**VIETNAM NATIONAL UNIVERSITY-HO CHI MINH CITY**

**UNIVERSITY OF SCIENCE**

**FACULTY OF INFORMATION TECHNOLOGY**

**------------------------------------------------**



**FINAL PROJECT**

**REPORT**

**Student : 20C14001 – Le Duong Tuan Anh**

**Teacher : Dr. Le Thi Nhan**

**Class : Master of Information System**

**Subject : Data Mining**

**Ho Chi Minh city, July 2021**

**Contents**

[1 General Information 3](#_Toc79306571)

[Student Information 3](#_Toc79306572)

[Report Information 3](#_Toc79306573)

[2 Problem Statement 4](#_Toc79306574)

[3 Data Description 5](#_Toc79306575)

[4 Data Understanding 7](#_Toc79306576)

[4.1 Description 7](#_Toc79306577)

[4.2 Overview by ScatterPlot 8](#_Toc79306578)

[4.3 Statistical Information 9](#_Toc79306579)

[5 Data Preprocessing 14](#_Toc79306580)

[5.1. Filling Zero-Values 14](#_Toc79306581)

[5.1.1. Fill by Mean method 15](#_Toc79306582)

[5.1.2. Fill by Regression Method 16](#_Toc79306583)

[5.2. Outliers Removal 19](#_Toc79306584)

[6 Solution 21](#_Toc79306585)

[6.1. Normalization 21](#_Toc79306586)

[6.2. Model Training 25](#_Toc79306587)

[6.3. Result 27](#_Toc79306588)

[6.3.1. Accuracy Score 27](#_Toc79306589)

[6.3.1. Confusion Matrix 28](#_Toc79306590)

[7 Conclusion 30](#_Toc79306591)

# General Information

## Student Information

|  |  |  |
| --- | --- | --- |
| Student ID | Full Name | Email |
| 20C14001 | Le Duong Tuan Anh | [leduongtuananh97@gmail.com](mailto:leduongtuananh97@gmail.com) |

## Report Information

This report is a summary of the **Final Project** based on **Diabetes prediction with given Pima Indians medical details.**

# Problem Statement

In this project, I use dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases, which contains data between relationship between some medical details and the result on Diabetes.

**The objective** of the dataset is to diagnostically predict whether a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

The dataset consists of **several medical predictor variables** and **one target variable**, ***Outcome***.

Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

# Data Description

Total instances (records): 768

Total columns: 9 (8 medical attributes + 1 Outcome Class (has diabetes or not)).

Attribute’s detail:

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Type | Datatype | Has missing value? |
| Pregnancies | Interval-scaled | Int64 | No |
| Glucose | Interval-scaled | Int64 | No |
| BloodPressure | Interval-scaled | Int64 | No |
| SkinThickness | Interval-scaled | Int64 | No |
| Insulin | Interval-scaled | Int64 | No |
| BMI | Interval-scaled | Float | No |
| DiabetesPedigreeFunction | Interval-scaled | Float | No |
| Age | Interval-scaled | Int64 | No |
| Outcome | Categorical Data Binary (0/1) | Int64 | No |

Attribute’s meaning:

**Pregnancies**: Number of times pregnant

**Glucose**: Plasma Glucose Concentration.

**BloodPressure**: Diastolic Blood Pressure.

**SkinThickness**: Estimate body fat.

**Insulin**: 2-Hour Serum Insulin.

**BMI**: Body Mass Index.

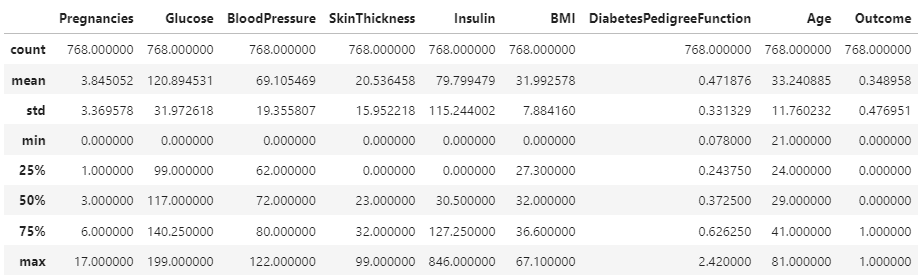
**DiabetesPedigreeFunction**: information about diabetes history in relatives and genetics.

