# MTH786P - Machine Learning with Python: Project on Diabetes Data Set

Registration Number: 220891802

Marc Grammersdorf

## 1 Introduction

In this report, the objective is to create (binary) regression/classification models that use any combination of patient characteristics as inputs to predict whether they have diabetes or not. It is assumed that our patients are females, since the column of pregnancies is given. Further down, two different models were used to analyse the data.

### 1.1 Data Set

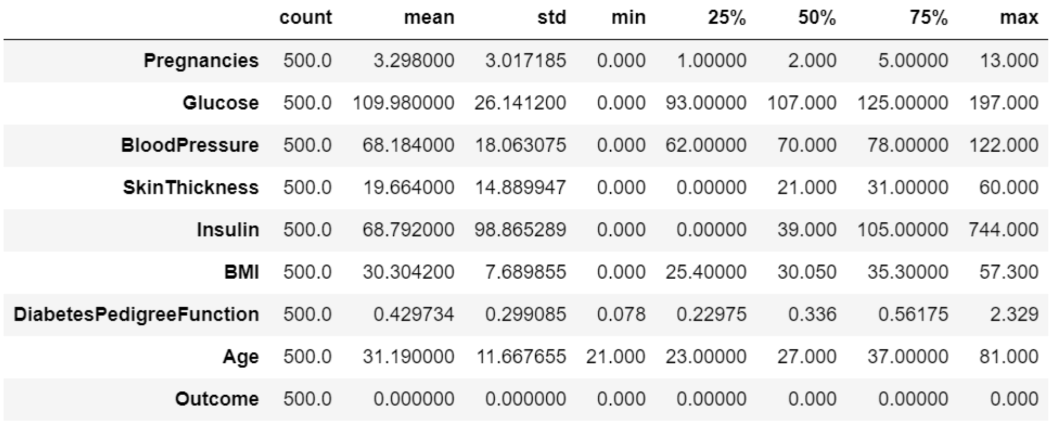
Columns Found in Data Set (Smith et al., 1988):

1. **Pregnancies:** How many times our patients were pregnant
2. **Glucose:** A Glucose Tolerance Test. (GTT) Checks how well the body processes blood sugar. The test is done by comparing the levels of glucose in the blood before and after drinking a sugary drink
3. **BloodPressure:** Diastolic Blood Pressure. Measures the arterial pressure when the heart rests between beats. (Normal range=[60-80mmHg])
4. **SkinThickness (SFT) :** Measurement of body fat
5. **Insulin:** A hormone that helps move glucose from the bloodstream into the cells. It is produced by the pancreas
6. **BMI:** Body Mass Index. A metric that calculates whether body weight is considered healthy based on the height and weight of the individual
7. **DiabetesPedigreeFunction:** A function that assesses the probability of having diabetes based on family history. [Measured from 0-1]
8. **Age:** Age of patient in years
9. **Outcome:** The result whether the patient is diabetic or not.[Yes=1 / No=0]

## 2 Visualisations

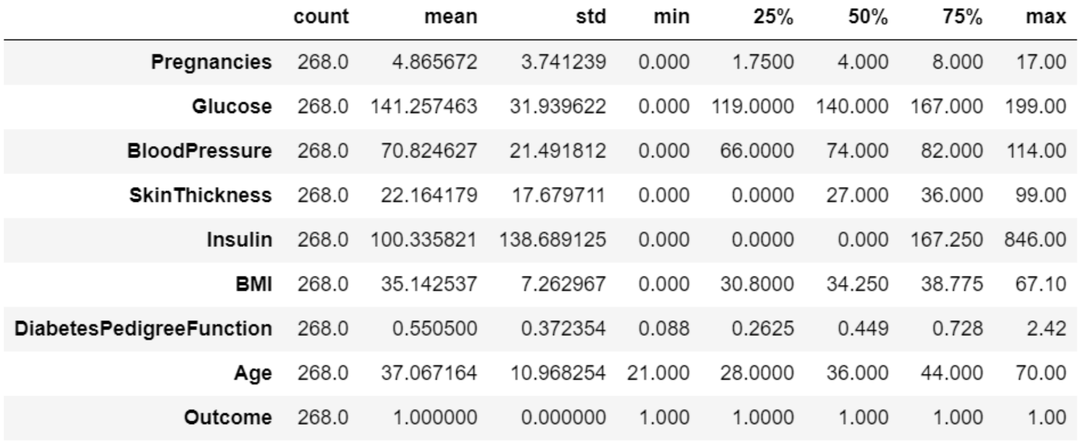
For this section, it was important to find out about the measures of central tendency(e.g. Mean,Min,Max) for each column. Before creating the table, the data was split into two sections; outcome 0 and outcome 1.

