PERFORMANCE ANALYSIS OF MULTIPLE SUPERVISED LEARNING METHODS FOR SOLVING A BINARY CLASSIFICATION PROBLEM

Rahul Kumar

University of Liverpool, Liverpool

Table of Contents

Abstract/Executive Summary

This report aims to investigate whether a patient has diabetes or not. The dataset contains 9 different features like age of the patient, existing medical conditions and their BMI. Overall, the result indicates that people with pre-medical conditions like high blood pressure, Insulin, Glucose levels and BMI are more vulnerable to diabetes. Report concludes that age, high Insulin level and Glucose levels contributes largely causing diabetes. Looking at the pattern, it is recommended that diabetes can be prevented by eating healthy, get more physical activity and by losing excess pounds. Additionally, Type 1 diabetes can't be prevented. However, the same healthy lifestyle choices that help treat prediabetes, type 2 diabetes and gestational diabetes can also help prevent.

Introduction

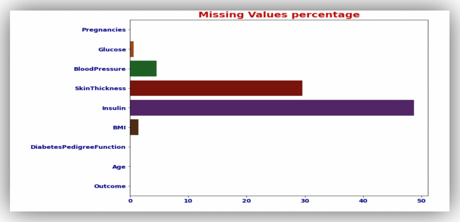
Diabetes mellitus refers to a group of diseases that affect how your body uses blood sugar (glucose). Glucose is vital to your health because it's an important source of energy for the cells that make up your muscles and tissues. It's also your brain's main source of fuel. It is important to know what are most possible reasons that can increase risk of diabetes and what can be done to lower the risk. Some of these factors cannot be controlled like age, Family history, Environmental Factors and Race and ethnicity. The risk of having diabetes increases if a parent or sibling has type 1 diabetes. On the other, there are other factors can be controlled to reduce the chance of having diabetes. This includes controlling lifestyle changes - such as eating healthy food, losing excess pounds and get more physical activities.

This report will discuss various risk factors contributing to diabetes and how this analysis can be used to make predictions and informed decisions.

Data is analysed by first importing dataset using Pandas and then applying checks for null values and white spaces to make sure that data is clean. Continuous numerical variables are explored using histogram chart while categorical variables are explored using count plot using matplotlib and seaborn. After that, we explored the relationship and corelation between various variables using boxplot and heatmap. After that we did feature engineering to analyse important variables/features and their impact on the target. Supervised learning techniques such as K- Nearest Neighbour Classifier, Decision Tree Classifier and Logistic Regression have been applied for data modelling while Hyper Parametric method is used for improving the performance of the model. We used different packages of sklearn to train, predict, plot the confusion matrix and classification report. At last, we compared the accuracy of classification methods. The given dataset is split into 70 and 30 ratios for training and testing respectively.

Data Cleaning

There is no missing value as such, but we have value = 0 for some variable where 0 is not possible, hence we will consider them as missing values and impute them with mean values. Columns that cannot have 0 as their value (and include 0): Glucose, BloodPressure, SkinThickness, Insulin, BMI. We will replace 0 with NaN values and then we will analysis the overall percentage of missing value in each feature.



Insulin has highest missing value percentage (around 49 %), SkinThickness and BloodPressure around 5 % missing values respectively and BMI and Glucose has around 2 % and around 1 % missing values respectively.

Data Exploration

We used box plot and heatmap to find the relationship between different features and target.
Below are our observations:

Pregnancies vs Outcome

1. Pregnant Nondiabetic and pregnant diabetic patients have the same Median.
2. Nondiabetic patients have more variance than diabetic patients.
3. Outliers are present in both groups, but non-diabetic patients is more skewed.

Glucose vs Outcome

1. Nondiabetic patient Glucose level have more spread when compared to Diabetic Patient glucose level.
2. Both have outliers but Nondiabetic patient Glucose level have a larger spread of outliers.
3. Both have same Median, but the Inter Quartile Range (IQR) of Diabetic patients is little higher than non-diabetic patients.