**Age**: Age (years).

**Outcome**: 0 = Diabetic, 1 = Not Diabetic

# Data Understanding

## Description

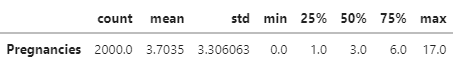


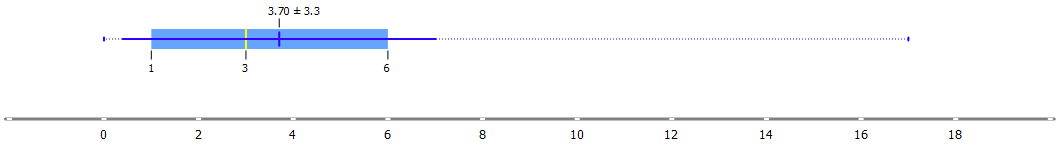
## Overview by ScatterPlot

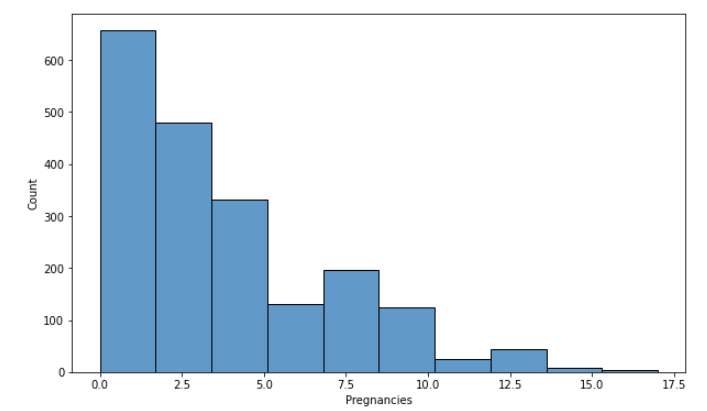


## Statistical Information

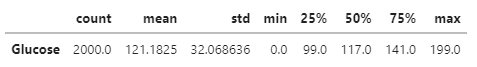
**2.1. Pregnancies**

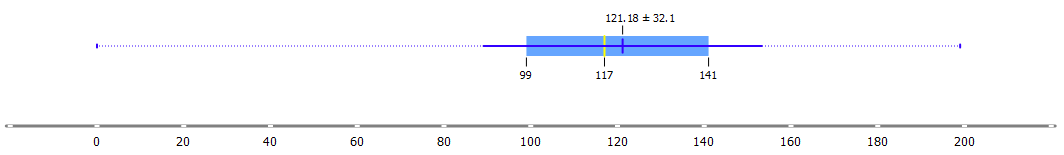


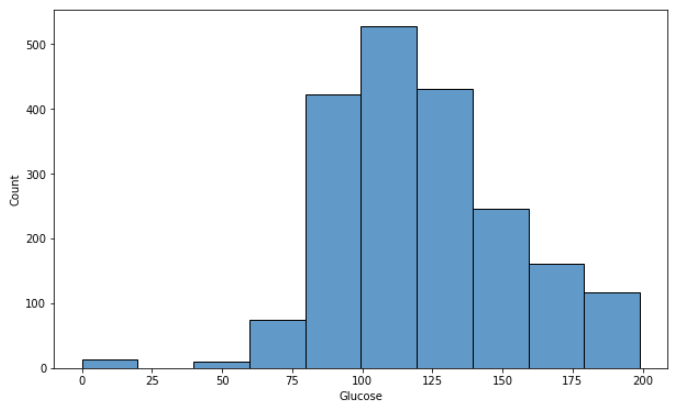




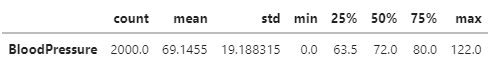
**2.2. Glucose**

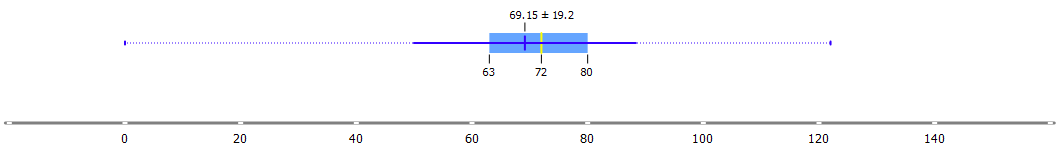


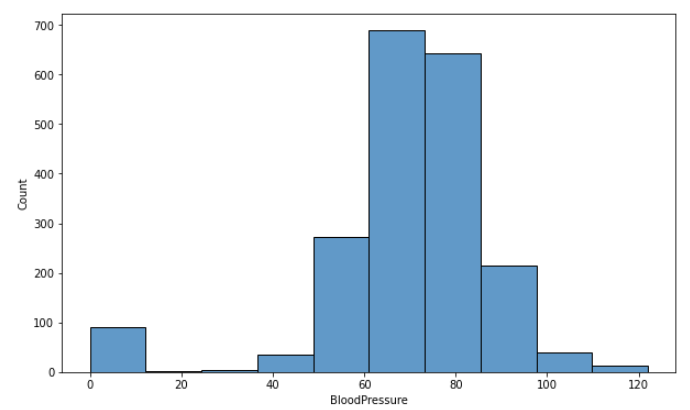




**2.3. BloodPressure**

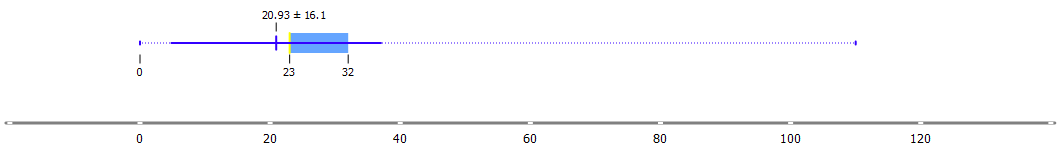


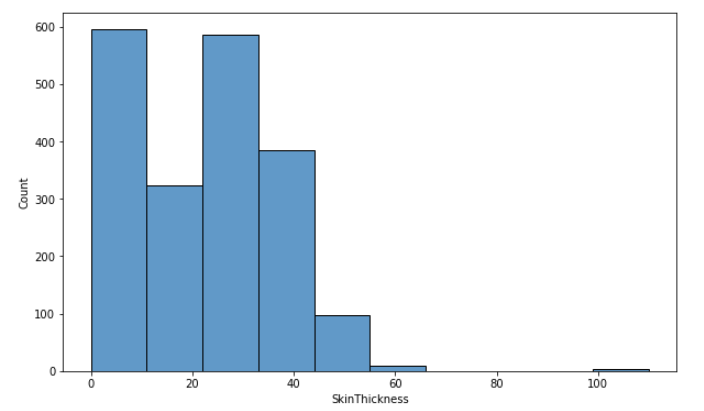