**Table for Outcomes with 0's:**



| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Pregnancies** | 500.0 | 3.298000 | 3.017185 | 0.000 | 1.00000 | 2.000 | 5.00000 | 13.000 |
| **Glucose** | 500.0 | 109.980000 | 26.141200 | 0.000 | 93.00000 | 107.000 | 125.00000 | 197.000 |
| **BloodPressure** | 500.0 | 68.184000 | 18.063075 | 0.000 | 62.00000 | 70.000 | 78.00000 | 122.000 |
| **SkinThickness** | 500.0 | 19.664000 | 14.889947 | 0.000 | 0.00000 | 21.000 | 31.00000 | 60.000 |
| **Insulin** | 500.0 | 68.792000 | 98.865289 | 0.000 | 0.00000 | 39.000 | 105.00000 | 744.000 |
| **BMI** | 500.0 | 30.304200 | 7.689855 | 0.000 | 25.40000 | 30.050 | 35.30000 | 57.300 |
| **DiabetesPedigreeFunction** | 500.0 | 0.429734 | 0.299085 | 0.078 | 0.22975 | 0.336 | 0.56175 | 2.329 |
| **Age** | 500.0 | 31.190000 | 11.667655 | 21.000 | 23.00000 | 27.000 | 37.00000 | 81.000 |
| **Outcome** | 500.0 | 0.000000 | 0.000000 | 0.000 | 0.00000 | 0.000 | 0.00000 | 0.000 |

**Table for Outcomes with 1's:**



| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Pregnancies** | 268.0 | 4.865672 | 3.741239 | 0.000 | 1.7500 | 4.000 | 8.000 | 17.00 |
| **Glucose** | 268.0 | 141.257463 | 31.939622 | 0.000 | 119.0000 | 140.000 | 167.000 | 199.00 |
| **BloodPressure** | 268.0 | 70.824627 | 21.491812 | 0.000 | 66.0000 | 74.000 | 82.000 | 114.00 |
| **SkinThickness** | 268.0 | 22.164179 | 17.679711 | 0.000 | 0.0000 | 27.000 | 36.000 | 99.00 |
| **Insulin** | 268.0 | 100.335821 | 138.689125 | 0.000 | 0.0000 | 0.000 | 167.250 | 846.00 |
| **BMI** | 268.0 | 35.142537 | 7.262967 | 0.000 | 30.8000 | 34.250 | 38.775 | 67.10 |
| **DiabetesPedigreeFunction** | 268.0 | 0.550500 | 0.372354 | 0.088 | 0.2625 | 0.449 | 0.728 | 2.42 |
| **Age** | 268.0 | 37.067164 | 10.968254 | 21.000 | 28.0000 | 36.000 | 44.000 | 70.00 |
| **Outcome** | 268.0 | 1.000000 | 0.000000 | 1.000 | 1.0000 | 1.000 | 1.000 | 1.00 |

Using the tables above, interpretation was done to identify which columns are the most important for the classification models.For instance, it can be concluded that glucose and age play a crucial role for the outcome of diabetes, since the difference in their means from each table differ a lot.(Age column means: 0s=31.19 and 1s=37.07. Glucose column means: 0s=109.98 and 1s=141.26) Giving a signal that these columns will be needed for our modelling to make the classification accuracy higher.

# 3 Methods

In this section, two different classification models were used which are the Binary Logistic Regression(BLR) and K-Nearest-Neighbors(KNN).

## 3.1 Binary Logistic Regression

Uses a logistic function to model the probability of a binary outcome given a set of independent variables. The parameters of the model are estimated from the training data using maximum likelihood estimation. Once the model is trained, it can be used to make predictions for new data.
To convert a binary classification prediction, represented by f(x, w), into a probability, the logistic function ((f(x, w))) was used instead of f(x, w).

$$\sigma(z) := \frac{1}{1+e^{-z}} = \frac{e^z}{1+e^z}$$

The output of the logistic function, f(x, w) can be used to define the probabilities of the input x, belonging to either class label zero or class label one.

$$\rho(1|x) := \sigma(f(x,w)),$$

$$\rho(0|x) := 1 - \sigma(f(x,w))$$

From this, it can be seen that

$$\rho(1|x) \geq 0, \rho(0|x) \geq 0$$

and

$$\rho(1|x) + \rho(0|x) = 1$$

Assuming a set of s samples, each represented by a pair (xi, yi), where the yi values are either 0 or 1 and are independently and identically distributed according to the probability density function previously defined, these samples can be used to train a model. According to the probability density, it is assumed that a set of samples s, which are represented by pairs $(x_i, y_i)_{i=1}^{s}$,where $y_i$ is either 0 or 1 and are identically distributed.The likelihood that corresponds to this situation can be expressed as follows.

$$\rho(y|X, w) = \prod_{i=1}^{s} \rho(y_i|x_i)$$

Further down, some examples are observed, using the Binary Logistic Regression, to find the best possible Classification Accuracy for the Training and Validation sets, together with box plots for the inputs removed to show how the outcome depends on the input.

