gVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgj0f5BK0gQTJEg  
1xwkGqYpZBHXo/iCVpAkmSJBqjhMMUhWzCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkia  
YIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+  
kkjTBBALszXGCQapiFkEduj9IJWmCCRKkmuMEg1TFLII6dH+QStIEEYRINccJBqmKWQR  
16P4glaQJJkiQao4TDFIVswjq0P1BKkkTTJAglRwnGKQqZhHUofuDVJImmCBBqjLOMEh  
VzCKoQ/cHqSRNMEGCVHOCyJCqmEVQh+4PUkmaYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1  
xgkGqYhZBHbo/SCVpggkSpJrjBINUXSyCONR/kErSBBMkSDXHCQapilKedej+IJWkCSZ  
IkGqOEwXSFbMI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6k  
kTTBBglRznGCQqphFUIfuD1JJmmCCBKnmOMEgVTGLoA7dH6SSNMEECVLNcYJBqmIWQR2  
6P0glaYIJEqSa4wSDVMUsgj0f5BK0gQTJEg1xwkGqYpZBHXo/iCVpAkmSJBqjhMMUhW  
zCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5x  
gkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALszXGCQapiFkEduj9IJWmCCRK  
kmuMEg1TFLII6dH+QStIEEYRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkk  
TTJAglRwnGKQqZhHUofuDVJImmCBBqjLOMEhVzCKoQ/cHqSRNMEGCVHOCyJCqmEVQh+4  
PUkmaYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUXSy  
CONR/kErSBBMkSDXHCQapilKedej+IJWkCSZIkGqOEwXSFbMI6tD9QSpJE0yQINUCJxi  
kKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnm  
OMEgVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgj0f5BK0gQ  
TJEg1xwkGqYpZBHXo/iCVpAkmSJBqjhMMUhWzCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4N  
UkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6A  
O3R+kkjTBBALszXGCQapiFkEduj9IJWmCCRKkmuMEg1TFLII6dH+QStIEEYRINccJBqm  
KWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkkTTJAglRwnGKQqZhHUofuDVJImmCBBqjLO  
MEhVzCKoQ/cHqSRNMEGCVHOCyJCqmEVQh+4PUkmaYIIEqeY4wSBVMYugDt0fpJI0wQQ  
JUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUXSyCONR/kErSBBMkSDXHCQapilKedej+IJW  
kCSZIkGqOEwXSFbMI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP  
3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnmOMEgVTGLoA7dH6SSNMEECVLNcYJBqmI  
WQR26P0glaYIJEqSa4wSDVMUsgj0f5BK0gQTJEg1xwkGqYpZBHXo/iCVpAkmSJBqjhM  
MUhWzCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJ  
Uc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALszXGCQapiFkEduj9IJWm  
CCRKkmuMEg1TFLII6dH+QStIEEYRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1  
BKkkTTJAglRwnGKQqZhHUofuDVJImmCBBqjLOMEhVzCKoQ/cHqSRNMEGCVHOCyJCqmEV  
Qh+4PUkmaYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBIN



UxSyCOmR/kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9QSpJE0yQINU  
cJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmC  
CBKnmOMEGVTGLoA7dH6SSNMEECVLncYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5D  
qev48PfbIXMLD0x8e90TPj/zN3/T4zL3rZvFSBqnmOMEGVTGLoA7dH6S6Hjcz3SJeyiD  
VHCcYpCpmEdSh+4NUV/fnxfpz0+Pjb+9pDw+PT8/Pu7vnx5qdx/Xp7hw0+e4e5ZPP3  
Gs9AieGGDVHOCYJCqmEVQh+4PUq3lZQP62QL00cPj81kWoZdnxyN8b7cXuo3dIV7fINU  
cJxikKmYR1KH7g1RL2u1nvFBz488tR075DNal7J7xIgf7/q1j9JnrTD/9YJ//HCeew1  
SFbMI6tD9Qap1/Xn60Xp291/1+vqqa6367QKXugg/V1uZqpkEdSh+4NUa7v4xeIEXzw  
nt7ICvNZB+ut++puN+zds/6kMd79V+fUPRF/+GHvcW5CqmEVQh+4Pui1vuJyd97PMnb8  
8oQf/u8sOvN5Bei4vC9qg/n/xR40/9d0vNhj3GKQqZhHUofuDVLdgtJyde0HafjKuZTV  
4xYP0nE6t/9/9qWTzUX+1zrnPIFUxi6AO3R+kuhGD5ey8S9LWE/FDzCa86EF6XieV/y+  
/Xfz5MX/7/WjuNkhVzCKoQ/cHqW7Hyb8Vfc4PND8/iXM+mhbE6x6k53bCbvbLlfip1H/  
9Rx7uN0hVzCKoQ/cHqW7JyW+cnetts897mZ9i1uGVD9Lz+776f7UYpz3c75c6dxykKmY  
R1KH7g1Q35sQ3zs7xRtb5P93RLeDFD9JL+G43+9V6POq0c9Q6dx2kKmYR1KH7g1Q350q  
7mXuZXvHyB+1lffP8v1iSF1jMnMk6ZhHUofuDvDfoCrvZ54f0Y8xWFECQXsg3b5v9Vs0  
fP8x5fgrhzoNUxSyCOmR/kOomffexDn7revJpLzvPhUo3gRoI0ov55ueSX/mJ4TKLmTN  
ZxyyCOmR/kOpGXW43+/xI7mXVqIIgVzXvSv8MJX+u94e5+yBVMYugDt0fpLpZl9nN/Bh  
TR6iDIL2kr982++fV7Pjuz1bx3H+QqphFUIfuD1LdsPPvZn6MqU8ohSC9qLN+onmxxcy  
ZrGMWQR26P0h10866m32+c/cyrTFJvin8fynUo8XsnDXPQwSpilkEdej+INWNO9tu9v1  
tCT/G1A71EKsXdq63zS64mDmTdcwiqEP3B6lu3j12s437dC/TKwoiSC/um7r/Wb0e3+1  
ZFzNns05ZBHXo/iDVHfjt3ezz/Z33AqWbQk0E6RV8/bbZT2r2uPLP/OMIjxKkKmYR1KH  
7g1R34Td3Mz/G1JeoiC9ht/+RPPCi5kzWccsgjp0f5DqTzvSbrZxN+5l+oC6CNKr+Kb  
qh6V7vOidvfJ5nCBVMYugDt0fpLobv7Cbfb4LP8bUMUojSK/kFz/RPLqrC5Q+jxSkKmY  
R1KH7g1R35B93s88XOfcyFUzxBom1/NrbZpdfzJzJOMYR1KH7g1R35cTdbON6tfE3/Rh  
TWyiPIL2ab2r+1A3raDG7SPHzWEGqYhZBHbo/SHVnfrab+XaZTkaFBokVfa7eQydv8nH  
bXOanEh4sSFXMIqhD9wep7s58N3Mv0wA1EqTX9E3Jf79mXWcxycybrmEVQh+4PUt2h0W6  
2ceMLXZd0m6iSIL2uf3vb7KgHLvZzCY8XpCpmEdSh+4NUd+nk3ezRt8s0RKEE6ZV9XfF  
fF/XRWne5DuABg1TFLII6dH+Q6k6dupsdcs/TdyiVIL22bwr+i/eBr7aYOZN1zCKoQ/c  
Hqe7WD3YzP8bU9yiWIL2+H36iefTXLtkDPGSQqphFUIfuD1LdseFu5l6mU1AuQbqAH71  
tdsXFzJmsYxZBHbo/SHXxtt/N/BhTJ6JignQJX79ttrF1HTfIZX864UGDVMUsgjp0f5D  
qzp26m/mGmU5ExQTpGr4u908/fhzd/NI/nvCwQapiFkEduj9Idfe+fh/hnbuZTkK9Bok  
ivv1J5GORX3kxcybrmEVQh+4PUhVwN9MvolqCdBmnf6J5dMvLf6DPAwepilkEdej+IFU  
FdzP9GmolSNdx6iear01xhdrnkYNUxSyCOmR/kKqEu5l+CZUSpCs56W2z6y9mzmQdswj  
q0P1Bqg7f/PrNIXczfYk6CdKlfl2a7d82W2AxcybrmEVQh+4PUjUY7GU77mb6AlUSpGv  
57hPN5yv/7v8rHj1IVcwiqEP3B6nu36kfZB640pVkt4AaCdLVTKr+WuXOwwepilkEdej  
+INWdG75d9sbdTH9BhQTpck6v/KvVo08fpCpmEdSh+4NUd23j6vT49HjaFcvdTJuojoyB  
d0I1vm13vw3ueQJCqmEVQh+4PUt2xz1em123r1HcT3M20geoI0hWdtJpd8bcqeQZBqmI  
WQR26P0h1rzbWr/dNy91MP0ZtBOMStqjza/7nLjyFIFUxi6AO3R+kuk9bH2N+uAS5m+m  
HqIwgXdQ3b5tdt7x5EkGqYhZBHbo/SHWPNI5Hn98acDfTj1AXQbqqL8v8yrXNswsSFbM  
I6tD9Qaq7s3Ep+ssFyN1MP0BVBOm6/v622bULm6cRpCpmEdSh+4NUd2Zj2/rin2nczTR

GTQTpww62ml3zV8z2eB5BqmIWQR26P0h1V076GPMDDzMNURFBurLtIr/6YuZM1jGLOA7  
dH6S6HxvXn1P2KXczjVAPQbq2+Y8s18AzCVIVswjq0P1BqnuXsWCdeulxN9MA1RCkizu  
u8SVqmecSpCpmEdSh+4NU9+Ef3xNwN9PJqIUgXdxRha9RyDyZIFUxi6AO3R+kugMbe9X  
4yuNuphNRCUG6tiUXM2eyjlkEdej+INXN29ipfvQrNO5mOgl1EKRL06rtBX7FbI+nE6Q  
qZhHUofuDVDfuN3+1+dTdbJkrm66BIgjSlS26mDmTdcwiqEP3B6lu2cYq9W/vaLmb6Vu  
UQJAubNXFzJmsYxZBHbo/SHW7Ntaof7/ouJvpGxRAkK7r49vKK30Yz1MKUhWzCORQ/UG  
qW3W+/wtNG/e8yd2sEy9/kC5r3cXMmaxjFkEduj9IdZN+/WPMj9zN9He8+EG6qoUXM2e  
yjlKedej+INUNOsvHmB+5m+lveOmDdFEfC3mxcuVZBamKWQR16P4g1c3ZWJrOccFxn9M  
2XvggXdPsi5kzWccsgjP0f5Dqtpz5Y8wP3M20hZc9Sfd01C3LFSrPK0hVzCKoQ/cHqW7  
Jxq5011+ccTfTz7zoQbqgj4vZWr9itsczC1IVswjq0P1Bqpux8XbZ+VeiU3ezBS970hN  
e8iBdz8faXbFCeWpBqmIWQR26P0h1I7ZWpEu8VbW1D25yN2vBCx6kqzkq3CXLk+cWpCp  
mEdSh+4NUt2BrPbrYpebk3cwpNTvwagfpWo6Lds3S5MkFqYpZBHxo/iDV8jY3o4u+BXD  
qR5ruZhV4rYN0JccFu+r7uTy9IFUxi6AO3R+kWtvz4+YbVhfegE5fzfxQ8/7xSgfpfl  
5fnp8eG+Fh4fHx6fnPlu19udzzyxbkzy/IFUxi6AO3R+kWtffP0a89KXm9A80d3zj7L7  
xMgfp+bzswjzUhpCd7W1F269v5AfWrUeeYJCqmEVQh+4PUq1p4yf/Axe/1gzeNnt5cDm  
7Y7zIQXous58KPlu5FHmKQapiFkEduj9ItZyXn/y/uxat/Z7Z3styuhHSPpHvMJBeib  
DnwmOrf3h0k8ySFXMIqhD9wepVvKXj2M2XPSdgJ9eHR+e3M7uEK9ukJ7Hvy1mq//SI08  
zSFXMIqhD9wep1nD6ThYXWnu+/lj1ew++eXZveGWD9Dz+ZTNb/yN1nmiQqphFUIfuD1J  
dz58Xz0/TjeyDh8en+fn3R1xn79k/9Q+/udw/+jtmfIIulm8okF6Hj/ezFZ/u2yP5xq  
kKmYR1KH7g1TX8Y+/PrPt398k+Ndfz7NTVw19Te8ikF6Hj8ryFv5L1B4ukGqYhZBHbo  
/SHUdbma6VbyKQXom84q8of8wmGccpCpmEdSh+4NUkiaYIEF6NpMfYm7slxp51kGqYhZ  
BHbo/SCVpggkSpGd02n+EcoP/qQnPPEhVzCKoQ/cHqSRNMEGC9Ly+/1Bzt5Td4gfkPP0  
gVTGLOA7dH6SSNMEECdLz+7P7v+By+F8L7/4/M930/8k8/h1BqmIWQR26P0glaYIJEqS  
a4wSDVMUsgjP0f5BK0gQTJEg1xwkGqYpZBHxo/iCVpAkmSJBqjhMMUhWzCORQ/UEqSRN  
MkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9  
SSZpgggSp5jjBIFUxi6AO3R+kkjTBBA1SzXGCQapiFkEduj9IJWmCCRkkmMEg1TFLII  
6dH+QStIEEYrINccJBqmKWQR16P4g1aQJJKiQao4TDFIVswjq0P1BKkkTTJAg1RwnGKQ  
qZhHUofuDvJImmCBBqj1OMehVzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkmaYIIeQeY  
4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUXSyConR/keRsbm  
kSDXHCQapilkEdej+IJWkCSZIkGqOEwxSfBmi6tD9QSpJE0yQINUCJxikKmYR1KH7g1S  
SJpggQao5TjBIVcwiqEP3B6kkTTBBg1RznGCQqphFUIfuD1JJmmCCBknOMEgVTGLOA7  
dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjP0f5BK0gQTJEg1xwkGqYp  
ZBHxo/iCVpAkmSJBqjhMMUhWzCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4  
wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBA1  
SzXGCQapiFkEduj9IJWmCCRkkmMEg1TFLII6dH+QStIEEYrINccJBqmKWQR16P4g1aQ  
JJKiQao4TDFIVswjq0P1BKkkTTJAg1RwnGKQqZhHUofuDvJImmCBBqj1OMehVzCKoQ/c  
HqSRNMEGCVHocYJCqmEVQh+4PUkmaYIIeQeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZ  
BHbo/SCVpggkSpJrjBINUXSyConR/keRsbmKSDXHCQapilkEdej+IJWkCSZIkGqOEwx  
SfBmi6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBg1R  
znGCQqphFUIfuD1JJmmCCBknOMEgVTGLOA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYI

```
JEqSa4wSDVMUsgjpp0f5BK0gQTJEg1xwkGqYpZBHXo/iCVpAkmSJBqjhMMUhWzCOrQ/UE
qSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVC
H7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBA1SzXGCQapiFkEduj9IJWmCCRKkmUEg1T
FLII6dH+QStIEEyRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkkTTJAglRw
nGKQqZhHUofuDVJImmCBBqjLOMEhVzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkmaYII
EgeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyConR/kEr
SBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9QSpJE0yQINUcJxikKmYR1KH
7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnMOMegVTG
LoA7dH6SSNMEECVLncYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjpp0f5BK0gQTJEg1xwk
GqYpZBHXo/iCVpAkmSJBqjhMMUhWzCOrQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEG
qOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjT
BBA1SzXGCQapiFkEduj9IJWmCCRKkmUEg1TFLII6dH+QStIEEyRINccJBqmKWQR16P4
glaQJJkiQao4TDFIVswjq0P1BKkkTTJAglRwnGKQqZhHUofuDVJImmCBBqjLOMEhVzCK
oQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkmaYIIEgeY4wSBVMYugDt0fpJI0wQQJUs1xgkG
qYhZBHbo/SCVpggkSpJrjBINUxSyConR/kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGq
OEwxSFbMI6tD9QSpJE0yQINUcJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTB
Bg1RznGCQqphFUIfuD1JJmmCCBKnMOMegVTGLoA7dH6SSNMEECVLncYJBqmIWQR26P0g
laYIJEqSa4wSDVMUsgjpp0f5BK0gQTJEg1xwkGqYpZBHXo/iCVpAkmSJBqjhMMUhWzCOr
Q/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKq
YRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBA1SzXGCQapiFkEduj9IJWmCCRKkmU
Eg1TFLII6dH+QStIEEyRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkkTTJA
glRwnGKQqZhHUofuDVJImmCBBqjLOMEhVzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkm
aYIIEgeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyConR
/kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9kqT1MK1VzCKoQ/dLktb
DpFYxi6AO3S9JWg+TWsUsgjpp0vyRpPUxqFbMI6tD9kqT1MK1VzCKQJP0Eq4TLhPSr7Ch
JkqRVuJlJkiStws1MkiRpFW5mkiRJq3AzkyRJWoWbmSRJ0irczCRJklbhZiZJkrQKNzN
JkqRVuJlJkiStws1MkiRpFW5mkiRJq3AzkyRJWoWbmSRJ0irczCRJklbhZiZJkrQKNzN
JkqRVuJlJkiStws1MkiRpFW5mkiRJq3AzkyRJWoWbmSRJ0irczCRJklbhZiZJkrQKNzN
JkqRVuJlJkiStws1MkiRpFW5mkiRJq3AzkyRJWoWbmSRJ0irczCRJklbhZiZJkrQKNzN
JkqRVuJlJkiStws1MkiRpDf/73/8BfrWdG/8VnMAAAAASUVORK5CYII=)
```

"""

```
X = data_clean.drop('Outcome', axis=1)
```

```
y = data_clean['Outcome']
```

```
"""#Metode 90 10
```

#Berdasarkan Jurnal Analysis and Prediction Of Pima Indian Diabetes Dataset Using SDKNN Classifier Technique

yang dikutip dari :  
<https://iopscience.iop.org/article/10.1088/1757-899X/1070/1/012059/meta>

"Our concept is implemented on Pima Indian Diabetes Dataset (PIDD). The analysis on Pima Indian Diabetes Dataset (PIDD) is carried out by splitting dataset in to 90% training data and 10% testing data. We have found that, in our proposed technique, average classification accuracy gives result 83.2%, a great improvement as compared to other conventional technique."

Untuk tahap train test split, saya akan membagi data train menjadi 90% dan data test menjadi 10% karena dapat menghasilkan akurasi yang lebih baik

#Berdasarkan A Novel Diabetes Healthcare Disease Prediction Framework Using Machine Learning Techniques

yang dikutip dari :  
<https://downloads.hindawi.com/journals/jhe/2022/1684017.pdf>

Foremost using

a function like the model selection train test split, the data set is divided into the training and testing data sets. Due to the limited data set source, about 90%, of the data set, is used for training purposes and the remaining 10% is used for testing by selecting the data randomly.



Alasan saya membagi data train dan test menjadi 90% dan 10% adalah karena terbatasnya jumlah data untuk di training

![xytraintest.png] (data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAzMAAA  
HMCAlAAABukmEEAAAAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADs  
MAAA7DAcadvqGQAABmPSURBVHhe7d09dtrM2wdgeNcCT5HzrACvAKdJlTYdlHaTLqW7NK  
Y0XdpUaWJWYK8gJ8UDe+GVRiMQH/bf39xY13VOTsQgJDCD9JuRNOoul8sOAAAB/F/+Hw  
CAQ5PMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAA  
CikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAK  
KQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAop  
DMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikM  
wAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzA  
AAopDMAACikMwAAKKQzAAAopDMAACikMwAAKLoLpflPEk7dLvdPAVAMhbK6DMDAThCMg  
MAiEIyAwCIQjIDAIjCFQcTs3UFgAoAPIETyUvx12SLPjMAgCgkMwCAKCQzAIAoJDMAgC  
gkMwCAKCQzAIAoJDMAgCgkMwCAKCQzAA5sNu6ujWe5FFpJMgMeaDE56Z5MFvkRvIwylZ  
12rpe1687p/05ns7H8xnsmlcGBFXFnW2vj+zxOsVXfjodXM6vhrmg0xlezS8H09P7Ks  
JicjHNk/DuSGZwYgc3y+X1qNMZ5U6DYq90e96PmE96xVu90ev1R69g9v38tnP787do1h  
qL3z9vO4PPHzdrVe/j50FZEwa52VJ2j5WpPU1Piol+UVGmp6s2TP1cYd2Ttiqsf0s599  
cHTss583Tj17ZeVMSfIO0gmUEA/Q+DPFXmnx+XxW7p/HvrDtbMfnWui4wqmrXI/O9tp/  
PvP9txv/fPv5307d+y2VK2W1LR2U3RaCkm/smFZVOmbCgUWar/91tq1RTF04sqUM3G/Z  
+f56ml0znpj2flXGXuP+//+pTnLJjdnq5/bY1F5ZeVpfdGJDMIJ+2WkqgVX/Uc7HQPrN  
v067KkeCKVpN1RKqn3MM0Zc9kdc27MWxXmOdOTjWdXb6NxEvd6KQ+3mPz68HU4/Fp2GW  
6n0sZ7Kd3zp+AdW/8wGmbfzZuXX6tjocOrKqwVVeNimnviemffyrz2+2MKdvVx07It1J  
z+819Rg1IHxhnYcMMq0rhrUlmEM1sfDrdEaf+nUr/6Lzo2jEb3UPrNv0s3H//N90KD  
T3JPzofEn71enpl/KV5Uk7qSNhmflS7MTKfoS6c6Fc3r45myuqZi3SYZoz2fc2in1hfR  
L3/PJDnvERFr//fij3p0lA1vRXM5rtfMA7/xQcm9RbvBuAFv/96XQGH/r54T3SnDtSV1  
yt2SP9P9UnFSSveuwe7iKZQRC5rd7tFrms2D1cDfPhm8Hlj3r/sLdNX+6a8j5s+G1UFp  
Xng1UZJu1ZVj0NjfpEyj1zyb45d/shho3jSnd1LRRFKR/1zs7Wp3M/0Oz730/VW9uJZr  
sfsCzUvfEu1GeUbX55+88+26uss7d/5/1RrQxjG6W7B0zvsNkogEOQzCCIzMO9caHatp  
02fblvy7u2MsPs3wU191LpGpZLbdXOef+fogNu28j5cgqK+30Xy0aRyP3dm7Nfk1zzM  
pBq7GDvPMD6t54B3rlaZW3G32eZR/pbW4sVHIa33tBZpnWp6erly8mk2Kyive5/7eIea  
NPD2krVK9aLWs2dpScw8hbNvojf/G1XMph1X1jGzkjq/rM0vHHUv1w33ypPFk9mY/7bU  
5X86XJoiwveN+czbK1uvSut1FJzZbe9ENcX27OX66puYjdD3j/e+D15S+klkufaP0Flz  
a/19Vzg8vLol5UT5c1JBWt6nC2rjarwqqo+XDvdLXWxjt5q+qVV1fLpbSYStA6+ddfy6  
UcVr13WO9S1srdRLO82m3Ue4zrUXou/7epXOJ6vvXerFFWvKqwd840sSotVntZF6ayfW  
8jz1Mqnt7zju625wOkNWyuYMvePwVvp/jrN+VSHi//BWu51BZTCVon//pruZTDSQmjt  
gPdpRSPs3UUSWSJM/cKElG16uS4oXNhayma4PRaP+cxWIbM5cFzWXuexvNd7HxKe637z  
M2F7Wx6qxe/M574A31v3wt1/J4+S9Yy6W0WLF416sD7dDtdvNUogIcv91k0j9bnZOzmJ  
x8/+fmnhPVjs+7/4BHyZbkpflLssUVAHDCZuPT88bp/Yvfp/990NnOR+Pdf0CAJn1mra