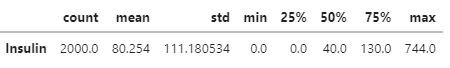
**2.4. SkinThickness**

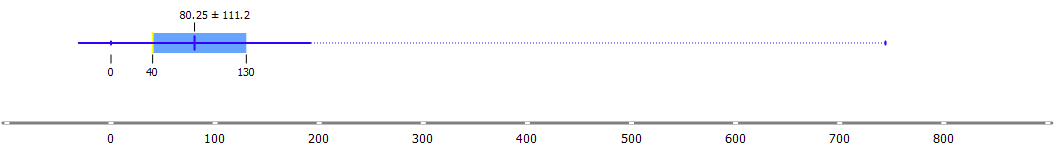


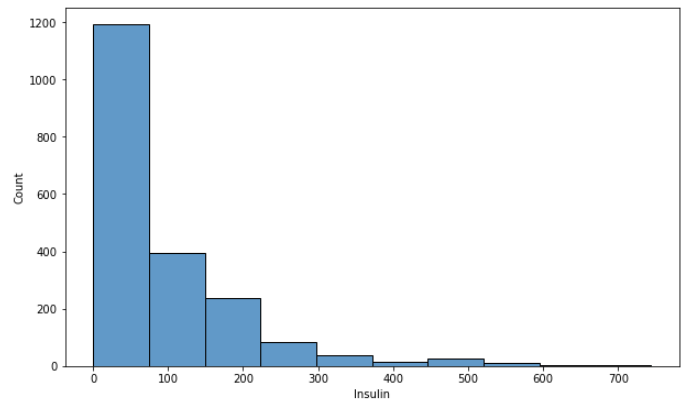




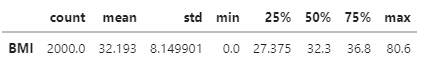
**2.5. Insulin**

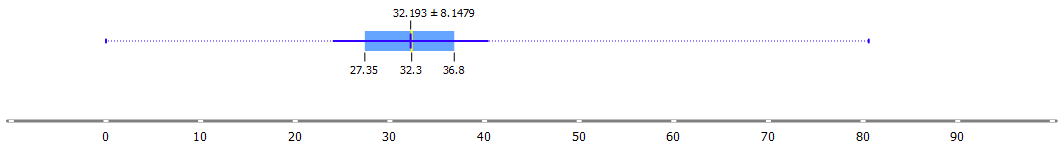


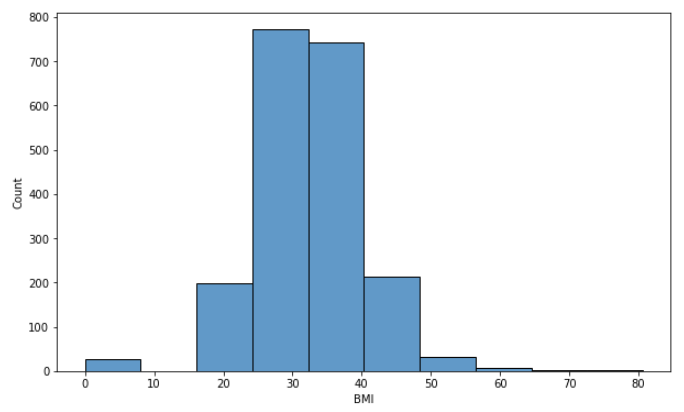




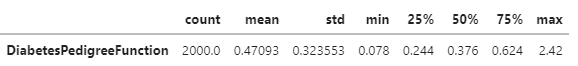
**2.6. BMI**

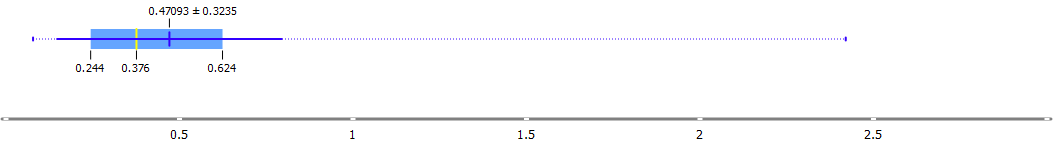


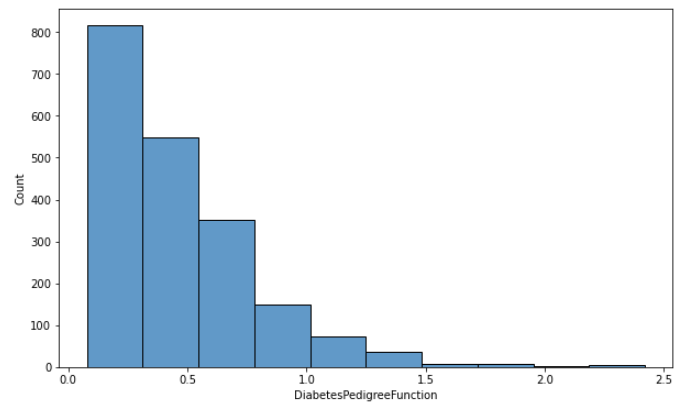




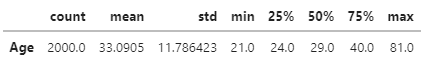
**2.7. DiabetesPedigreeFunction**

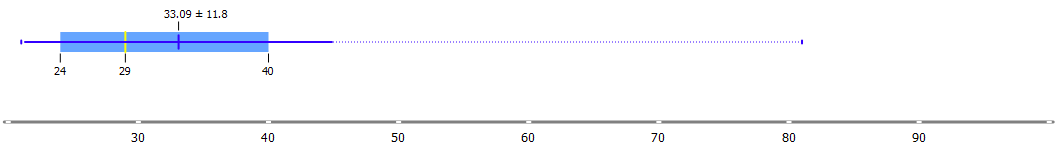


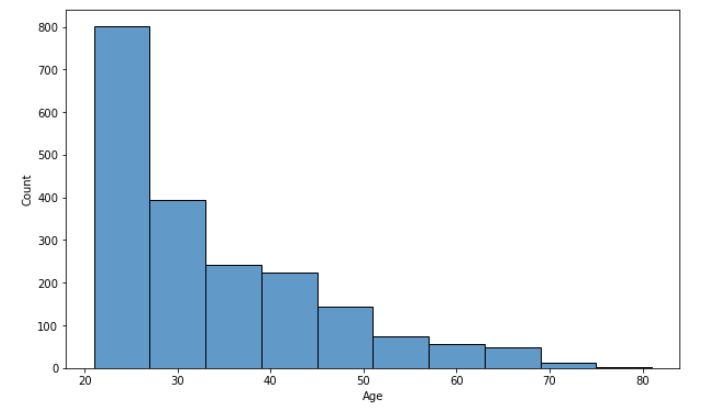




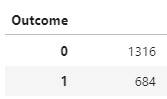
**2.8. Age**

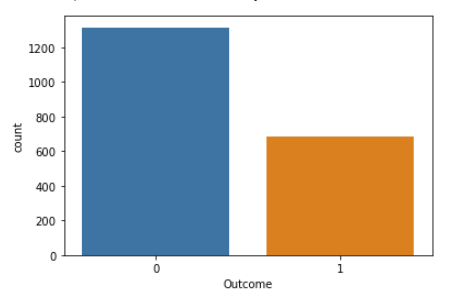






