Prediction of Diabetes using Machine Learning Algorithms

Description:

Diabetes is considered as one of the deadliest and chronic diseases which causes an increase in blood sugar level. There are many complications, if diabetes remains untreated and unidentified. Sometimes it is tedious in identifying if the person is having diabetes or not. This critical problem is solved by the rise in machine learning approaches. The motive of this Project is to design a model which can prognosticate the likelihood of diabetes in patients with maximum accuracy. Therefore, two machine learning classification algorithms namely Decision Tree, and K- Nearest Neighbors are used in this experiment to detect diabetes at an early stage. Experiments are performed on Pima Indians Diabetes Database (PIDD) which is sourced from from NIDDK. The performances of these two algorithms are evaluated on various measures like Precision, Accuracy, F-Measure, and Recall. Results obtained are verified using Receiver Operating Characteristic (ROC) curves in a proper and systematic manner and The Project also generalizes the selection of optimal features from dataset to improve the classification accuracy.

About Dataset:

The datasets consist 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 more. This dataset Consists of 8 attributes and 768 instances.

The attributes in the dataset are as follows:

- **Pregnancies**: Number of times pregnant
- **Glucose**: Plasma glucose concentration, 2 hours in an oral glucose tolerance test
- **BloodPressure**: Diastolic blood pressure (mm Hg)
- **SkinThickness**: Triceps skin fold thickness (mm)
- **Insulin**: 2-Hour serum insulin (mu U/ml)
- **BMI**: Body mass index (weight in kg/(height in m)^2)
- **DiabetesPedigreeFunction**: Diabetes pedigree function
- **Age**: Age (years)
- **Outcome**: Class variable (0 or 1)

Loading packages:

First, we are loading required packages for performing our descriptive analysis, Building models like Decision tree and K-NN.

Performing Exploratory Data Analysis

In the Dataset, some columns consist of zero, which does not make sense and thus indicates missing value. Hence first we need to replace those Zeroes in the dataset, then only the accuracy of our project will be better. By using the method FFILL we replace those missing values.

Ffill:

It is a function which is used to fill the missing value in the dataset. 'ffill' stands for 'forward fill' and will propagate last valid observation forward.

After replacing the missing values, we are visualizing all the attributes using histograms.

Understanding relationship between pair of variables:

This is done with the help of pair plots.

Pair Plots:

Pair plot is used to understand the best set of features to explain a relationship between two variables or to form the most separated clusters. It also helps to form some simple classification models by drawing some simple lines or make linear separation in our dataset. A "pairs plot" is also known as a scatterplot, in which one variable in the same data row is matched with another variable's value.

From, the above graph we can say that there is some kind of linear line between Pregnancies and age.

Blood Pressure and age have little relation. Most of the aged people have Blood Pressure.

Insulin and Glucose have some relation.

Correlation between Variables:

Correlation is a bivariate analysis that measures the strength of association between two variables and the direction of the relationship. In terms of the strength of relationship, the value of the correlation coefficient varies between +1 and -1. A value of ± 1 indicates a perfect degree

of association between the two variables. As the correlation coefficient value goes towards 0 and vice versa. Variables within a dataset can be related for lots of reasons. It can be useful in data analysis and modeling to better understand the relationships between variables. That relationship between two variables is called as correlation. A correlation could be positive, meaning both variables move in the same direction, or negative, meaning that when one variable's value increases, the other variables' values decrease. Correlation can also be neural or zero, meaning that the variables are unrelated.