N99u7Mxo0rLQeX83d31eK7/4BHyZbkpflLskUyax1bAeD5bEleir8kWxzNBACIQjIDAI  
hCMgMAiElyAwCIQjIDAIhCMgMAiElyAwCIQjIDAIhCMgMAiElyAwCIwt2Z2mXrNiAABG  
TX3G6bzFpELAM4CjbXbSaZAQBEIzKBAEQhmbWIEExcAjoLNdzU5AqB1tk5fUAGAJ7A1eS  
n+kmzRZwYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZA  
YAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBg  
AQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGA  
B  
CFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEI  
V  
kBgAQhWQGABCFZAY81Wx80u12T8az/BiA55LM4E0tJmWYuctGxpmNc+mGk8kiP39ws1  
/T2+K/2+1FnPcEcoQkM3hTvbOb5XI+vx4NBrmkMricL5fLq2F+WBpeFSXFnPnhYHR5Xc  
xzc9bLBQc3/DQqP8Rg9C3OewI4cpIZvL1eb3h1czO/bISzf//ZH25yaRHcbq7Oho8IQI  
vJ+NV7sopPUYTHm408CcBzSGZwKL2zmlWHWgd6uu8w5WJycjpNseyxvVKz7+d/8iQAx0  
MygwMaXq07zm7Pv2xls9m4f37bGV0//gDmYnIxzZMAHBPJDA6qd/bjjmw2G1fdZy8/Vr  
iYfCkSHQBHSDKDA2selFxls+ow5h06yxaz8UnZ0wbAUZLM4PCGV5vZLB3GfHx3WRHnTv  
qnaSSL0u15f3ekjcvSug5CVg3PsUjT5Yhkjb66NEdjaI+T4tnZ5mHW2qLMgXsG8qhWUh  
dXM1XLGt+xJIjFWH0c0JKWyV98LZdycOuLAZLRdS5/vPrUtWogjtr8+jKNcVEpFr+5wj  
xzfm390uJF+TWbCysH81gP+9F4r1leFq8H/Vh5xicjkPx11nLpe7Gqt1sV/zXkNdVyKS  
2mzwxiaF4MUOwOvr7sSBSLyZeLn38aBzl/jS8+FLucOq8NPn/s1ZccFGv/kQ+i9ob1aX  
C359/XvQez8ZeLP53dQ6blWv7crst/F/N9KAdhK8wv875u+ks3BNEZq48Dkswgit7Zt7  
pXa/c6zWfqnd0UVt1pnWnnUzqFrYhe1Zhk5YPFf3sG2uj982+eWithYyuWttMfltayKr  
79+ffTzU09CFvv7FMu//OfI5o8h7H6eN8kM4iiHOpi1d/04t1s0+jTnj107+Pndf/Zjg  
cHqv6HKv4NPm92/NXlt3/n6X94EmP18c5JZhBDGupidD1fdWu9cjbBj906an3/p3SCf3  
mN6B3qqAVvx1h9vHuSGUSQzvAaxV8Ne40Bzg6QzSr5As4vX359+vFj9xx+OBRj9dEckh  
kcXBq8bLAAJOog2SwNcdE//fnh23x5c1MmxfwEHFhZN9M1KvC+SWZwYlm7rDmm7KGyWR  
ERY/HQrtfL1Yn78FzjajC7huYoYbPm03cOH2asPlpEMoND2uouWz1ENqsPFO29OgCe6m  
p9TXA5JMt8uWxW92H9dPnMnZdClpf9prFXqgWtRxxkrWzQ5kvVPz3N0K9JeP0/ftuvfT2  
p3nE87n/MwLteXg9vi2dP+VvpKZ1d2+40cWMnlq5UUj4th67A4Pe0bl5YXIZnB4eTusn  
r0sA1vn83mf6udjFEteGHNO5DtUda89SB6j2WspT4byQwOo2xxplTjXt113HAjm2037B  
/g9ufvqq+gaOs/5sXrIjH6CTauhFtMGkd/6ih3B6NjsDb8uicGVWa/it/B00d0NVYf74  
1kBm+tOo01X41Gcd+mfnGMPmU4G88WD9jw52HJ0kvKE2D6Fx+qfoLF7Ht9/vS+xn2971  
y9ML3ypo6HqfBL52MdJBf1DjUpd7br9zX7/XO9ksbbXdxRTjuUAYmnie2614LZC5BN1  
YfRy/1xdIi+Yuv5VLeynZTvLJzL8nGiTnb1qfY3K28fWwee1Qdcdl/MGlnWfPVzTUHo8  
t8rKZ4bSqrF5VK0jybdu7FmZSr2PdpHvIxCCx/j7Vceq/ds8QKZ158L1U9y6hsqpkD1  
rY/K47v2Z3rOiu9T9u5VuqV67kUlpMJWid/Ouv5VKAx8hbkFouvd9ugEk1D48zdyWjwg  
PDUT77bFC2M1b325fMCMXRTADexM4RzfIQ94fvv8u+SwCY/URnWQGwBupT2acXpRXma  
Rgtv/Ur5dmrD6Oh2QGwFupu83SZSNvFsyM1ccxkcvAeDvDPMTE9LQcze/po2U8hrH6OC  
aSQBvaD08y5P7sIzVx7uWrwSgNfIXX8ulAI+RtyC1XPpA+aLIx1/JuLoIMqsvklxdZ1  
nYt9Tt11Wv3ChtXG+5ubT1ZzjX6/mbxc3lbJQ/TH51LZfsYipB6+RffY2XajxG3oLUcu  
lDpSzzlCEm3uFYfflvtVxKi3WLF7k60A7dbjdPJSOA8ATP25LMxt2LD/N046TWs01mi/  
PMAHhTi8nF9I1Gy4DjI5k8JbecBgzoEKOZraOnnPg+Z6+JSkHff37bX11aLGKbTJb9J  
k8B8rK2yJ1uyfjNO7/5Mt5561uyATHSJ9Z62ifAc/3iC3JbNw9bYwcNrrWX9Zkm8wWfW



```

yX56oMykE18nP3aYy1njTXsf1cViw6z1DK6y3X1x7nIdI33+t9a+Edyt9zLZc+3boC3X
s15R6rV+Y6V/6g8tT+6l1p1NCyhm9V2OqljcK71/Jc1WJXciktphK0Tv7113IpB7K/Yb
6za7pnD/OQvcM6+63SVUP57Hrxe+4peD0q3lAKkPVse288eP9aeGeqr3ollz5HbgE8Np
jt/jyav4k7fjqR2h55ebVcSoupBK2Tf/21XArwGHkLUsulz5Kyz+ODWeGo2x7VIldyKS
3Wlf7l6kA7dLvdPJWoAMATvMKWZDbuXnyYNweMaQXbZLa4AgCAwyvH133J0TLgWEImAB
zcSw9jBkfL0czW0XMOPN8Lb0kWk5P+32/1jSxbxTaZLfrMAHh7s/FJkU1O0jj8i8mX88
6L3ZAJjps+s9bRPgOe77lbkmpo5ZXrdSv7y0q2yWzRZwbAmxt+rW9QmYagaGssg136zF
pH+wx4PluSl+IvyRZ9ZgAAUUhmAABRSGYAAFFIZgAAUUhmAABRSGYAAFFIZgAAUUhMAA
BRSGYAAFFIZgAAUUhmAABRSGYAAFFIZgAAUUhmAABRSGYAAFFIZgAAUUhmAABRSGYAAF
FIZgAAUUhmAABRSGYAAFF018tlnqQdutlungIgdTl9JkBAEQhmQEARGZAQBElzkBAE
ThCgAAnmJ1OZH9CLWgyQwAIApHMwEAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzA
AAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAA
CikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAK
KQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAop
DMAACikMwAAKKQzAAAopDMAACikMwAAGLodP4fsJaSB2q93ToAAAAASUVORK5CYII=)
"""

#test size 10% and train size 90%
#Random state = Mengontrol pengacakan yang diterapkan ke data sebelum
menerapkan pemisahan agar hasil yang didapatkan tetap sama
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.1,
random_state=42) #split test 10% dan train sisa dari test yaitu 90%

"""#Decision Tree"""

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(random_state=42) #Mengontrol keacakan
estimator agar hasil yang didapatkan selalu tetap
dtree.fit(X_train, y_train)

from sklearn.metrics import confusion_matrix

```



```
""#Berdasarkan Jurnal Machine learning models for non-invasive glucose
measurement: towards diabetes management in smart healthcare
yang          dikutip          dari          :
https://link.springer.com/article/10.1007/s12553-022-00690-7
```

```
**Precision:** Precision is measurement of how many positive
predictions are made for actual positive values.
```

```
**Recall/true positive rate/sensitivity:** Recall is measurement of
correct positive predictions from all positive predictions made
```

```
**F-1 Score:** F-1 Measure is a combination of precision and recall
```

```
#Penjelasan
```

```
**Precision:** Terdapat 76% pasien yang sebenarnya terkena diabetes
dibandingkan dengan keseluruhan pasien yang diprediksi terkena
diabetes.
```

```
**Recall : ** Terdapat 87% pasien yang diprediksi terkena diabetes
dibandingkan dengan keseluruhan pasien yang sebenarnya terkena
diabetes.
```

```
**F-1 Score : ** jika F1-Score punya skor yang baik mengindikasikan
bahwa model klasifikasi punya precision dan recall yang baik
```

```
#Berdasarkan jurnal The advantages of the Matthews correlation
coefficient (MCC) over F1 score and accuracy in binary
classification evaluation
```

```
"Checking only F1, one would read a good value (0.66 in
the [0, 1] interval)."
```

nilai F1-Score bisa diilang bagus jika value F1 berada diantara 0,66 hingga 1

#Alasan menggunakan Precision Recall

yang dikutip dari : Jurnal The relationship between Recall and Precision

[https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/\(SICI\)1097-4571\(199401\)45:1%3C12::AID-ASI2%3E3.0.CO;2-L](https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1097-4571(199401)45:1%3C12::AID-ASI2%3E3.0.CO;2-L)

"Recall and Precision and, in particular, Recall-Precision plots, have been used for many years to characterize document retrieval performance."

Alasan saya menggunakan Recall dan precision adalah untuk melihat kinerja model

#Berdasarkan Jurnal Diagnose Diabetic Mellitus Illness Based on IoT Smart Architecture

yang dikutip dari : <https://www.hindawi.com/journals/wcmc/2022/7268571/>

"for the classification considering the evaluative measures like accuracy, precision, recall, and F1-score on PIDD and claimed to have achieved comparatively enhanced results on binary classifications"

Alasan kedua saya menggunakan precision recall adalah karena precision recall lebih cocok digunakan untuk binary classifications

""

```
y_pred = dtree.predict(X_test)
```

```
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score
```

```

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

"""#Berdasarkan Jurnal Diabetes Mellitus Prediction using
Classification Techniques
Yang dikutip dari :
https://www.researchgate.net/profile/Anny-Leema/publication/34407266
7_Diabetes_Mellitus_Prediction_using_Classification_Techniques/links
/5f50f16da6fdcc9879c511e9/Diabetes-Mellitus-Prediction-using-Classif
ication-Techniques.pdf

"Confusion matrix is used to visualize the performance of the
algorithms which cross tabulates the observed and predicted
classes with associated statistics, evaluation metrics like
sensitivity, specificity, precision and accuracy are used to
evaluate the performance of the method. Factors like True
Positive (TP), True Negative (TN), False Positive (FP) and
False Negative (FN) are used"

Alasan saya menggunakan confusion matrix sebagai visualisasi data
adalah untuk memvisualisasikan performa dari sebuah algoritma
klasifikasi

![format confusion
matrix.png](data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAOoAAACNCA
IAAAA2B3H7AAAAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA
7DAcdvqQQAABCoSURBVHhe7Z1teeTGEoUvhWAIBXMwhGAwhWWwDMIgCIXgCSyBMAiH3H
d12sclpZlJ4im1JG+9P+ap/1Cp+/RRS7PW2v/7u21OS9u3OTFt3+bEtH2bE/Nu3/81zR
kYf124sO+ImiJa0nLavvNoSctp+86jJS2n7TuPlrSctu88TiTp9+/fX15efvn111E+Ko
ez77dv3zj1HUa/EzJ/8FLsKl+/fh2droF9f/vtN7qN81FJI3wv7DV07Mt1/9dff6nIMJ
6fnxX/+eefT09Pis/ILpL+8ccfUUNAWxS+b1+gw14e+PekEe5v3yRrkv719dXOPh27SK
q7WdRQ/P777yO6Qdu3gLX02Bfpf/31Vxbmy5cvdCDmE2R9P37EK4GYJzkq2b+5M47auX

```

D2EU3klN2FdmIpg4xs1aNHZV+aLCCHjNplQ5H+6fA5xBHCezLI4SURUS0o5UijqTGfk  
1FegJLpccPlmGXLXwXSdf25aF2RIsy2A5ZEEQPu1Ym2hc9EY1uxOwdzoZf0VOHk4r+nE  
5Nc0iSnsC+oCVJ1zo19iu4iPSskCoBx9P0j7fOLeC8I5qI70WR0bZY2W6WMvZftC+VUU  
Mrb08DDqa/s80hzgXeC6lhLxjGLftaaEHNvftqGRJsFeo2E847oolIK2uIydZff7Egj2  
F6NrhqX47CviSjmitzIrp8ApXXRAuf077rDLuwi6TSKioQVcKXMi63ozu7L9CTGngSS1  
8elFmte5EG8F7YfWSCYTxuX2LUV71gqUY0EYXxooms7Ruhns1Y4iRVk30FPXk80DOD+i  
clJwubRnhu+7KRKEZlirKvvsbFpwUknqyyYBgjmoi0umVfqwRJ1WhfKl0PPgof2/2Ap5  
1tDh6hOJZ9+YrAMHicSnsndzrq07cEuqEmvuQoSc+aSXR6Kg/1+PjWWm7NLpJKw2iYiP  
2H8/RPB2gLNEk0PyfQTLySb1iLQRJ2DhgVJbw9lHtHAg+2IyxmB8WcuaIhoRQdGRSu  
3B01Tfi9GUWK0s0mSJDWcf0Sx+aBRYb42YG1kAI6IVAZKiTxSfo9DWNXjU//IAHEgN9X  
SQ12fCeUe0cCD7fj5a0nLavvNoSctP+86jJS2n7TuPlrSctu88WtJy2r7zaEnLafvOoy  
Utp+07j5a0nLbvPFrScn4i+35dXpja60ducBZJ7wsVf2i809vaN/6AN/6kEaSC2UIOvR  
U1Cp/RvklD8fT0xExHjw+xFigunU66xXp9AEYyooVi+wIe0ssfKLL+mThNMArVvLy8HE  
RlUSVpQq/ayLJWm0q1Pg7effB62I4kab19hd7qwMF6h8agy0bSIDpn/Bnsi4Bktozsmu  
hMTckLNCQh20Zr9DhJ0q3s+/z8LD+1N/d+mDdIglh6swNJ9OKY4BDtMRfTmByi/QZIrj  
V7Xf7jgMHE1OulM1p1+xOc7scJlvnGIh10yaUXrB6BbCMqBQHJHGwUhq7xS2F8xm6oar  
UJdKlHofQOGh0Ex1KjbOqsw4VqNBqXbyloFWQe0cKG9uVTr4cqFswWFDNV66KebSKX8j  
0Q0CSbAodgX+pRlphW1ZOHbsqmolyuoq8lFYEMHM4nMa00kZCiLhvneQSfvRZUIr01Ai  
biMWMazYVYqvq5AkFkKVRpo/5JKIjJkV2nU2cvR3z9n5yciODog1YRwxkXgx5RBbas3G  
AFmY+nhGpxn0OXuFm6SUUFRR+JBQjnYaubtBMc4lZgSIg7CotlfQpyar2BFeIoxqbiI8  
SzF6J5yYi+3lCPIrMgjt6SNXXxo4CnTE9rlYQittqQhFXRecBrfWtBC4njhItBj6iCuO  
NKQXkOXSwN1WuQFa0J3G2tF1CpZQPXEftlIINbIeWxQVWfiEb/MOQZUSmaFYPEHAR8Io  
VuI2rypQh6rJKYLAGdidXZJKHcX6yFjRsB9d5N6LYmDuZxSDihYtBj6iCaF/fcbAOul  
gaaqIoEdYDgTRzYkS34lSSnITK5mGvVY6tgpzSnW4ehg5UXMtGaTUvjz+iJj9ogWYXJ4  
t6MrFqQEeNwjLs2KoMfI7y25OYNgJ2AS9N6rYFcZxwMegRVRDtC8wQ36Aas7U0nFF3QO  
07Hv3pySH0IZXXg3ptNipG3dcqp1UB6c5ZyOBdQQfGGy6k4sdIZ69C84oOM2qKg9fsvE  
EKOiCjxU9CpeRrYYG1YQdBQ68F003WglYRwxkXgx5RBcm+gAWRLEqDQamJW4W14Pckl0  
hSRt3XKqdVEZwR3aPooMq4wcf1+zDrs5egeV0doZ67ovgYiBpdq/GQKE6MISVfCwtcD1  
TKwaPq7oJWEccJF4Me0cPIqXFiQuJaGhWBSVLJBW2NqKTIMoBalaRDJAo1tFJkhcDJca  
E6oyY11P/I+AYdqLy6GIyZzQM4ta38COqcUSlMn8y3nKFWbbcsAQ7zjohc0gcQh6Lqk1  
DooCZU5VPipP2bDO5m7ixofSQf0UK9fWUpwRxG7Rt6YB2FXYLoS0+mKrEE81d9RFqGM  
SShkOIvTxaORXjMKKI6J5EFywPOelMa3L8hyHbiIpgIsuE3olTMyiMt2h1RtF26KM5Ap  
ZFCirXQukRi3o6kEpNkFYTnddnP+bqglZB5hEt1Nv3cVDNjhTSMcl3fI4j6afhBPblul  
871UtZX3VPRNu3nBPYl7sP9z7uero38Ylx0358Ctq+5ZzAvnp48FMam/Hp9l1xHEk/DU  
nSI9r309CSltP2nUdLWk7bdx4taTlt33m0pOW0fefRkpbT9p1HS1pO23ceLWk5bd95tK  
TltH3n0ZKW0/adR0taTtt3HilpOW3febSk5bR959GS1tP2nUdLWk7bdx4taTlt33m0pO  
W0fefRkpbT9p1HS1pO23ceLWk5bd95tKTltH3n0ZKW0/adR0tazj37Ns3xGX5duLDviJ  
oiWtJy2r7zaEnLafvOoyUtp+07j5a0nM9g3+/fv78sf/BilI9K27ecA9mXM97i/q/yxb  
76heCjFft2GuG31W+xTox+JyQN/r2wy6z8i7xHefn1kvrV8qN8Azocfxn2GiH2RUOUVD  
EqrD8XoPiMJE13tq/2iWhfkf6Swpq27x3SxZ8Ufn19tbNPR5L0oPYV2on9Zxrib/1N9q



VJ3dhaOGTULkul3xOcDp9DHOGOMIykMMLqb3mg/5e3vxXCJ8j6fvyIVwKxRa768x//Fc  
4+ooXD2ZeH2hG9/SkR7nfIrYddbxvRvkiJrPpDRqyKs+FXhNbhpkI/plPTHaRdE1SGN  
BEAiILVzitFJGRGvs1FekJKNHDwT3WswkSXoI+yZG22JluxmVabL/on2pxOWKwbusPQ  
1oTX9nm0Ocy44wjGRfkPLpjksN/QouYuWosNbiHx/wtoDzjmjhvZAA5iARLS4mW3+xi  
Lc4HTbumpfjkJckrgVlDkR12ACnHFEU8Iwbtk3KgbUXLWv1E6wGavbTDjviBaOZV+I8u  
FLGZcL/c7uC/Skhp6k0mOZMqt1L3YfgEgKi/9q33WGXUISvhdSwxzW9o1Qz2aMNYmT3M  
m+gp48HuiZQf0x/WhbSMWt2UXSNVcVTnoKam7Zl1gLYSaLKRjGiBYObV/LB0nuaf8qXQ  
8+Ch/b/aDvK4rn4BHUC8P49/b1HyBDNyupr3HxaQHvtn3Hjy2iySL2H85DO3ryFKFvDO  
yyFP2cQDetRFwVehKThCUBHnyvnmU7OPuI9kMKr+cucdJ3WbohF77kKG0Q+F5iSnA6UM  
9a3NpxtizJuqd9OWNkvTUiImoCWuNgAlRjGfgcxyxHoa9r0Dd+m+ZAaqing7w+E847op  
2IQoEVljUFfVQJKInIVGoPJiZAeWJkJlar/gVtOWI2nH1EC3va99PTkpbT9p1HS1pO23  
ceLWk5bd95tKTltH3n0ZKW0/adR0taTtt3Hi1pOW3febSk5bR959GSlvMT2ffr8g7aXj  
8fgrNIel+o+HP43dnWvvGnkfGHtyAVzBzy6EWzUfim9k0aiqenJ2Y6enyItVBx7XTSLd  
brAzCSES0U2xfwkF7vQJH1awY0wShU8/LychCVRZwKCb29JmtabSrV+jh6X2cUDkaStN  
6+Qi/K4GC98GHQZSNpEJ0z/gz2RUAYW0Z2TXSmpuSdJJKQbaM1epwk6Vb2fx5+lp/Sy5  
A/zBukQSy9FYVkestJcIj2mIq2GA7RfgMk15q9Lv8Xw2Bi6vWGFK26/Qm/YJWKZNA119  
5ZewSyjaqUBCRz1FEausbv2fEzu6Gq1SbQpR6FYq8hoIPgWGqUTz11uFCNBuPirQWtgs  
wjWtjQvnzqpVLFgtmCYqZqXdtTTTfhSvgcCmmRT4BDsSz3KEtOqevLQtdlU1MtV9LWkIp  
CBw/kkppUmElLUZEM8j+Cz14JKZLZwWQ8ZkyjuRBLVT9XIIGsRSt91D8JBTE5sut06u  
zliO+qk5MTEdxZ0CriOOFi0COqWJaVG6wg8/GUUC3uc+gSN0s3qei j6COxAOE8bHWTdo  
JD3AcMCXFHYbGsT0FOrTewQhzF2FR8hHj2QjQvGdHXG+prZBbE0Vuypi5+FPCU6Wmtk1  
DEVhuSsCo6D3itbyloIXGccDHOEVUQdlwpKM+hi6Whcg2yojWBU631Aiq1b0AaYqsmZ  
ArpDw2qOoT0egfhjwjKkXzYoSYg4BPpNBtRE2+FEGPVRKTJaAzsTqbJTT7i7WwcSOg3r  
sJ3dbEwTwOCUe0cDHOEVUQ7es7DtZBF0tDTRQlwnogkGZOjOhWnEqSk1DZPOy1yrFVkf  
06083D0IGKa9korebl8UfU5Act0OziZFFPJLYN6KhRWIYdW5WBz1F+exLTRSu4KVJ3b  
YgjhMuBj2iCqJ9gRniG1RjtpaGM+oOaHzXoz89OYQ+ppJ6UK/NRswO+1rltCog3TkLg  
wr6MB4w4VU/Bjp7FVoXtFhRk1x8JqdN0hBB2S0+EmolHwtLLA07CB06LUAut1a0CriOO  
Fi0COqINkXsCCSRWkwKDVxq7AWHJ70EknKqPta5bQqgjOieXQdVBk3+Lh+H2Z99hI0r6  
sj1HNXFB8DUaNR4SxYkxpORrYYHrgUo5eFTdXdAq4jjhYtAjehg5NU5MSFxLoyIwSS  
q5oK0R1RRZBlCrmnSIRKGGVoqsEDg5L1Rn1KSG+h8Z36AD1VcXgzGzeQCntpUfgZwjKo  
Xpk/mWM9Sq7ZYlwGHeEZFL+gDiUFR9Egod1ISqfEqctH+Twd3MnQWtguQjWqi3rywlmM  
OofUMPrKOWWBB96clUJZZg/qQPSAsWgljScAix10crp2IcRhQR3ZPoguUhJ51pTY7/MG  
QbURFMZJnQO3FqBoXxFq3MKNoOfTRHwLJIQeVaKD1iUU8HUqkJ0mqi8/rs1Fxd0CrIPK  
KFevs+DqrZkUI6JvmoZ3Ek/TSwL5c92uncinrq+6JaPuWcwL7cvfh3sddT/cmPjFu2o  
9PQdu3nBPYVw8PfkpjMz7dviuOI+mnIUL6Rpt+GlrSctq+82hJy2n7zqMlLaftO4+WtJ  
y27zxa0nLavvNoSctP+86jJS2n7TuPlrSctu88WtJy2r7zaEnLafvOoyUtp+07j5a0nL  
bvPfrSctq+82hJy2n7zqMlLaftO4+WtJy27zxa0nLavvNoSctP+86jJS2n7TuPlrSctu  
88WtJy2r7zaEnLuWffpjK+w68LvT00J6bt25yYtm9zYtq+zWn5++//A3Xnpu/lu9g3AA  
AAAE1FTkSuQmCC)

```

#Beberapa definisi dari bagian confusion matrix
True positive: 41 kasus di mana mesin memperkirakan jika pasien
    memiliki diabetes, dan mereka memang memiliki diabetes.

False-positive: 13 kasus dimana mesin memperkirakan jika pasien
    memiliki diabetes, padahal sebenarnya tidak memiliki diabetes

False-negative: 6 kasus dimana mesin memperkirakan jika pasien tidak
    memiliki diabetes, padahal sebenarnya memiliki diabetes

True Negative: 35 kasus di mana mesin memperkirakan jika pasien tidak
    memiliki diabetes, dan mereka memang memiliki diabetes.
"""

cm = confusion_matrix(y_test, y_pred) #memasukkan y_test dan y_pred
plt.figure(figsize=(5,5)) #ukuran plotting heatmap
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label') #Mendefinisikan sumbu y untuk Actual label
plt.xlabel('Predicted label') #Mendefinisikan sumbu x untuk predicted
    label
all_sample_title = 'Accuracy Score: {0}'.format(dtree.score(X_test,
    y_test)) #melakukan kalkulasi score akurasi, {0} berfungsi untuk
    menampilkan angka akurasi

plt.title(all_sample_title, size = 15) #setting ukuran tulisan
    "Accuracy Score: 0.8"

""">#XGBoost"""

from xgboost import XGBClassifier

```

```

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5,annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
    y_test)) # menampilkan score akurasi
plt.title(all_sample_title, size = 15) #format title untuk accuracy
    score

"""#Metode 80 20

#Berdasarkan Jurnal Improving the Classification Accuracy using
    Recursive Feature Elimination with Cross-Validation
yang dikutip dari :
    http://www.puneetmisra.com/admin/uploads/journals/5f136d202b8ba1.186
    44117.pdf

"In
this study, we have used the train_test_split() method of

```

the Scikit-Learn library of python. Through this function, we divide the dataset into a different ratio. However, the 80/20 (train/test) rule is mostly used in the studies."

