UNIVERSITY OF LIVERPOOL

COMP 534 – Applied Artificial Intelligence

Assignment 1 - Performance analysis of multiple supervised learning methods for solving a binary classification problem.

Submitted by,

Name	Student Id	Email
Alwin Joseph Christopher	201594340	sgachri4@liverpool.ac.uk
Mohamed Muradh Maricair	201602133	sgmkader@liverpool.ac.uk

Introduction:

We have developed various classification models for diabetes data using our AI knowledge in this assignment. We need to classify the patient as diabetes or non-diabetes based on specific categorical features.

We used below libraries to realise the supervised learning models.

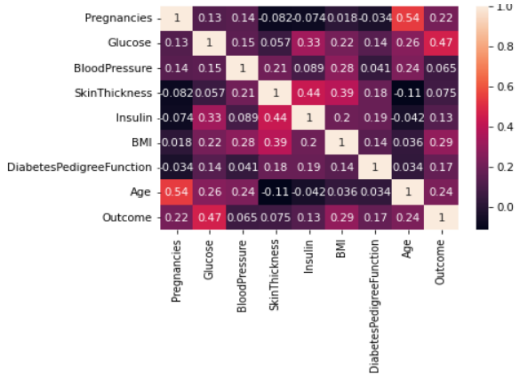
- Scikit-learn
- Pandas
- NumPy
- Matplotlib
- Seaborn
- Random

We have decided to use the below models as we understand the working of each model in detail along with various hyperparameters.

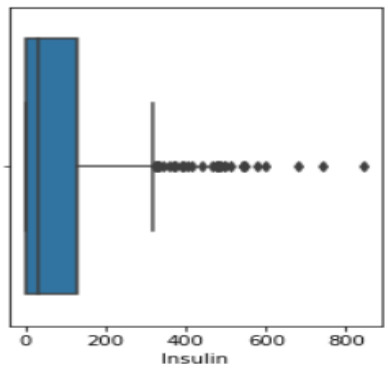
- Decision Tree
- Logistic Regression
- K-Nearest Neighbors

Data Visualization:

Before pre-processing the data, we did a pair plot using seaborn library to visualize correlation matrix. Based on this correlation plot, we identified that Glucose and BMI feature has a more significant impact on determining the Outcome results. Therefore, we are using this feature to plot visualisation for the Logistic Regression model.



Boxplot has been plotted for all features to understand each feature's values range. Also, It helps us to use the mean or median method to fill the Zero-value data. E.g., 50% percentile data is closer to the Q1 Quartile in the Insulin feature. Therefore, we are trying to replace the Zero values with median values. We used a similar technique to replace mean or median values for each feature.



Data Pre-processing:

- Pregnancies: Value greater than 8 is abnormal, so we assigned all the values greater than eight as 8
- Age: We clubbed the age value into five different bins. Age less than 30 is considered as one and value between 30 and 40 is regarded as two etc. similarly, the age greater than 60 is considered as 5
- Insulin: For the Zero values, we try to fill with a different range of values, but the measurement of the levels is based on the Blood test time taken. Therefore, we filled with median values based on the box plot. [1]
- Zero values in Blood Pressure, Skin Thickness, Glucose and BMI features have been replaced with its mean value.
- Data has been split into 75: 25 ratio and Cross-validation is not used for the model
- Xtrain and Xtest values have been standardised for Knn and Logistic Regression models

Grid Search CV:

- Applied Grid search cv for all three models with various parameters. The below table shows the best accuracy and best parameter for each model:

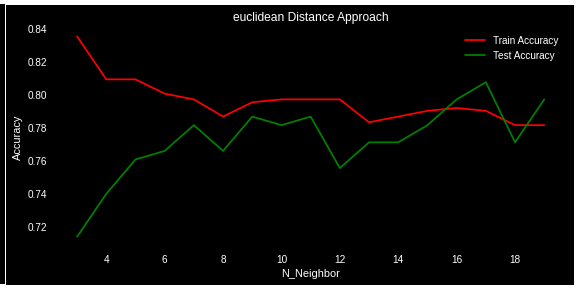
	Logistic Regression	KNeighbors Classifier	Decision Tree Classifier
Best Accuracy	76.57%	77.26%	73.96%
Best Parameters	{'C': 0.25, 'random_state': 42}	{'metric': 'manhattan', 'n_neighbors': 12, 'weights': 'distance'}	{'criterion': 'entropy', 'max_depth': 3}

- Alternatively, we observed the accuracy improved if we used different parameters instead of the best parameters suggested from Grid Search CV.