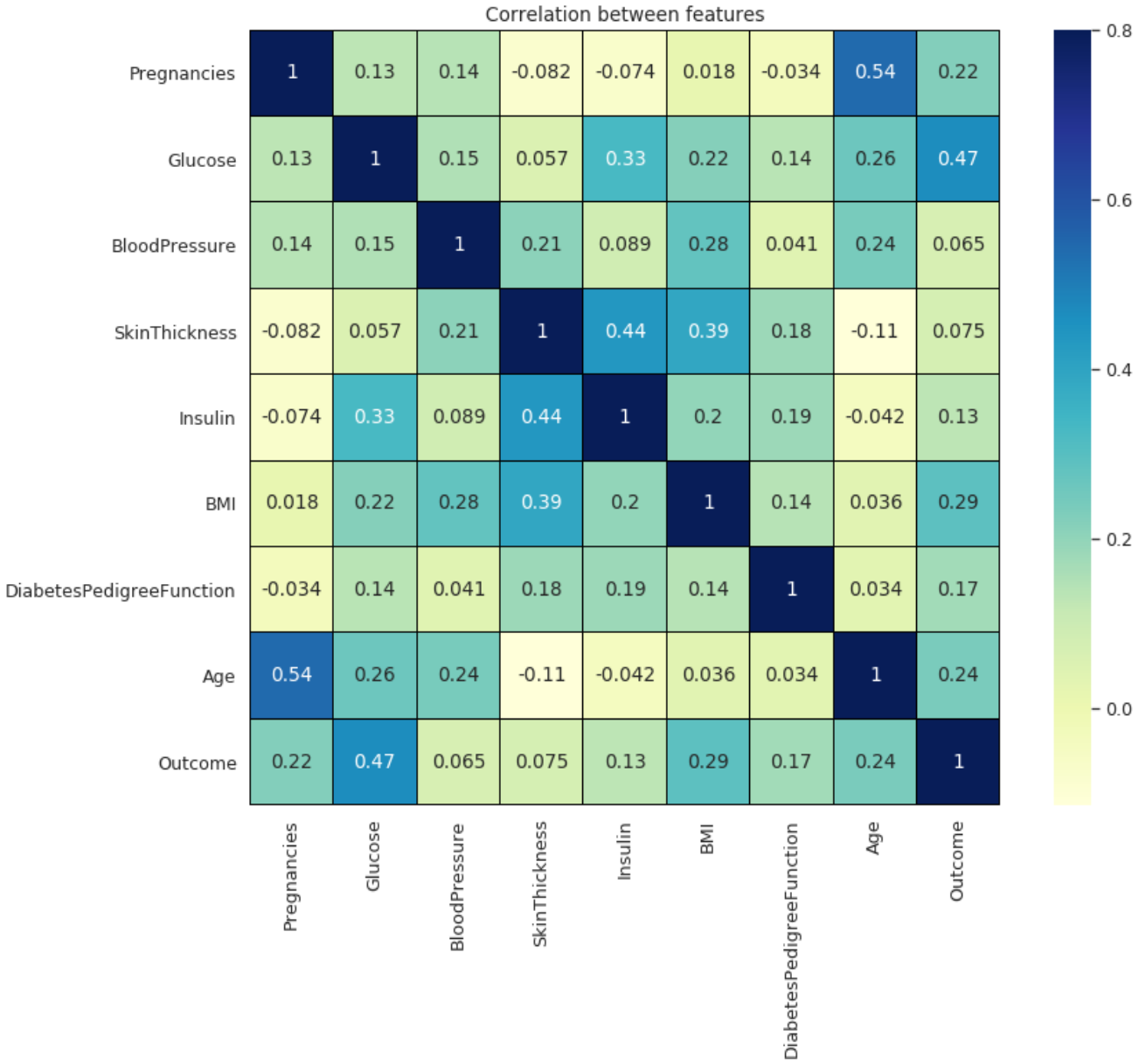
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.128749	0.193807	0.088314	0.051574	0.022731	-0.033523	0.544341	0.221898
Glucose	0.128749	1.000000	0.214426	0.177872	0.317222	0.228348	0.137932	0.267282	0.490122
BloodPressure	0.193807	0.214426	1.000000	0.177016	0.069794	0.268753	0.001775	0.319892	0.153802
SkinThickness	0.088314	0.177872	0.177016	1.000000	0.101794	0.467647	0.089064	0.147379	0.167967
Insulin	0.051574	0.317222	0.069794	0.101794	1.000000	0.109349	0.084946	0.122629	0.149867
BMI	0.022731	0.228348	0.268753	0.467647	0.109349	1.000000	0.156857	0.022044	0.309940
DiabetesPedigreeFunction	-0.033523	0.137932	0.001775	0.089064	0.084946	0.156857	1.000000	0.033561	0.173844
Age	0.544341	0.267282	0.319892	0.147379	0.122629	0.022044	0.033561	1.000000	0.238356
Outcome	0.221898	0.490122	0.153802	0.167967	0.149867	0.309940	0.173844	0.238356	1.000000

These are the correlation values between data.