Blood Pressure vs Outcome

1. Blood Pressure of Diabetic Patients and Nondiabetic patients have the same median and IQR.
2. They both have outliers and approximately the same spread.
3. Patients who have Diabetes seem to have extreme low blood pressure.

SkinThickness vs Outcome

1. SkinThickness of Diabetic Patients and Nondiabetic patients have same median, however, IQR of Non-diabetic patients is a little higher than diabetic patients.
2. There are outliers in both groups, however, Diabetic Patient seem to have a very high value of SkinThikness.

Insulin vs Outcome

1. Insulin value in non-Diabetic patients has high variance compared to Insulin value in Diabetic Patients.
2. Non-Diabetic patients has a slightly higher insulin median and a lower IQR than Diabetic patients.
3. There are more outliers in non-diabetic patients than in diabetic patients.

BMI vs Outcome

1. BMI of nondiabetic patients has the same median and a slightly higher IQR than diabetic patients.
2. There are outliers present in both, however, non-diabetic patients have more extreme outliers.

DiabetesPedigreeFunction vs Outcome

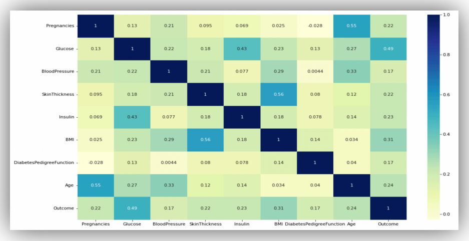
1. DiabetesPedigreeFunction in non-diabetic patients are almost similar in terms of median and IQR.
2. Outliers are present in both, but non-diabetic patients have more outliers and variance.

Age vs Outcome

1. Age of Diabetic patients has higher median.
2. Diabetic patients age has more outliers, higher IQR and variance which shows that diabetes is observed in both young and old people.

Outlier Treatment

- Using Capping and Flooring method to replace outliers.



Outcome (Target Variable) doesn't have much correlation with other variables.

- Pregnancies and Age are correlated.
- Skin Thickness and BMI are highly correlated.
- Glucose and Insulin are highly correlated.

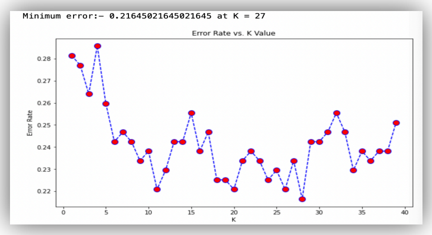
Data Modelling

Used below Classification model for data modelling because classifier method works on both continuous and categorical data and “diabetes” data set contains continuous as well as categorical numerical data.

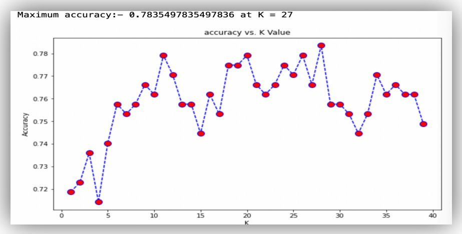
K-NEAREST NEIGHBOR CLASSIFIER

K-Nearest Neighbor classifier is a simple supervised classification algorithm. KNN does not make any assumptions on the data distribution, hence it is non-parametric.

- We have selected 8([' 'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin','BMI', 'DiabetesPedigreeFunction', 'Age']) as data features into the X data-frame and target field ([‘DEATH_EVENT’] into a y data-frame.
- After splitting the data, we take 70% data for training and remaining 30% for testing purposes.
- Then we plotted two graphs first one as K value vs Error rate and second as K value vs accuracy to find out the optimal value of K for which KNN model prediction accuracy is highest.