Tujuan penggunaan train test split menjadi 80 dan 20 adalah karena rasio tersebut paling sering digunakan dalam penelitian

```
"""
```

```
#test size 20% and train size 80%
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2,
    random_state=42)

""">#Decision Tree""
from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(random_state=42)
dtree.fit(X_train, y_train)

y_pred = dtree.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
```



```

plt.ylabel('Actual label')

plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {0}'.format(dtreescore(X_test,
    y_test))

plt.title(all_sample_title, size = 15)

"""#XGBoost"""

from xgboost import XGBClassifier

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')

plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
    y_test))

plt.title(all_sample_title, size = 15)

"""#Metode 70 30

```

```
#Berdasarkan Jurnal A Comparative Analysis for Diabetic Prediction  
Based on Machine Learning Techniques
```

```
Yang dikutip dari : https://www.iasj.net/iasj/download/a371daadb33b96fd
```

```
Disimpulkan bahwa Train Test split dengan rasio 70 dan 30 dapat  
menghasilkan akurasi yang lebih besar dibandingkan dengan metode  
K-Fold Cross Validation
```

```
"""  
  
#test size 30% and train size 70%  
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,  
random_state=42)  
  
""">#Decision Tree""  
  
from sklearn.tree import DecisionTreeClassifier  
  
dtree = DecisionTreeClassifier(random_state=42)  
dtree.fit(X_train, y_train)  
  
y_pred = dtree.predict(X_test)  
  
from sklearn.metrics import accuracy_score, f1_score, precision_score,  
recall_score  
  
print('F-1 Score : ',(f1_score(y_test, y_pred)))  
print('Precision Score : ',(precision_score(y_test, y_pred)))  
print('Recall Score : ',(recall_score(y_test, y_pred)))  
  
cm = confusion_matrix(y_test, y_pred)
```

```

plt.figure(figsize=(5,5))

sns.heatmap(data=cm,linewidths=.5, annot=True,  cmap = 'Blues')

plt.ylabel('Actual label')

plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {0}'.format(dtree.score(X_test,
    y_test))

plt.title(all_sample_title, size = 15)

"""#XGBoost"""

from xgboost import XGBClassifier

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(5,5))

sns.heatmap(data=cm,linewidths=.5, annot=True,  cmap = 'Blues')

plt.ylabel('Actual label')

plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
    y_test))

plt.title(all_sample_title, size = 15)

```

```

"""#Metode 60 40"""

#test size 40% and train size 60%
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.4,
    random_state=42)

"""#Decision Tree"""

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(random_state=42)
dtree.fit(X_train, y_train)

y_pred = dtree.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score
print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {}'.format(dtree.score(X_test,
    y_test))

```



```

plt.title(all_sample_title, size = 15)

"""#XGBoost"""

from xgboost import XGBClassifier

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)
from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score
print('F-1 Score : ', (f1_score(y_test, y_pred)))
print('Precision Score : ', (precision_score(y_test, y_pred)))
print('Recall Score : ', (recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
    y_test))
plt.title(all_sample_title, size = 15)

```

### Test 3 : Random state = 0, Imbalanced dataset

```
# -*- coding: utf-8 -*-

"""Diabetes_Prediction_Using_Decision_Tree_and_XGBoost_Algorithm
    imbalanced.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/18FPmJy1JCi2zITXIGLrtNoR7lq1
    QGVTO

#Importing Library and Preparing the dataset
"""

# Commented out IPython magic to ensure Python compatibility.

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(color_codes=True)
from sklearn.model_selection import train_test_split
# %matplotlib inline

diabetes_df = pd.read_csv('diabetes.csv')

diabetes_df

"""#Exploratory Data Analysis"""

#Column in Dataset
```

```

diabetes_df.columns

#Show data types and null value each column
diabetes_df.info()

#Menampilkan 10 baris pertama
diabetes_df.isnull().head(10)

#Checking if there is null value
diabetes_df.isnull().sum()

#Checking if there is zero value

#replace 0 value with NaN
diabetes_df_copy = diabetes_df.copy(deep = True) #deep = True -> Buat
    salinan indeks dan data dalam dataframe
diabetes_df_copy[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', '
    BMI']] =
    diabetes_df_copy[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin
    ', 'BMI']].replace(0, np.NaN)

# Showing the Count of NANs
print(diabetes_df_copy.isnull().sum())

"""#Berdasarkan Jurnal A model for early prediction of diabetes
oleh Talha Mahboob Alam et al, yang bersumber dari :
https://reader.elsevier.com/reader/sd/pii/S2352914819300176?token=7D
4F21A1900E1B47FA971E110A1469CE2A8882E14CFBF3A4C3790507841542704D1370
C180ADD4341ED02B627B619B3F&originRegion=eu-west-1&originCreation=202
20928035824

```

"Data cleaning consists of filling the missing values and removing noisy data. Noisy data contains outliers which are removed to resolve inconsistencies. In our dataset, glucose, blood Pressure, skin thickness, insulin, and BMI have some zero values. \*\*Thus, all the zero values were replaced with the median value of that attribute.\*\*"

Jadi di tahap ini yang saya lakukan adalah mengisi value 0 (missing values) pada beberapa kolom tersebut dengan nilai median untuk mengatasi nilai inconsistent

```
"""
```

```
#Fill null value with median
```

```
diabetes_df_copy['Glucose'].fillna(diabetes_df_copy['Glucose'].median()  
, inplace = True)
```

```
diabetes_df_copy['BloodPressure'].fillna(diabetes_df_copy['BloodPressure']  
.median(), inplace = True)
```

```
diabetes_df_copy['SkinThickness'].fillna(diabetes_df_copy['SkinThickness']  
.median(), inplace = True)
```

```
diabetes_df_copy['Insulin'].fillna(diabetes_df_copy['Insulin'].median()  
, inplace = True)
```

```
diabetes_df_copy['BMI'].fillna(diabetes_df_copy['BMI'].median(),  
inplace = True)
```

```
#inplace = True -> untuk menyimpan hasil modifikasi pada dataframe  
diabetes_df_copy
```

```
"""#Berdasarkan Jurnal Deteksi Dini Penyakit Diabetes Menggunakan  
Machine Learning dengan Algoritma Logistic Regression
```

yang dikutip dari :  
<https://jurnal.ugm.ac.id/v3/JNTETI/article/view/3586/1646>



"Pada tahap ini juga dilakukan pengecekan terhadap data yang tidak seimbang. Penanganan terhadap data yang tidak seimbang dilakukan menggunakan synthetic minority over-sampling technique (SMOTE) "

Pada tahap ini yang akan saya lakukan adalah oversampling minority value, yakni value 1 pada variabel outcome yang berfungsi untuk menyeimbangkan data yang nantinya akan digunakan untuk prediksi

#Berdasarkan Jurnal SMOTE: Synthetic Minority Over-sampling Technique yang dikutip dari :  
<https://www.jair.org/index.php/jair/article/view/10302/24590>

"However, the nature of the application requires a fairly high rate of correct detection in the minority class and allows for a small error rate in the majority class in order to achieve this."

Alasan saya menggunakan teknik Oversampling adalah untuk meningkatkan sampel kelas minoritas dan untuk memperkecil tingkat kesalahan di kelas mayoritas

#Berdasarkan PENERAPAN SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE (SMOTE) TERHADAP DATA TIDAK SEIMBANG PADA PEMBUATAN MODEL KOMPOSISI JAMU yang dikutip dari :  
<https://journal.ipb.ac.id/index.php/xplore/article/view/12424/9491>

"the model with SMOTE is more accurate than model without SMOTE because has higher AUC value."

Keunggulan dari menerapkan metode Oversampling minority adalah bisa meningkatkan nilai AUC

```

"""
#Counting 1 and 0 Value in Outcome column
sns.countplot(diabetes_df_copy['Outcome']) #membuat bar plot
perbandingan jumlah value
print(diabetes_df_copy.Outcome.value_counts()) #menampilkan jumlah
value 0 dan 1

"""#Berdasarkan Jurnal Prediksi Risiko Penyakit Diabetes menggunakan
Algoritma Regresi Logistik
oleh Qatrunnada Refa Cahyani, Mochammad Januar Finandi, Jathu Rianti,
Devi Lestari Arianti, Arya Dwi Pratama Putra yang dapat diakses
melalui
https://journal.literasisains.id/index.php/jomlai/article/view/598/470
, kita melakukan tahap preprocessing yaitu pengecekan outliers
pada variabel. Outliers dihapus dengan menggunakan Z-Score.
Beberapa data dihapus karena nilainya jauh dari Z-Score. Maka jumlah
data yang awalnya 1000 menjadi 938

#Berdasarkan Detection of Spatial Outlier by Using Improved Z-Score
Test
yang dapat diakses melalui : https://booksc.org/book/77788899/9f8c40 ,
"The zscore test has long been used to detect outliers in data."

Fungsi Tes Z-Score adalah untuk mendeteksi apakah terdapat outlier
dalam data.

#Alasan Menggunakan Z-Scores

##Berdasarkan Outlier Detection in Multivariate Time Series Data Using
a Fusion of K-Medoid, Standardized Euclidean Distance and Z-Score
yang dapat diakses melalui :
https://link.springer.com/chapter/10.1007/978-3-030-69143-1\_21

```

Z-score technique produced a better outlier detection result of 0.9978 F-measure as compared to inter-quartile of 0.8571 F-measure.

Alasan saya menggunakan Z Score adalah karena Teknik Z-score menghasilkan hasil deteksi outlier yang lebih baik

```
#Checking Outliers using Box Plot
```

```
"""
```

```
sns.boxplot(x=diabetes_df_copy["Pregnancies"])
```

```
sns.boxplot(x=diabetes_df_copy["Glucose"])
```

```
sns.boxplot(x=diabetes_df_copy["BloodPressure"])
```

```
sns.boxplot(x=diabetes_df_copy["SkinThickness"])
```

```
sns.boxplot(x=diabetes_df_copy["Insulin"])
```

```
sns.boxplot(x=diabetes_df_copy["BMI"])
```

```
sns.boxplot(x=diabetes_df_copy["DiabetesPedigreeFunction"])
```

```
sns.boxplot(x=diabetes_df_copy["Age"])
```

```
""">#Removing Outliers using Z Scores
```

```
#Berdasarkan jurnal Robust statistics for outlier detection
```

yang dikutip dari :  
<https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.2>

"Data preprocessing is a useful step that helps to remove noise, inconsistencies and redundancy to achieve high quality data which improves the performance. During the data pre-processing, all the missing values are filled by the mean of the corresponding feature. Formatting of the given dataset is ensured to be consistent. All the incorrect data types of the features changed to their required datatype. Data normalization is performed on the given dataset to make its range consistent. We used z-score normalization on all the features to restrict the range of values between 3 to - 3"

"Generally, some of machine learning algorithms does not handle categorical data. Diagnosis feature in this dataset contains two categories M(malignant) and B(benign) is replaced numeric with M by 1 and N by 0."

Menurut jurnal tersebut. ambang batas Z-Score yang dapat diterapkan adalah antara -3 dan 3 untuk 2 kategori diagnosis (True or False)

""

```
import scipy.stats as stats

z = np.abs(stats.zscore(diabetes_df_copy))

data_clean = diabetes_df_copy[(z<3).all(axis = 1)] #print all of rows
that have z<3 (z score below 3)

data_clean.shape
```



```

#z<3 = print semua data yang Z-Scorenya kurang dari 3

#.all(axis=1) = melihat di bagian kolom untuk setiap baris (melihat
kolom z score setelah di proses dengan print z kurang dari 3)

#Cleaned Outliers data using Z Scores

data_clean

"""#Data Correlation
Berdasarkan Diabetes Prediction Based on XGBoost Algorithm oleh Mingqi
Li et al 2020 bersumber dari
https://iopscience.iop.org/article/10.1088/1757-899X/768/7/072093/pdf

Setelah data preprocessing, kita harus mempelajari apakah data tersebut
diproses dengan baik atau tidak dan bagaimana caranya
banyak korelasi antara data tersebut. Koefisien korelasi dapat
digunakan untuk mencerminkan kedekatan
hubungan antar variabel. Koefisien korelasi dihitung dengan metode
selisih.
Hal ini juga didasarkan pada dispersi kedua variabel dan rata-ratanya
masing-masing. keduanya
perbedaan dikalikan untuk mencerminkan tingkat korelasi antara dua
variabel

#Berdasarkan Jurnal Prediksi Risiko Penyakit Diabetes menggunakan
Algoritma Regresi Logistik

Jika nilai korelasi > 0 maka terdapat korelasi positif. Sementara
nilai satu variabel meningkat, nilai variabel lainnya juga
meningkat. Jika persamaan korelasi = 0 maka tidak ada korelasi.
Jika korelasi < 0 maka ada korelasi negatif. Sementara satu
variabel meningkat, variabel lainnya menurun. Ketika korelasi
diperiksa, ada 2 variabel yang bertindak sebagai korelasi positif
terhadap variabel dependen outcome, variabel tersebut adalah
glucose. Seiring peningkatan ini, variabel dependen juga

```

meningkat. Dengan demikian, semua variabel digunakan untuk modelling karena korelasinya berdekatan.

```
#Berdasarkan Complex heatmaps reveal patterns and correlations in multidimensional genomic data
```

Yang dapat dikutip dari :  
<https://academic.oup.com/bioinformatics/article/32/18/2847/1743594>

Heatmaps are a fundamental visualization method that is broadly used to unravel patterns hidden in genomic data. They are especially popular for gene expression analysis (Eisen et al., 1998) and methylation profiling (Sturm et al., 2012). With the increasing availability of genomic datasets, **visualization methods that effectively show relations within multidimensional data are urgently needed.**

Alasan saya menggunakan heatmap adalah karena heatmap correlation dapat digunakan untuk menemukan hubungan potensial antara variabel dan untuk memahami kekuatan hubungan ini dan mendeteksi hubungan linier dan nonlinier

```
#Fungsi Heatmap Correlation Matrix
```

```
Berdasarkan Jurnal Implementasi Seleksi Fitur Klasifikasi Waktu Kelulusan Mahasiswa Menggunakan Correlation Matrix With Heatmap
```

Yang dikutip dari :  
<https://ejurnal.ung.ac.id/index.php/jjee/article/view/14403/4663#>

"atribut yang memiliki warna merah muda adalah atribut yang relevan dengan atribut aoutput dan sebaliknya atribut yang berwarna merah tua adalah atribut yang kurang relevan dengan atribut aoutput."

Atribut berwarna merah tua yang dimaksud oleh penulis adalah atribut yang memiliki korelasi negatif tinggi dengan kolom output

"dari 13 atribut awal yang digunakan terseleksi menjadi 9 atribut yang memiliki relevansi atau berkontribusi terhadap hasil output yaitu : jenis kelamin, kelas, umur, SKS1, IPS1, SKS2, SKS3, SKS4 dan SKS5, sedangkan atribut input yang kurang relevan terdapat 4 atribut yaitu : IPS2, IPS3, IPS4 dan IPS5"

#Alasan menggunakan Correlation Matrix Heatmap pada penelitian ini Untuk melihat relevansi antar kolom. Disini target kolom saya adalah kolom "Output", dan relevansi antar kolomnya memiliki nilai positif. Maka saya akan menggunakan semua kolom sebagai indikator

```
"""
```

```
sns.heatmap(data_clean.corr(), annot=True)
```

```
#.corr() = correlation matrix
```

```
#annot=True = memberikan value korelasi antar kolom dalam bentuk angka
```

```
"""#Machine Learning Model Building
```

```
axis=1 = Remove column
```

Dataset X = menggunakan semua kolom untuk kriteria dan ciri ciri pada tubuh pasien

Dataset y = sebagai hasil apakah pasien memiliki diabetes atau tidak

```
![xy.png](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAzMAAAHMCAIAAABukmEEAAAAAXNSR0IArs4c6QAAARnQUlBAACxjwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAACUdSURBVHhe7d1hQmJbskThOy4H5HgcjZOpwXQLr1DEU0paAhtifX/aCimgNpl
```

5UvC+99//JEMStAY3M0mSpFW4mUmSJK3CzUySjGkVbmaSjEmrcDOTJELahZuZJEnSKtz  
MjEmSVuFmJkmStAo3M0mSpFW4mUmSJK3CzUySjGkVbmaSjEmrcDOTJELahZuZJEnSKtz  
MjEmSVuFmJkmStAo3M0mSpFW4mUmSJK3CzUySjGkVbmaSjEmrcDOTJELahZuZJEnSKtz  
MjEmSVuFmJkmStAo3M0mSpFW4mUmSJK3CzUySjGkVbmaSjEmrcDOTJELahZuZJEnSKtz  
MjEmSVuFmJkmStAo3M0mSpFW4mUmSJK3CzUySjGkVbmaSjEmrcDOTJELahZuZJEnSKtz  
MjEmSVuFmJkmStAo3M0mSpFW4mUmSJK3CzUySjGkVbmaSjEmrcDOTJELahZuZJEnSKtz  
MjEmSVuFmJkmStAo3M0mSpFW4mdX5T5K0Kia1ilkEdeh+SdJ6mNqQzHhUofslSethUqu  
YRVCH7pckrYdJrWIWQR26X5K0Hia1ilkEdeh+IjWkCSZIkGqOEwxSfBmI6tD9QSpJE0y  
QINUCjXikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1J  
JmmCCBKnMOMegVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp  
0f5BK0gQTJEG1xwkGqYpZBHXo/iBVrz9PD/89PP3hT9JpmCBBqj1OMehVzCKoQ/cHqX7  
by7pzbNH15/ybmbvfPaKqg/Snnh+5n53HZ9IO/KuDVMUsgjp0f5DqHHZxm1xkdtvJssv  
ZebledF3N7szuRT1A+hP7AjnYxo7/vOn58W72t/35vSNVMYugDt0fpDqH3Tb2fv143c3  
u5nJyst0ldHetdTW7L68D5A3p2L4tjmtjMzz0sbVuHCcYpCpmEdSh+4NU53B0+Xh7K2C  
XPzw9764+ufzsL0V779ej92zv5Rv75PE538h9H96Q7C+3/Hdb15Bb7r958N23p/H6jtf  
e+72c7s/T4+6ePjxKHDyXnS+OQuvhRQRsQf2r/fmFJn7tkdfKSV08Pr0VyEbNHBtZW5i  
73wUvX6eid7fk64Mn8H5Xn5/Vmfb4QapiFkEduj9IdQ67If92oXi9BrwvSweD/+B2++/  
uv97d/jXkq6NLxu6PH794veXLl3+75f4rHuj1podXvsPv7u9h9/XLF0QvXz7x1QCL2cE  
9vvn0D9zZeg5a0r5w3pFO7V77rVf5PT+qjk/hYc3s4vdeeC/6l+/v/2fv7Q4Ov369g4O  
7en+ss9s/k3ekKmYR1KH7g1Tn8H4xeJU5v8u5fuwc32z/vcPbvFlkDq4hh+m7r2+59Tf  
eb/nXp/EPF6iDXwba39PBHW3+AzeFg5bEKxSkU7vq26qv9zz1mfQ4fPnyc5UcVtf+xrs  
/HIabX1+p/HiwIFUxi6AO3R+kOofdoN+46Hy4KrzYvtn7jQ6+v3mVerX74w53vHHL04e  
N3HL7abzme5+++fadF1t/8+05vTu42fvTeX/kvz0HrYdXNEin9jX0uSoP4o1Kpgzfy+j  
Q7vtvaW5yeNPNr3dfXKH8OMegVTGLoA7dH6Q6h78N+sOrwovdH7dut8/33r65eZV6vd3  
+y5eM09665WH2LunfnSar/XcPnvQpnp8+3n73SId38fkf+PVz0Ep46YJ0bLowDsOtSj4  
MD7//5kMh7f6w+3r3v3mkza8//K3L4QSDVMUsgjp0f5DqHI43kTi8KuzsLwhvV4Tnx/3  
3+J+Ptq5SR9nL33qxecv9F2/py8Puf3Hs7btbT+Pg18tevr3xjP5u4x/w4cq3+Q/cPAq  
t6OV10kT6Ax9f8xcfy3T3p9cyOLjhW9Hy9fvNqdj9bd//2v7buy9SUNtf7//WwR3nFuf  
18pCHSFXMIqhD9wepf+t+xMfbZWNnfyHZORz7r5eEPW58kOx9/G8HDu/k7et4eHzcVux  
L3R7ceBcc3ufW0zh8Fh/+FV/b+jce3tWHh0bu/tNz0JJ4jYL0hz4Ww8fX/e17D09P+7r  
afZsKo76OqvrV3yr/5Y+bX78+6hXKj4cLUhWzCOroQ/UGq5Xz8KPDlenFva8rd/wPvHBM  
kSDXHCQapilkEdeh+INvidj/IH2wq97e33P0/804xQYJUc5xgkKqYRVCH7g9SLeftQ5a  
d949o7sfd/wPvHK9ckGqOEwxSfBmI6tD9QSpJE0yQINUCjXikKmYR1KH7g1SSJpggQao  
5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnMOMegVTGLoA7dH6SSNME  
ECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEG1xwkGqYpZBHXo/iC  
VpAkmSJBqjhMMUhWzCOroQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqh  
D9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBA1szXGCQap  
iFkEXW1+StDBGtir58heh4yVJy2Nwq4+vfrHaxZK0Paa3+vjaF6HdJUNLy3Crj699FzP  
ekrQwRrYq+fLXoe+DVJImmCBBqj1OMehVzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkm



aYIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyCO nR /kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9QSpJE0yQINUCJxikKmY R1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnmOME gVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEG 1xwkGqYpZBHxO/iCVpAkmSJBqjhMMUhWzCO rQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkia YIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+ kkjTBBAlSzXGCQapiFkEduj9IJWmCCRKkmuMEg1TFLII6dH+QStIEEyRINccJBqmKWQR 16P4glaQJJkiQao4TDFIVswjq0P1BKkkTTJAglRwnGKQqZhHUofuDVIImmCBBqjlOMeh VzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkmaYIEqeY4wSBVMYugDt0fpJI0wQQJUs1 xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyCO nR/kErSBBMkSDXHCQapilkEdej+IJWkCSZ IkGqOEwxSFbMI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6k kTTBBglRznGCQqphFUIfuD1JJmmCCBKnmOMEgVTGLoA7dH6SSNMEECVLNcYJBqmIWQR2 6P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEG1xwkGqYpZBHxO/iCVpAkmSJBqjhMMUhW zCO rQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5x gkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBAlSzXGCQapiFkEduj9IJWmCCRK kmuMEg1TFLII6dH+QStIEEyRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkk TTJAglRwnGKQqZhHUofuDVIImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4 PUkmaYIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSy CO nR/kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9QSpJE0yQINUCJxi kKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBK n mOMEgVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQ TJEG1xwkGqYpZBHxO/iCVpAkmSJBqjhMMUhWzCO rQ/UEqSRNMkCDVHCcYpCpmEdSh+4N UkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6A O3R+kkjTBBAlSzXGCQapiFkEduj9IJWmCCRKkmuMEg1TFLII6dH+QStIEEyRINccJBqm KWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkkTTJAglRwnGKQqZhHUofuDVIImmCBBqjl OMEhVzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkmaYIEqeY4wSBVMYugDt0fpJI0wQQ JUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyCO nR/kErSBBMkSDXHCQapilkEdej+IJW kCSZIkGqOEwxSFbMI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP 3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnmOMEgVTGLoA7dH6SSNMEECVLNcYJBqmI WQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEG1xwkGqYpZBHxO/iCVpAkmSJBqjhM MUhWzCO rQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJ UC5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBAlSzXGCQapiFkEduj9IJWm CCRKkmuMEg1TFLII6dH+QStIEEyRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1 BKkkTTJAglRwnGKQqZhHUofuDVIImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHocYJCqmEV Qh+4PUkmaYIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBIN UxSyCO nR/kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9QSpJE0yQINU cJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmC CBKnmOMEgVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5D qev48PfbIXMLD0x8e90TPj/zN3/T4zL3rZvFSBqnmOMEgVTGLoA7dH6S6Hjcz3SJeyiD VHCcYpCpmEdSh+4NUV/fnxfpz0+Pjb+9pDw+PT8/Pu7vnkX5qdx/Xp7hw0+e4e5ZPP3