**2.9. Outcome**

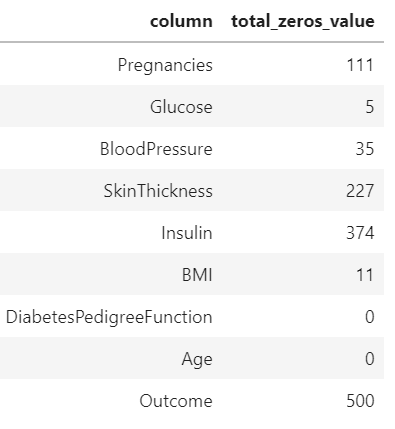




# Data Preprocessing

As we don’t have any null-values, so we will check on zero-values and remove outliers (if needed).

## 5.1. Filling Zero-Values



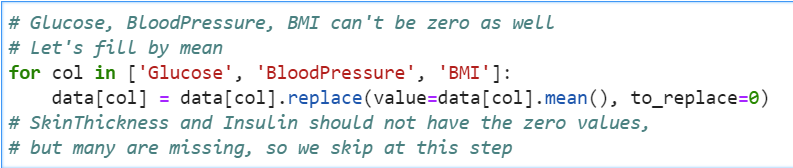
The **Pregnancies** column can be zero, as the gender might be male. There are 5 columns has zeros values: **Glucose, BloodPressure, SkinThickness, Insulin and BMI**. For example, SkinThickness cannot be zero, and BMI also. The other medical details cannot be zero, at least there must be very small values.

So, let’s apply different type of zero-filling methods here.

### 5.1.1. Fill by Mean method

We just fill by mean on columns that have not much zero-values.