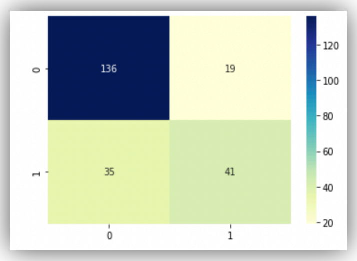


```
Minimum error:- 0.21645021645021645 at K = 27
```



```
Maximum accuracy:- 0.7835497835497836 at K = 27
```

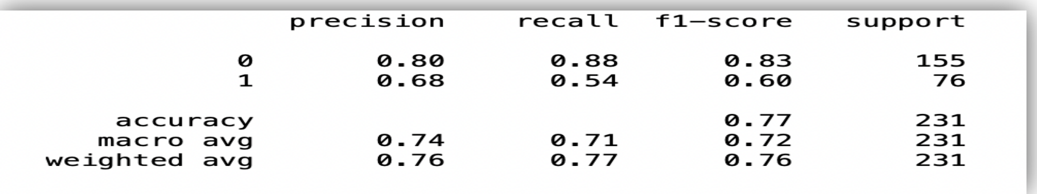
- We import the classifier model from the sklearn library and fit the model and achieved an accuracy of 77%.



- We used confusion matrix to measure the performance of the model. Which is show here:
- From confusion matrix, we can state that the 136 times model predicted True positive and 41 times it predicted True Negative. We

can also state that 19 times model predicted False Negative and 35 times it predicted False Positive.

- We used classification report to measure the quality of predictions from our KNN model.



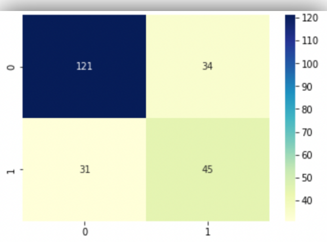
```
              precision    recall  f1-score   support

           0       0.80      0.88      0.83       155
           1       0.68      0.54      0.60        76

    accuracy                           0.77       231
   macro avg       0.74      0.71      0.72       231
weighted avg       0.76      0.77      0.76       231
```

DECISION TREE

A type of supervised learning algorithm that is mostly used in classification problems. It works for both categorical and continuous input and output variables. It splits the population or sample into two or more homogeneous sets (or sub-populations), based on most significant splitter / differentiator in input variables.

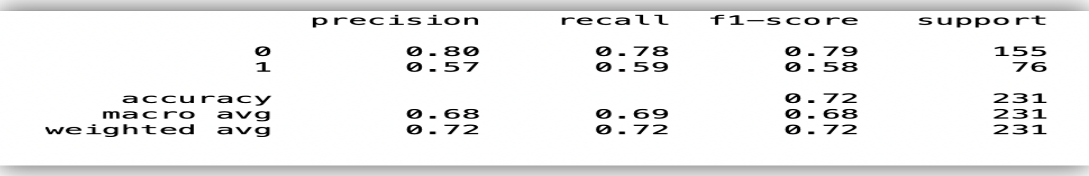
- We have selected 8([' 'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin','BMI', 'DiabetesPedigreeFunction', 'Age']) as data features into the X data-frame and target field ([‘DEATH_EVENT’] into a y data-frame.
- After splitting the data, we take 70% data for training and remaining 30% for testing purposes.

- We import the classifier model from the sklearn library and fit the model and achieved an accuracy of 72%.



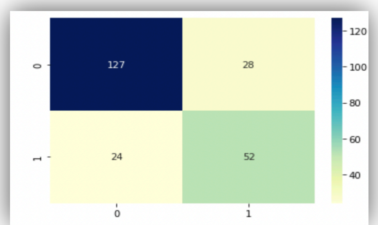
- We used confusion matrix to measure the performance of the model. Which is show here:
- From confusion matrix, we can state that the 121 times model predicted True positive and 45 times it predicted True Negative. We can also state that 34 times model predicted False Negative and 31 times it predicted False Positive.

- We used classification report to measure the quality of predictions from our KNN model.