### 3.1.1 Testing for the whole Data Set

<u>**Results:**</u>

$$\textbf{Optimal weights=}\begin{bmatrix} -0.85555271 & 0.51884788 & 1.16153491 & -0.24411194 & 0.02302342 \\ -0.08663452 & 0.74731685 & 0.32794261 & 0.14865107 & \end{bmatrix}$$

**Classification Accuracy:**

**Training set=** 78.50162866449512%
**Validation set=** 74.02597402597402%

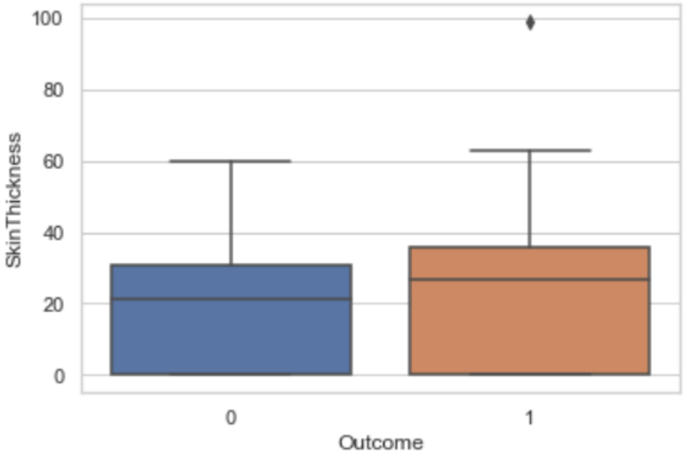
### 3.1.2 Testing without Skin Thickness

<u>**Results:**</u>

$$\textbf{Optimal weights=}\begin{bmatrix} -0.87274955 & 0.44896597 & 1.19997106 & -0.2446524 \\ -0.09638635 & 0.61210588 & 0.31123762 & 0.08856728 \end{bmatrix}$$

**Classification Accuracy:**

**Training set=** 77.52442996742671%
**Validation set=** 79.22077922077922%



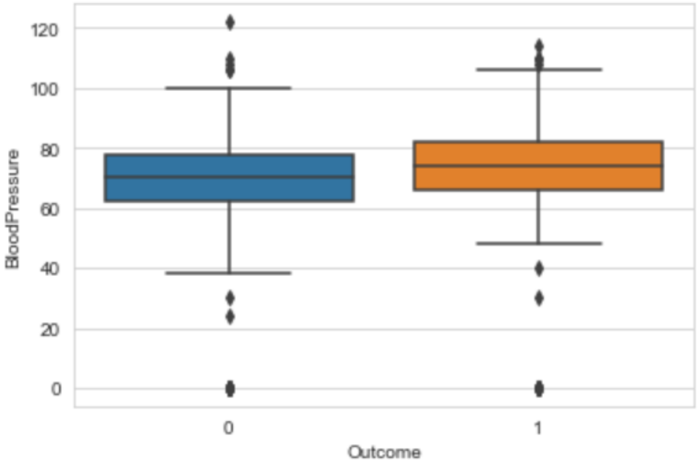
#### 3.1.3 Testing without Skin Thickness and Blood Pressure

<u>**Results:**</u>

$$\textbf{Optimal weights}= \begin{bmatrix} -0.86031326 & 0.45968383 & 1.01538612 & -0.18155923 \\ 0.67171425 & 0.26461609 & 0.09390444 & \end{bmatrix}$$

