UDEM

Logistic Classification Report

Profesor: Andrés Hernández

520195 Santiago Cantú Gómez

Hora: 14:30 - 16:00

San Pedro Garza García, N.L. 20 de abril del 2020

Instructions:

Elaborate a report, which can be completed as a Jupyter Notebook or as a PDF file. This must include an introduction to logistic classification and diabetes prediction, the data exploration stage (visualise the histogram of each feature) and a short discussion on these plots, the building, training and testing of your model, as well as a discussion

What is Logistic classification?

Also known as logistic regression, logistic classification is a binary classification model in which the algorithm predicts only two different type of values, this values are 0 and 1, and depending on which value was predicted means the classification of the sample value. It also refers to the process of using the logistic function as hypothesis function instead of using the regression hypothesis function because the last one is not robust to outlier data, as they may modify the hypothesis function and produce missclassifications on the testing samples, in order to fix this error, the logistic function is used so all the algorithm possible values are between 0 and 1.

The logistic function is defined as:

$$L_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}} = P(y|\mathbf{x};\mathbf{w})$$

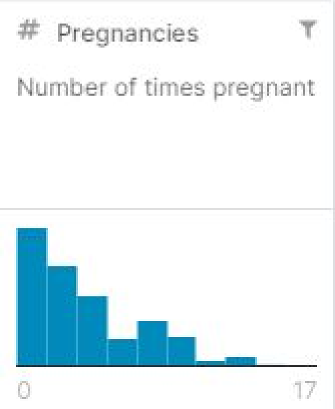
What is diabetes?

It is a disease that happens when the blood glucose or blood sugar is high on the body of a person, it is the main source of energy which comes from all the food that we eat everyday. Insulin is a hormone made by the pancreas organ, it helps all the glucose from the food to get into the human cells for energy, the problem with diabetes is that the pancreas doesn't produce enough insulin.

Diabetes prediction

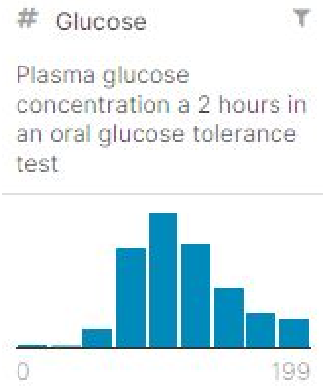
The National Institute of Diabetes and Digestive and Kidney Diseases in order to know if a person has diabetes or not, they follow a procedure of gathering data to evaluate the pacient, the data was taken from a csv file in https://www.kaggle.com/uciml/pima-indians-diabetes-database having a total of 768 patients, all the data that the institute need to collect are:

- Number of the times that the person has been pregnant



Less than the half of the females had never been pregnant and more than the half had been pregnant for at least 1 time.

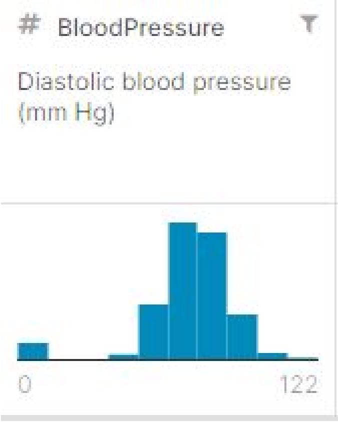
- The plasma glucose concentration a 2 hours in an oral glucose tolerance test



Plasma glucose concentration a 2 hours in an oral glucose tolerance test

Most of the females had an average of the plasma glucose, the maximum glucose concentration in two hours was 199.

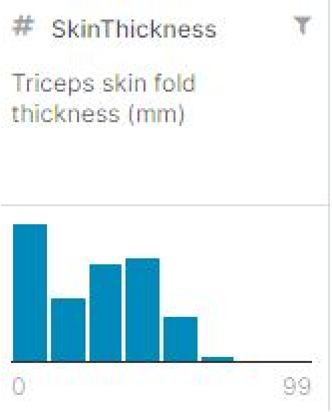
- The blood pressure



Diastolic blood pressure (mm Hg)

Most of the patients had an average blood pressure, few had very low blood pressure and few high, there was outline data.