**The applied columns: Glucose, BloodPressure, BMI.**

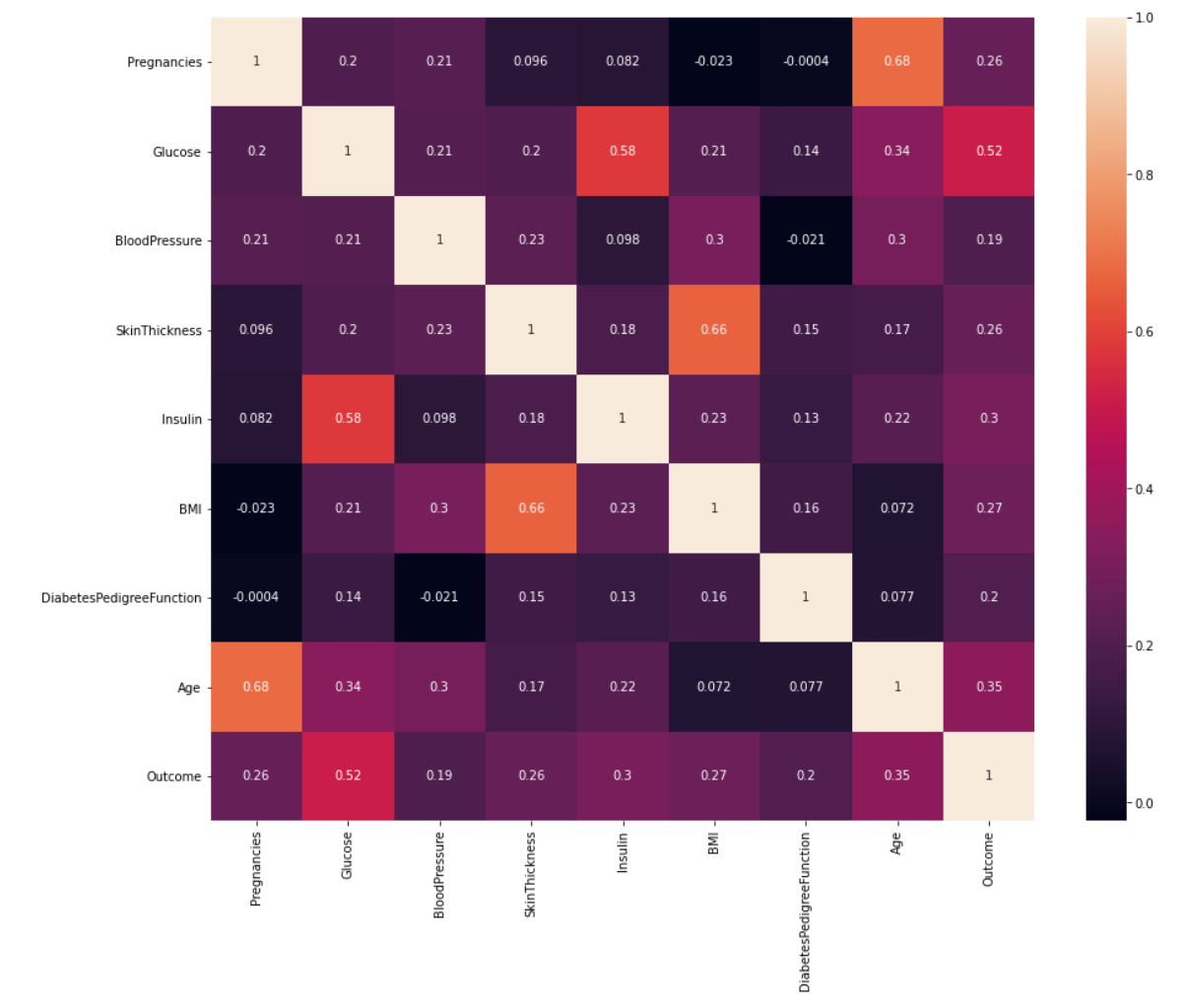


|  |  |  |
| --- | --- | --- |
| **Column** | **Before filling zeros** | **After filling zeros** |
| Glucose |  |  |
| BloodPressure |  |  |
| BMI |  |  |

### 5.1.2. Fill by Regression Method

The regression method is a very useful when the columns that having zero-values are very correlated with the other columns. So we will try this method on these columns: **Insulin, SkinThickness.**

Correlation Matrix, **Pearson metric** *(on dataset that have Insulin != 0 and SkinThickness != 0):*



We can see that there are some high correlations between:

|  |  |
| --- | --- |
| Insulin and Glucose (0.58). | SkinThickness and BMI (0.66). |

We try to estimate the zero-values by Linear Regression:

|  |  |
| --- | --- |
| **Insulin = Glucose \* 2.23952761** | **SkinThickness = BMI \* 0.99549695** |

Now the filling result:

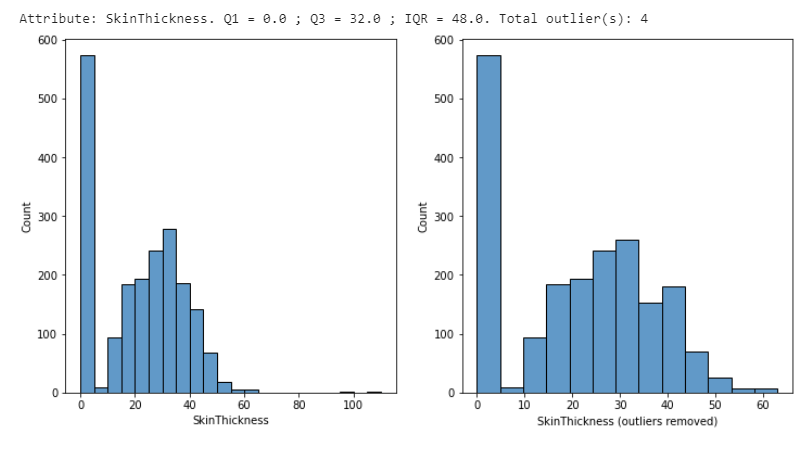
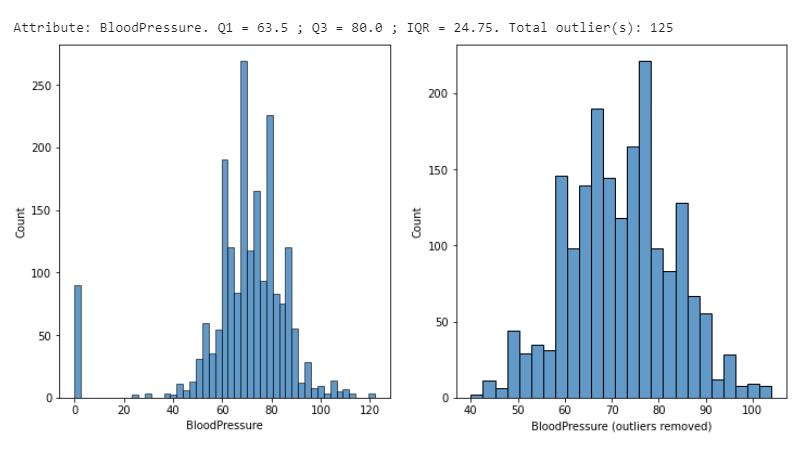
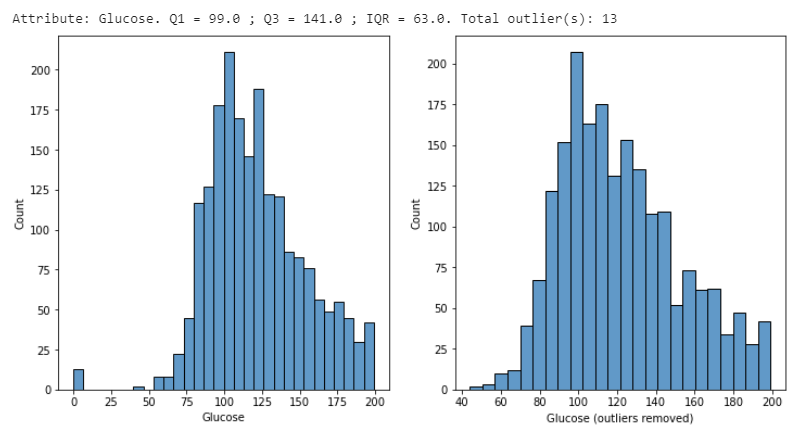
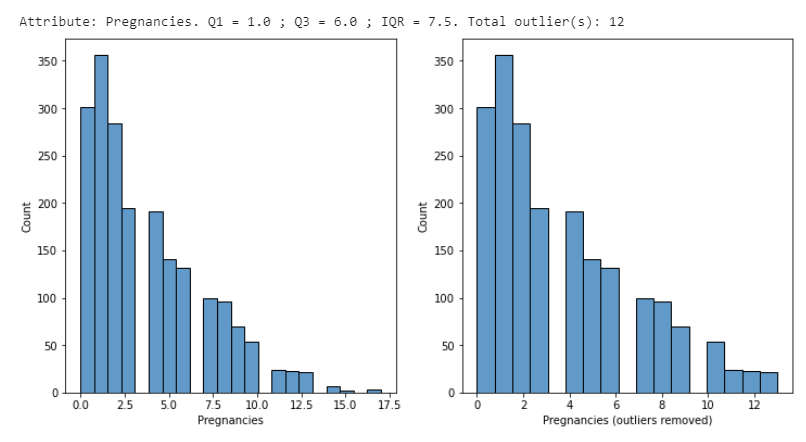
|  |  |  |
| --- | --- | --- |
| **Column** | **Before filling zeros** | **After filling zeros** |
| Insulin |  |  |
| SkinThickness |  |  |