Gs9AieGGDVHocYJCqmEVQh+4PUq3lZQP62QL00cPj81kWoZdnxyN8b7cXuo3dIV7fINU  
cJxikKmYR1KH7g1RL2u1nvFBz488tR075DNa17J7xIgf7/q1j9JnrTD/9YJ//HCeew1  
SFbMI6tD9Qap1/Xn60Xp291/1+vqgea6367QKXugg/V1uZqpkEdSh+4NUa7v4xeIEXzw  
nt7ICvNZB+ut++puN+zds/6kMd79V+fUPRf/+GHvcW5CqmEVQh+4PUi1vuJyd97PMnb8  
8oQf/u8sOvN5Bei4vC9qg/n/xR40/9d0vNhj3GKQqZhHUofuDVLdgtJyde0HafjKuZTV  
4xYP0nE6t/9/9qWTzUX+1zrnPIFUxi6AO3R+kuhGD5ey8S9LWE/FDzCa86EF6XieV/y+  
/Xfz5MX/7/WjuNkhVzCKoQ/cHqW7Hyb8Vfc4PND8/iXM+mhbE6x6k53bCbvbL1fip1H/  
9Rx7uN0hVzCKoQ/cHqW7JyW+cnetts897mZ9i1uGVD9Lz+776f7UYpz3c75c6dxykKmY  
R1KH7g1Q35sQ3zs7xRtb5P93RLeDFD9JL+G43+9V6P0q0c9Q6dx2kKmYR1KH7g1Q350q  
7mXuZXvHyB+1lffP8v1iSF1jMnMk6ZhHUofuDVDfoCrvZ54f0Y8xWFECQXsg3b5v9Vs0  
fP8x5fgrhzoNUxSyCOmR/kOomffexDn7revJpLzvPhUo3gRoIOv55ueSX/mJ4TKLmTN  
ZxyyCOmR/kOpGXW43+/xI7mXVqIIgvZxvSv8MJX+u94e5+yBVMYugDt0fpLpZ19nN/Bh  
TR6iDIL2kr982++fV7Pjuz1bx3H+QqphFUIfuD1LdsPPvZn6MqU8ohSC9qLN+onmxxcy  
ZrGMWQR26P0h10866m32+c/cyrTFJvin8fynUo8XsnDXPQwSpilkEdej+INWNO9tu9v1  
tCT/G1A7LEKSXdq63zS64mDmTdcwiqEP3B6lu3j12s437dC/TKwoiSC/um7r/Wb0e3+1  
ZFzNns05ZBHXo/iDVHfjt3ezz/Z33AqWbQk0E6RV8/bbZT2r2uPLP/OMIjxKkKmYR1KH  
7g1R34Td3Mz/G1JeoiC9ht/+RPPCi5kzWccsgjp0f5DqTvzSbrZxN+5l+oC6CNKr+Kb  
qh6V7vOidvfJ5nCBVMYugDt0fpLobv7Cbfb4LP8bUMUojSK/kFz/RPLqrC5Q+jxSkKmY  
R1KH7g1R35B93s88XOfcyFUzxB0m1/NrbZpdfzJzJOMYR1KH7g1R35cTdbON6tfe3/Rh  
TWyiPIL2ab2r+1A3raDG7SPHzWEGqYhZBHbo/SHVnfrab+XaZTkaFBOkVfa7eQydv8nH  
bXOanEh4sSFXMIqhD9wep7s58N3Mv0wA1EqTX9E3Jf79mXWxcybrmEVQh+4PUt2h0W6  
2ceMLXZd0m6iSIL2uf3vb7KgHLvZzCY8XpCpmEdSh+4NUd+nk3ezRt8s0RKEE6ZV9Xff  
ff/XRWne5DuABg1TFLII6dH+Q6k6dupsdcs/TdyiVIL22bwr+i/eBr7aYOZN1zCKoQ/c  
Hqe7WD3YzP8bU9yiWIL2+H36iefTXLtkDPGSQqphFUIfuD1LdseFu5l6mU1AuQbqAH71  
tdsXFzJmsYxZBHbo/SHXXTt/N/BhTJ6JignQJX79ttrF1HTfIZX864UGDVMUsgjp0f5D  
qzp26m/mGmU5ExQTpGr4u908/fhzd/NI/nvCwQapiFkEduj9Idfe+fh/hnbuZTkK9Bok  
ivv1J5GORX3kxcybrmEVQh+4PUhVwN9MvolqCdBmnf6J5dMvLf6DPAwepilkEdej+IFU  
FdzP9GmolSNdx6ieaR01xhdrnkYNUxSyCOmR/kKqEu5l+CZUSpCs56W2z6y9mzmQdswj  
q0P1Bqg7f/PrNIXczfYk6CdK1fL2a7d82W2AxcybrmEVQh+4PUjUY7GU77mb6A1USpGv  
57hPN5yv/7v8rHj1IVcwiqEP3B6nu36kfzB640pVKt4AaCdLVTKr+WuXOwwepilkEdej  
+INWdG75d9sbdTH9BhQTPck6v/KvVOo8fpCpmEdSh+4NUd23j6vT49HjaFcvdTJuoJyB  
d0I1vm13vw3ueQJCqmEVQh+4PUt2xz1em123r1HcT3M20geoIOhWdtJpd8bcqeQZBqmI  
WQR26P0h1rzbWr/dNy91MP0ZtBOmSTqjza/7nLjyFIFUxi6AO3R+kuk9bH2N+uAS5m+m  
HqIwgXdQ3b5tdt7x5EkGqYhZBHbo/SHWPni5Hn98acdftj1AXQbqqL8v8yrXNswHSfBM  
I6tD9Qaq7s3Ep+ssFyN1MP0BVBOm6/v622bULm6cRpCpmEdSh+4NUd2Zj2/riN2nczTR  
GTQTpwv62m13zV8z2eB5BqmIWQR26P0h1V076GPMDDzMNURFBurLrIr/6YuZM1jGLOA7  
dH6S6HxvXn1P2KXczjVAPQbq2+Y8s18AzCVIVswjq0P1BqnuxsWCdeu1xN9MA1RCkizu  
u8SVqmeCSpCpmEdSh+4NU9+Ef3xNwN9PJqIUgXdxRha9RyDyZIFUxi6AO3R+kugMbe9X  
4yuNuphNRCUG6tiUXM2eyjlkEdej+INXN29ipfvQrNO5mOgl1EKRL06rtBX7fBI+nE6Q  
qZhHUofuDVDfuN3+1+dTdbJkrm66BIgjSlS26mDmTdcwiqEP3B6lu2cYq9W/vaLmb6Vu

UQJAubNXFzJmsYxZBHbo/SHW7Ntaof7/ouJvpGxRAkK7r49vKK30Yz1MKUhWzCORQ/UG  
qW3W+/wtNG/e8yd2sEy9/kC5r3cXMmaxjFkEduj9IdZN+/WPMj9zN9He8+EG6qoUXM2e  
yjlkEdeJ+INUNOsvHmB+5m+lveOmDdFEFC3mxcuVZBamKWQR16P4g1c3ZWJrOccFxN9M  
2XvqgXdPSi5kzWccsgjp0f5Dqtpz5Y8wP3M20hZc9Sfd01C3LFSrPK0hVzCKoQ/cHqW7  
Jxq5011+ccTfTZ7zoQbqgj4vZWr9itsczC1IVswjq0P1Bqpux8XbZ+VeiU3ezBS970hN  
e8iBdz8faXbFCeWpBqmIWQR26P0h1I7ZWpEu8VbW1D25yN2vBCx6kqzq3CXLk+cWpCp  
mEdSh+4NUt2BrPbrYpebk3cwPNTvwagfpWo6Lds3S5MkFqYpZBHxo/iDV8jY3o4u+BXD  
qR5ruZhV4rYN0JccFu+r7uTy9IFUxi6AO3R+kWtvz4+YbVhfegE5fzfxQ8/7xSgfpel  
5fnp8eG+Fh4fHx6fnPlu19udzzyxbkzy/IFUxi6AO3R+kWtffP0a89KXm9A80d3zj7L7  
xMgfp+bzsWjzUhpdc7W1F269v5AfWrUeeYJCqmEVQh+4PUq1p4yf/Axe/1gzeNNt5cDm  
7Y7zIQXous58KPlu5FHmKQapiFkEduj9ItZyXn/y/uxat/Z7Z3stytuhHSPpHvMJBeib  
DnwmOrf3hOk8ySFXMIqhd9wepVvKXj2M2XPSdgJ9eHR+e3M7uEK9ukJ7Hvy1mq//SI08  
zSFXMIqhd9wep1nd6ThYXWnu+/ljl1ew++eXZveGWD9Dz+ZTNb/yN1nmiQqphFUIifuD1J  
dz58Xz0/TjeyDh8enp+fn3R1xn79k/9Q+/udw/+jtmfIIulm8okF6Hj/ezFZ/u2yP5xq  
kKmYR1KH7g1TX8Y+/PrPt398k+Ndfz7NTVw19Te8ikF6Hj8ryFv5L1B4ukGqYhZBHbo  
/SHUdbma6VbyKQXom84q8of8wmGccpCpmEdSh+4NUkiaYIEF6NpMfYm7slxp51kGqYhZ  
BHbo/SCVpggkSpGd02n+EcoP/qQnPPehVzCKoQ/cHqSRNMEGC9Ly+/1Bzt5Td4gfkPP0  
gVTGLoA7dH6SSNMEECdLz+7P7v+By+F8L7/4/M930/8k8/h1BqmIWQR26P0glaYIJEqS  
a4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHxo/iCVpAkmSJBqjhMMUhWzCORQ/UEqSRN  
MkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhd9wepJE0wQYJUc5xgkKqYRVCH7g9  
SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALszXGCQapiFkEduj9IJWmCCRKkmMEg1TFLII  
6dH+QStIEEyRINccJBqmKWQR16P4glaQJJKiQao4TDFIVswjq0P1BKkkTTJAg1RwnGKQ  
qZhHUofuDVJImmCBBqjLOMEhVzCKoQ/cHqSRNMEGCVHOcYJCqmEVQh+4PUkmaYIIIEqeY  
4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyConR/keRrSBBM  
kSDXHCQapilkEdeJ+IJWkCSZIkGqOEwxSfBmI6tD9QSpJE0yQINUCJxikKmYR1KH7g1S  
SJpggQao5TjBIVcwiqEP3B6kkTTBBg1RznGCQqphFUIifuD1JJmmCCBknOMEgVTGLoA7  
dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYp  
ZBHxo/iCVpAkmSJBqjhMMUhWzCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4  
wSFXMIqhd9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBAL  
szXGCQapiFkEduj9IJWmCCRKkmMEg1TFLII6dH+QStIEEyRINccJBqmKWQR16P4glaQ  
JJkiQao4TDFIVswjq0P1BKkkTTJAg1RwnGKQqZhHUofuDVJImmCBBqjLOMEhVzCKoQ/c  
HqSRNMEGCVHOcYJCqmEVQh+4PUkmaYIIIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZ  
BHbo/SCVpggkSpJrjBINUxSyConR/keRrSBBMkSDXHCQapilkEdeJ+IJWkCSZIkGqOEwx  
SfBmI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBg1R  
znGCQqphFUIifuD1JJmmCCBknOMEgVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYI  
JEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHxo/iCVpAkmSJBqjhMMUhWzCORQ/UE  
qSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhd9wepJE0wQYJUc5xgkKqYRVC  
H7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALszXGCQapiFkEduj9IJWmCCRKkmMEg1T  
FLII6dH+QStIEEyRINccJBqmKWQR16P4glaQJJKiQao4TDFIVswjq0P1BKkkTTJAg1Rw  
nGKQqZhHUofuDVJImmCBBqjLOMEhVzCKoQ/cHqSRNMEGCVHOcYJCqmEVQh+4PUkmaYII



```
EqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyCOnR/kEr
SBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9QSpJE0yQINUCJxikKmYR1KH
7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIifuD1JJmmCCBKNmOMEGVTG
LoA7dh6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEG1xwk
GqYpZBHXo/iCVpAkmSJBqjhMMUhWzCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEG
qOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjT
BBALSzXGCQapiFkEduj9IJWmCCRKkmuMEg1TFLII6dH+QStIEEyRINccJBqmKWQR16P4
glaQJJkiQao4TDFIVswjq0P1BKkkTTJAglRwnGKQqZhHUofuDVJImmCBBqjlOMehVzCK
oQ/cHqSRNMEGCVHOCYJCqmEVQh+4PUkmaYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkG
qYhZBHbo/SCVpggkSpJrjBINUxSyCOnR/kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGq
OEwxSFbMI6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTB
Bg1RznGCQqphFUIifuD1JJmmCCBKNmOMEGVTGLoA7dh6SSNMEECVLNcYJBqmIWQR26P0g
laYIJEqSa4wSDVMUsgjp0f5BK0gQTJEG1xwkGqYpZBHXo/iCVpAkmSJBqjhMMUhWzCOR
Q/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKq
YRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALSzXGCQapiFkEduj9IJWmCCRKkmuM
Eg1TFLII6dH+QStIEEyRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkkTTJA
glRwnGKQqZhHUofuDVJImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHOCYJCqmEVQh+4PUkm
aYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyCOnR
/kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9kqT1MK1VzCKoQ/dLktb
DpFYxi6AO3S9JWg+TWsUsgjp0vyRpPUxqFbMI6tD9kqT1MK1VzCKQJP0Eq4TLhPSr7Ch
JkqRVuJlJkiStws1MkiRpFW5mkiRJq3AzkyRJWoWbmSRJ0irczCRJklbhZiZJkrQKNzN
JkqRVuJlJkiStws1MkiRpFW5mkiRJq3AzkyRJWoWbmSRJ0irczCRJklbhZiZJkrQKNzN
JkqRVuJlJkiStws1MkiRpFW5mkiRJq3AzkyRJWoWbmSRJ0irczCRJklbhZiZJkrQKNzN
JkqRVuJlJkiStws1MkiRpFW5mkiRJq3AzkyRJWoWbmSRJ0irczCRJklbhZiZJkrQKNzN
JkqRVuJlJkiStws1MkiRpDf/73/8BfrWdG/8VnMAAAAASUVORK5CYII=)
```

```
"""
```

```
X = data_clean.drop('Outcome', axis=1)
y = data_clean['Outcome']
```

```
"""#Metode 90 10
```

```
#Berdasarkan Jurnal Analysis and Prediction Of Pima Indian Diabetes
Dataset Using SDKNN Classifier Technique
```



yang dikutip dari :  
<https://iopscience.iop.org/article/10.1088/1757-899X/1070/1/012059/meta>

"Our concept is implemented on Pima Indian Diabetes Dataset (PIDD). The analysis on Pima Indian Diabetes Dataset (PIDD) is carried out by splitting dataset in to 90% training data and 10% testing data. We have found that, in our proposed technique, average classification accuracy gives result 83.2%, a great improvement as compared to other conventional technique."

Untuk tahap train test split, saya akan membagi data train menjadi 90% dan data test menjadi 10% karena dapat menghasilkan akurasi yang lebih baik

#Berdasarkan A Novel Diabetes Healthcare Disease Prediction Framework Using Machine Learning Techniques

yang dikutip dari :  
<https://downloads.hindawi.com/journals/jhe/2022/1684017.pdf>

Foremost using a function like the model selection train test split, the data set is divided into the training and testing data sets. Due to the limited data set source, about 90%, of the data set, is used for training purposes and the remaining 10% is used for testing by selecting the data randomly.

Alasan saya membagi data train dan test menjadi 90% dan 10% adalah karena terbatasnya jumlah data untuk di training

![xytraintest.png] (data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAzMAAAHMCAIAAABukmEEAAAAAXNSR0IArs4c6QAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADs

MAAA7DAcdvqGQAABmPSURBVHhe7d09dtrM2wdgeNcCT5HzrACvAKdJlTYdlHaTLqW7NK  
Y0XdpUaWJWYK8gJ8UDe+GVRiMQH/bf39xY13VOTsQgJDCD9JuRNOoul8sOAAAB/F/+Hw  
CAQ5PMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAA  
CikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAK  
KQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAop  
DMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikM  
wAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzA  
AAopDMAACikMwAAKKQzAAAopDMAACikMwAAKLoLpfLPEk7dLvdpAVAMHbK6DMDAihCMg  
MAiEiYAwCIQjIDAIjCFQcTs3UFgAoAPIEtyUvx12SLPjMAgCgkMwCAKCQzAIAoJDMAgC  
gkMwCAKCQzAIAoJDMAgCgkMwCAKCQzAA5sNu6ujWe5FFpJMgMeaDE56Z5MFvkRvIwylZ  
12rpe1687p/05ns7H8xns1mcGBFXFnW2vj+zxOsVXfjodXM6vhrmg0xlezS8H09P7Ks  
JicjHNk/DuSGZwYGc3y+X1qNMZ5U6DYq90e96PmE96xVu90evlR69g9v38tnP787do1h  
qL3z9v04PPHzdrVe/j50FZEWA52VJ2j5WpPU1Piol+UVGmp6s2TP1cYd2Ttiqsf0s599  
cHTss583Tj17ZeVMSfIO0gmUEA/Q+DPFXmnx+XxW7p/HvrDtbMfnWui4wqmrXI/O9tp/  
PvP9txv/fPv5307d+y2VK2WLLR2U3RaCkm/smFZVombCgUWar/91tq1RTF04sqUM3G/Z  
+f56ml0zvnvj2flXGXuP+//+pTnLJJdnq5/by1F5ZeVpfdGJDMIJ+2WkqqVX/Uc7HQPrN  
v067KkeCKVpN1RKqn3MM0Zc9kdc27MWxXmOdOTjWdXb6Nx:Evd6KQ+3mPz68HU4/Fp2GW  
6n0sZ7Kd3zp+AdW/8wGmbfzZuXX6tjocOrKqVVeNimnviemffyrz2+2MKdvVx07It1J  
z+819Rg1IHXhnyCmWMq0rhrUlmEM1sfDrtdeEaf+nUr/6Lzo2jEb3UPrNv0s3H//N90KD  
T3JPzofEn7lenpl/KV5Uk7qSNhmflS7MTKfoS6c6Fc3r45myuqZi3SYZoz2fc2in1hfR  
L3/PJDnvERFr//fij3p01A1vRXM5rtfMA7/xQcm9RbvBuAFv/96XQGH/r54T3SnDtSV1  
yt2SP9P9UnFSSveuwe7iKZQRC5rd7tFrms2D1cdFPhm8Hlj3r/sLdNX+6a8j5s+G1UFp  
Xng1UZJu1ZVj0NjfPEyjlzyb45d/shho3jSnd1LRRFKR/1zs7Wp3M/0Oz730/VW9uJZr  
sfsCzUvfEulGeUbX55+88+26uss7d/5/1RrQxjG6W7B0zvsNkogEOQzCCIzMO9caHatp  
02fb1vy7u2MsPs3wU191LpqGPZLbdXOef+fogNu28j5cgqK+30Xy0aRyP3dm7Nfk1zzM  
pBq7GDvPMD6t54B3rlaZW3G32eZR/pbW4sVHIa33tBZpnWp6erly8mk2Kyive5/7eIea  
NPD2krVK9aLWs2dpScw8hbNvojf/G1XMphlX1jGzkjq/rM0vHHUv1w33ypPFk9mY/7bU  
5X86XJoiwveN+czbK1uvSut1FJzzbe9ENcX27OX66puYjdD3j/e+D15S+klkufaP0Flz  
a/19Vzg8vLol5UT5c1JBWt6nC2rjarwqqo+XDvdLXWxjt5q+qVV1fLpbSYStA6+ddfy6  
UcVr13WO9S1srdRLO82m3Ue4zrUXou/7epXOJ6vvXerFFWvKqwd840sSotVntZF6ayfW  
8jz1Mqnt7zju625wOkNWyuYMvePwVvp/jrN+VSHi//BWu51BZTCVon//pruZTDSQmjt  
pDpRSPSS3UUSWSJM/cKElG16uS4oXNhayma4PRaP+cxWIbM5cFzWXuexvNd7HxKe637z  
M2F7Wx6qxe/M574A3lv3wtl/J4+S9Yy6W0WLF416sD7dDtdvNUogIcv91k0j9bnZOzmJ  
x8/+fmnhPVjs+7/4BHyZbkpFhLssUVAHdcZuPT88bp/Yvfp/990NnOR+Pdf0CAJn1mra  
N99u7Mxo0rLQeX83d3leK7/4BHyZbkpFhLskUyax1bAeD5bEleir8kWxzNBACIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIwt2Z2mXrNiAABG  
TX3Gb6zFpELAM4CjbXbSaZAQBEIZkBAEQhmbWIExcAjoLNdZu5AqB1tk5fUAGAJ7AleS  
n+kmzRZwYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZA  
YAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBg  
AQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGAB

CFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEI  
VkBGAQhWQGABCFZAY81Wx80u12T8az/BiA55LM4E0tJmWYuctGxpmNc+mGk8kiP39ws1  
/T2+K/2+lFnPcEcoQkM3hTvbOb5XI+vx4NBrmkMricL5fLq2F+WBpeFSXFnPnhYHR5Xc  
xzc9bLBQc3/DQqP8Rg9C3OewI4cpIZvL1eb3h1cz0/bISzf//ZH25yaRHcbq7Oho8IQI  
vJ+NV7sopPUYTHm408CcBzSGZwKL2zm1WHWGd6uu8w5WJycjpNseyxvVKz7+d/8iQAx0  
MygwMaXq07zm7Pv2x1s9m4f37bGV0//gDmYnIxxZMAHBPJDA6qd/bj jmw2G1fdZY8/Vr  
iYfCkSHQBHSDKDA2se1Fx1s+ow5h06yxaz8UnZ0wbAUZLM4PCGV5vZLB3GfHx3WRHnTv  
qnaSSL0u15f3ekjcvSug5CVg3PsUjT5Yhkjb66NEджаI+T4tnZ5mHW2qLMgXsG8qhWUh  
dXM1XLGt+xJIjFWH0c0JKWyV98LZdycOuLAZLRdS5/vPrUtWogjtr8+jKNcVEpFr+5wj  
xzfm390uJF+TWbCysH81gP+9F4r1lleFq8H/Vh5xicjkPx11nLpe7Gqt1sv/zXkNdVyKS  
2mzwxiaF4MUOwOvr7sSBSLyZeLn38aBzl/jS8+FLucOq8NPn/s1ZccFGv/kQ+i9ob1aX  
C359/XvQez8ZeLP53dQ6b1Wv7crst/F/N9KAdhK8wv875u+ks3BNEZq48Dkswgit7Zt7  
pXa/c6zWfqnd0UVt1pnWnnUzqFrYhelZhk5YPPf3sG2uj982+eWithYyuWttMfltayKr  
79+fftZu09CFvv7FMu//OfI5o8h7H6eN8kM4iiHOpi1d/04t1s0+jTnj107+Pndf/zjg  
cHqv6HKv4NPm92/NXlt3/n6X94EmP18c5JZhBDGupid1fdWu9cjbBj906an3/p3SCf3  
mN6B3qqAVvx1h9vHuSGUSQzvAaXV8Ne40Bzg6QzSr5As4vX359+vFj9xx+OBRj9dEckh  
kcXBq8bLAAJOog2SwNcdE//fnh23x5c1MmxfwEHFhZN9M1KvC+SWZwYLM7rDmm7KGyWR  
ERY/HQRtfl1Yn78FzjajC7huYoYbPm03cOH2asPlpEMOND2uouWz1ENqsPFO29OgCe6m  
p9TXA5JMt8uWxW92H9dPnMnZdClpf9prFXqgWtRxxrWzQ5kvVPz3N0K9JeP0/ftuvfT2  
p3nE87n/MwLteXg9vi2dP+VvpKZ1d2+40cWMnlq5UUj4tH67A4Pe0bl5YXIZnB4eTusn  
r0sA1vn83mf6udjFEteGHNO5DtUda89SB6j2WspT4byQwOo2xxpl1tjXt113HAjm2037B  
/g9ufvqq+gaOs/5sXrIJh6CTauhFtMGkd/6ih3B6NjsDb8uicGVWa/it/B00d0NVYf74  
1kBm+tOo0lX41Gcd+mfNGMPmU4G88WD9jw52HJ0kvKE2D6Fx+qfoLF7Ht9/vS+xn2971  
y9ML3ypo6HqfBL52MdJBf1DjUpd7br9zX7/XO9ksbbXdxRTjuUAYmnie2614LZC5BN1  
YfRy/1xdIi+Yuv5VLeynZTvLJzL8nGiTnb1qfy3K28fWwee1Qdcd1/MGlnWfPVzTUHo8  
t8rKZ4bSqrF5VK0jybdu7FmZSr2PdpHvIxCCx/j7Vceq/ds8QKZZ158L1U9y6hsqpkD1  
rY/K47v2Z3rOiu9T9u5VuqV67kUlpMJWid/Ouv5VKAx8hbK Fouvd9ugEk1D48zdyWjwg  
PDUT77bFC2M1b325fMCMXRTADexM4RzfIQ94fvv8u+SwCY/URnWQGwBupT2acXpRXma  
Rgtv/Ur5dmrD6Oh2QGwFupu83SZNvFsyM1ccxkcvAeDvDPMTE9LQcze/po2U8hrH6OC  
aSQBvaD08y5P7sIzVx7uWrwSgNfIXX8ulAI+RtyC1XPPa+aLIx1/JuLoIMqsvklxdZ1  
nYt9Tt11Wv3ChTXG+5ubT1ZzjX6/mbxc3lbJQ/TH51LZfSYipB6+RffY2XajxG3oLUcu  
lDpSzzlCEm3uFYfflvTvxKi3WLF7k60A7dbjdPJSOa8ATP25LMxt2LD/N046TWs01mi/  
PMAHhTi8nF9I1Gy4DjI5k8JbecBgzoEKOZraOnnPg+Z6+JSkHff37bX11aLGKbTJb9J  
kB8KrK2yJ1uyfjNO7/5Mt5561uyATHSJ9Z62ifAc/3iC3JbNw9bYwcNrrWX9Zkm8wWfW  
YAvKbh13rAijQ6hVgG99Jn1jraZ8Dz2ZK8FH9JtugzAwCIQjIDAihCMgMAiEiYAwCIQj  
IDAihCMgMAiEiYAwCIQjIDAihCMgMAiEiYAwCIQjIDAihCMgMAiEiYAwCIQjIDAihCMg  
MAiEiYAwCIQjIDAihCMgMAiEiYAwCIQjIDAihCMgMAiEiYAwCIQjIDAihCMgMAiEiYAw  
CIQjIDAihCMgMAiEiYAwCIQjIDAihCMgMAiEiYAwCIQjIDAihCMgMAiEiYAwCIQjIDA  
ihCMgMAiEiYAwCIQjIDAihCMgMAiEiYAwCIQjIDAihCMgMAiEiYAwCIQjIDAihCMgMAiE  
iYAwCIQjIDAihCMgMAiEiYAwCIQjIDAihCMgMAiEiYAwCIQjIDAihCMgMAiEiYAwCIQj





```

6L3ZAJjps+s9bRPgOe771bkmpo5ZXRdSv7y0q2yWzRZwbAmxt+rW9QmYagaGssg136zF
pH+wx4PluSl+IvyRZ9ZgAAUUhmAABRSGYAAFFIZgAAUUhmAABRSGYAAFFIZgAAUUhMAA
BRSGYAAFFIZgAAUUhmAABRSGYAAFFIZgAAUUhmAABRSGYAAFFIZgAAUUhmAABRSGYAAF
FIZgAAUUhmAABRSGYAAFF018tlnqQdutlungIgdTl9JkBAEQhmQEARCGZAQBEIZkBAE
ThCgAAnmJlOZH9CLwgyQwAIpHMwEAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzA
AAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAA
CikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAK
KQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAop
DMAACikMwAAKKQzAAAopDMAACikMwAAGLodP4fSJaSB2q93ToAAAAASUVORK5CYII=)
"""

#test size 10% and train size 90%
#Random state = Mengontrol pengacakan yang diterapkan ke data sebelum
menerapkan pemisahan agar hasil yang didapatkan tetap sama
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.1,
random_state=0) #split test 10% dan train sisa dari test yaitu 90%

"""#Decision Tree"""

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(random_state=0) #Mengontrol keacakan
estimator agar hasil yang didapatkan selalu tetap
dtree.fit(X_train, y_train)

from sklearn.metrics import confusion_matrix

"""#Berdasarkan Jurnal Machine learning models for non-invasive glucose
measurement: towards diabetes management in smart healthcare
yang dikutip dari :
https://link.springer.com/article/10.1007/s12553-022-00690-7

```

**\*\*Precision:\*\*** Precision is measurement of how many positive predictions are made for actual positive values.

**\*\*Recall/true positive rate/sensitivity:\*\*** Recall is measurement of correct positive predictions from all positive predictions made

**\*\*F-1 Score:\*\*** F-1 Measure is a combination of precision and recall

#Penjelasan

**\*\*Precision:\*\*** Terdapat 76% pasien yang **\*\*sebenarnya\*\*** terkena diabetes dibandingkan dengan keseluruhan pasien yang **\*\*diprediksi\*\*** terkena diabetes.

**\*\*Recall ::\*\*** Terdapat 87% pasien yang **\*\*diprediksi\*\*** terkena diabetes dibandingkan dengan keseluruhan pasien yang **\*\*sebenarnya\*\*** terkena diabetes.

**\*\*F-1 Score ::\*\*** jika F1-Score punya skor yang baik mengindikasikan bahwa model klasifikasi punya precision dan recall yang baik

#Berdasarkan jurnal The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation

"Checking only F1, one would read a good value (0.66 in the [0, 1] interval)."

nilai F1-Score bisa diilang bagus jika value F1 berada diantara 0,66 hingga 1

#Alasan menggunakan Precision Recall

yang dikutip dari : Jurnal The relationship between Recall and Precision

[https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/\(SICI\)1097-4571\(199401\)45:1%3C12::AID-ASI2%3E3.0.CO;2-L](https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1097-4571(199401)45:1%3C12::AID-ASI2%3E3.0.CO;2-L)

"Recall and Precision and, in particular, Recall-Precision plots, have been used for many years to characterize document retrieval performance."

Alasan saya menggunakan Recall dan precision adalah untuk melihat kinerja model

#Berdasarkan Jurnal Diagnose Diabetic Mellitus Illness Based on IoT Smart Architecture

yang dikutip dari : <https://www.hindawi.com/journals/wcmc/2022/7268571/>

"for the classification considering the evaluative measures like accuracy, precision, recall, and F1-score on PIDD and claimed to have achieved comparatively enhanced results on binary classifications"

Alasan kedua saya menggunakan precision recall adalah karena precision recall lebih cocok digunakan untuk binary classifications

"""

```
y_pred = dtree.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))
```

```
""#Berdasarkan Jurnal Diabetes Mellitus Prediction using  
Classification Techniques
```

```
Yang dikutip dari :  
https://www.researchgate.net/profile/Anny-Leema/publication/344072667\_Diabetes\_Mellitus\_Prediction\_using\_Classification\_Techniques/links/5f50f16da6fdcc9879c511e9/Diabetes-Mellitus-Prediction-using-Classification-Techniques.pdf
```

```
"Confusion matrix is used to visualize the performance of the  
algorithms which cross tabulates the observed and predicted  
classes with associated statistics, evaluation metrics like  
sensitivity, specificity, precision and accuracy are used to  
evaluate the performance of the method. Factors like True  
Positive (TP), True Negative (TN), False Positive (FP) and  
False Negative (FN) are used"
```

```
Alasan saya menggunakan confusion matrix sebagai visualisasi data  
adalah untuk memvisualisasikan performa dari sebuah algoritma  
klasifikasi
```

```
![format confusion  
matrix.png] (data:image/png;base64,iVBORw0KGgoAAAANSUUhEUgAAAOoAAACNCA  
IAAAA2B3H7AAAAXNSR0I Ars4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA  
7DAcdvqGQAABCosURBVHhe7Z1teeTGEoUvhWAIBXMwhGAwhWWwDMIgCIxgCSyBMAiH3H  
d12sc1pZlJ4im1JG+9P+ap/lCp+/RRS7PW2v/7u21OS9u3OTFt3+bEtH2bE/Nu3/81zR  
kYfl124sO+ImiJa0nLavvNoSctp+86jJS2n7TuPlrSctu88TiTp9+/fX15efvn11E+Ko  
ez77dv3zj1HUa/EzJ/8FLsKl+/fh2droF9f/vtN7qN81FJI3wv7DV07Mt1/9dff6nIMJ  
6fnxX/+eefT09Pis/ILpL+8ccfUUNAWxS+b1+gw14e+PekEe5v3yRrkv719dXOPh27SK  
q7WdRQ/P777yO6Qdu3gLX02Bfpf/31Vxbmy5cvdCDmE2R9P37EK4GYJzkq2b+5M47auX  
D2EU3kln2FdmIpg4xs1aNhZV+aLCCHjNplQ5H+6fA5xBHCCezLI4SURUS0o5UijqTGfk  
1FegJLpccPlmGXLXwXSdf25aF2RIsy2A5ZEEQPu1Ym2hc9EYluxOwdzoZf0VOHk4r+nE  
5Nc0iSnsC+oCVJ1zo19iu4iPSskCoBx9P0j7fOLEC8I5qI70WR0bZY2W6WMvZftC+VUU  
Mrb08DDqa/s80hzgXeC6lhLxjGLftaaEHNvftqGRJsFeo2E847oolIK2uIydZff7Egj2  
F6NrhqX47CviSJmitzIrp8ApxxRAuf077rDLuwi6TSKioQVcKXMi63ozu7L9CTGnqSS1
```



8elFmte5EG8F7YfWSCYTxuX2LUV71gqUY0EYYxooms7Ruhns1Y4iRVk30FPXk80DOD+i  
clJwubRnhu+7KRKEZlirKvvsbFpwUknqyyYBgmjoi0umVfqwRJ1WhfKl0PPgof2/2Ap5  
1tDh6hOJZ9+YrAMHicSnsndzrq07cEuqEmvuQoSc+aSXR6Kg/1+PjWWm7NLpJKw2iyiP  
2H8/RPB2gLNEk0PyfQTWLySbliLQRJ2DhgVJbw9lHtHAg+2IyxmB8WcuaIhoRQdGRSu  
3B01Tfi9GUWK0s0mSJDWcf0Sx+aBRYb42YG1kAI6IVAZKiTxSfo9DWNXjU//IAHEgN9X  
SQ12fCeUe0cCD7fj5a0nLavvNoSctp+86jJS2n7TuPlrSctu88WtJy2r7zaEnLafvOoy  
Utp+07j5a0nLbvPFRScn4i+35dXpja60ducBZJ7wsVf2i809vaN/6AN/6kEaSC2UIOvR  
U1Cp/Rvk1D8fT0xExHjw+xFiqunU66xXp9AEYyooVi+wIe0ssfKLL+mThNMArVvLy8HE  
RlUSVpQq/ayLJWm0q1Pg7effB62I4kab19hd7qwMF6h8agy0bSIDpn/Bnsi4Bktozsmu  
hMTckLNCQh20Zr9DhJ0q3s+/z8LD+1N/d+mDdIglh6swNJ9OKY4BDtMRFtMByi/QZIrj  
V7Xf7jgMHElOulM1p1+xOc7scJlvnGIh10yaUXrB6BbCMqBQHJHGwUhq7xS2F8xm6oar  
UJdKlHofQOGH0Ex1KjbOqsw4VqNBgXby1oFWQe0cKG9uVTr4cqFswWFDNV66KebSXX8j  
0Q0CSbAodgX+pRlphW1ZOHbsqmolyuoq8lFYEMHM4nMa00kZCiLhvneQSfvRZUIr01Ai  
biMWMazYVYqvq5AkFkKVrpo/5JKIjJkV2nU2cvR3z9n5yciODOglYRwxkXgx5RBbas3G  
AFmY+nhGpxn0OXuFm6SUUFRR+JBQjnYaubtBMc4lZgSIg7CotlfQpyar2BFeIoxqbiI8  
SzF6J5yYi+3lCPIrMgjt6SNXXxo4CnTE9rlyQittqQhFXRecBrfWtBC4njhItBj6iCuO  
NKQXkOXSwN1WuQFa0J3G2tF1CpZQPXEftlIINbIeWxQVWfiEb/MOQZUSmaFYPEHAR8Io  
VuI2rypQh6rJKYLAGdidXZJKHcX6yFjRsB9d5N6LYmDuZxSDiihYtBj6iCaF/fcbAOul  
gaaqIoEdYDgTRzYkS34lSSnITK5mGvVY6tgpzSnW4ehg5UXMtGaTUvjz+iJj9ogWYXJ4  
t6MrFqQEeNwjLs2KoMfI7y25OYNgJ2AS9N6rYFcZxwMegRVRDtC8wQ36Aas7U0nFF3QO  
O7Hv3pySH0IZXXg3ptNipG3dcqp1UB6c5ZyOBdQQfGGy6k4sdIZ69C84oOM2qKg9fsvE  
EK0iCjxU9CpeRrYYGLYQdBQ68F0O3Wg1YRwxkXgx5RBcm+gAWRLEqDQamJW4W14Pck10  
hSRt3XKqdVEZwR3aPooMq4wcf1+zDrs5egeV0doZ67ovgYiBpdq/GQKE6MISVfCwtcD1  
TKwaPq7oJWEccJF4Me0cPIqXFiQuJaGhWBSVLJBW2NqKTIMoBa1aRDJAo1tFJkhdJca  
E6oyY11P/I+AYdqLy6GIyZzQM4ta38COQcUS1Mn8y3nKFWbbcsAQ7zjohc0gcQh6Lqk1  
DooCZU5VPipP2bDO5m7ixofSQf0UK9fWUpwRxG7Rt6YB2FxyLoS0+mKrEE81d9RFqwgM  
SShkOIvTxaORXjMKKI6J5EFywPOelMa3L8hyHbiIpgI sue3olTMyiMt2hlrtF26KM5Ap  
ZFCirXQukRi3o6kEpNkFYTnddnP+bqglZB5hEt1Nv3cVDNjhTSMcl3fi4j6afhBPblul  
871UtZX3VPRNu3nBPY17sP9z7uero38Y1x0358Ctq+5ZzAvnp48FMam/Hp911xHEk/DU  
nSI9r309CS1tP2nUdLWk7bdx4taTlt33m0pOW0fefRkpbT9p1HS1pO23ceLWk5bd95tK  
Tlth3n0ZKW0/adR0taTtt3Hi1pOW3febSk5bR959GS1tP2nUdLWk7bdx4taTlt33m0pO  
W0fefRkpbT9p1HS1pO23ceLWk5bd95tKTlth3n0ZKW0/adR0tazj37Ns3xGX5duLDviJ  
oiWtJy2r7zaEnLafvOoyUtp+07j5a0nM9g3+/fv78sf/BilI9K27ecA9mXM97i/q/yxb  
76heCjffT2GuG31W+xTox+JyQN/r2wy6z8i7xHefn1kvrV8qN8Azocfxn2GiH2RUOUVD  
EqrD8XoPiMJE13tq/2iWhfkf6Swpq27x3SxZ8Ufn19tbNPR5L0oPYV2on9Zxrib/1N9q  
VJ3dhaOGTULku13xOcDp9DHOGOMIykMMLqb3mg/5e3vxXCJ8j6fvYIVwKxRa768x//Fc  
4+ooXD2ZeH2hg9/SkR7nfIrYddbvxRvkiJrPpDRqyKs+FXhNbhpkI/p1PTHaRde1SGN  
BEAiILVzitFJGRGvs1FekJKNHDWt3WswkSXoI+yZG22JluxmVabL/on2pxOWKwbusPQ  
1oTX9nm0Ocy44wjGRfkPLpjksN/QouYuWosNbiHx/wtoDzjmjhvZaa5iARLS4mW3+XwI  
Lc4HTbumpfjkJckrgVldkR12ACnHFEu8IwbtK3KgbUXLWv1E6wGavbTDjviBaOZV+I8u

FLGZcL/c7uC/Skhp6k0mOZMqt1L3YfgEgKi/9q33WGXUISvhdSwxzW9o1Qz2aMNYmT3M  
m+gp48HuiZQf0x/WhbSMWt2UXSNVcVTnoKam7Zl1gLYSaLKRjGiBYObV/LB0nuaf8qXQ  
8+Ch/b/advK4rn4BHuc8P49/b1HyBDNyupr3HxaQHvtn3Hjy2iySL2H85DO3ryFKFvDO  
yyFP2cQDetRFwVehKThCUBHnyvnmU7OPuI9kMKr+cucdJ3WbohF77kKG0Q+F5iSnA6UM  
9a3NpxtiZJuqd9OWNkvTUiImoCWuNgAlRjGfgcxyxHoa9r0Dd+m+ZAaqing7w+E847op  
2IQoEV1jUFfVQJKInIVGoPJiZAeWJkJ1ar/gVtOWI2nH1EC3va99PTkpbT9p1HS1pO23  
ceLWk5bd95tKT1tH3n0ZKW0/adR0taTtt3Hi1pOW3febSk5bR959GSlvMT2ffr8g7aXj  
8fgrNIel+o+HP43dnWvvGnkfgHtyAVzBZy6EWzUfiM9k0aiqenJ2Y6enyItVBx7XTSLd  
brAzCSES0U2xfwkF7vQJH1awY0wShU8/LychCVRZwKCb29JmtabSrV+jh6X2cUDkaStN  
6+Qi/K4GC98GHQZSNpEJ0z/gz2RUAYW0Z2TXSmpuSdJKQbaM1epwk6Vb2fx5+lp/Sy5  
A/zBukQSy9FYVkestJcIj2mIg2GA7RfgMk15q9Lv8Xw2Bi6vWGFK26/Qm/YJWKZNA119  
5ZewSyjagUBCRz1FEausbv2fEzu6Gq1SbQpR6FYq8hoIPgWGqUTZ11uFCNBuPirQWtgs  
wjWtjQvnzqpVLFgtmCYqZqXdTTTfhSvgeCmmRT4BDsSz3KEtOqevLQTDlU1MltV9LWkIp  
CBw/kkppUmELLUZeM8j+Cz14JKZLZWwEQ8ZkyjuRBLVT9XIIGsRSt91D8JBTE5sut06u  
zliO+qk5MTEdxZ0CriOOFi0COqwJaVG6wg8/GUUC3uc+gSN0s3qei j6COxAOE8bHWTdo  
JD3AoMCXFHYbGsT0FOrTewQhzF2FR8hHj2QjQvGdHXG+pRZBbE0Vuypi5+FPCU6Wmtkl  
DEVhuSsCo6D3itby1oIXGccDHoEVUQdlwpKM+hi6Whcg2yojWBU631Aiq1bOAaYqsMZH  
ArpDw2qOoT0egfhjwjKkXzYoSYg4BPpNBtRE2+FEGPVRKTJaAzsTqbJJT7i7WwcSOg3r  
sJ3dbEwTwOCUe0cDHoEVUQ7es7DtZBF0tDTRQlwnogkGZOjOhWnEqSk1DZPOy1yrFvKf  
O6083D0IGKa9korebl8UfU5Act0OziZFFPJLYN6KhRWIYdW5WBz1F+exLTRsAu4KVJ3b  
YgjhMuBj2iCqJ9gRniG1RjtpaGM+oOaHzXoz89OYQ+pPJ6UK/NRswO+1rltCog3TkLGB  
wr6MB4w4VU/Bjp7FVoXtFhRk1x8JqdN0hBB2S0+EmolHwtLLA07CB06LUAut1a0CriOO  
Fi0COqINkXsCCSRWkwKDVxq7AWHJ70EknKqPta5bQqgjOieXQdVBk3+Lh+H2Z99hI0r6  
sj1HNXFB8DUaNR4SxYkxpORrYYHrgUo5eFTdXdAq4jjhYtAjehg5NU5MSFxLoyIwSS  
q5oK0R1RRZBlCrmnSIRKGGVoqsEDg5LlRn1KSG+h8Z36AD1VcXgzGzeQCntpUfgZwjKo  
Xpk/mWM9Sq7ZYlwGHeEZFL+gDiUFR9Egod1ISqfEqctH+Twd3MnQWtguQjWqi3rywlmM  
OofUMPrKOWBB96clUJZZg/qcPSAsWgljScAixl0crp2IcRhQR3ZPoguUhJ51pTY7/MG  
QbURFMZJnQO3FqBoXxFq3MKNoOfTRHwLJIQeVaKD1iUU8HUqkJ0mqi8/rs1Fxd0CrIPK  
KFevs+DqrZkUI6JvmOz3Ek/TScwL5c92uncinrq+6JaPuWcwL7cvfh3sddT/cmPjFu2o  
9PQdu3nBPYVw8PfkpjMz7dviuOI+mnIUl6RPt+GlrSctq+82hJy2n7zqMlLaft04+WtJ  
y27zxa0nLavvNoSctp+86jJS2n7TuPlrSctu88WtJy2r7zaEnLafvOoyUtp+07j5a0nL  
bvPfrSctq+82hJy2n7zqMlLaft04+WtJy27zxa0nLavvNoSctp+86jJS2n7TuPlrSctu  
88WtJy2r7zaEnLuWffpjkw68LvT00J6bt25yYtm9zYtq+zWn5+//A3Xnpu/lu9g3AA  
AAAE1FTkSuQmCC)

#Beberapa definisi dari bagian confusion matrix

True positive: 41 kasus di mana mesin memperkirakan jika pasien memiliki diabetes, dan mereka memang memiliki diabetes.

