MANI BARATHI SP S(1832031)

DEPARTMENT OF DATA SCIENCE

CLASSIFICATION OF DIABETIC PATIENT

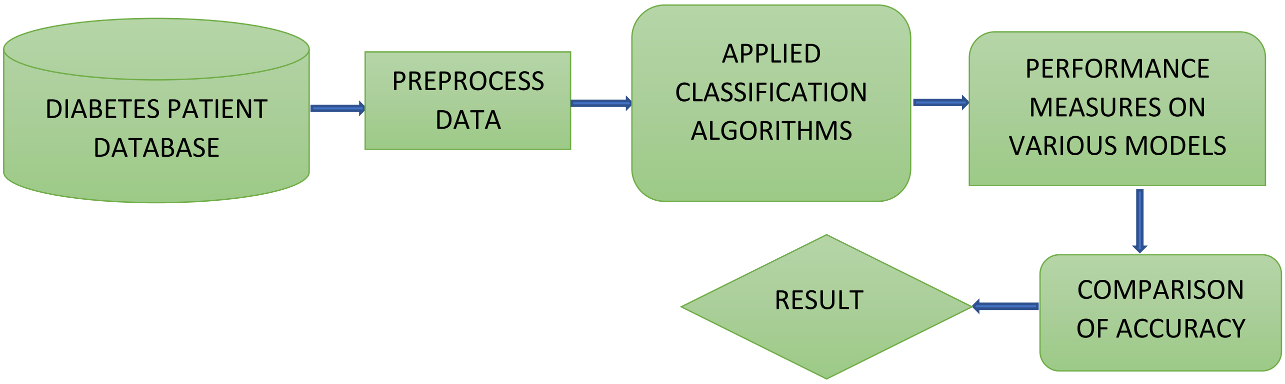
PROBLEM DESCRIPTION:

To Predict the patient who has diabetes from Diabetes Database.csv and also to understand the dataset attributes and to figure out type ML model suits. Fitting Machine Learning to process and transform Diabetes data to create a prediction model. This model must predict which people are likely to develop diabetes with > 90% accuracy (i.e. accuracy in the confusion matrix).

INTRODUCTION:

Classification strategies are broadly used in the medical field for classifying data into different classes according to some constrains comparatively an individual classifier. Diabetes is an illness which affects the ability of the body in producing the hormone insulin, which in turn makes the metabolism of carbohydrate abnormal and raise the levels of glucose in the blood. In Diabetes a person generally suffers from high blood sugar. Intensify thirst, Intensify hunger and Frequent urination are some of the symptoms caused due to high blood sugar. Many complications occur if diabetes remains untreated. Some of the severe complications include diabetic ketoacidosis and nonketotic hyperosmola coma. Diabetes is examined as a vital serious health matter during which the measure of sugar substance cannot be controlled. Diabetes is not only affected by various factors like height, weight, hereditary factor and insulin but the major reason considered is sugar concentration among all factors. The early identification is the only remedy to stay away from the complications. Machine learning algorithms gain its strength due to the capability of managing a large amount of data to combine data from several different sources and integrating the background information in the study.

METHODOLOGY:


DIABETES PATIENT DATABASE
PREPROCESS DATA
APPLIED CLASSIFICATION ALGORITHMS
PERFORMANCE MEASURES ON VARIOUS MODELS
COMPARISON OF ACCURACY
RESULT

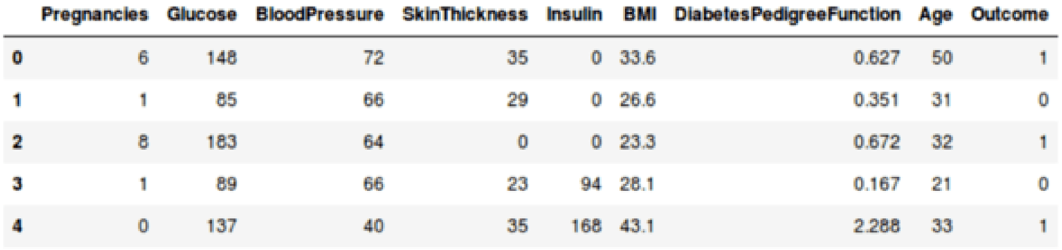
DATA DESCRIPTION:

1. Pregnancies decribes the number of times the person has been pregnant.
2. Gluose describes the blood glucose level on testing.
3. Blood pressure describes the diastolic blood pressure.
4. Insulin describes the amount of insulin in a 2hour serum test.
5. BMI describes he body mass index.
6. DiabetesPedigreeFunction describes the family history of the person.
7. Age describes the age of the person
8. Outcome describes if the person is predicted to have diabetes or not.
9. It should also be noted that the dataset has no missing values and thus, filling up the dataset using algorithms will not be necessary.