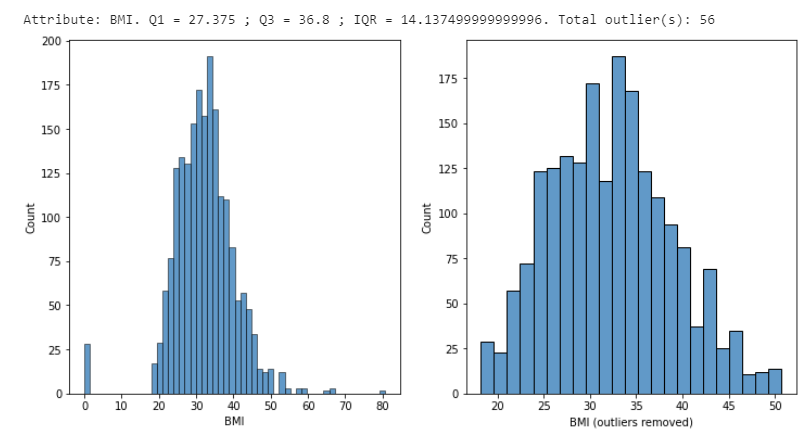
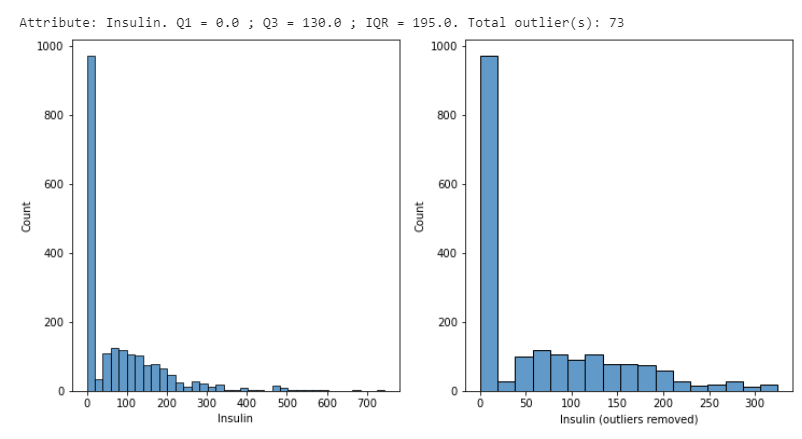
## 5.2. Outliers Removal

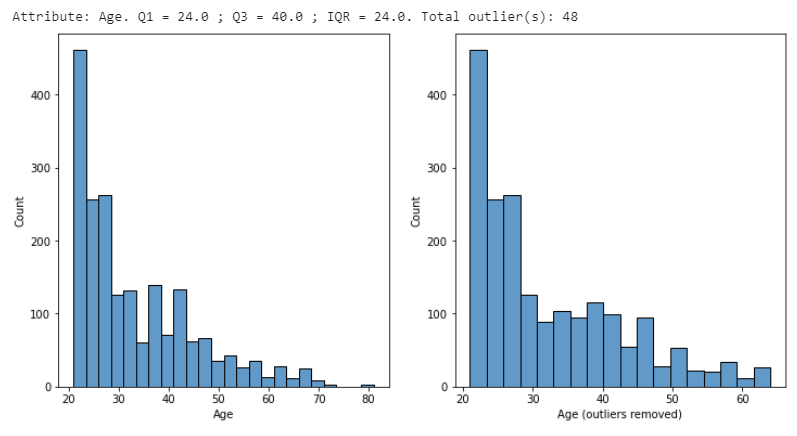
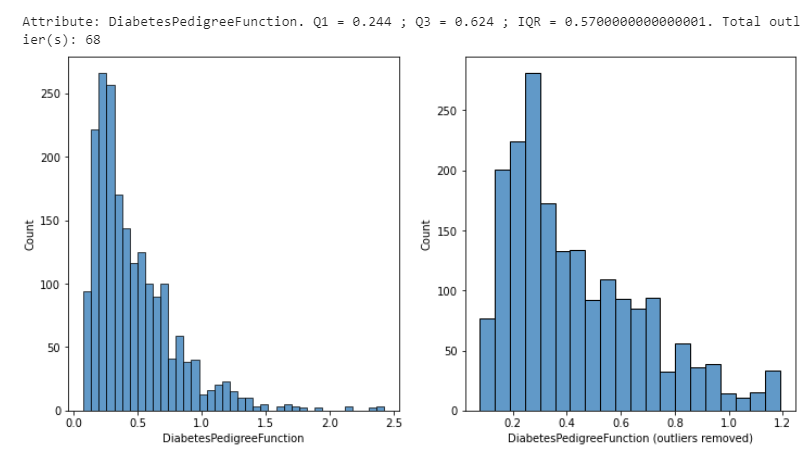
We would consider outliers by using **Inter-quartile Range** method. In case the data point is out of range