False-positive: 13 kasus dimana mesin memperkirakan jika pasien memiliki diabetes, padahal sebenarnya tidak memiliki diabetes

False-negative: 6 kasus dimana mesin memperkirakan jika pasien tidak memiliki diabetes, padahal sebenarnya memiliki diabetes

True Negative: 35 kasus di mana mesin memperkirakan jika pasien tidak memiliki diabetes, dan mereka memang memiliki diabetes.

```
"""
```

```
cm = confusion_matrix(y_test, y_pred) #memasukkan y_test dan y_pred
plt.figure(figsize=(5,5)) #ukuran plotting heatmap
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label') #Mendefinisikan sumbu y untuk Actual label
plt.xlabel('Predicted label') #Mendefinisikan sumbu x untuk predicted
label
all_sample_title = 'Accuracy Score: {0}'.format(dtree.score(X_test,
y_test)) #melakukan kalkulasi score akurasi, {0} berfungsi untuk
menampilkan angka akurasi
plt.title(all_sample_title, size = 15) #setting ukuran tulisan
"Accuracy Score: 0.8"

"""#XGBoost"""
```

```
from xgboost import XGBClassifier
```

```
xgb_model = XGBClassifier()
```

```
xgb_model.fit(X_train, y_train)
```

```
y_pred = xgb_model.predict(X_test)
```

```

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))

print('Precision Score : ',(precision_score(y_test, y_pred)))

print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(5,5))

sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')

plt.ylabel('Actual label')

plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
    y_test)) # menampilkan score akurasi

plt.title(all_sample_title, size = 15) #format title untuk accuracy
    score

"""#Metode 80 20

#Berdasarkan Jurnal Improving the Classification Accuracy using
    Recursive Feature Elimination with Cross-Validation
yang dikutip dari :
    http://www.puneetmisra.com/admin/uploads/journals/5f136d202b8ba1.186
    44117.pdf

"In
this study, we have used the train_test_split() method of
the Scikit-Learn library of python. Through this function, we divide
the dataset into a different ratio. However, the 80/20 (train/test)
rule is mostly used in the studies."

Tujuan penggunaan train test split menjadi 80 dan 20 adalah karena
rasio tersebut paling sering digunakan dalam penelitian

```



```

"""
#test size 20% and train size 80%
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2,
    random_state=0)

""">#Decision Tree"""

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(random_state=0)
dtree.fit(X_train, y_train)

y_pred = dtree.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score
print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {0}'.format(dtree.score(X_test,
    y_test))
plt.title(all_sample_title, size = 15)

```

```

"""#XGBoost"""

from xgboost import XGBClassifier

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score
print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
    y_test))
plt.title(all_sample_title, size = 15)

"""#Metode 70 30

#Berdasarkan Jurnal A Comparative Analysis for Diabetic Prediction
    Based on Machine Learning Techniques

Yang dikutip dari : https://www.iasj.net/iasj/download/a371daadb33b96fd

```

Disimpulkan bahwa Train Test split dengan rasio 70 dan 30 dapat menghasilkan akurasi yang lebih besar dibandingkan dengan metode K-Fold Cross Validation

```
"""  
  
#test size 30% and train size 70%  
  
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,  
    random_state=0)  
  
""">#Decision Tree""  
  
from sklearn.tree import DecisionTreeClassifier  
  
dtree = DecisionTreeClassifier(random_state=0)  
dtree.fit(X_train, y_train)  
  
y_pred = dtree.predict(X_test)  
  
from sklearn.metrics import accuracy_score, f1_score, precision_score,  
    recall_score  
  
print('F-1 Score : ',(f1_score(y_test, y_pred)))  
print('Precision Score : ',(precision_score(y_test, y_pred)))  
print('Recall Score : ',(recall_score(y_test, y_pred)))  
  
cm = confusion_matrix(y_test, y_pred)  
  
plt.figure(figsize=(5,5))  
  
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')  
  
plt.ylabel('Actual label')  
plt.xlabel('Predicted label')
```

```

all_sample_title = 'Accuracy Score: {0}'.format(dtree.score(X_test,
    y_test))

plt.title(all_sample_title, size = 15)

""""#XGBoost""""

from xgboost import XGBClassifier

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
    y_test))

plt.title(all_sample_title, size = 15)

""""#Metode 60 40""""

#test size 40% and train size 60%

```



```

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.4,
    random_state=0)

"""#Decision Tree"""

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(random_state=0)
dtree.fit(X_train, y_train)

y_pred = dtree.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {}'.format(dtree.score(X_test,
    y_test))

plt.title(all_sample_title, size = 15)

"""#XGBoost"""

from xgboost import XGBClassifier

```

```

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
    y_test))
plt.title(all_sample_title, size = 15)

"""#Hasil Perbandingan Akurasi Train Test Split

Decision Tree      XGBoost

86.36              87.50      Train Test 90 10

86.28              86.85      Train Test 80 20

83.58              86.64      Train Test 70 30

```

```
80.22          84.52  Train Test 60 40
```

#### Test 4 : Random state = 42, Imbalanced dataset

```
# -*- coding: utf-8 -*-

"""Diabetes_Prediction_Using_Decision_Tree_and_XGBoost_Algorithm_imbalanced_random_state42.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1kIOsG-tGXhTfiKnTkIqz-MTVnDGbQjvr

#Importing Library and Preparing the dataset
"""

# Commented out IPython magic to ensure Python compatibility.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(color_codes=True)
from sklearn.model_selection import train_test_split

# %matplotlib inline

diabetes_df = pd.read_csv('diabetes.csv')
diabetes_df
```

```

"""#Exploratory Data Analysis"""

#Column in Dataset
diabetes_df.columns

#Show data types and null value each column
diabetes_df.info()

#Menampilkan 10 baris pertama
diabetes_df.isnull().head(10)

#Checking if there is null value
diabetes_df.isnull().sum()

#Checking if there is zero value

#replace 0 value with NaN
diabetes_df_copy = diabetes_df.copy(deep = True) #deep = True -> Buat
salinan indeks dan data dalam dataframe

diabetes_df_copy[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', '
BMI']] =
diabetes_df_copy[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin
', 'BMI']].replace(0, np.NaN)

# Showing the Count of NaNs
print(diabetes_df_copy.isnull().sum())

"""#Berdasarkan Jurnal A model for early prediction of diabetes
oleh Talha Mahboob Alam et al, yang bersumber dari :
https://reader.elsevier.com/reader/sd/pii/S2352914819300176?token=7D4F21A1900E1B47FA971E110A1469CE2A8882E14CFBF3A4C3790507841542704D1370

```

```
C180ADD4341ED02B627B619B3F&originRegion=eu-west-1&originCreation=2020928035824
```

```
"Data cleaning consists of filling the missing values and removing noisy data. Noisy data contains outliers which are removed to resolve inconsistencies . In our dataset, glucose, blood Pressure, skin thickness, insulin, and BMI have some zero values. **Thus, all the zero values were replaced with the median value of that attribute.**"
```

Jadi di tahap ini yang saya lakukan adalah mengisi value 0 (missing values) pada beberapa kolom tersebut dengan nilai median untuk mengatasi nilai inconsistent

```
"""
```

```
#Fill null value with median
```

```
diabetes_df_copy['Glucose'].fillna(diabetes_df_copy['Glucose'].median(), inplace = True)
```

```
diabetes_df_copy['BloodPressure'].fillna(diabetes_df_copy['BloodPressure'].median(), inplace = True)
```

```
diabetes_df_copy['SkinThickness'].fillna(diabetes_df_copy['SkinThickness'].median(), inplace = True)
```

```
diabetes_df_copy['Insulin'].fillna(diabetes_df_copy['Insulin'].median(), inplace = True)
```

```
diabetes_df_copy['BMI'].fillna(diabetes_df_copy['BMI'].median(), inplace = True)
```

```
#inplace = True -> untuk menyimpan hasil modifikasi pada dataframe diabetes_df_copy
```

```
"""#Berdasarkan Jurnal Deteksi Dini Penyakit Diabetes Menggunakan Machine Learning dengan Algoritma Logistic Regression
```



yang dikutip dari :  
<https://jurnal.ugm.ac.id/v3/JNTETI/article/view/3586/1646>

"Pada tahap ini juga dilakukan pengecekan terhadap data yang tidak seimbang. Penanganan terhadap data yang tidak seimbang dilakukan menggunakan synthetic minority over-sampling technique (SMOTE) "

Pada tahap ini yang akan saya lakukan adalah oversampling minority value, yakni value 1 pada variabel outcome yang berfungsi untuk menyeimbangkan data yang nantinya akan digunakan untuk prediksi

#Berdasarkan Jurnal SMOTE: Synthetic Minority Over-sampling Technique

yang dikutip dari :  
<https://www.jair.org/index.php/jair/article/view/10302/24590>

"However, the nature of the application requires a fairly high rate of correct detection in the minority class and allows for a small error rate in the majority class in order to achieve this."

Alasan saya menggunakan teknik Oversampling adalah untuk meningkatkan sampel kelas minoritas dan untuk memperkecil tingkat kesalahan di kelas mayoritas

#Berdasarkan PENERAPAN SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE (SMOTE) TERHADAP DATA TIDAK SEIMBANG PADA PEMBUATAN MODEL KOMPOSISI JAMU

yang dikutip dari :  
<https://journal.ipb.ac.id/index.php/xplore/article/view/12424/9491>

"the model with SMOTE is more accurate than model without SMOTE because has higher AUC value."

Keunggulan dari menerapkan metode Oversampling minority adalah bisa meningkatkan nilai AUC

```
"""
```

```
#Counting 1 and 0 Value in Outcome column
```

```
sns.countplot(diabetes_df_copy['Outcome']) #membuat bar plot  
perbandingan jumlah value
```

```
print(diabetes_df_copy.Outcome.value_counts()) #menampilkan jumlah  
value 0 dan 1
```

```
"""#Berdasarkan Jurnal Prediksi Risiko Penyakit Diabetes menggunakan  
Algoritma Regresi Logistik
```

```
oleh Qatrunnada Refa Cahyani, Mochammad Januar Finandi, Jathu Rianti,  
Devi Lestari Arianti, Arya Dwi Pratama Putra yang dapat diakses  
melalui
```

```
https://journal.literasisains.id/index.php/jomlai/article/view/598/470,  
kita melakukan tahap preprocessing yaitu pengecekan outliers  
pada variabel. Outliers dihapus dengan menggunakan Z-Score.  
Beberapa data dihapus karena nilainya jauh dari Z-Score. Maka jumlah  
data yang awalnya 1000 menjadi 938
```

```
#Berdasarkan Detection of Spatial Outlier by Using Improved Z-Score  
Test
```

```
yang dapat diakses melalui : https://booksc.org/book/77788899/9f8c40,  
"The zscore test has long been used to detect outliers in data."
```

Fungsi Tes Z-Score adalah untuk mendeteksi apakah terdapat outlier dalam data.

```
#Alasan Menggunakan Z-Scores
```

```
##Berdasarkan Outlier Detection in Multivariate Time Series Data Using  
a Fusion of K-Medoid, Standardized Euclidean Distance and Z-Score
```

yang dapat diakses melalui :  
[https://link.springer.com/chapter/10.1007/978-3-030-69143-1\\_21](https://link.springer.com/chapter/10.1007/978-3-030-69143-1_21)

Z-score technique produced a better outlier detection result of 0.9978 F-measure as compared to inter-quartile of 0.8571 F-measure.

Alasan saya menggunakan Z Score adalah karena Teknik Z-score menghasilkan hasil deteksi outlier yang lebih baik

```
#Checking Outliers using Box Plot
"""
sns.boxplot(x=diabetes_df_copy["Pregnancies"])
sns.boxplot(x=diabetes_df_copy["Glucose"])
sns.boxplot(x=diabetes_df_copy["BloodPressure"])
sns.boxplot(x=diabetes_df_copy["SkinThickness"])
sns.boxplot(x=diabetes_df_copy["Insulin"])
sns.boxplot(x=diabetes_df_copy["BMI"])
sns.boxplot(x=diabetes_df_copy["DiabetesPedigreeFunction"])
sns.boxplot(x=diabetes_df_copy["Age"])
""">#Removing Outliers using Z Scores
```

```

#Berdasarkan jurnal Robust statistics for outlier detection

yang          dikutip          dari          :
      https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.2

>Data preprocessing is a useful step that helps to remove
noise, inconsistencies and redundancy to achieve high
quality data which improves the performance. During the
data pre-processing, all the missing values are filled by the
mean of the corresponding feature. Formatting of the given
dataset is ensured to be consistent. All the incorrect data
types of the features changed to their required datatype.
Data normalization is performed on the given dataset to
make its range consistent. We used z-score normalization
on all the features to restrict the range of values between 3
to - 3"

"Generally, some of machine learning algorithms does
not handle categorical data. Diagnosis feature in this
dataset contains two categories M(malignant) and B(benign) is replaced
numeric with M by 1 and N by 0."

Menurut jurnal tersebut, ambang batas Z-Score yang dapat diterapkan
adalah antara -3 dan 3 untuk 2 kategori diagnosis (True or False)
"""

import scipy.stats as stats

z = np.abs(stats.zscore(diabetes_df_copy))

data_clean = diabetes_df_copy[(z<3).all(axis = 1)] #print all of rows
that have z<3 (z score below 3)

```

```

data_clean.shape

#z<3 = print semua data yang Z-Scorenya kurang dari 3
#.all(axis=1) = melihat di bagian kolom untuk setiap baris (melihat
kolom z score setelah di proses dengan print z kurang dari 3)

#Cleaned Outliers data using Z Scores

data_clean

"""#Data Correlation
Berdasarkan Diabetes Prediction Based on XGBoost Algorithm oleh Minggu
Li et al 2020 bersumber dari
https://iopscience.iop.org/article/10.1088/1757-899X/768/7/072093/pdf

Setelah data preprocessing, kita harus mempelajari apakah data tersebut
diproses dengan baik atau tidak dan bagaimana caranya
banyak korelasi antara data tersebut. Koefisien korelasi dapat
digunakan untuk mencerminkan kedekatan
hubungan antar variabel. Koefisien korelasi dihitung dengan metode
selisih.
Hal ini juga didasarkan pada dispersi kedua variabel dan rata-ratanya
masing-masing. keduanya
perbedaan dikalikan untuk mencerminkan tingkat korelasi antara dua
variabel

#Berdasarkan Jurnal Prediksi Risiko Penyakit Diabetes menggunakan
Algoritma Regresi Logistik

Jika nilai korelasi > 0 maka terdapat korelasi positif.Sementara
nilai satu variabel meningkat, nilai variabel lainnya juga
meningkat. Jika persamaan korelasi = 0 maka tidak ada korelasi.
Jika korelasi < 0 maka ada korelasi negatif. Sementara satu
variabel meningkat, variabel lainnya menurun. Ketika korelasi

```



diperiksa, ada 2 variabel yang bertindak sebagai korelasi positif terhadap variabel dependen outcome, variabel tersebut adalah glucose. Seiring peningkatan ini, variabel dependen juga meningkat. Dengan demikian, semua variabel digunakan untuk modelling karena korelasinya berdekatan.

#Berdasarkan Complex heatmaps reveal patterns and correlations in multidimensional genomic data

Yang dapat dikutip dari :  
<https://academic.oup.com/bioinformatics/article/32/18/2847/1743594>

Heatmaps are a fundamental visualization method that is broadly used to unravel patterns hidden in genomic data. They are especially popular for gene expression analysis (Eisen et al., 1998) and methylation profiling (Sturm et al., 2012). With the increasing availability of genomic datasets, \*\*visualization methods that effectively show relations within multidimensional data are urgently needed.\*\*

Alasan saya menggunakan heatmap adalah karena heatmap correlation dapat digunakan untuk menemukan hubungan potensial antara variabel dan untuk memahami kekuatan hubungan ini dan mendeteksi hubungan linier dan nonlinier

#Fungsi Heatmap Correlation Matrix

Berdasarkan Jurnal Implementasi Seleksi Fitur Klasifikasi Waktu Kelulusan Mahasiswa Menggunakan Correlation

Matrix With Heatmap

Yang dikutip dari :  
<https://ejurnal.ung.ac.id/index.php/jjee/article/view/14403/4663#>

"atribut yang memiliki warna merah muda adalah atribut yang relevan dengan atribut aoutput dan sebaliknya atribut yang berwarna merah tua adalah atribut yang kurang relevan dengan atribut aoutput."

Atribut berwarna merah tua yang dimaksud oleh penulis adalah atribut yang memiliki korelasi negatif tinggi dengan kolom output

"dari 13 atribut awal yang digunakan terseleksi menjadi 9 atribut yang memiliki relevansi atau berkontribusi terhadap hasil output yaitu : jenis kelamin, kelas, umur, SKS1, IPS1, SKS2, SKS3, SKS4 dan SKS5, sedangkan atribut input yang kurang relevan terdapat 4 atribut yaitu : IPS2, IPS3, IPS4 dan IPS5"

#Alasan menggunakan Correlation Matrix Heatmap pada penelitian ini

Untuk melihat relevansi antar kolom. Disini target kolom saya adalah kolom "Output", dan relevansi antar kolomnya memiliki nilai positif. Maka saya akan menggunakan semua kolom sebagai indikator

"""

```
sns.heatmap(data_clean.corr(), annot=True)
```

```
#.corr() = correlation matrix
```

```
#annot=True = memberikan value korelasi antar kolom dalam bentuk angka
```

```
"""#Machine Learning Model Building
```

```
axis=1 = Remove column
```

Dataset X = menggunakan semua kolom untuk kriteria dan ciri ciri pada tubuh pasien

Dataset y = sebagai hasil apakah pasien memiliki diabetes atau tidak



ECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHxo/iC  
VpAkmSJBqjhMMUhWzCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqh  
D9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALSzXGCQap  
iFkEXWl+StDBGtir58heh4yVJy2Nwq4+vfRHaxZK0PAa3+vjaF6HdJUUnLY3Crj699Fzp  
ekrQwRrYq+fLXoe+DVJImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkm  
aYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyCONR  
/kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxFbMI6td9QSpJE0yQINUcJxikKmY  
R1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnOME  
gVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg  
1xwkGqYpZBHxo/iCVpAkmSJBqjhMMUhWzCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkia  
YIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+  
kkjTBBALSzXGCQapiFkEduj9IJWmCCRKkmuMEg1TFLII6dH+QStIEEYRINccJBqmKWQR  
16P4glaQJJkiQao4TDFIVswjq0P1BKkkTTJAglRwnGKQqZhHUofuDvJImmCBBqjlOMeh  
VzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkmaYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1  
xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyCONR/kErSBBMkSDXHCQapilkEdej+IJWkCSZ  
IkGqOEwxFbMI6td9QSpJE0yQINUcJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6k  
kTTBBglRznGCQqphFUIfuD1JJmmCCBKnOMEgVTGLoA7dH6SSNMEECVLNcYJBqmIWQR2  
6P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHxo/iCVpAkmSJBqjhMMUhW  
zCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5x  
gkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALSzXGCQapiFkEduj9IJWmCCRK  
kmuMEg1TFLII6dH+QStIEEYRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkk  
TTJAglRwnGKQqZhHUofuDvJImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4  
PUkmaYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSy  
CONR/kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxFbMI6td9QSpJE0yQINUcJxi  
kKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKn  
mOMEgVTGLoA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQ  
TJEg1xwkGqYpZBHxo/iCVpAkmSJBqjhMMUhWzCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4N  
UkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6A  
O3R+kkjTBBALSzXGCQapiFkEduj9IJWmCCRKkmuMEg1TFLII6dH+QStIEEYRINccJBqm  
KWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkkTTJAglRwnGKQqZhHUofuDvJImmCBBqjl  
OMehVzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkmaYIIEqeY4wSBVMYugDt0fpJI0wQQ  
JUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyCONR/kErSBBMkSDXHCQapilkEdej+IJW  
kCSZIkGqOEwxFbMI6td9QSpJE0yQINUcJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP  
3B6kkTTBBglRznGCQqphFUIfuD1JJmmCCBKnOMEgVTGLoA7dH6SSNMEECVLNcYJBqmI  
WQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHxo/iCVpAkmSJBqjhM  
MUhWzCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJ  
Uc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBALSzXGCQapiFkEduj9IJWm  
CCRKkmuMEg1TFLII6dH+QStIEEYRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1  
BKkkTTJAglRwnGKQqZhHUofuDvJImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHocYJCqmEV  
Qh+4PUkmaYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBIN



UxSyCOmR/kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9QSpJE0yQINU  
cJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBg1RznGCQqphFUIfuD1JJmmC  
CBKnmOMEGVTGLoA7dH6SSNMEECVLncYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5D  
qev48PfbIXMLD0x8e90TPj/zN3/T4zL3rZvFSBqnmOMEGVTGLoA7dH6S6Hjcz3SJeyiD  
VHCcYpCpmEdSh+4NUV/fnxfpz0+pjb+9pDw+PT8/Pu7vnxX5qdx/Xp7hw0+e4e5ZPP3  
Gs9AieGGDVHOCYJCqmEVQh+4PUq3lZQP62QL00cPj81kWoZdnxyN8b7cXuo3dIV7fINU  
cJxikKmYR1KH7g1RL2u1nvFBz488tR075DNal7J7xIgf7/q1j9JnrTD/9YJ//HCeew1  
SFbMI6tD9Qap1/Xn60Xp291/1+vqqea6367QKXugg/V1uZqpkEdSh+4NUa7v4xeIEXzw  
nt7ICvNZB+ut++puN+zds/6kMd79V+fUPRF/+GHvcW5CqmEVQh+4PUi1vuJyd97PMnb8  
8oQf/u8sOvN5Bei4vC9qg/n/xR40/9d0vNhj3GKQqZhhUofuDVLdgtJyde0HafjKuZTV  
4xYP0nE6t/9/9qWTzUX+1zrnPIFUxi6AO3R+kuhGD5ey8S9LWE/FDzCa86EF6Xiev/y+  
/Xfz5MX/7/WjuNkhVzCKoQ/cHqW7Hyb8Vfc4PND8/iXM+mhbE6x6k53bCbvbLlfip1H/  
9Rx7uN0hVzCKoQ/cHqW7JyW+cnetts897mZ9i1uGVD9Lz+776f7UYpz3c75c6dxykKmY  
R1KH7g1Q35sQ3zs7xRtb5P93RLeDFD9JL+G43+9V6POq0c9Q6dx2kKmYR1KH7g1Q350q  
7mXuZXvHyB+1lffP8v1iSF1jMnMk6ZhhUofuDVDfoCrvZ54f0Y8xWFECQXsg3b5v9Vs0  
fP8x5fgrhzoNUxSyCOmR/kOomffexDn7revJpLzvPhUo3gRoI0ov55ueSX/mJ4TKLmTN  
ZxyyCOmR/kOpGXW43+/xI7mXVqIIgvZxvSv8MJX+u94e5+yBVMYugDt0fpLpZl9nN/Bh  
TR6iDIL2kr982++fV7Pjuz1bx3H+QqphFUIfuD1LdsPPvZn6MqU8ohSC9qLN+onmxxcy  
ZrGMWQR26P0h10866m32+c/cyrTFJvin8fynUo8XsnDXPQwSpilkEdej+INWNO9tu9v1  
tCT/G1A71EKsXdq63zs64mDmTdcwiqEP3B6lu3jl2s437dC/TKwoiSC/um7r/Wb0e3+1  
ZFzNns05ZBHXo/iDVhfjt3ezz/Z33AqWbQk0E6RV8/bbZT2r2uPLP/OMIjxKkKmYR1KH  
7g1R34Td3Mz/G1JeoiC9ht/+RPPCi5kzWccsgjp0f5DqT vzSbrZxN+5l+oC6CNKr+Kb  
qh6V7vOidvfJ5nCBVMYugDt0fpLobv7Cbfb4LP8bUMUojSK/kFz/RPLqrC5Q+jxSkKmY  
R1KH7g1R35B93s88XOfcyFUzxBom1/NrbZpdfzJzJomYR1KH7g1R35cTdbON6tfE3/Rh  
TWyiPIL2ab2r+1A3radG7SPHzWEGqYhZBHbo/SHVnfrab+XaZTkaFBokVfa7eQydv8nH  
bXOanEh4sSFXMIqhD9wep7s58N3Mv0wA1EqTX9E3Jf79mXWcxcybrmEVQh+4PUt2h0W6  
2ceMLXZd0m6iSIL2uf3vb7KgHLvZzCY8XpCpmEdSh+4NUd+nk3ezRt8s0RKEE6ZV9XfF  
fF/XRWne5DuABg1TFLII6dH+Q6k6dupsdcs/TdyiVIL22bwr+i/eBr7aYOZN1zCKoQ/c  
Hqe7WD3YzP8bU9yiWIL2+H36iefTXLtkDPGSQqphFUIfuD1LdseFu5l6mU1AuQbqAH71  
tdsXFzJmsYxZBHbo/SHXxtt/N/BhTJ6JignQJX79ttrF1HTfIZX864UGDVMUsgjp0f5D  
qzp26m/mGmU5ExQTPGr4u908/fhzd/NI/nvCwQapiFkEduj9Idfe+fh/hnbuZTkK9Bok  
ivv1J5GORX3kxcybrmEVQh+4PUhVwN9MvolqCdBmnf6J5dMvLf6DPAwepilkEdej+IFU  
FdzP9GmolSNdx6iear01xhdrnkYNUxSyCOmR/kKqEu5l+CZUSpCs56W2z6y9mzmQdswj  
q0P1Bqg7f/PrNIXczfYk6CdKlfl2a7d82W2AxcybrmEVQh+4PUjUY7GU77mb6AlUSpGv  
57hPN5yv/7v8rHj1IVcwiqEP3B6nu36kfZB640pVKt4AaCdLVTKr+WuXOwwepilkEdej  
+INWdG75d9sbdTH9BhQTpck6v/KvVo08fpCpmEdSh+4NUd23j6vT49HjaFcvdTJuoJyB  
d0I1vm13vw3ueJCqmEVQh+4PUt2xzlem123r1HcT3M20geoI0hWdtJpd8bcqeQZBqmI  
WQR26P0h1rzbWr/dNy91MP0ZtBOMStqjza/7nLjyFIFUxi6AO3R+kuk9bH2N+uAS5m+m  
HqIwgXdQ3b5tdt7x5EkGqYhZBHbo/SHWPNI5Hn98acdftJ1AXQbqqL8v8yrXNswsSFbM  
I6tD9Qaq7s3Ep+ssFyN1MP0BVBOm6/v622bULm6cRpCpmEdSh+4NUd2Zj2/riN2nczTR