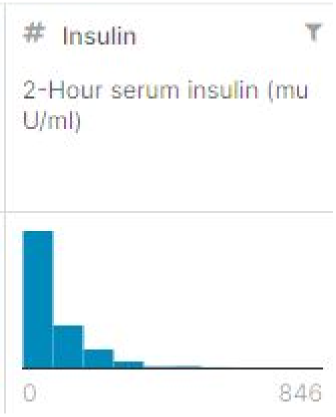
- The skin thickness of the triceps skinfold in mm



Triceps skin fold thickness (mm)

Almost half of the females had the triceps skin very thick and the other half didn´t.

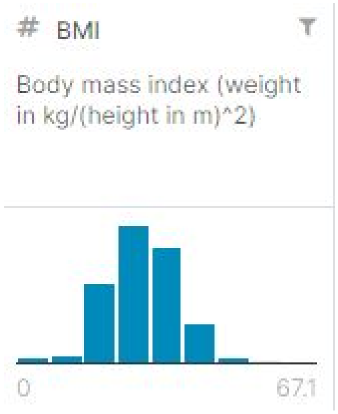
- How much insulin the pancreas produce in 2 hours



2-Hour serum insulin (mu U/ml)

Most of the females had a 0 production of insulin from the pancreas which means they had diabetes.

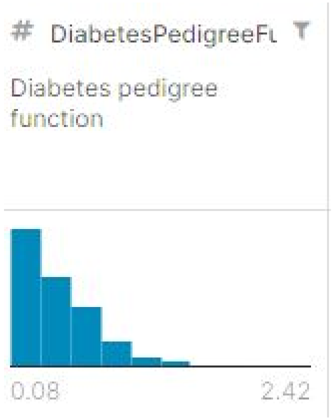
- The body mass index



Body mass index (weight in kg/(height in m)^2)

Most of the females had an average BMI, some of them were outline data because it is not possible to have 0 BMI, and some had a very high BMI.

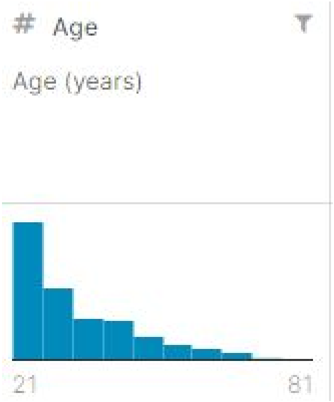
- The diabetes pedigree function



Diabetes pedigree function

The values of the diabetes pedigree function stands between 1 and 0.08 for most cases.

- The age of the patient



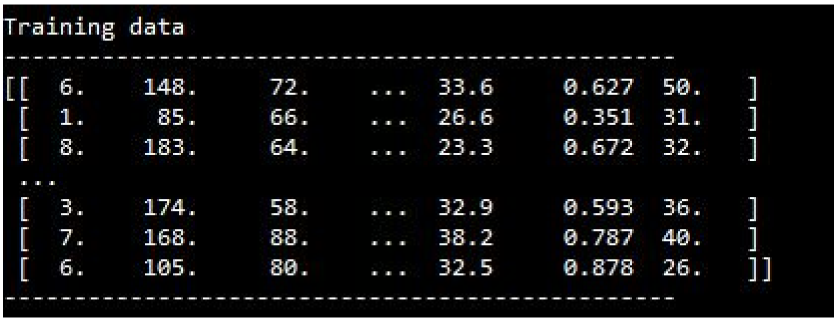
Most of the females had 21 years and less than 30.

After having all the data, they classify the person as a person with diabetes with a 1 or a person without diabetes with a 0.

