APPENDIX

IMPORT LIBRARY

```
import time

import pandas as pd

from sdv.timeseries import PAR

import matplotlib.pyplot as plt

import numpy as np

from sklearn.svm import SVR

from sklearn.ensemble import RandomForestRegressor
```

INITIALIZE

```
ins=[1, 5, 10, 15, 20, 25]

split=0.8 #insert desired percentage for training data over test data

col=6  #1 for anak-anak, 4 for penyakit dalam, 6 for bedah, 19 for forensik

detail=[1, 0, 0]

#array 0 for detail and plot

#array 1 for original data training only without sdv

#array 2 for original data testing only
```

READ AND MODIFY DATA

```
data=pd.read_csv("./dataset_SKRIPSI/DATASET_SKRIPSI_TRANSPOSED2.csv",
delimiter=';')

data["Tahun"] = pd.to_datetime(data["Tahun"])
```

DATA SPLIT

```
n=split*len(data)

df_1 = data.iloc[:int(n),:]

df_2 = data.iloc[int(n):,:]

print(len(df_1))

column_names = data.columns.values.tolist()
```

SDV GENERATION AND ANALYZE FUNCTION

```
def SDVGen_and_Analyze(df_1):


    model = PAR(sequence_index="Tahun")
    model.fit(df_1)


    time_data=[]
    start_time = time.time()
    df_1 = model.sample(num_sequences=ins[2])
    time_data.append(time.time() - start_time)
    print("running time = ", (time.time() - start_time), " seconds")


    stat1 = data.describe()
    stat2 = df_1.describe()
    plt.figure(figsize=(14,10))
    x_desc = ["original data", "5 seq"]
    y_desc=[stat1.Anestesi[0],stat2.Anestesi[0]]
    plt.subplot(2,2,1)
    plt.plot(x_desc,y_desc,"-b",marker = '|',label='number of data')
    plt.legend(loc="upper left")
    x_desc = ["original data", "5 seq"]
    y_desc = [stat1.Anestesi[1],stat2.Anestesi[1]]
    plt.subplot(2,2,2)
    plt.plot(x_desc,y_desc,"-r",marker = '|',label='mean')
    plt.legend(loc="upper left")


    return(df_1)
if detail[1]==0:
    df_1=SDVGen_and_Analyze(df_1)
```

TESTING DATA DECLARATION

```
test_value=[[0]*200 for i in range(21)]

i=0

for _ in range(20):

       if detail[2]==0: test_value[i]=df_2[column_names[i+1]].values

       elif detail[2]==1: test_value[i]=data[column_names[i+1]].values

       i=i+1
```

REGRESSION MODEL FITTING

```
x=[]

year_init=2015-len(df_1)

year=year_init

for _ in range(int(len(df_1))):

       year=year+1

       x.append(year)

if detail[2]==0:

       if detail[1]==1:

              x=[]

              x=df_1['Tahun'].values

x = np.array(x)

x = x.reshape(-1, 1)


y = df_1.iloc[:,(col+1):(col+2)].values.astype(int)

print(y)

regressor = SVR()

regressor.fit(x,y.ravel())

regressor2 = RandomForestRegressor()

regressor2.fit(x, y.ravel())
```

TESTING YEAR DECLARATION

```
test_year=[]

year=2015

for j in test_value[col]:

	year=year+1

	test_year.append(year)

if detail[1]==1:

	test_year=df_2['Tahun'].values

elif detail[2]==1:

	test_year=data['Tahun'].values
```

REGRESSION

```
y_pred = []

for j in test_year:

	y_pred.append(int(regressor.predict([[j]])))

Y_pred = []

for j in test_year:

	Y_pred.append(int(regressor2.predict(np.array(j).reshape(1, 1))))
```

MAPE FUNCTION

```
def MAPE(test_data, predicted_data):

	result = np.empty(test_data.shape)

	for j in range(test_data.shape[0]):

		result[j] = (test_data[j] - predicted_data[j]) / test_data[j]

	result = np.mean(np.abs(result)) * 100

	return result
```

EVALUATION

```
if detail[0]==1: print("Test Value = ", test_value[col])

if detail[0]==1: print("Jumlah Training Data =", len(df_1))

if detail[0]==1: print("Jumlah Test Data =", len(test_value[col]))
```

```
#calculate mse for svr and rf prediction
mse_svr = np.mean(((test_value[col] - np.array(y_pred)) ** 2))
print("Mean Square Error SVR =", mse_svr)


mse_rf = np.mean(((test_value[col] - np.array(Y_pred)) ** 2))
print("Mean Square Error RFR =", mse_rf)
#caculate mape for svr and rf prediction
mape_svr = MAPE(np.asarray(test_value[col]),np.asarray(y_pred))
print("Mean Absolute Percentage Error SVR =", mape_svr)
mape_rf = MAPE(np.asarray(test_value[col]),np.asarray(Y_pred))
print("Mean Absolute Percentage Error RFR =", mape_rf)
```

PAPER NAME
18.K1.0046_Binar Eko Harijanto

AUTHOR
Binar Eko Harijanto

WORD COUNT
5091 Words

CHARACTER COUNT
27672 Characters

PAGE COUNT
19 Pages

FILE SIZE
360.6KB

SUBMISSION DATE
Jun 7, 2022 2:30 PM GMT+7

REPORT DATE
Jun 7, 2022 2:31 PM GMT+7

4% Overall Similarity

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

- 3% Internet database
- 3% Publications database
- Crossref database
- Crossref Posted Content database
- 0% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 10 words)