GTQTpww62ml3zV8z2eB5BqmIWQR26P0h1V076GPMDDzMNURFBurLtIr/6YuZM1jGLOA7  
dH6S6HxvXn1P2KXczjVAPQbq2+Y8s18AzCVIVswjq0P1BqnuXsWCdeulxN9MA1RCkizu  
u8SVqmeCSpCpmEdSh+4NU9+Ef3xNwN9PJqIUgXdxRha9RyDyZIFUxi6AO3R+kugMbe9X  
4yuNuphNRCUG6tiUXM2eyjlkEdej+INXN29ipfvQrNO5mOgl1EKRL06rtBX7FbI+nE6Q  
qZhHUofuDVDfuN3+1+dTdbJkrm66BIgjs1S26mDmTdcwiqEP3B6lu2cYq9W/vaLmb6Vu  
UQJAubNXFzJmsYxZBHbo/SHW7Ntaof7/ouJvpGxRakK7r49vKK30Yz1MKUhWzCORQ/UG  
qW3W+/wtNG/e8yd2sEy9/kC5r3cXMmaxjFkEduj9IdZN+/WPMj9zN9He8+EG6qoUXM2e  
yjlkeEdej+INUNOsvHmB+5m+lveOmDdFEfC3mxcuVZBamKWQR16P4g1c3ZWJrOccFxn9M  
2XvggXdPsi5kzWccsgj0f5Dqtpz5Y8wP3M20hZc9Sfd01C3LFSrPK0hVzCKoQ/cHqW7  
Jxq5011+ccTfTz7zoQbqgj4vZWr9itsczC1IVswjq0P1BqpuX8XbZ+VeiU3ezBS970hN  
e8iBdz8faXbFCeWpBqmIWQR26P0h1I7ZWpEu8VbW1D25yN2vBCx6kqzkq3CXLk+cWpCp  
mEdSh+4NUt2BrPbrYpebk3cwpNTvwagfpWo6Lds3S5MkFqYpZBHxo/iDV8jY3o4u+BXD  
qR5ruZhV4rYN0JccFu+r7uTy9IFUxi6AO3R+kWtvz4+YbVhfegE5fzfxQ8/7xSgfpf1  
5fnp8eG+Fh4fHx6fnP1ul9udzzyxbkzy/IFUxi6AO3R+kWtffP0a89KXm9A80d3zj7L7  
xMgfp+bzswjzUhpCd7W1F269v5AfWrUeeYJCqmEVQh+4PUq1p4yf/Axe/1gzeNNt5cDm  
7Y7zIQXous58KPlu5FHmKQapiFkEduj9ItZyXn/y/uxat/Z7Z3styuhHSPpHvMJBeib  
DnwmOrf3hOk8ySFXMIqhD9wepVvKXj2M2XPSdgJ9eHR+e3M7uEK9ukJ7Hvy1mq//SI08  
zSFXMIqhD9wep1nD6ThYXWnu+/lj1ew++eXZveGWD9Dz+ZTNb/yN1nmiQqphFUIfuD1J  
dz58Xz0/TjeyDh8en+fn3R1xn79k/9Q+/udw/+jtmfIIulm8okF6Hj/ezFZ/u2yP5xq  
kKmYR1KH7g1TX8Y+/PrPt398k+Ndfz7NTVw19Te8ikF6Hj8ryFv5L1B4ukGqYhZBHbo  
/SHUdbma6VbyKQXom84q8of8wmGccpCpmEdSh+4NUkiaYIEF6NpMfYm7slxp51kGqYhZ  
BHbo/SCVpGGkSpGd02n+EcoP/qQnPEhVzCKoQ/cHqSRNMEGC9Ly+/1Bzt5Td4gfkPP0  
gVTGLOA7dH6SSNMEECdLz+7P7v+By+F8L7/4/M930/8k8/h1BqmIWQR26P0glaYIJEqS  
a4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHxo/iCVpAkmsJBqjhMMUhWzCORQ/UEqSRN  
MkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9  
SSZpgggSp5jjBIFUxi6AO3R+kkjTBBA1SzXGCQapiFkEduj9IJWmCCRKkmMEg1TFLII  
6dH+QStIEEyRINccJBqmKWQR16P4glaQJJKiQao4TDFIVswjq0P1BKkkTTJAg1RwnGKQ  
qZhHUofuDvJImmCBBqj1OMehVzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkmaYIIEqeY  
4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpGGkSpJrjBINUxSyConR/keRsbBM  
kSDXHCQapilkEdej+IJWkCSZIkGqOEwxSfBmi6tD9QSpJE0yQINUCJxikKmYR1KH7g1S  
SJpggQao5TjBIVcwiqEP3B6kkTTBBg1RznGCQqphFUIfuD1JJmmCCBKnOMEgVTGLOA7  
dH6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYp  
ZBHxo/iCVpAkmsJBqjhMMUhWzCORQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4  
wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBBA1  
SzXGCQapiFkEduj9IJWmCCRKkmMEg1TFLII6dH+QStIEEyRINccJBqmKWQR16P4glaQ  
JJKiQao4TDFIVswjq0P1BKkkTTJAg1RwnGKQqZhHUofuDvJImmCBBqj1OMehVzCKoQ/c  
HqSRNMEGCVHocYJCqmEVQh+4PUkmaYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZ  
BHbo/SCVpGGkSpJrjBINUxSyConR/keRsbBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwx  
SfBmi6tD9QSpJE0yQINUCJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBg1R  
znGCQqphFUIfuD1JJmmCCBKnOMEgVTGLOA7dH6SSNMEECVLNcYJBqmIWQR26P0glaYI

```
JEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHXo/iCVpAkmSJBqjhMMUhWzCOrQ/UE
qSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVC
H7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBbALSzXGCQapiFkEduj9IJWmCCRKkmuMEg1T
FLII6dH+QStIEEyRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkkTTJAglRw
nGKQqZhHUofuDvJImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkmaYII
EqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyConR/kEr
SBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9QSpJE0yQINUcJxikKmYR1KH
7g1SSJpggQao5TjBIVcwiqEP3B6kkTTBBglRznGCQqphFUIfuDlJJmmCCBKnMOMegVTG
LoA7dh6SSNMEECVLNcYJBqmIWQR26P0glaYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwk
GqYpZBHXo/iCVpAkmSJBqjhMMUhWzCOrQ/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEG
qOU4wSFXMIqhD9wepJE0wQYJUc5xgkKqYRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjT
BBALSzXGCQapiFkEduj9IJWmCCRKkmuMEg1TFLII6dH+QStIEEyRINccJBqmKWQR16P4
glaQJJkiQao4TDFIVswjq0P1BKkkTTJAglRwnGKQqZhHUofuDvJImmCBBqjlOMehVzCK
oQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkmaYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkG
qYhZBHbo/SCVpggkSpJrjBINUxSyConR/kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGq
OEwxSFbMI6tD9QSpJE0yQINUcJxikKmYR1KH7g1SSJpggQao5TjBIVcwiqEP3B6kkTTB
BgLRznGCQqphFUIfuDlJJmmCCBKnMOMegVTGLoA7dh6SSNMEECVLNcYJBqmIWQR26P0g
laYIJEqSa4wSDVMUsgjp0f5BK0gQTJEg1xwkGqYpZBHXo/iCVpAkmSJBqjhMMUhWzCOr
Q/UEqSRNMkCDVHCcYpCpmEdSh+4NUkiaYIEGqOU4wSFXMIqhD9wepJE0wQYJUc5xgkKq
YRVCH7g9SSZpgggSp5jjBIFUxi6AO3R+kkjTBbALSzXGCQapiFkEduj9IJWmCCRKkmuM
Eg1TFLII6dH+QStIEEyRINccJBqmKWQR16P4glaQJJkiQao4TDFIVswjq0P1BKkkTTJA
glRwnGKQqZhHUofuDvJImmCBBqjlOMehVzCKoQ/cHqSRNMEGCVHocYJCqmEVQh+4PUkm
aYIIEqeY4wSBVMYugDt0fpJI0wQQJUs1xgkGqYhZBHbo/SCVpggkSpJrjBINUxSyConR
/kErSBBMkSDXHCQapilkEdej+IJWkCSZIkGqOEwxSFbMI6tD9kqT1MK1VzCKoQ/dLktb
DpFYxi6AO3S9JWg+TWsUsgjp0vyRpPUxqFbMI6tD9kqT1MK1VzCKQJP0Eq4TLhPSr7Ch
JkqRVuJlJkiStws1MkiRpFW5mkiRJq3AzkyRJWoWbmSRJ0irczCRJklbhZiZJkrQKNzN
JkqRVuJlJkiStws1MkiRpFW5mkiRJq3AzkyRJWoWbmSRJ0irczCRJklbhZiZJkrQKNzN
JkqRVuJlJkiStws1MkiRpFW5mkiRJq3AzkyRJWoWbmSRJ0irczCRJklbhZiZJkrQKNzN
JkqRVuJlJkiStws1MkiRpFW5mkiRJq3AzkyRJWoWbmSRJ0irczCRJklbhZiZJkrQKNzN
JkqRVuJlJkiStws1MkiRpDf/73/8BfrWdG/8VnMAAAAASUVORK5CYII=)
```

```
"""
```

```
X = data_clean.drop('Outcome', axis=1)
```

```
y = data_clean['Outcome']
```

```
"""#Metode 90 10
```

#Berdasarkan Jurnal Analysis and Prediction Of Pima Indian Diabetes Dataset Using SDKNN Classifier Technique

yang dikutip dari :  
<https://iopscience.iop.org/article/10.1088/1757-899X/1070/1/012059/meta>

"Our concept is implemented on Pima Indian Diabetes Dataset (PIDD). The analysis on Pima Indian Diabetes Dataset (PIDD) is carried out by splitting dataset in to 90% training data and 10% testing data. We have found that, in our proposed technique, average classification accuracy gives result 83.2%, a great improvement as compared to other conventional technique."

Untuk tahap train test split, saya akan membagi data train menjadi 90% dan data test menjadi 10% karena dapat menghasilkan akurasi yang lebih baik

#Berdasarkan A Novel Diabetes Healthcare Disease Prediction Framework Using Machine Learning Techniques

yang dikutip dari :  
<https://downloads.hindawi.com/journals/jhe/2022/1684017.pdf>

Foremost using

a function like the model selection train test split, the data set is divided into the training and testing data sets. Due to the limited data set source, about 90%, of the data set, is used for training purposes and the remaining 10% is used for testing by selecting the data randomly.

Alasan saya membagi data train dan test menjadi 90% dan 10% adalah karena terbatasnya jumlah data untuk di training

![xytraintest.png] (data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAzMAAA  
HMCAIAAABukmEAAAAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADs  
MAAA7DAcadvqGQAABmPSURBVHhe7d09dtrM2wdgeNcCT5HzrACvAKdJlTYdlHaTLqW7NK  
Y0XdpUaWJWYK8gJ8UDe+GVRiMQH/bf39xY13VOTsQgJDCD9JuRNOoul8sOAAAB/F/+Hw  
CAQ5PMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAA  
CikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAK  
KQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAop  
DMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikM  
wAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzA  
AAopDMAACikMwAAKKQzAAAopDMAACikMwAAKLoLpflPEk7dLvdPAVAMHbK6DMDAItCMg  
MAiEiyAwCIQjIDAIjCFQcTs3UFgAoAPIETyUvx12SLPjMAgCgkMwCAKCQzAIAoJDMAGC  
gkMwCAKCQzAIAoJDMAGCgkMwCAKCQzAA5sNu6ujWe5FFpJMgMeaDE56Z5MFvkRvIwylZ  
12rpe1687p/05ns7H8xns1mcGBFXFnW2vj+zxOsVXfjodXM6vhrmg0xlezS8H09P7Ks  
JicjHNk/DuSGZwYgc3y+X1qNMZ5U6DYq90e96PmE96xVu9OevLR69g9v38tnP787do1h  
qL3z9vO4PPHzdrVe/j50FZEwa52VJ2j5WpPU1Piol+UVGmp6s2TP1cYd2Ttiqsf0s599  
cHTss583Tj17ZeVMSfIO0gmUEA/Q+DPFXmX+XxW7p/HvrDtbMfnWui4wqmrXI/O9tp/  
PvP9txv/fPv5307d+y2VK2W1LR2U3RaCkm/smFZVOmbCgUWar/91tq1RTF04sqUM3G/Z  
+f56ml0znpj2flXGXuP+//+pTnLJjdnq5/by1F5ZeVpfdGJDMIJ+2WkqgVX/Uc7HQPrN  
v067KkeCKVpN1RKqn3MM0Zc9kdc27MWxXmOdOTjWdXb6NxEvd6KQ+3mPz68HU4/Fp2GW  
6n0sZ7Kd3zp+AdW/8wGmbfzZuXX6tjocOrKqwVVeNimnviemffyrz2+2MKdvVx07It1J  
z+819Rg1IHxhnYcmWmq0rhrUlmEM1sfDrtdeaf+nUr/6Lzo2jEb3UPrNv0s3H//N90KD  
T3JPzofEn7lenpl/KV5Uk7qSNhmflS7MTKfoS6c6Fc3r45myuqZi3SYZoz2fc2in1hFR  
L3/PJDnvERFr//fij3p01A1vRXM5rtfMA7/xQcm9RbvBuAFv/96XQGH/r54T3SnDtSV1  
yt2SP9P9UnFSSveuwe7iKZQRC5rd7tFrms2D1cDfPhm8Hlj3r/sLdNX+6a8j5s+G1UFp  
Xng1UZJu1ZVj0NjfpEYj1zyb45d/shho3jSnd1LRRFKR/1zs7Wp3M/0Oz730/VW9uJZr  
sfsCzUvfEu1GeUbX55+88+26uss7d/5/1RrQxjG6W7B0zvsNkogEOQzCCIzMO9caHatp  
02fblvy7u2MsPs3wU191LpqGPZLbdXOef+fogNu28j5cgqK+30Xy0aRyP3dm7Nfk1zzM  
pBq7GDvPMD6t54B3rlaZW3G32eZR/pbW4sVHIa33tBZpnWp6erly8mk2Kyive5/7eIea  
NPD2krVK9aLWs2dpScw8hbNVojf/G1Xmph1X1jGzkjq/rM0vHHUv1w33ypPFk9mY/7bU  
5X86XJoiwveN+czbK1uvSut1FJzZbe9ENcX27OX66puYjd3j/e+D15S+klkufap0Flz  
a/19Vzg8vLol5UT5c1JBWt6nC2rjarwqgo+XDvdLXWxjt5q+qVV1fLpbSYStA6+ddfy6  
UcVr13W09S1srdRLO82m3Ue4zrUXou/7epXOJ6vvXerFFWvKqwd840sSotVntZF6ayfW  
8jz1Mqnt7zju625wOkNWyuYMvePwVvp/jrN+VSHi//BWu51BZTCVon//pruzTDSQmjtq  
pDpRSPs3UUSWSJM/cKE1G16uS4oXNhayma4PRaP+cxWIbM5cFzWXuexvNd7HxKe637z  
M2F7Wx6qxe/M574A31v3wt1/J4+S9Yy6W0WLF416sD7dtdvNUogIcv91k0j9bnZOzmJ  
x8/+fmnhPVjs+7/4BHyZbkpflLssUVAHDCZuPT88bp/Yvfp/990NnOR+Pdf0CAJn1mra



N99u7Mxo0rLQeX83d3leK7/4BHyZbkpFhLskUyax1bAeD5bEleir8kwxzNBACIQjIDAI  
hCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIQjIDAIhCMgMAiEiYAwCIwt2Z2mXrNiAABG  
TX3G6zFpELAM4CjbXbSaZAQBEIzkBAEQhmbWIExcAjoLNdZu5AqB1tk5fUAGAJ7A1eS  
n+kzmRZwYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZA  
YAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBg  
AQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGAB  
CFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEIVkBgAQhWQGABCFZAYAEI  
VkBGAQhWQGABCFZAY81Wx80u12T8az/BiA55LM4E0tJmWYuctGxpmNc+mGk8kip39ws1  
/T2+K/2+1FnPcEcoQkM3hTvbOb5XI+vx4NBrmkMricL5fLq2F+WBpeFSXFnPnhYHR5Xc  
xzc9bLBQc3/DQqP8Rg9C3OewI4cpIZvL1eb3h1cz0/bISzf//ZH25yaRHcbq7Oho8IQI  
vJ+NV7sopPUYTHm408CcBzSGZwKL2zmlWHWGd6uu8w5WJycjpNseyxvVKz7+d/8iQAx0  
MygwMaXq07zm7Pv2xls9m4f37bGV0//gDmYnIxxZMAHBPJDA6qd/bjmw2G1fdZY8/Vr  
iYfCkSHQBHSDKDA2selFxls+ow5h06yxaz8UnZ0wbAUZLM4PCGV5vZLB3GfHx3WRHnTv  
qnaSSL0u15f3ekjcvSug5CVg3PsUjT5Yhkjb66NEdjaI+T4tnZ5mHW2qLMgXsG8qhWUh  
dXM1XLGt+xJIjFWH0c0JKWyV98LZdycOuLAZLRdS5/vPrUtWogjtr8+jKNcVEpFr+5wj  
xzf390uJF+TWbCysH81gP+9F4r1leFq8H/Vh5xicjkPx11nLpe7Gqt1sv/zXkNdVyKS  
2mzwxiaF4MUOwOvr7sSBSLyZeLn38aBzl/jS8+FLucOq8NPn/s1ZccFGv/kQ+i9ob1aX  
C359/XvQez8ZeLP53dQ6blWv7crst/F/N9KAdhK8wv875u+ks3BNEZq48Dkswgit7Zt7  
pXa/c6zWfqnd0UVt1pnWnnUzqFrYhe1Zhk5YPff3sG2uj982+eWithYyuWttMfltayKr  
79+fftZU09CFvv7FMu//OfI5o8h7H6eN8kM4iiHOpi1d/04t1s0+jTnj107+Pndf/Zjg  
cHqv6HKv4NPm92/NX1t3/n6X94EmP18c5JZhBDGupid1fdWu9cjbBj906an3/p3SCf3  
mN6B3qqAVvx1h9vHuSGUSQzvAaXV8Ne40Bzg6QzSr5As4vX359+vFj9xx+OBRj9dEckh  
kcXBq8bLAAJOog2SwNcdE//fnh23x5clMmxfwEHFhZN9M1KvC+SWZwYlm7rDmm7KGyWR  
ERY/HQRtfl1Yn78FzjajC7huYoYbPm03cOH2asPlpEMoND2uouWz1ENqsPFO29OgCe6m  
p9TXA5JMt8uWxW92H9dPnMnZdClpf9prFXqgWtRxxkrWzQ5kvVPz3N0K9JeP0/ftuvfT2  
p3nE87n/MwLteXg9vi2dP+VvpKZ1d2+40cWMnlq5UUj4th67A4Pe0bl5YXIZnB4eTusn  
r0sA1vn83mf6udjFEteGHNO5DtUda89SB6j2WspT4byQwOo2xxpl1tjXt113HAjm2037B  
/g9ufvqq+gaOs/5sXrIJh6CTauhFtMGkd/6ih3B6NjsDb8uicGVWa/it/B00d0NVYf74  
1kBm+tOo01X41Gcd+mfNGMPmU4G88WD9jw52HJ0kvKE2D6Fx+qfoLF7Ht9/vS+xn2971  
y9ML3ypo6HqfBL52MdJBf1DjUpd7br9zX7/XO9ksbbXdxRTjuUAYmnie2614LZCxBN1  
YfRy/1xdIi+Yuv5VLeynZTvLJzL8nGiTnb1qfY3K28fWwee1Qdcdl/MGlnWfPVzTUHo8  
t8rKZ4bSqrF5VK0jybdu7FmZSr2PdpHvIxCCx/j7Vceq/ds8QKZ158L1U9y6hsqpkD1  
rY/K47v2Z3rOiu9T9u5VuqV67kUlpMJWid/Ouv5VKAx8hbkFouvd9ugEk1D48zdyWjwg  
PDUT77bFC2M1b325fMCMXRTADexM4RzfIQ94fvv8u+SwCY/URnWQGwBupT2acXpRXma  
Rgtv/Ur5dmrD6Oh2QGwFupu83SZSNvFsyM1ccxkcvAeDvDPMTE9LQcze/po2U8hrH6OC  
aSQBvaD08y5P7sIzVx7uWrwSgNfIXX8ulAI+RtyC1XPpA+aLIx1/JuLoIMqsvklxdZ1  
nYt9Tt11Wv3ChtXG+5ubT1ZZjX6/mbxc3lbJQ/TH51LZfSYipB6+RffY2XajxG3oLUcu  
lDpSzzlCEm3uFYfflvtVxKi3WLF7k60A7dbjdPJSOA8ATP25LMxt2LD/N046TWS01mi/  
PMAHhTi8nF9I1Gy4DjI5k8JbecBgzoEKOZraOnnPg+Z6+JSkHff37bX11aLGKbTJb9J  
kB8KrK2yJ1uyfjNO7/5Mt5561uyATHSJ9Z62ifAc/3iC3JbNw9bYwcNrrWX9Zkm8wWfW



YAvKbh13rAiq6hVgG99Jn1jraZ8Dz2ZK8FH9JtugzAwCIQjIDAIhCMgMAiEiyAwCIQj  
IDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMg  
MAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAw  
CIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAI  
hCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiE  
IyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQj  
IDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMg  
MAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAw  
CIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAI  
hCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiE  
IyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQj  
IDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMg  
MAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAw  
CIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAI  
hCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiEiyAwCIQjIDAIhCMgMAiE  
IyAwCIQjIDAIhCMgMAiEiyAwCIortcLvMk7dDtdvMUACHZNbeZPjMAiEUTus0kMwCAKC  
QzAIAoJDMAiMV5Zm3mCoDW2Tp9QQUAnsCW5KX4S7JFnxAQBSSGQBAFJIZAEAUkhkAQB  
SSGQBAFJIZAEAUkhkAQBSSGQBAFJIZAEAUkhkAQBSSGQBAFJIZAEAUkhkAQBSSGQBAFJ  
IZAEAUkhkAQBSSGQBAFJIZAEAUkhkAQBSSGRzSYnLSvdt4lmcrzca5dMPJZJGfB+AdkM  
zqkHpnN8vlfH49GgxySWVwOV8u11fD/LA0vCpKijnzW8Ho8rqY5+aslwsAeAckMzi4Xm  
94dXmzv2yEs3//2R+4cmkR3G6uzoaPCGWlyfgNetfeZi0A75hkBkH0zm5WHWKd6em+w5  
SLycnpNMWyx/aUzb6f/8mTr+ht1gJ7aHvwbkhmEMfwat1xdnv+ZWsXMBv3z287o+vHH8  
BcTC6mefIVvc1aYB9tD94PyQwi6Z39uCOBzCZVd9nGuWcPspH8KRLda3ubtcA+2h68J5  
IZxNI8qLnKZtVhzCd0ly1m45Oyp+11vc1aYD9tD94XyQzCGV5tZrN0GPPx3WVFnDvnp0  
7rncntex//SBuL2WR8sh684+RkPJntnkqzKGBKM6TnF5NJGtPjoWuB16DtwTu0pGXyF1  
/LpUSzvhggGV3n8serT12rBuLYksbrGKTxN+qHaeatud07GeW55tfVEjfe071r4R1KX/  
daLn2wzfpdatanjep/Z+UvK1196H/TVi0samxzWJpBHnFmy7yYKc+Qnp9fXqYVP3QtT5  
YXV8ultJhK0Dr511/LpcSzhjgebuAezJTuf/bKV7tFFc7xLSEzfnKmSSzVktf91oufY  
T/UcGrpx9Sn+6te0fQ9kiLXcultJijmRBU7+xbvRPZvU7zJaRrCkbfk9dG37K0Wx6kV  
c6/3tbvIe/8/SgMvy63q3CU2wME7OrrHWDyx/PG015Nu6f/vn842Y19185dGC11tvz/u  
oeG+nU/sH11zxXb3j/W4PXJZ1BUOXeYnViytks9mv8jqz6Wk+LWytyGuVzTA2vRg3Tj  
/rffz8IU/CE9UB//b8e/NGZKWyeu42Gx5H24PjJLBS01CsNH1fHXI56Wz2eK/NDDTne  
fwlPIlB/0Ple5zetrVnhTxLL2N3tnZo8fvgE11x3Camp7ajGYpmH16Xg3T9uBISWYQUL  
oac3R9Nez11gOcvWw2S90ED9N4DymeVfksF8Bz1H1Tq/6r5CWCmbYHx0oyg2jS4GWD1S  
AZr5bNku2+iv3KG6+vL13Lu7DVWTrwdL2Pn6tYtD6imU77em4w0/bgaElmEEvuLmuOKf  
uq2Wyzr6JpntmHwN7ZVRHPyvEHcsH+e3vC4+wc0Vz8/nlbn06fHjybtgdHRzKDQLa6y1  
ZeIZutToO+Y4GLyUXnnyodzsarXVRvWOSz1cgDtz9/i2Y82+YRzRTMPn981rn/TdoeHB  
vJDMI3WV7Bwp4+Wy2imbFAvvdka4/4vZ5KR//u/6cNJmv0NveJV7OeA11N1m5RHNEw  
tm2h4cK8kMQlJmXuU1Y+ms/1y0ZSOb9R/fmK93MuWNZtKLN8YFuD0/rW+r1032T8u7QT  
UPJ23301SnTO/Zge6sBR6gjlHT07J18szRMjJtD45VvkKF1shffc2Xcjjz9bDk9w8wXs

```

yX56oMykE18nP3aYy1njTXsf1cViw6z1DK6y3X1x7nIdI33+t9a+Edyt9zLZc+3boC3X
s15R6rV+Y6V/6g8tT+6l1p1NCyhm9V2OqljcK71/Jc1WJXciktphK0Tv7113IpB7K/Yb
6za7pnD/OQvcM6+63SVUP57Hrxe+4peD0q3lAKkPVse288eP9aeGeqr3ollz5HbgE8Np
jt/jyav4k7fjqR2h55ebVcSoupBK2Tf/21XArwGHkLUsulz5Kyz+ODWeGo2x7VIldyKS
3Wlf716kA7dLvdPJWoAMATvMKWZDbuXnyYNweMaQXbZLa4AgCAwyvH133J0TLgWEImAB
zcSw9jBkfL0czW0XMOPN8Lb0kWk5P+32/1jSxbxTaZLfrMAHh7s/FJkU100jj8i8mX88
6L3ZAJjps+s9bRPgOe771bkmpo5ZXRDsv7y0q2yWzRZwbAmxt+rW9QmYagaGssg136zF
pH+wx4PluSl+IvyRZ9ZgAAUUhMAABRSGYAAFFIZgAAUUhMAABRSGYAAFFIZgAAUUhMAA
BRSGYAAFFIZgAAUUhMAABRSGYAAFFIZgAAUUhMAABRSGYAAFFIZgAAUUhMAABRSGYAAFF
FIZgAAUUhMAABRSGYAAFF018tlnqQdutlungIgdTl9JkBAEQhmQEARCGZAQBEIZkBAE
ThCgAAAnmJlOZH9CLwgyQwAIApHMwEAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzA
AAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAA
CikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAK
KQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAopDMAACikMwAAKKQzAAAop
DMAACikMwAAKKQzAAAopDMAACikMwAAGLodP4fsJaSB2q93ToAAAAASUVORK5CYII=)
"""

#test size 10% and train size 90%
#Random state = Mengontrol pengacakan yang diterapkan ke data sebelum
menerapkan pemisahan agar hasil yang didapatkan tetap sama
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.1,
random_state=42) #split test 10% dan train sisa dari test yaitu 90%

"""#Decision Tree"""

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(random_state=42) #Mengontrol keacakan
estimator agar hasil yang didapatkan selalu tetap
dtree.fit(X_train, y_train)

from sklearn.metrics import confusion_matrix