**Classification Accuracy:**

**Training set**= 77.0358306188925%
**Validation set**= 79.87012987012987%



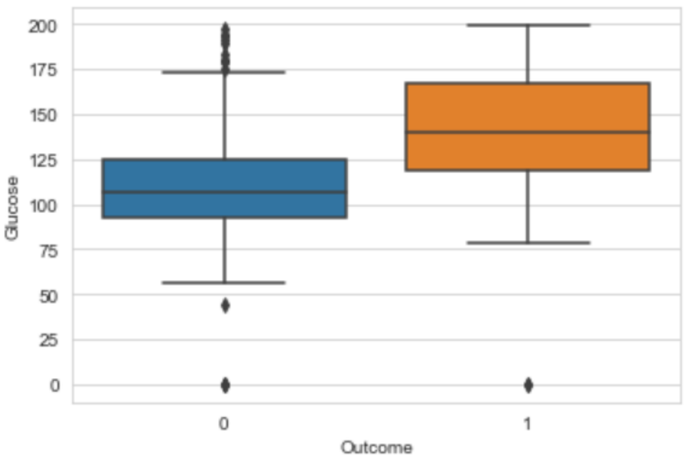
### 3.1.4 Testing without Glucose

<u>**Results:**</u>

$$\textbf{Optimal weights=}\begin{bmatrix}-0.78910151 & 0.33637148 & -0.17734861 & -0.16757339\\ 0.23755581 & 0.8026379 & 0.28298792 & 0.44622498\end{bmatrix}$$

**Classification Accuracy:**

**Training set=** 69.54397394136808%
**Validation set=** 70.77922077922078%



### 3.1.5 Comments:

Four different results were generated and three different box plots. In the first test, the whole data was used, where the classification accuracy for the validation set was 74.03% and 78.50% for the training set.
In the next test the skin thickness was removed. According to the results after the removal of skin thickness, the classification accuracy for the validation set has increased and this can also be observed from the box plot, which indicates that diabetes does not depend on skin thickness.
For the next test, two data inputs were removed, skin thickness and blood pressure. There is a slight increase in classification accuracy, having a result of 79.87% for the validation set.
However, the last test was done by removing the glucose input, that is seen to be one of the main components affected in diabetes. All of the results were observed to decrease, concluding that glucose is an essential input that does not need to be removed from the data set.

### 3.2 K-Nearest-Neighbors(KNN)

It is a non-parametric algorithm that does not make any assumptions about the underlying distribution of the data. Instead, a probability is assigned to the unknown output label based on the labels of the K nearest neighbors. After calculating the probabilities for all the class labels, the class label with the highest probability is assigned as the output of the classifier. Additionally, KNN is considered a "lazy learning" algorithm as it does not learn from the training data but memorizes all the examples.
The probability's form is:

$$\rho(y = c|x, K) := \frac{1}{K} \sum_{l \in N_K(x)} j(y_l = c),$$

with j being

$$j(z) := \begin{cases} 1 & \text{if z is true} \\ 0 & \text{if z is false} \end{cases} \tag{1}$$

$N_k$ denotes the neighbourhood of x, which includes the K nearest neighbours of x.After calculating the probabilities for all class labels, the class label with the highest probability is assigned as the output of the classifier, represented by f.

$$f(x) := argmax\rho(y = c|x, K)$$

Where c $\in$C$_0, C_1, ..., C_n$

This method was used and implemented in the data set diabetes.
The K was assigned at 5 and took a range of Knn from 1-28, as the data set consists of 768 rows of data.[28 was found by square rooting 768 (number of samples).]

#### 3.2.1 Testing for the whole Data Set

**Optimal number of neighbours:** 19
**Prediction Error:** 23.95%.

#### 3.2.2 Testing Without Skin Thickness

**Optimal number of neighbours:** 14
**Prediction Error:** 22.52%.

#### 3.2.3 Testing without Skin Thickness and Blood Pressure

**Optimal number of neighbours:** 21
**Prediction Error:** 22.53%

#### 3.2.4 Testing Without Glucose

**Optimal number of neighbours:** 7
**Prediction Error:** 28.91%

#### 3.2.5 Comments:

The lowest prediction error can be found when removing the input skin thickness, with a percentage of 22.52% for the Prediction Error.
As for the results when removing both skin thickness and blood pressure, it can be seen that the percentage prediction error is similar to the results generated by only removing skin thickness.
For the last test, when removing the input glucose, a higher prediction error was observed as well as a very low optimal number of neighbours.

## 4 Conclusion

In conclusion, based on the results generated by both methods, it can be concluded that a better classification accuracy is produced when using the Binary Logistic Regression. However, the models cannot be compared as they differ from each other. The most important difference is that Binary Logistic Regression is a parametric algorithm that makes assumptions about the data and uses a logistic function to model the probability of a binary outcome whereas, KNN is a non-parametric algorithm that does not make assumptions about the data and compares the feature values of an unknown sample to the nearest labeled samples in the training data set.

## 5 References:

Used Qmul lecture notes and Coursework 9 Solutions.

Smith, J.W. et al. (1988) Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, Proceedings of the Annual Symposium on Computer Application in Medical Care. U.S. National Library of Medicine. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2245318/