Model Implementation:

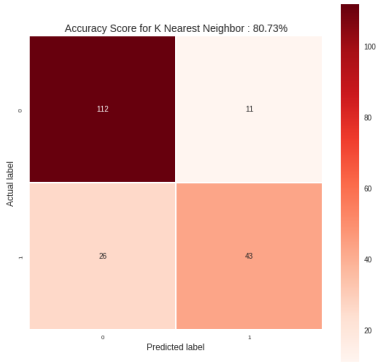
KNN Neighbours:

We plotted the n_neighbor vs accuracy graph to find the best n_neighbor value for Manhattan and Euclidean distance metrics. We observed that n_neighbor = 17 in the Euclidean distance metric is the best parameter for the KNN model.



Metrics:

- Euclidean distance at n_neighbor = 13
  - Train accuracy = 0.79
  - Test accuracy = 0.80
- F1 Score: 69.92
- Recall: 62.32
- Precision: 79.63
- Accuracy: 80.73%



Based on the Confusion matrix results, we can conclude that model has been performing well as its neither overfitting nor underfitting. Since it's a diabetics dataset, we cannot afford to have a high number of false-negative classes. Out of 192 test samples, only 11 are classified as false negatives.
Also, we got the high value of precision which explains that 79% of our positive predicted values are positive in actual data.

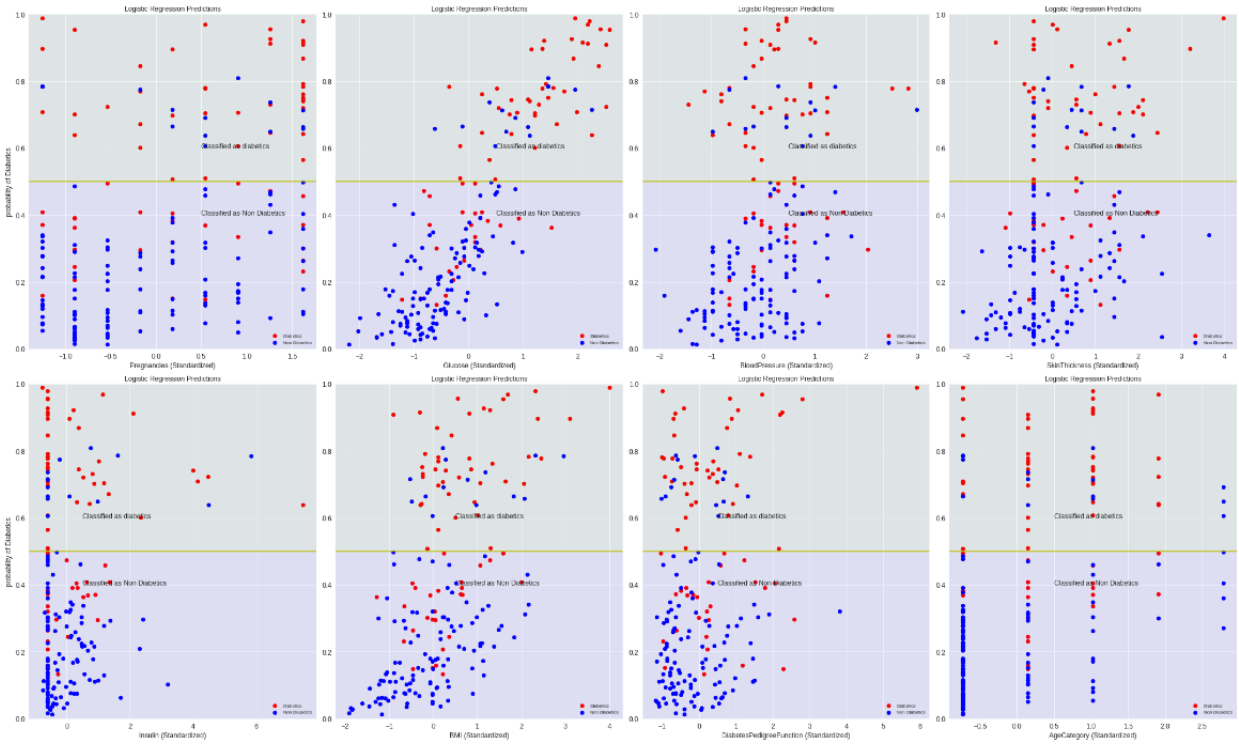
Logistic Regression:

we have implemented the logistic regression model in two ways. our first model is trained with all the features, and the second model is trained with only two features as it enables us to visualise the plot for our model. features have been selected based on the correlation plot.

Model trained with all features:

The Accuracy obtained for this approach is 78.65%, and we got a decent F1_score of 67.2. F1_score gives us the harmonic mean of precision and recall and denotes the quality of our model prediction.

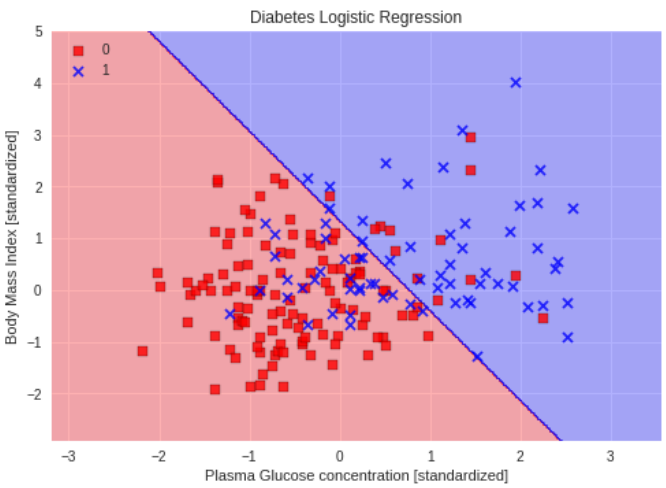
In the below feature vs Probability Plot, we tried to observe the misclassification and linearity of the feature for our model fit. It is evident that Glucose is highly correlated with Outcome, and it is linearly separable.

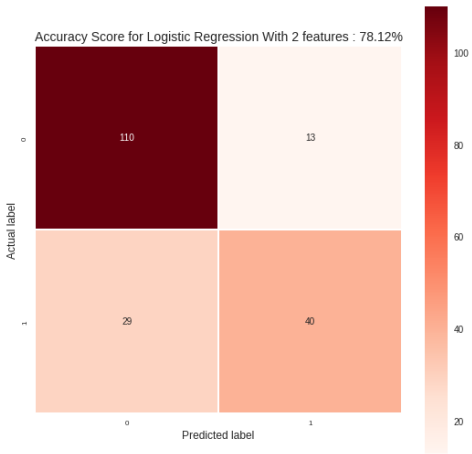


Model with Two features:

Logistic regression plot shows the classification of Outcome and its boundary linearly separating the two classes. Glucose and BMI data have been used to plot this model.