To visualize the correlation between variable we use heatmap.

Heatmap:

A heat map (or heatmap) is a graphical representation of data where values are depicted by color. Heat maps make it easy to visualize complex data and understand it at a glance**:**



From this Heatmap we can say that,

- Pregnancies and age have some kind of a linear line.
- BloodPressure and age have little relation.
- Most of the aged people have BloodPressure.
- Insulin and Glucose have some relation.
- Glucose, Age BMI and Pregnancies are the most Correlated features with the Outcome.
- Insulin and DiabetesPedigreeFunction have little correlation with the outcome.
- BloodPressure and SkinThickness have tiny correlation with the outcome.
- Age and Pregnancies, Insulin and Skin Thickness, BMI and Skin Thickness, Insulin and Glucose are little correlated.

From all this Understanding we can say that, the middle-aged women are most likely to be diabetic than the young women. As the percentage of diabetic women are 48% and 59% in the age group of 31-40 and 41-55. After Pregnancy people have more chance of diabetes. People with high Glucose level are more likely to have diabetes. People with high BloodPressure have more chance of diabetes. People with high Insulin level are more likely to have Diabetes.

After doing all these analyses, now we have a clear understanding about the attributes and to which other variable they are related to. So now we need to build the model.

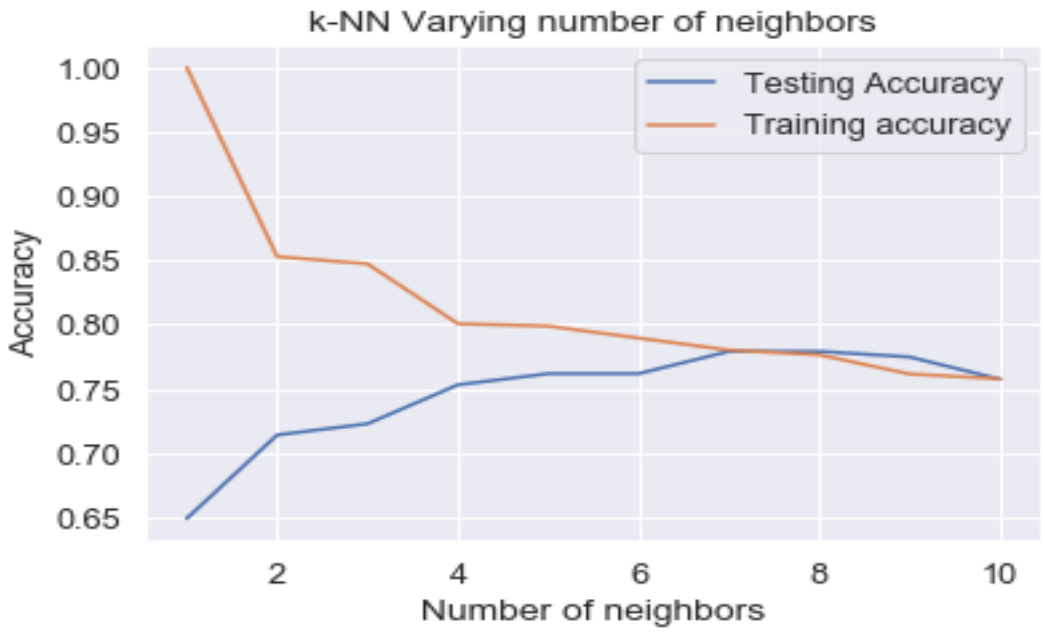
K-Nearest Neighbors:

The k-nearest neighbors (KNN) algorithm is a simple and one of the most commonly used supervised machine learning algorithm. This can be used to solve both classification and regression problems. A **supervised machine learning** algorithm is one that relies on labeled input data to learn a function that produces an appropriate output when given new unlabeled data.