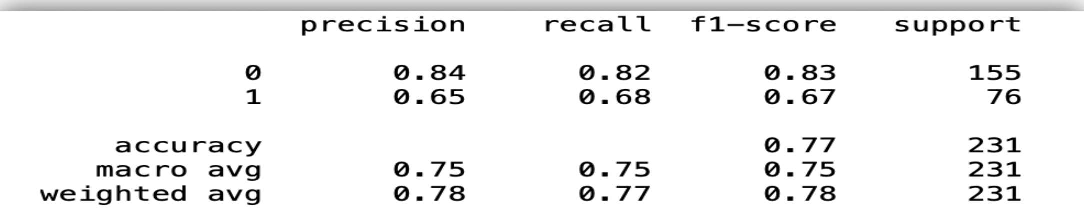


```
              precision    recall  f1-score   support

           0       0.80      0.78      0.79       155
           1       0.57      0.59      0.58        76

    accuracy                           0.72       231
   macro avg       0.68      0.69      0.68       231
weighted avg       0.72      0.72      0.72       231
```

- We optimized your model by changing the max_leaf_nodes value to get maximum accuracy.
- After optimizing the model, we achieved an accuracy of 77%.

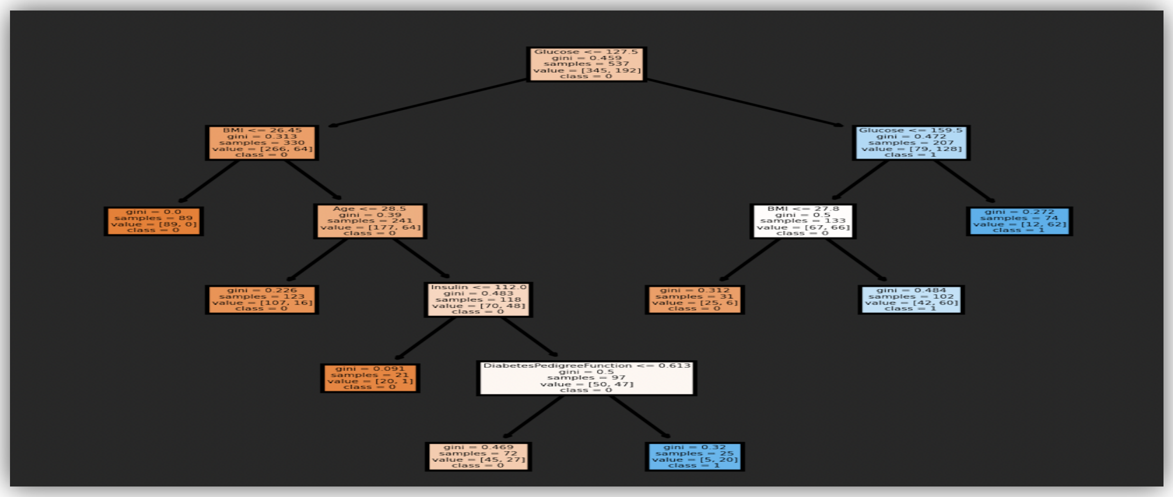


- Again, used confusion matrix to measure the performance of the model.
- From confusion matrix, we can state that the 127 times model predicted True positive and 52 times it predicted True Negative. We can also state that 28 times model predicted False Negative and 24 times it predicted False Positive.
- We used classification report to measure the quality of predictions from our decision tree model.



```
              precision    recall  f1-score   support

           0       0.84      0.82      0.83       155
           1       0.65      0.68      0.67        76

    accuracy                           0.77       231
   macro avg       0.75      0.75      0.75       231
weighted avg       0.78      0.77      0.78       231
```

TREE GRAPH

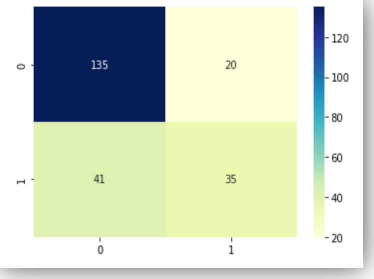


- Above we can see the tree built after training. During the training phase, the Decision Tree add nodes, split them into branches that lead to leaves.