[ Q1 – 1.5\*IQR, Q3 + 1.5\*IQR ]

it might be **outliers**.







**Total removed outliers in dataset: 11,2%**

# Solution

In this section, I will do some different classification methods to classify if the people have diabetes or not, based on their medical details.

Before we do the classification tasks, we need to normalize the input first.

## 6.1. Normalization

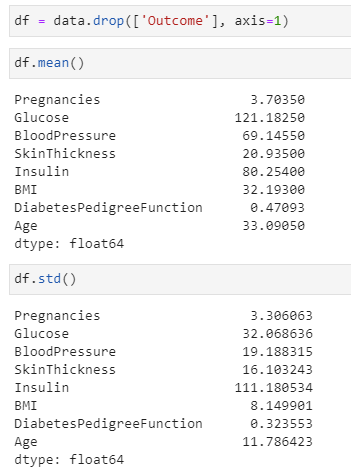
We standardize features by removing the mean, then scaling dataset to unit variance. Note that we will not standardize the **Outcome** column, as it is a **categorical column**.

The standardized score is calculated as below:

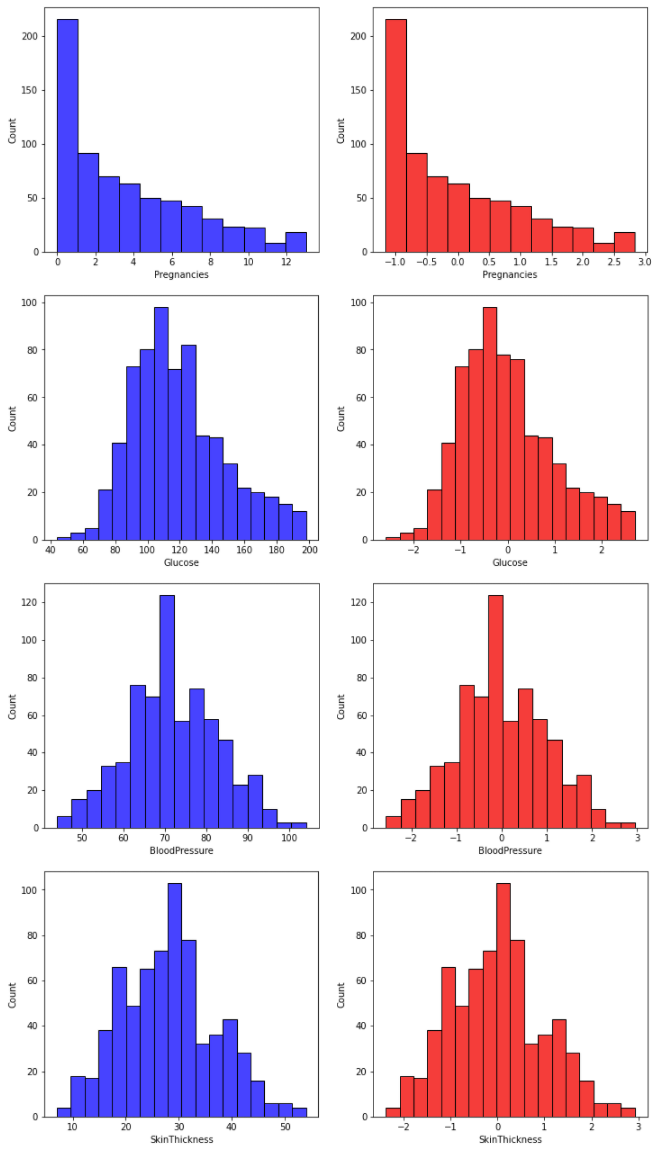
**Z = (X – U) / S**

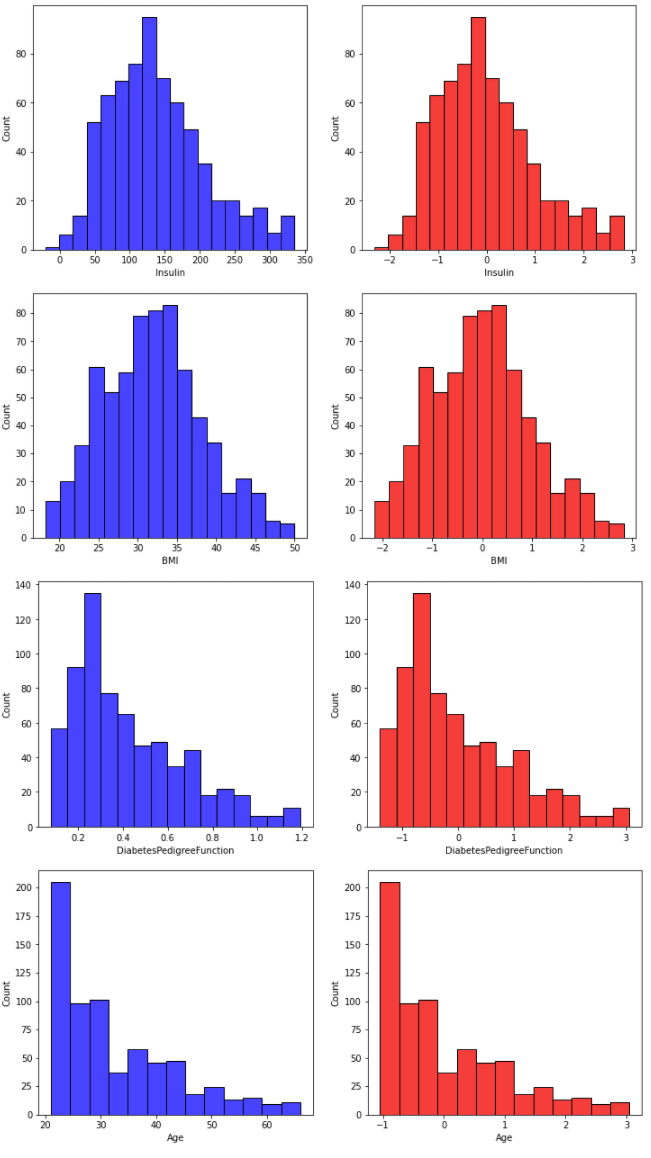
**Where:**

* X: The value of sample in dataset.
* U: The mean of dataset.
* S: The standard deviation of dataset.



The result after normalization *(the blue shows original, the red shows normalized)*





## 6.2. Model Training

In this project, I have done on some different kind of classification methods as below:

* Logistic Regression

https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html

* Decision Tree

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier

* Random Forest

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

* SVM (Linear Kernel)

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

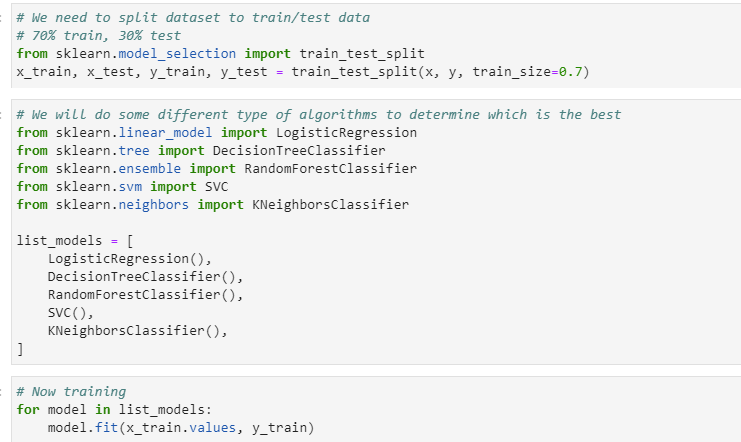
* kNN

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