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

It finds the similar items by calculating the distance between them. There are various distant measures that can be used for appropriate dataset.

For performing KNN algorithm, first we need to split our dataset into train and test set. Then we need to choose the number of clusters we need (i.e., K value) or else we can find the best K-value by iterating.



In this graph, we can see that the score of training and test set meet at the point 7, Hence the best k value is said to be 7. Even at point 10, there is an intersection between train and test score but it's accuracy is less than point 7. Hence we consider point 7 as the best cluster value.

Accuracy Score for first 10 Clusters:

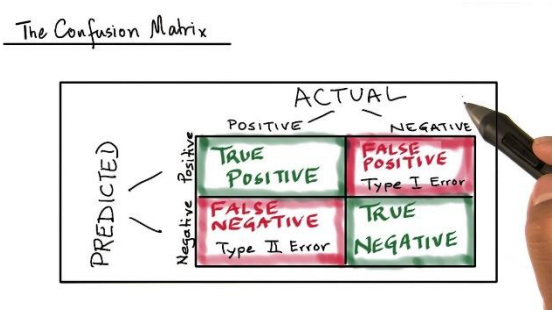
K = 1 Train_accuracy: 1.0
Test_accuracy: 0.6493506493506493
K = 2 Train_accuracy: 0.8528864059590316
Test_accuracy: 0.7142857142857143
K = 3 Train_accuracy: 0.8472998137802608
Test_accuracy: 0.7229437229437229
K = 4 Train_accuracy: 0.8007448789571695
Test_accuracy: 0.7532467532467533
K = 5 Train_accuracy: 0.7988826815642458
Test_accuracy: 0.7619047619047619
K = 6 Train_accuracy: 0.7895716945996276
Test_accuracy: 0.7619047619047619
K = 7 Train_accuracy: 0.7802607076350093
Test_accuracy: 0.7792207792207793

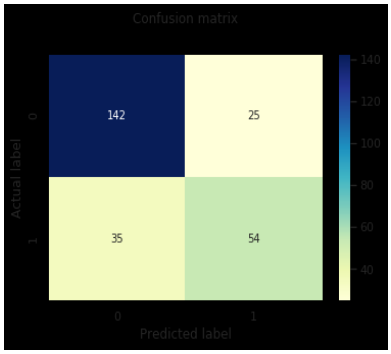
K = 8 Train_accuracy: 0.776536312849162
Test_accuracy: 0.7792207792207793
K = 9 Train_accuracy: 0.7616387337057728
Test_accuracy: 0.7748917748917749
K = 10 Train_accuracy: 0.7579143389199255
Test_accuracy: 0.7575757575757576

Model Performance Analysis:

Confusion Matrix

The confusion matrix is a technique used for summarizing the performance of a classification algorithm i.e. it has binary outputs. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.





Classification Report:

A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report. It includes Precision, Recall and F1 score.

