CHAPTER 4
ANALYSIS AND DESIGN

4.1 Analysis

To process the dataset, firstly first the data should pass some step so the result of the prediction could be effective, those steps are:

1. Data Preparation & Cleaning: The data preparation step is to find the correlation between all of the data that matters to the prediction result, the data cleaning step is to make the data clean from the unused data and checking if there are any empty data that need to be removed.

2. Feature Transformation: In this project, feature transformation is used to fill the unknown values of the variables into the mean of the all value. This step is to make the models performs better without deleting the variables because of the unusual data.

3. Feature Engineering: Feature engineering step is to see and handle the data so it can improve the model. Here are some advantages of doing feature engineering:

1. Remove irrelevant feature (reduce overfitting).
2. Increase the model performance.
3. Faster training (Reduce time training).
4. Easier to debug.
5. Easier to build.
6. Faster to understand.

In this project, feature engineering is implemented to change the scale of the data in the range between 0 and 1 and to select the important features in the data.

4. Data Evaluation: To train and evaluate the data by modifying the model to get the best performance of the model.

Supervised Learning

Supervised learning means that the models are trained using labeled data, in other word, it needs to be supervised to train the model to give an output data. Supervised learning is used to solve classification and regression problems, since this method classify a new data to the old data. This method usually requires two types of data, one is training data and the other is testing data. The trained data will be used to find the perfect matches model for the test data, so it can predict a new input.

Unsupervised Learning

Unsupervised Learning means that the patterns inferred from the unlabeled input data. Unsupervised learning is used to find the structure and pattern from the input data without any supervision, but finds pattern from the data by its own. It works by searching the hidden structure and find their correlation between one and others, that is why this method usually used as clustering.

After looking intensely at the literature and soliciting the experience of human experts on pathology, the factors will be recognized which have a big impact on the result of this project. Logistic regression and Naïve Bayes will be used to process the data with all of the factors that matter to the diabetes impact. Because the output has only two possible outcomes, logistic regression and Naïve Bayes Classifier are the suitable algorithms to solve the problem. Logistic regression in this project follows similar steps as linear regression, the data is divided to minimize the error as the example image shows below.

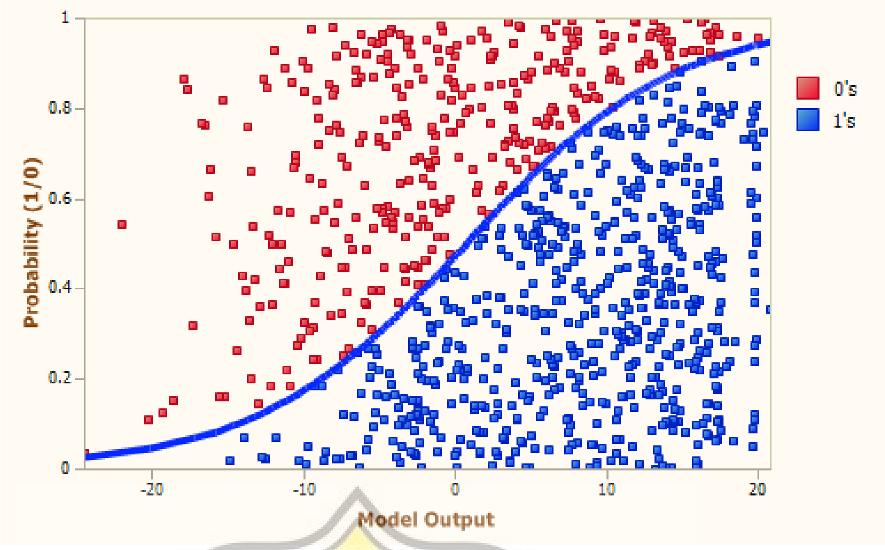

Illustration 4.2. Logistic Regression Graph (Abin, 2019)

And the Naïve Bayes algorithm will be shown as the image below.