Training and testing the model

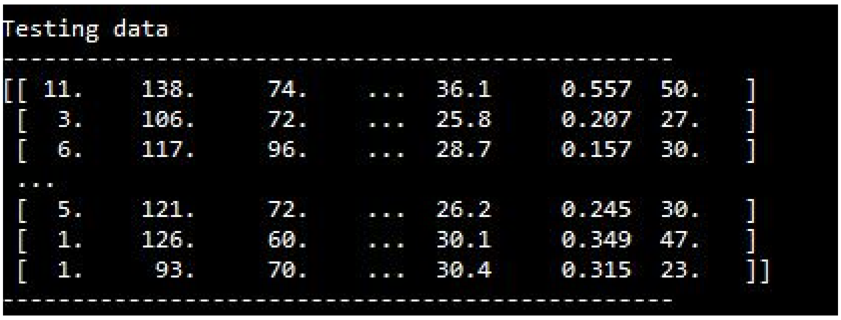
The first step was to load all the data from https://www.kaggle.com/uciml/pima-indians-diabetes-database, after having all the data in the program, it was necessary to separate the data into training samples and testing samples, 80% of the data was used for training and the other 20% was used to test the trained algorithm.

The data of 614 patients was used to train the algorithm.



```
Training data
-------------------------------------------------
[[  6.    148.     72.    ...  33.6     0.627  50.    ]
 [  1.     85.     66.    ...  26.6     0.351  31.    ]
 [  8.    183.     64.    ...  23.3     0.672  32.    ]
 ...
 [  3.    174.     58.    ...  32.9     0.593  36.    ]
 [  7.    168.     88.    ...  38.2     0.787  40.    ]
 [  6.    105.     80.    ...  32.5     0.878  26.    ]]
-------------------------------------------------
```

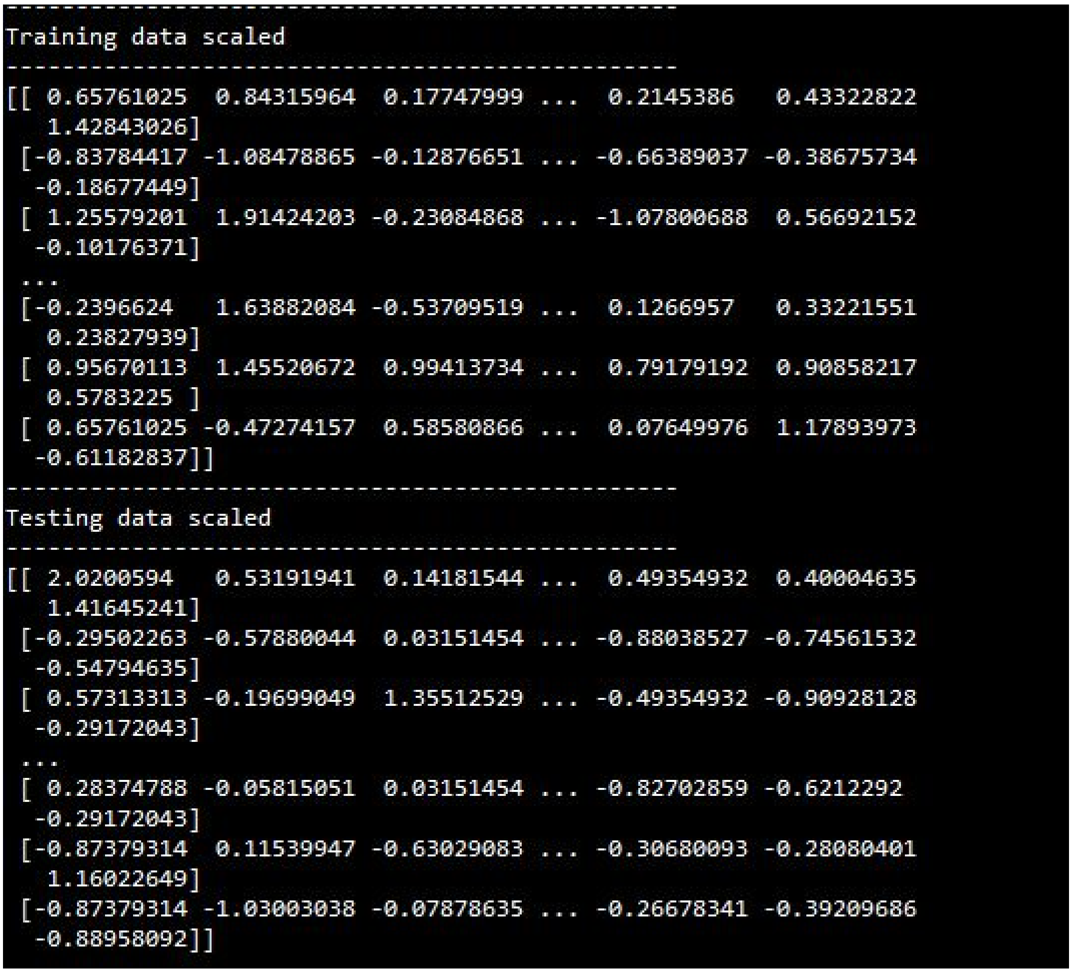
The data of 154 patients was used to test the trained algorithm



```
Testing data
-------------------------------------------------
[[ 11.    138.     74.    ...  36.1     0.557  50.    ]
 [  3.    106.     72.    ...  25.8     0.207  27.    ]
 [  6.    117.     96.    ...  28.7     0.157  30.    ]
 ...
 [  5.    121.     72.    ...  26.2     0.245  30.    ]
 [  1.    126.     60.    ...  30.1     0.349  47.    ]
 [  1.     93.     70.    ...  30.4     0.315  23.    ]]
-------------------------------------------------
```