The implementation has been done by scikit-learn library (Python), so I just use the model in the library.

The data is split to **trainset** and **testset**, with ratio **7/3**: 70% data for training and 30% for testing, respectively.

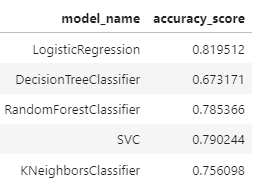
*The splitting data and training code:*

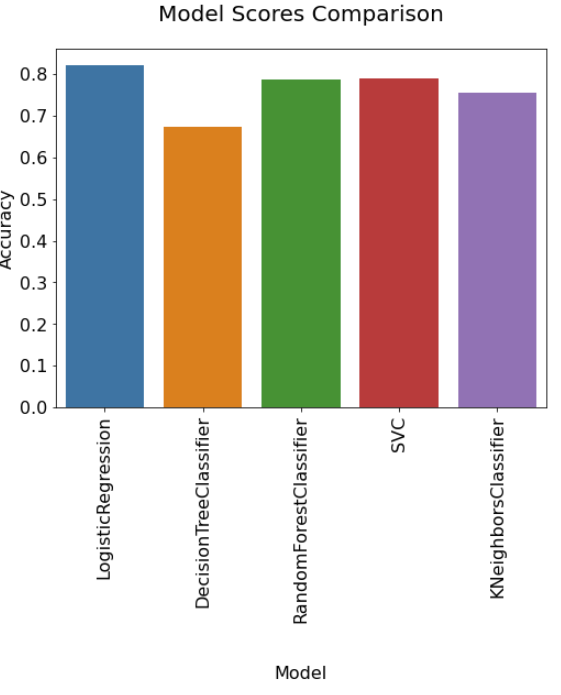
****

## 6.3. Result

The testing result are calculated on 30% data (testset).

### 6.3.1. Accuracy Score





With accuracy metrics, the **LogisticRegression** gave the best result.

### 6.3.1. Confusion Matrix

To analyze more in detail on the prediction of classification problems, we use confusion matrix.

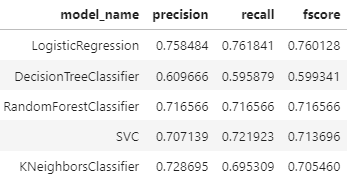
|  |  |
| --- | --- |
| Model | Confusion Matrix |
| LogisticRegression |  |
| DecisionTreeClassifier |  |
| RandomForestClassifier |  |
| SVM |  |
| kNN |  |

This classification is done on **medical examination**, which is the False Negative is one of the most importance metrics. Based on this result, Logistic Regression method gave the best result with **8.78% (lower is better).**

# Conclusion

In this project, I have done end-to-end flow to analyze, process and do some classification methods on the Diabetes Prediction task.

The precision, recall and f-score of all methods are below:



In conclusion, **the simple Logistic Regression** performs the **best result** on the provided dataset.

This result might be improved if we use other complex methods, e.g., Deep Learning.