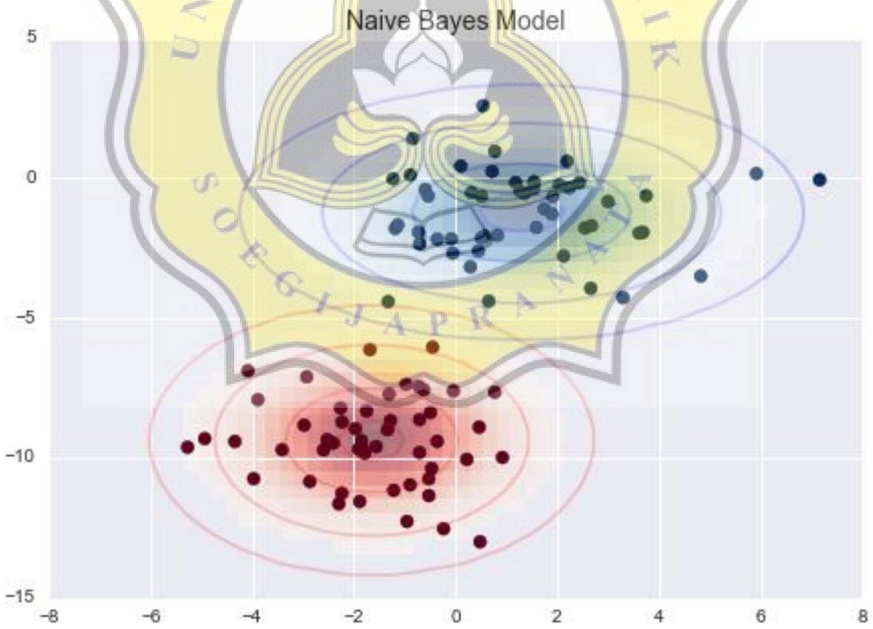

Illustration 4.3. *Naïve Bayes classifier explained in image (Jake, 2016)*

4.1.1. Scikit Library of Gaussian Naïve Bayes

In this section, the data will be processed in the Sci-Kit Learn library using Naïve Bayes. With the library, the first thing to do is load the data, and then train the model, finding accuracy and predicting pass or fail. This algorithm [9] itself is a set of supervised learning algorithms that applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. In Sci-kit learn the regular Naïve Bayes has the cost function as follows:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

where the parameters σ_y and μ_y are estimated using maximum like hood.

The final result of the Naïve Bayes from Scikit Library to predict the onset of diabetes returned the value of accuracy is 0.77 which has a higher value than the logistic regression code without the library.

4.1.2. Logistic Regression without Library

In this section will talk about the logistic regression algorithm without a library, on the other words it's need to define a class to make the logistic regression. In this step the logistic regression algorithm using sigmoid function and a gradient descent technique. Sigmoid function is a real function that is defined for all real input values and has a non-negative derivative at each point. Sigmoid function defined by the formula:

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} = 1 - S(-x).$$

where S equals the sigmoid function, for the more explanation of a sigmoid function, the graph image is show below:

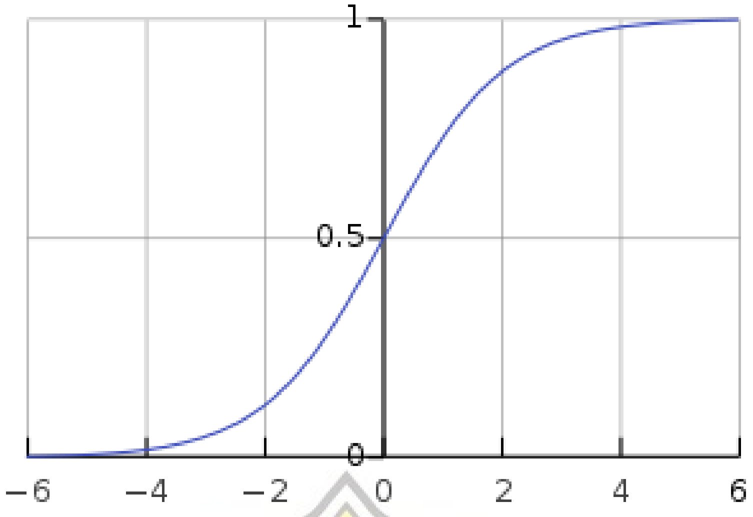

Illustration 4.4. *Sigmoid function graph. (Qef, 2008)*

The images show that sigmoid function is always between 1 and 0, because of the biggest number is 1.

Beside sigmoid function, gradient descent is also an important step in this logistic regression to reach the optimized network weight and bias value. It works by iteratively trying to minimize the cost function. In simple way it works by differentiate the value until get the minimum peak, the picture bellow to illustrate the words.

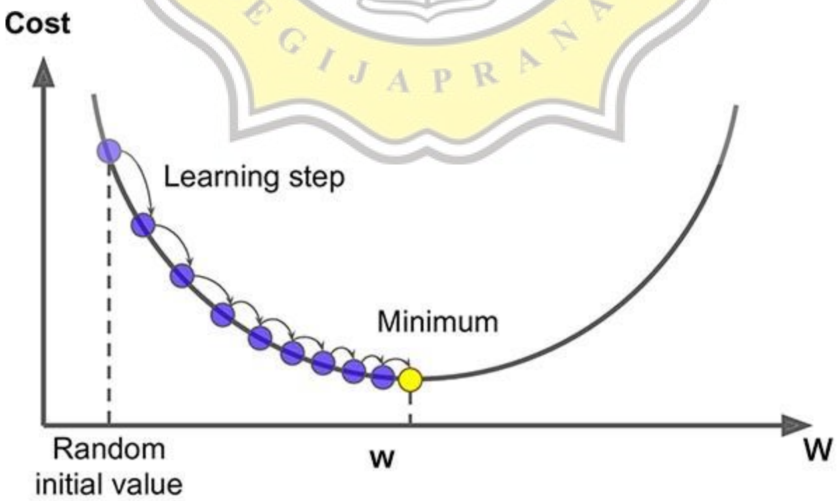

Illustration 4.5. *Gradient Descent Graph illustrated (2020)*

The gradient descent is defined by the formula:

-

$$y = m\,x + b$$

With assumption to search the optimal value of parameters m and b,

$$input\ (x) \rightarrow model(y = m\,x + b) \rightarrow predicted(\hat{y}) \rightarrow actual\ true\ (y)$$

From the predicted value, if it cannot get the actual true, then it will return into the model by update the weight parameters and the error could be define by $(\hat{y} - y)$. With the lost function's formula:

$$Cost\ Function\ f(m, b) = \frac{1}{n}\sum_{i+1}^{n}(error)^2 = \frac{1}{n}\sum_{i+1}^{n}(\hat{y} - y)^2$$

The gradient descent works as follow:

$$new\ weight = \ old\ weight - stepsize$$

The step size itself come with formula:

$$stepsize = learning\ rate\ \times gradient = \ \alpha\frac{d_{loss}}{d_w}$$

So, the gradient descent formula could be written:

$$W_{new} = W_{old} - \alpha\frac{d_{loss}}{d_w}$$

After the logistic regression algorithm been run, it returns accuracy of 0.76. From the result, the logistic regression without the library already works effectively since it has a little difference with the accuracy of the Naïve Bayes model.

4.2 Desain

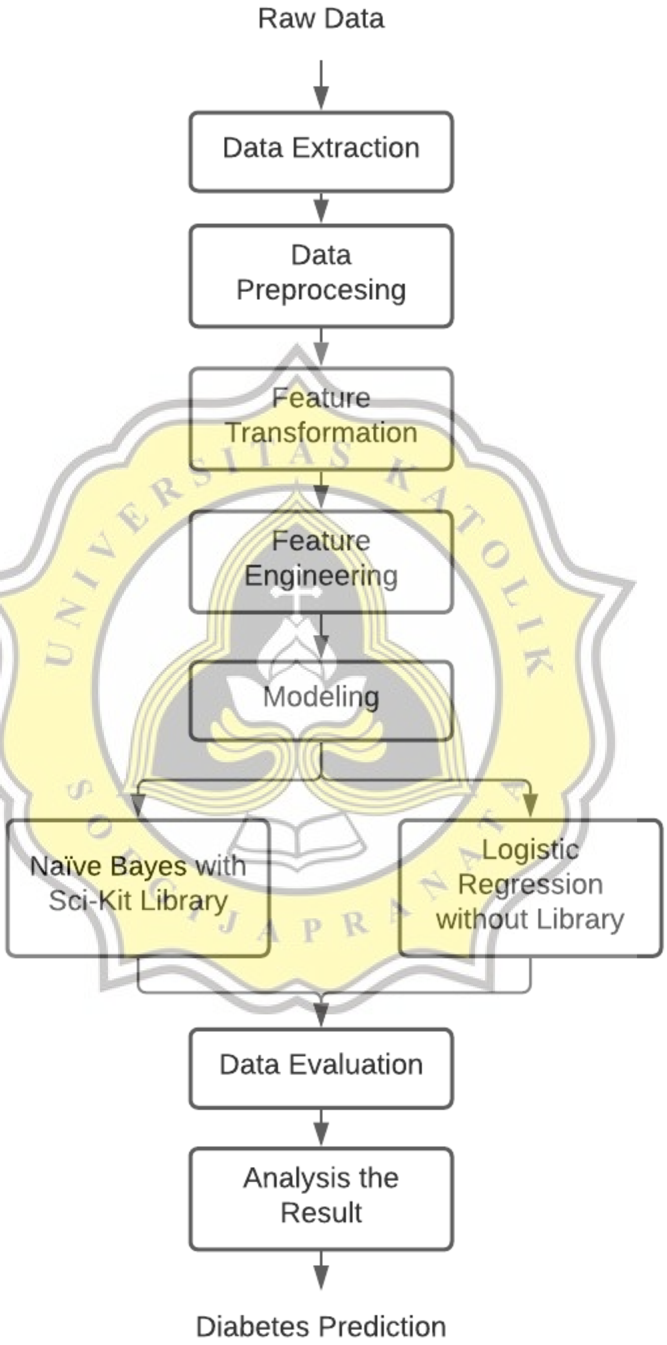

Illustration 4.6. *Flow chart of this project*

The design of this project was shown in the picture above. It will start with getting the raw data and then bring it to data extraction, to make sure that the data could be read by the system. The next step is Data preprocessing which includes data preparation and data cleaning, where the data need to be cleaned from the unused data and to find the correlation of the data so the model can perform better. The next step is feature transformation to transform the unknown values in the dataset. Then starts in the feature engineering as it's been explained before.

After that, start to model the algorithm, get the learning rate, and how many iterations will be used. The next step is to run the algorithm, in this case, two methods used to gain the best result. One is using Naïve Bayes with the Sci-Kit library and the other is using logistic regression without using the library. After the algorithm works, data evaluation is needed to make sure that the result returned from the best performance. When the result has come, analyzing the result is an important thing to make sure that there is no mistake. Lastly, the diabetes prediction will appear.