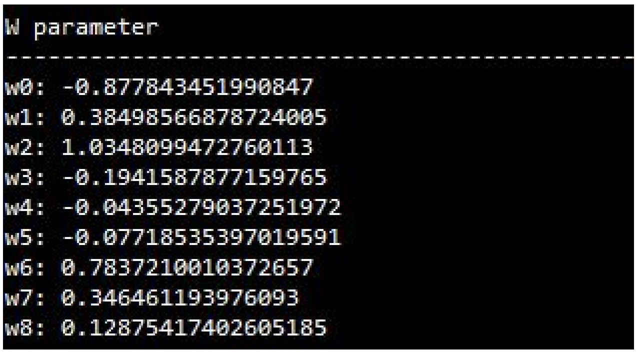
Having the 80% of the data for training, the next step was to train our algorithm using the cost function and the gradient descent, the learning rate was set to 0.0005 and the stopping criteria to 0.01. In order to make the training faster and avoid the diverge of the algorithm, the training data features were escalated using the training data mean and standard deviation. Once having the trained algorithm and all the optimal values for every feature of the data (w's), the next step was to use this data and apply the logistic function as the hypothesis function in order to predict if the patient had or not diabetes.

The training and testing data scaled in order to converge the algorithm and avoid diverge



```
---------------------------------------------
Training data scaled
---------------------------------------------
[[ 0.65761025  0.84315964  0.17747999 ...  0.2145386   0.43322822
   1.42843026]
 [-0.83784417 -1.08478865 -0.12876651 ... -0.66389037 -0.38675734
  -0.18677449]
 [ 1.25579201  1.91424203 -0.23084868 ... -1.07800688  0.56692152
  -0.10176371]
 ...
 [-0.2396624   1.63882084 -0.53709519 ...  0.1266957   0.33221551
   0.23827939]
 [ 0.95670113  1.45520672  0.99413734 ...  0.79179192  0.90858217
   0.5783225 ]
 [ 0.65761025 -0.47274157  0.58580866 ...  0.07649976  1.17893973
  -0.61182837]]
---------------------------------------------
Testing data scaled
---------------------------------------------
[[ 2.0200594   0.53191941  0.14181544 ...  0.49354932  0.40004635
   1.41645241]
 [-0.29502263 -0.57880044  0.03151454 ... -0.88038527 -0.74561532
  -0.54794635]
 [ 0.57313313 -0.19699049  1.35512529 ... -0.49354932 -0.90928128
  -0.29172043]
 ...
 [ 0.28374788 -0.05815051  0.03151454 ... -0.82702859 -0.6212292
  -0.29172043]
 [-0.87379314  0.11539947 -0.63029083 ... -0.30680093 -0.28080401
   1.16022649]
 [-0.87379314 -1.03003038 -0.07878635 ... -0.26678341 -0.39209686
  -0.88958092]]
```