LOADING DATA:

We have our data saved in a CSV file called diabetes.csv. We first read our dataset into a pandas dataframe called diabetesDF, and then use the head() function to show the first five records from our dataset.

SAMPLE DATA:



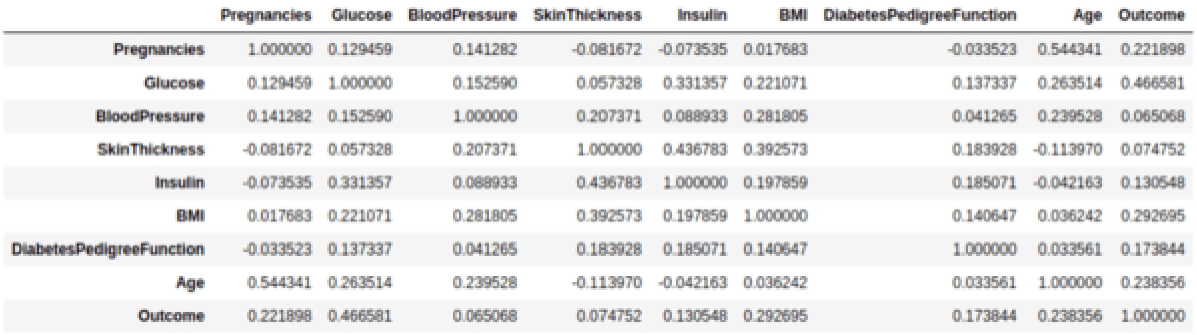
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

DATA EXPLORATION:

Data exploration is the initial step in data analysis, where users explore a large data set in an unstructured way to uncover initial patterns, characteristics, and points of interest. Data exploration can use a combination of manual methods and automated tools such as data visualizations, charts, and initial reports.

CORRELATION:

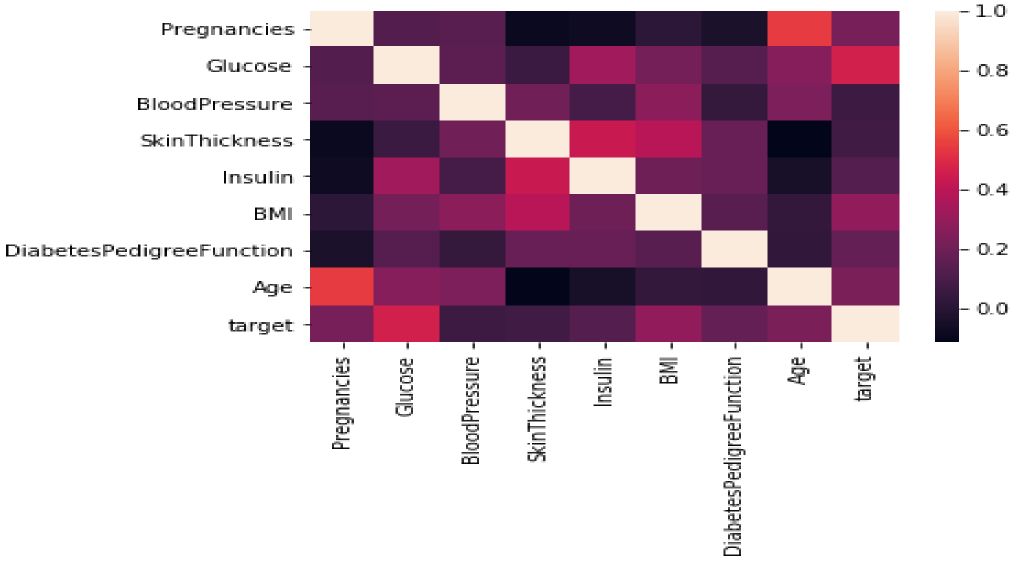
Correlation is an indication about the changes between two or more variables. The Correlation says the relation between the target variable and other independent variables.