LOGISTIC REGRESSION

Logistic regression is a process of modelling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.

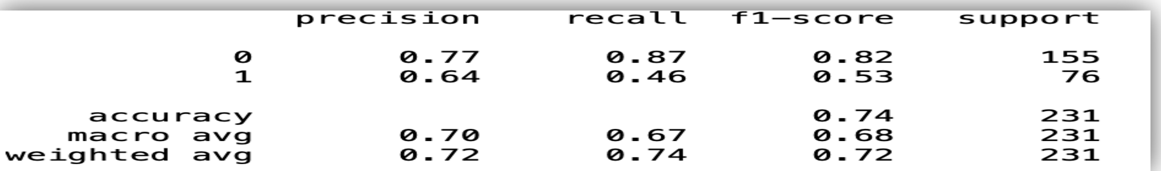
- We have selected 8([' 'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin','BMI', 'DiabetesPedigreeFunction', 'Age']) as data features into the X data-frame and target field ([‘DEATH_EVENT’] into a y data-frame.
- After splitting the data, we take 70% data for training and remaining 30% for testing purposes.
- We import the classifier model from the sklearn library and fit the model and achieved an accuracy of 74%.



- We used confusion matrix to measure the performance of the model. Which is shown here:

- From confusion matrix, we can state that the 135 times model predicted True positive and 35 times it predicted True Negative. We can also state that 20 times model predicted False Negative and 41 times it predicted False Positive.

- We used classification report to measure the quality of predictions from our logistic Regression model.



```
              precision    recall  f1-score   support

           0       0.77      0.87      0.82       155
           1       0.64      0.46      0.53        76

    accuracy                           0.74       231
   macro avg       0.70      0.67      0.68       231
weighted avg       0.72      0.74      0.72       231
```

EVALUATION

Please find below table that compares precision, recall, f1-score and accuracy of all three models:

S. No.	Precision	Recall	f1-score	Accuracy
K-Nearest Neighbour classifier	0.80	0.88	0.83	0.77
Decision Tree	0.84	0.82	0.83	0.77
Logistic Regression	0.77	0.87	0.82	0.74

KNN is comparatively slower than Logistic Regression. KNN supports non-linear solutions where LR supports only linear solutions. LR can derive confidence level (about its prediction), whereas KNN can only output the labels. Decision trees simplify such relationships. A logistic regression can, with appropriate feature engineering, better account for such a relationship. A second limitation of a decision tree is that it is very expensive in terms of sample size.

Conclusion

Type 1 diabetes can't be prevented. However, the same healthy lifestyle choices that help treat prediabetes, type 2 diabetes and gestational diabetes can also help prevent them:

Eat healthy foods. Choose foods lower in fat and calories and higher in fibber. Focus on fruits, vegetables and whole grains. Strive for variety to prevent boredom.

Get more physical activity. Aim for about 30 minutes of moderate aerobic activity on most days of the week, or at least 150 minutes of moderate aerobic activity a week.

Lose excess pounds. If you're overweight, losing even 7% of your body weight — for example, 14 pounds (6.4 kilograms) if you weigh 200 pounds (90.7 kilograms) — can reduce the risk of diabetes.

References

- http://pandas.pydata.org/pandas-docs/
- https://www.who.int/news-room/fact-sheets/detail/diabetes
- https://www.mayoclinic.org/diseases-conditions/diabetes/symptoms-causes/syc-20371444
- https://www.kaggle.com/uciml/pima-indians-diabetes-database
- https://seaborn.pydata.org/generated/seaborn.heatmap.html
- https://seaborn.pydata.org/generated/seaborn.histplot.html
- https://seaborn.pydata.org/generated/seaborn.countplot.html
- https://seaborn.pydata.org/tutorial/categorical.html
- https://seaborn.pydata.org/generated/seaborn.lineplot.html
- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html