The optimal feature values calculated with the cost function and gradient descent of the training data

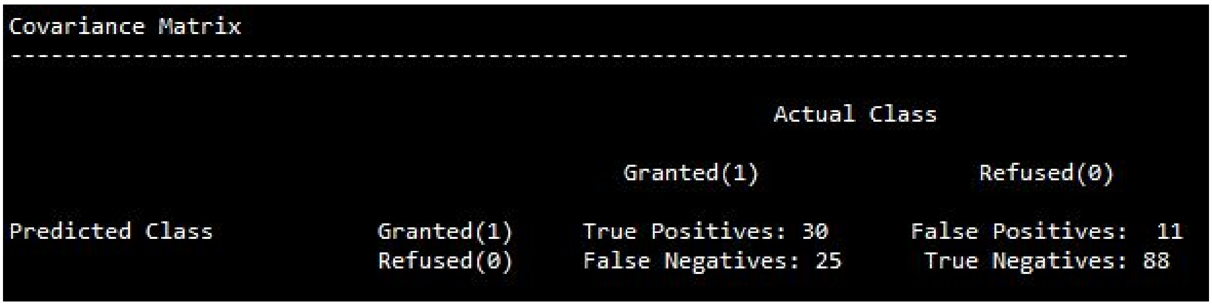


```
W parameter
-------------------------------------------
w0: -0.877843451990847
w1: 0.38498566878724005
w2: 1.0348099472760113
w3: -0.1941587877159765
w4: -0.04355279037251972
w5: -0.07718535397019591
w6: 0.7837210010372657
w7: 0.346461193976093
w8: 0.12875417402605185
```

Before using the logistic function it was necessary to escalate also the testing data using the mean and standard deviation of the training data because the training data was already escalated. With all the predicted values from the hypothesis function, the next step is to classify them into a patient with diabetes (1) and a patient without diabetes (0), in order to make the classification, all the values of the predicted hypothesis were compared, if they were less than 0, the patient was classified as a patient without diabetes, if the value was 0 or more, the patient was classified as a patient with diabetes.

Having all the data classified, the next step was to make a confusion matrix in order to calculate the accuracy, precision, recall, specificity and f1_score of the data, this calculations were made using the classification made by the National Institute of Diabetes and Digestive and Kidney Diseases and the values classified by the algorithm trained before.

Confusion matrix



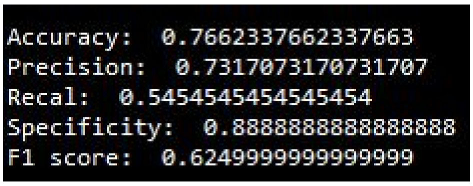
```
Covariance Matrix
---------------------------------------------------------------------------------

                                                        Actual Class

                                              Granted(1)              Refused(0)

Predicted Class             Granted(1)     True Positives: 30      False Positives:  11
                            Refused(0)     False Negatives: 25      True Negatives: 88
```

Calculations done with the values of the confusion matrix



```
Accuracy:  0.7662337662337663
Precision:  0.7317073170731707
Recal:  0.5454545454545454
Specificity:  0.8888888888888888
F1 score:  0.6249999999999999
```

After comparing the predicted values from the actual values in the confusion matrix, the algorithm had a 76.62% of accuracy, 73.17 of precision, a recall of 54.54%, a specificity of 88.88% and a f1_score of 62.49%, this means that the trained algorithm that we used had three fourths of the patients classified correctly and the others were classified incorrectly. This doesn't mean that the algorithm is wrong, it can be better but also we don’t know if all the predictions made by the National Institute of Diabetes were done correctly, so it is probable that the algorithm is working very good.

I declare that I have done this activity following the code of honor of the UDEM.