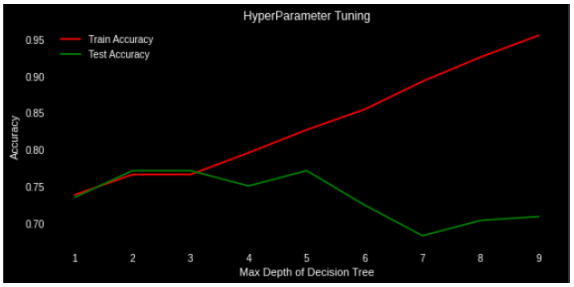
The confusion matrix shows a good result for the model, and the accuracy obtained is 78.12%



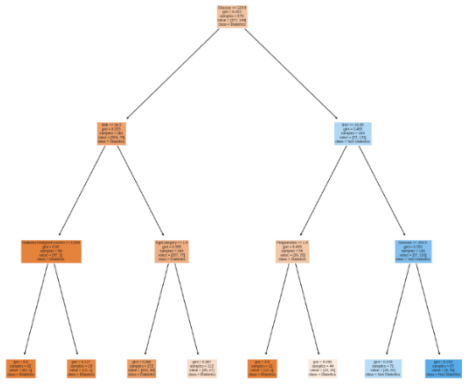


Decision Tree:

We tried to plot max_depth vs Accuracy to find the best value for this parameter to run the model. Train and test Accuracy is suitable for max depth of 2 and 3, beyond which model becomes overfitting, which reduces the accuracy of testing set.



Based on the tree graph, it is evident that only 2 leaf nodes classify the data as non-diabetic, whereas 6 leaf nodes classify the data as diabetic. Two pure nodes are organising the data as Diabetic as the Gini impurity value is zero.

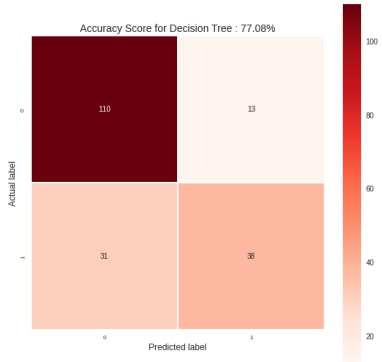


The decision tree model identifies Glucose as the most crucial feature through Information Gain, and the first node is split based on this feature. Then nodes are divided based on the BMI feature as the model identifies it as the next important feature.

The accuracy for the Decision tree model is 77.08%, and it obtained an F1_score of 63.33%.

The table below shows each feature's importance for our decision tree model, sorted in descending order.

Index	Feature	Importance
1	Glucose	0.6
5	BMI	0.254
7	AgeCategory	0.098
0	Pregnancies	0.047
6	DiabetesPedigreeFunction	0.001
2	BloodPressure	0.0
3	SkinThickness	0.0
4	Insulin	0.0



Result:

Model	Accuracy %	Precision %	Recall %	F1 Sccore %
Decision Tree	77.08	74.5	55.07	63.33
K Nearest Neighbor	80.72	79.63	62.32	69.92
Logistic Regression	78.65	75	60.87	67.2
Logistic Regression with 2 features	78.12	75.4	57.97	65.57

We can see from the table above that KNN performed reasonably well with this dataset. Data were pre-processed and re-scaled with a standard scaler, which normalised the data with a mean of -0 and a standard deviation of -2. We used 17 neighbours to determine an observation class. Predictions for new instances are made by searching through the entire training data set for the k most similar instances (neighbours) and summarising the output variable for those k instances using the Euclidean measure of distance. We achieved 78.99 percent training accuracy and 80.72 percent test accuracy even with 8-dimensional features. Decision tree, which is the foundation of machine learning models and predicts the class based on a series of conditional statements, used only a few parameters to make its prediction, so it performed worse than KNN. Logistic Regression, on the other hand, is simpler to implement, interpret, and train, but it assumes linearity between the dependent and independent variables. Finding linear separation in an 8-dimensional data mix of independent and dependent variables may be difficult, so it performed less well than KNN.

Conclusion:

Problems encountered:

- Examining a large set of Diabetics data to determine the significance of each feature in predicting the outcome
- Data cleaning entails removing or replacing zero-value rows with mean or median values.
- Selecting hyperparameters for each Supervised Machine Learning process

Project Task Allocation:

We divided the tasks evenly amongst ourselves and used Google Collab for scripting. We discussed and implemented various data pre-processing ideas. Muradh was in charge of the Logistic Regression model and report creation. Alwin wrote the scripts for the decision tree and the KNN model.