```

```
""#Berdasarkan Jurnal Machine learning models for non-invasive glucose
measurement: towards diabetes management in smart healthcare
yang          dikutip          dari          :
https://link.springer.com/article/10.1007/s12553-022-00690-7
```

```
**Precision:** Precision is measurement of how many positive
predictions are made for actual positive values.
```

```
**Recall/true positive rate/sensitivity:** Recall is measurement of
correct positive predictions from all positive predictions made
```

```
**F-1 Score:** F-1 Measure is a combination of precision and recall
```

```
#Penjelasan
```

```
**Precision:** Terdapat 76% pasien yang sebenarnya terkena diabetes
dibandingkan dengan keseluruhan pasien yang diprediksi terkena
diabetes.
```

```
**Recall : ** Terdapat 87% pasien yang diprediksi terkena diabetes
dibandingkan dengan keseluruhan pasien yang sebenarnya terkena
diabetes.
```

```
**F-1 Score : ** jika F1-Score punya skor yang baik mengindikasikan
bahwa model klasifikasi punya precision dan recall yang baik
```

```
#Berdasarkan jurnal The advantages of the Matthews correlation
coefficient (MCC) over F1 score and accuracy in binary
classification evaluation
```

```
"Checking only F1, one would read a good value (0.66 in
the [0, 1] interval)."
```

nilai F1-Score bisa diilang bagus jika value F1 berada diantara 0,66 hingga 1

#Alasan menggunakan Precision Recall

yang dikutip dari : Jurnal The relationship between Recall and Precision

[https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/\(SICI\)1097-4571\(199401\)45:1%3C12::AID-ASI2%3E3.0.CO;2-L](https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1097-4571(199401)45:1%3C12::AID-ASI2%3E3.0.CO;2-L)

"Recall and Precision and, in particular, Recall-Precision plots, have been used for many years to characterize document retrieval performance."

Alasan saya menggunakan Recall dan precision adalah untuk melihat kinerja model

#Berdasarkan Jurnal Diagnose Diabetic Mellitus Illness Based on IoT Smart Architecture

yang dikutip dari : <https://www.hindawi.com/journals/wcmc/2022/7268571/>

"for the classification considering the evaluative measures like accuracy, precision, recall, and F1-score on PIDD and claimed to have achieved comparatively enhanced results on binary classifications"

Alasan kedua saya menggunakan precision recall adalah karena precision recall lebih cocok digunakan untuk binary classifications

""

```
y_pred = dtree.predict(X_test)
```

```
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score
```

```

print('F-1 Score : ',(f1_score(y_test, y_pred)))

print('Precision Score : ',(precision_score(y_test, y_pred)))

print('Recall Score : ',(recall_score(y_test, y_pred)))

"""#Berdasarkan Jurnal Diabetes Mellitus Prediction using
Classification Techniques
Yang dikutip dari :
https://www.researchgate.net/profile/Anny-Leema/publication/34407266
7_Diabetes_Mellitus_Prediction_using_Classification_Techniques/links
/5f50f16da6fdcc9879c511e9/Diabetes-Mellitus-Prediction-using-Classif
ication-Techniques.pdf

"Confusion matrix is used to visualize the performance of the
algorithms which cross tabulates the observed and predicted
classes with associated statistics, evaluation metrics like
sensitivity, specificity, precision and accuracy are used to
evaluate the performance of the method. Factors like True
Positive (TP), True Negative (TN), False Positive (FP) and
False Negative (FN) are used"

Alasan saya menggunakan confusion matrix sebagai visualisasi data
adalah untuk memvisualisasikan performa dari sebuah algoritma
klasifikasi

![format confusion
matrix.png] (data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAOoAAACNCA
IAAAA2B3H7AAAAAXNSR0IArs4c6QAAARnQU1BAACxjwv8YQUAAAACJcEhZcwAADsMAAA
7DAcdvqQQAABCoSURBVHhe7Z1teeTGEoUvhWAIBXmwhGAwhWWwDMIgCIxgCSyBMAiH3H
d12sclpZlJ4im1JG+9P+ap/1Cp+/RRS7PW2v/7u21OS9u3OTFt3+bEtH2bE/Nu3/81zR
kYf124sO+ImiJa0nLavvNoSctp+86jJS2n7TuPlrSctu88TiTp9+/fX15efvn11E+Ko
ez77dv3zj1HUa/EzJ/8FLsKl+/fh2droF9f/vtN7qN81FJI3wv7DV07Mt1/9dff6nIMJ
6fnxX/+eefT09Pis/ILpL+8ccfUUNAWxS+b1+gw14e+PekEe5v3yRrkv719dXOPh27SK
q7WdRQ/P777yO6Qdu3gLX02Bfpf/31Vxbmy5cvdCDmE2R9P37EK4GYJzkq2b+5M47auX

```



D2EU3klN2FdmIpg4xs1aNHZV+aLCCHjNplQ5H+6fA5xBHCCEzLI4SURUS0o5UijqTgfk  
1FegJLpccPlmGXLXwXSdf25aF2RI5y2A5ZEEQPulYm2hc9EY1uxOwdzoZf0VOHk4r+nE  
5Nc0iSnsC+oCVJ1zo19iu4iPSskCoBx9P0j7fOLeC8I5qI70WR0bZY2W6WMvZftC+VUU  
Mrb08DDqa/s80hzgXeC6lhLxjGLftaaEHNvftqGRJsFeo2E847oolIK2uIydZff7Egj2  
F6NrhqX47CviSjmitzIrp8ApxxRAuf077rDLuwi6TSKioQVcKXMi63ozu7L9CTGngSS1  
8elFmte5EG8F7YfWSCYTxuX2LUV71gqUY0EYXxooms7Ruhns1Y4iRVk30FPXk80DOD+i  
clJwubRnhu+7KRKEZlirKvvsbFpwUknqyyYBgmjmoi0umVfqwRJ1Whfk10PPgof2/2Ap5  
1tDh6hOJZ9+YrAMHicSnsndzrq07cEuqEmvuQoSc+aSXR6Kg/1+PjWWm7NLpJKw2iyiP  
2H8/RPB2gLNEk0PyfQTWLySb1iLQRJ2DhgVJbw9lHtHAg+2IyxmB8WcuaIhoRQdGRSu  
3B01Tfi9GUWK0s0mSJDWcf0Sx+aBRYb42YG1kAI6IVAZKiTxSfo9DWNXjU//IAHEgN9X  
SQ12fCeUe0cCD7fj5a0nLavvNoSctP+86jJS2n7TuPlrSctu88WtJy2r7zaEnLafvOoy  
Utp+07j5a0nLbvPFRScn4i+35dXpja60ducBZJ7wsVf2i809vaN/6AN/6kEaSC2UIOvR  
U1Cp/Rvk1D8fT0xExHjw+xFiquN66xXp9AEYyooVi+wIe0ssfKLL+mThNMArVvLy8HE  
RLUSVpQq/ayLJWm0q1Pg7effB62I4kab19hd7qwMF6h8agy0bSIDpn/Bnsi4Bktozsmu  
hMTckLNCQh20Zr9DhJ0q3s+/z8LD+LN/d+mDdIglh6swNj9OKY4BDtMRFtMByi/QZIrj  
V7Xf7jgMHE1OulM1p1+xOc7scJlvnGIh10yaUXrB6BbCMqBQHJHGwUhq7xS2F8xm6oar  
UJdK1HofQOGh0Ex1KjbOqsw4VqNBgXby1oFWQe0cKG9uVTr4cqFswWFDNV66KebSKX8j  
0Q0CSbAodgX+pRlphW1ZOHbsqmolyuoq8lFYEMHM4nMa00kZCiLhvneQSfvRZUIr01Ai  
biMWMazYVYqvq5AkFkKVRpo/5JKIjJkV2nU2cvR3z9n5yciODog1YRwxkXgx5RBbas3G  
AFmY+nhGpxn0OXuFm6SUUFRR+JBQjnYaubtBMc4lZgSIg7CotlfQpyar2BFeIoxqbiI8  
SzF6J5yYi+3lCPIrMgjt6SNXXxo4CnTE9rlyQittqQhFXRecBrfWtBC4njhItBj6iCuO  
NKQXkOXSwN1WuQFa0J3G2tF1CpZQPXEftlIINbIeWxQVWfiEb/MOQZUSmaFYPEHAR8Io  
VuI2rypQh6rJKYLAGdidXZJKHcX6yFjRsB9d5N6LYmDuZxSDiihYtBj6iCaF/fcbAOul  
gaaqIoEdYDgTRzYkS34lSSnITK5mGvVY6tgpzSnW4ehg5UXMtGaTUvjz+iJj9ogWYXJ4  
t6MrFqQEeNwjLs2KoMfI7y25OYNgJ2AS9N6rYFczxwMegRVRDtC8wQ36Aas7U0nFF3QO  
07Hv3pySH0IZXXg3ptNipG3dcqp1UB6c5ZyOBdQQfGGy6k4sdIZ69C84oOM2qKg9fsvE  
EKOiCjxU9CpeRrYYG1YQdBQ68F003WglYRwxkXgx5RBcm+gAWRLEqDQamJW4W14Pck10  
hSRt3XKqdVEZwR3aPooMq4wcf1+zDrs5egeV0doZ67ovgYiBpdq/GQKE6MISVfCwtcD1  
TKwaPq7oJWEccJF4Me0cPIqXFiQuJaGhWBSVLJBW2NqKTIMoBalarDJAo1tFJkhdJca  
E6oyY11P/I+AYdqLy6GIyZzQM4ta38COqcUSlMn8y3nKFWbbcsAQ7zjohc0gcQh6Lqk1  
DooCZU5VPipP2bDO5m7ixofSQf0UK9fWUpwRxG7Rt6YB2FxYLoS0+mKrEE81d9RFqwGM  
SShkOIvTxaORXjMKKI6J5EFywPOelMa3L8hyHbiIpgIsue3olTMyiMt2hlRtF26KM5Ap  
ZFCirXQukRi3o6kEpNkFYTnddnP+bqglZB5hEt1Nv3cVDNjhTSMcl3fi4j6afhBPblul  
871UtZX3VPRNu3nBPY17sP9z7uero38Ylx0358Ctq+5ZzAvnp48FMam/Hp911xHEk/DU  
nSI9r309CS1tP2nUdLWk7bdx4taT1t33m0pOW0fefRkpbT9p1HS1pO23ceLWk5bd95tK  
T1tH3n0ZKW0/adR0taTtt3HilpOW3febSk5br959GS1tP2nUdLWk7bdx4taT1t33m0pO  
W0fefRkpbT9p1HS1pO23ceLWk5bd95tKT1tH3n0ZKW0/adR0tazj37Ns3xGX5duLDviJ  
oiWtJy2r7zaEnLafvOoyUtp+07j5a0nM9g3+/fv78sf/BilI9K27ecA9mXM97i/q/yxb  
76heCjffT2GuG31W+xTox+JyQN/r2wy6z8i7xHefn1kvrV8qN8Azocfxn2GiH2RUOUVD  
EqrD8XoPiMJE13tq/2iWhfkf6Swppq27x3SxZ8Ufn19tbNPR5L0oPYV2on9Zxrib/1N9q

VJ3dhaOGTULkul3xOcDp9DHOGOMIykMMLqb3mg/5e3vxXCJ8j6fvyIVwKxRa768x//Fc  
4+ooXD2ZeH2hG9/SkR7nfIrYddbvxRvkiJrPpDRqyKs+FXhNbhpkI/plPTHaRdE1SGN  
BEAiILVzitFJGRGvs1FekJKKnHDwT3WswkSXoI+yZG22JluxmVabL/on2pxOWKwbusPQ  
1oTX9nm0Ocy44wjGRfkPLpjkSN/QouYuWosNbiHx/wtoDzjmjhvZAA5iARLS4mW3+wxI  
Lc4HTbumpfjkJckrgVlDkR12ACnHFEu8Iwbtk3KgbUXLWv1E6wGavbTDjviBaOZV+I8u  
FLGZcL/c7uC/Skhp6k0mOZMqt1L3YfgEgKi/9q33WGXUISvhdSwxzW9o1Qz2aMNYmT3M  
m+gp48HuiZQf0x/WhbSMWt2UXSNVcVTnoKam7Zl1gLYSaLKRjGiBYObV/LB0nuaf8qXQ  
8+Ch/b/aDvK4rn4BHUC8P49/b1HyBDNyupr3HxaQHvtn3Hjy2iySL2H85DO3ryFKFvDO  
yyFP2cQDetRFwVehKThCUBHnyvnmU7OPuI9kMKr+cucdJ3WbohF77kKG0Q+F5iSnA6UM  
9a3NpxtizJuqd9OWNkvTUiImoCWuNgAlRjGfgcxyxHoa9r0Dd+m+ZAaqing7w+E847op  
2IQoEVljUffVQJKInIVGoPJiZaEWJkJlar/gVtOWI2nH1EC3va99PTkpbT9p1HS1pO23  
ceLWk5bd95tKTltH3n0ZKW0/adR0taTtt3Hi1pOW3febSk5bR959GSlvMT2ffr8g7aXj  
8fgrNIel+o+HP43dnWvvGnkfGHtyAVzBZy6EWzUfiM9k0aiqenJ2Y6enyItVBx7XTSLd  
brAzCSES0U2xfwkF7vQJH1awY0wShU8/LychCVRZwKCb29JMTabSrV+jh6X2cUDkaStN  
6+Qi/K4GC98GHQZSNpEJ0z/gz2RUAYW0Z2TXSmpuSdJJKQbaM1epwk6Vb2fX5+lp/Sy5  
A/zBukQSy9FYVkestJcIj2mIq2GA7RfgMk15q9Lv8Xw2Bi6vWGFK26/Qm/YJWKZNA119  
5ZewSyjagUBCRz1FEausbv2fEzu6Gq1SbQpR6FYq8hoIPgWGqUTZ11uFCNBuPirQWtgs  
wjWtjQvnzqpVLFgtmCYqzqXdTTTfhSvvcCmmRT4BDsSz3KEtOgevLQtdlU1MtV9LWkIp  
CBw/kkppUmELLUZEM8j+Cz14JKZLZwWwEQ8ZkyjuRBLVT9XIIGsRSt91D8JBTE5sut06u  
zliO+qk5MTEdxZ0CriOOFi0COqwJaVG6wg8/GUUC3uc+gSN0s3qei j6COxAOE8bHWTdo  
JD3AoMCXFHYbGsT0FOrTewQhzF2FR8hHj2QjQvGdHXG+prZBbE0Vuypi5+FPCU6Wmtkl  
DEVhuSsCo6D3itbyloIXGccDHoEVUQdlwpKM+hi6Whcg2yoyWBU631Aiq1b0AaYqsmZ  
ArpDw2qOoT0egfhjwjKkXzYoSYg4BPpNBtRE2+FEGPVRKTJaAzsTqbJTT7i7WwcSOg3r  
sJ3dbEwTwOCUe0cDHoEVUQ7es7DtZBF0tDTRQlwnogkGZOjOhWnEqSk1DZPOy1yrFvKf  
06083D0IGKa9korebl8UfU5Act0OziZFFPJLYN6KhRWIYdW5WBz1F+exLTRSu4KVJ3b  
YgjhMuBj2iCqJ9gRniG1RjtpaGM+oOaHzXoz89OYQ+ppJ6UK/NRsw0+1rltCog3TkLgb  
wr6MB4w4VU/Bjp7FVoXtFhRk1x8JqdN0hBB2S0+EmolHwtLLA07CB06LUAut1a0CriOO  
Fi0COqINkXsCCSRWkwKDVxq7AWHJ70EknKqPta5bQqgjOieXQdVBk3+Lh+H2Z99hI0r6  
sj1HNXFB8DUaNRNR4SxYkxpORrYYHrgUo5eFTdXdAq4jjhYtAjehg5NU5MSFxLoyIwSS  
q5oK0R1RRZBlCrmnSIRKGGVoqsEDg5L1Rn1KSG+h8Z36AD1VcXgzGzeQCnptUfgZwjKo  
Xpk/mWM9Sg7ZYlwGHeEZFL+gDiUFR9Egod1ISqfEqctH+Twd3MnQWtguQjWqi3rywlmM  
OofUMPrKOWWBB96clUJZZg/qQPSAsWgljScAix10crp2IcRhQR3ZPoguUhJ51pTY7/MG  
QbURFMZJnQO3FqBoXxFq3MKNoOfTRHwLJIQeVaKD1iUU8HUqkJ0mqi8/rs1Fxd0CrIPK  
KFevs+DqrZkUI6JvmOz3Ek/TScwL5c92uncinrq+6JaPuWcwL7cvfh3sddT/cmPjFu2o  
9PQdu3nBPYVw8PfkpjMz7dviuOI+mnIUL6Rpt+GlrSctq+82hJy2n7zqM1LaftO4+WtJ  
y27zxa0nLavvNoSctP+86jJS2n7TuPlrSctu88WtJy2r7zaEnLafvOoyUtp+07j5a0nL  
bvPfrSctq+82hJy2n7zqM1LaftO4+WtJy27zxa0nLavvNoSctP+86jJS2n7TuPlrSctu  
88WtJy2r7zaEnLuWffpjK+w68LvT00J6bt25yYtm9zYtq+zWn5+//A3Xnpu/lu9g3AA  
AAAE1FTkSuQmCC)

```

#Beberapa definisi dari bagian confusion matrix

True positive: 41 kasus di mana mesin memperkirakan jika pasien
    memiliki diabetes, dan mereka memang memiliki diabetes.

False-positive: 13 kasus dimana mesin memperkirakan jika pasien
    memiliki diabetes, padahal sebenarnya tidak memiliki diabetes

False-negative: 6 kasus dimana mesin memperkirakan jika pasien tidak
    memiliki diabetes, padahal sebenarnya memiliki diabetes

True Negative: 35 kasus di mana mesin memperkirakan jika pasien tidak
    memiliki diabetes, dan mereka memang memiliki diabetes.
"""

cm = confusion_matrix(y_test, y_pred) #memasukkan y_test dan y_pred
plt.figure(figsize=(5,5)) #ukuran plotting heatmap
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label') #Mendefinisikan sumbu y untuk Actual label
plt.xlabel('Predicted label') #Mendefinisikan sumbu x untuk predicted
    label
all_sample_title = 'Accuracy Score: {}'.format(dtree.score(X_test,
    y_test)) #melakukan kalkulasi score akurasi, {} berfungsi untuk
    menampilkan angka akurasi
plt.title(all_sample_title, size = 15) #setting ukuran tulisan
    "Accuracy Score: 0.8"

"""#XGBoost"""

from xgboost import XGBClassifier

```

```

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
    y_test)) # menampilkan score akurasi
plt.title(all_sample_title, size = 15) #format title untuk accuracy
    score

"""#Metode 80 20

#Berdasarkan Jurnal Improving the Classification Accuracy using
    Recursive Feature Elimination with Cross-Validation
yang dikutip dari :
    http://www.puneetmisra.com/admin/uploads/journals/5f136d202b8ba1.186
    44117.pdf

"In
this study, we have used the train_test_split() method of

```

the Scikit-Learn library of python. Through this function, we divide the dataset into a different ratio. However, the 80/20 (train/test) rule is mostly used in the studies."

Tujuan penggunaan train test split menjadi 80 dan 20 adalah karena rasio tersebut paling sering digunakan dalam penelitian

```
"""
```

```
#test size 20% and train size 80%
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2,
    random_state=42)

"""#Decision Tree"""

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(random_state=42)
dtree.fit(X_train, y_train)

y_pred = dtree.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
```



```

plt.ylabel('Actual label')

plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {0}'.format(dtreescore(X_test,
    y_test))

plt.title(all_sample_title, size = 15)

"""#XGBoost"""

from xgboost import XGBClassifier

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(5,5))

sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')

plt.ylabel('Actual label')

plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
    y_test))

plt.title(all_sample_title, size = 15)

"""#Metode 70 30

```

```
#Berdasarkan Jurnal A Comparative Analysis for Diabetic Prediction  
Based on Machine Learning Techniques
```

```
Yang dikutip dari : https://www.iasj.net/iasj/download/a371daadb33b96fd
```

```
Disimpulkan bahwa Train Test split dengan rasio 70 dan 30 dapat  
menghasilkan akurasi yang lebih besar dibandingkan dengan metode  
K-Fold Cross Validation
```

```
"""  
  
#test size 30% and train size 70%  
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,  
random_state=42)  
  
"""#Decision Tree"""  
  
from sklearn.tree import DecisionTreeClassifier  
  
dtree = DecisionTreeClassifier(random_state=42)  
dtree.fit(X_train, y_train)  
  
y_pred = dtree.predict(X_test)  
  
from sklearn.metrics import accuracy_score, f1_score, precision_score,  
recall_score  
  
print('F-1 Score : ',(f1_score(y_test, y_pred)))  
print('Precision Score : ',(precision_score(y_test, y_pred)))  
print('Recall Score : ',(recall_score(y_test, y_pred)))  
  
cm = confusion_matrix(y_test, y_pred)
```

```

plt.figure(figsize=(5,5))

sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')

plt.ylabel('Actual label')

plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {0}'.format(dtree.score(X_test,
y_test))

plt.title(all_sample_title, size = 15)

"""#XGBoost"""

from xgboost import XGBClassifier

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(5,5))

sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')

plt.ylabel('Actual label')

plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
y_test))

plt.title(all_sample_title, size = 15)

```

```

"""#Metode 60 40"""

#test size 40% and train size 60%
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.4,
    random_state=42)

"""#Decision Tree"""

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(random_state=42)
dtree.fit(X_train, y_train)

y_pred = dtree.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score
print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))

sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {}'.format(dtree.score(X_test,
    y_test))
plt.title(all_sample_title, size = 15)

```

```

"""#XGBoost"""

from xgboost import XGBClassifier

xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)

from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score

print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

all_sample_title = 'Accuracy Score: {0}'.format(xgb_model.score(X_test,
    y_test))
plt.title(all_sample_title, size = 15)

"""#Hasil Perbandingan Akurasi Train Test Split

Decision Tree      XGBoost

86.36              87.50  Train Test 90 10

```



86.28                    86.85    Train Test 80 20

83.58                    86.64    Train Test 70 30

80.22                    84.52    Train Test 60 40



PAPER NAME

TA-19.K1.0018.docx

WORD COUNT

6428 Words

CHARACTER COUNT

33324 Characters

PAGE COUNT

29 Pages

FILE SIZE


372.2KB

SUBMISSION DATE

Dec 13, 2022 4:42 PM GMT+7

REPORT DATE

Dec 13, 2022 4:43 PM GMT+7



- **13% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 11% Internet database
- 10% Publications database
- Crossref database
- Crossref Posted Content database
- 8% Submitted Works database

- **Excluded from Similarity Report**

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less than 10 words)
- Manually excluded text blocks