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

Precision = TP/TP+FP

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class

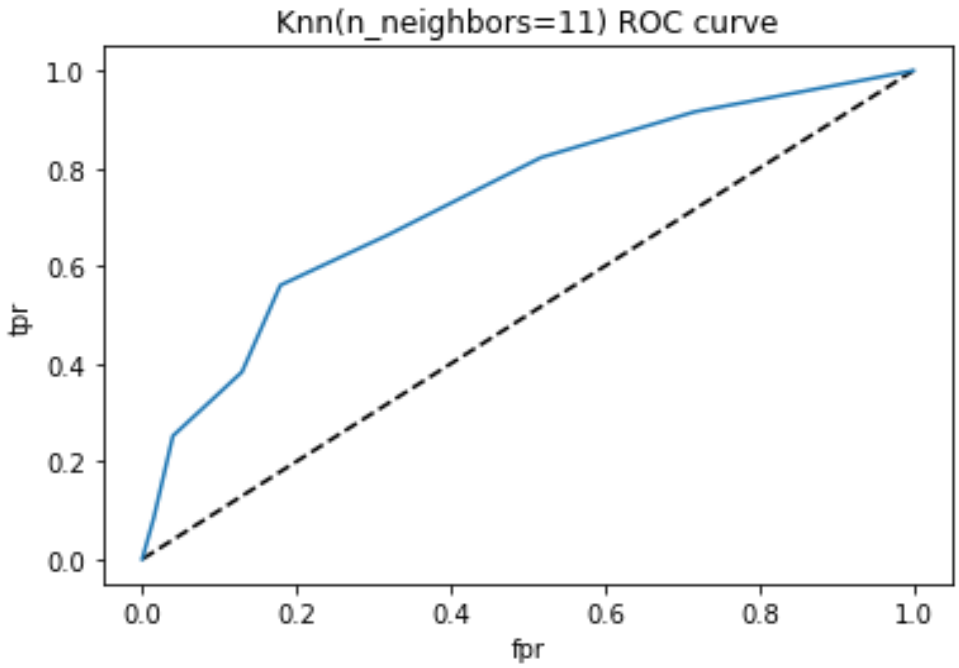
Recall = TP/TP+FN

F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

	precision	recall	f1-score	support
0	0.80	0.85	0.83	167
1	0.68	0.61	0.64	89
micro avg	0.77	0.77	0.77	256
macro avg	0.74	0.73	0.73	256

ROC – AUC Curve:

AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between various attributes.



From this we can tell, that K-NN algorithm is 73% accurate.

Decision tree:

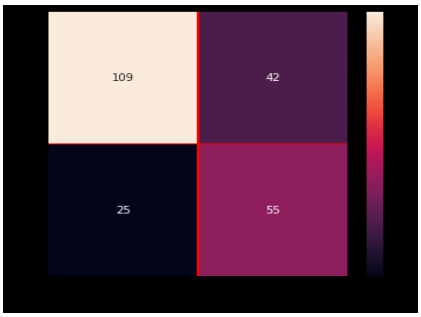
Decision Tree is a supervised machine learning algorithm used to solve classification problems. The main objective of using Decision Tree in this Project is the prediction of target class using decision rule taken from prior data. It uses nodes and internodes for the prediction and classification. Root nodes classify the instances with different features. Root nodes can have two or more branches while the leaf nodes represent classification. In every stage, Decision tree chooses each node by evaluating the highest information gain among all the attributes.

For calculating information gain we can use either Gini index or entropy. Information gain can also be used for feature selection, by evaluating the gain of each variable in the context of the target variable. In this slightly different usage, the calculation is referred to as mutual information between the two random variables.

Accuracy of Decision tree:

Accuracy : 77.48917748917748

Confusion Matrix for Decision Tree:



Classification Report for Decision tree:

Report :	precision	recall	f1-score	support
0	0.77	0.90	0.83	142
1	0.78	0.57	0.66	89
accuracy			0.77	231
macro avg	0.78	0.74	0.75	231

Comparative Performance of Classification Algorithms on Various Measures.

From all the above analysis, we can tell that Decision tree showing the maximum accuracy. So the Decision tree can predict the chances of diabetes with more accuracy as compared to other classifiers.

One of the important real-world medical problems is the detection of diabetes at its early stage. In this study, systematic efforts are made in designing a system which results in the prediction of disease like diabetes. During this work, machine learning classification algorithms are studied and evaluated on various measures. Experiments are performed on Pima Indians Diabetes Database. Experimental results determine the adequacy of the designed system with an achieved accuracy of 77.48 % using the Naive Bayes classification algorithm. In future, the designed system with the used machine learning classification algorithms can be used to predict or diagnose other diseases. The work can be extended and improved for the automation of diabetes analysis including some other machine learning algorithms.