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221898
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

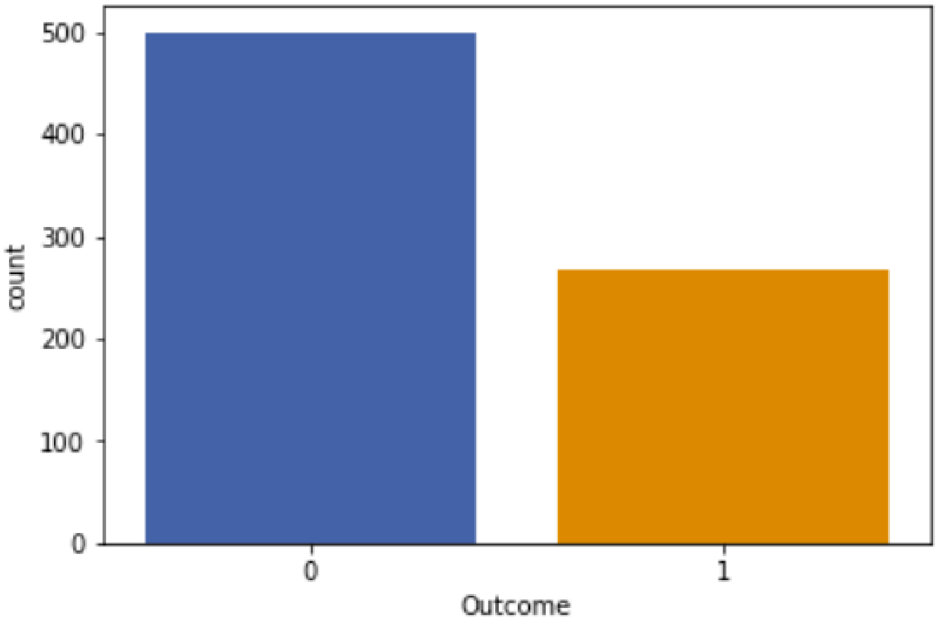
Output of feature (and outcome) correlations

Heat Map:



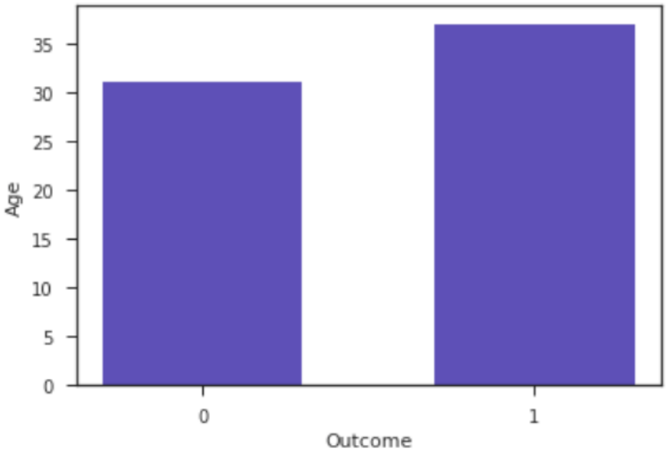
In the above heatmap, brighter colors indicate more correlation. As we can see from the table and the heatmap, glucose levels, age, BMI and number of pregnancies all have significant correlation with the outcome variable. Also notice the correlation between pairs of features, like age and pregnancies, or insulin and skin thickness.

BAR PLOT:



Barplot visualization of number of non-diabetic (0) and diabetic (1) people in the dataset

- From the bar plot we can say that non-diabetic persons are more in count than the persons with diabetics.



Average age of non-diabetic and diabetic people in the dataset

- It is also helpful to visualize relations between a single variable and the outcome.
- The relation between age and outcome.
- The figure is a plot of the mean age for each of the output classes.
- We can see that the mean age of people having diabetes is higher.

DATASET PREPARATION:

The data set consists of record of 768 patients in total. To train our model we will be using 650 records. We will be using 100 records for testing, and the last 17 records to cross check our model.

```
Number of True cases:  268 (34.90%)
Number of False cases: 500 (65.10%)
```

There are 34.90% of patients have diabetics and 65.10% people are non-diabetic

- Next separate the label and features (for both training and test dataset).
- Also convert them into NumPy arrays as our machine learning algorithm process data in NumPy array format.
- As the final step before using machine learning , will normalize our inputs.
- Machine Learning models often benefit substantially from input normalization.
- It also makes it easier for us to understand the importance of each feature later, looking at the model weights.

- Normalize the data such that each variable has 0 mean and standard deviation of 1.

Training and Evaluating Machine Learning Model:

Now train our classification model using a machine simple learning model

called **logistic regression.**

```
69.92% in training set
30.08% in testing set
```

```
The training set accuracy is 69.92% and testing set accuracy is 30.08%
```

```
Original True  : 268 (34.90%)
Original False : 500 (65.10%)

Training True  : 188 (35.01%)
Training False : 349 (64.99%)

 Test True      : 80 (34.63%)
Test False     : 151 (65.37%)
```

Before and after training and testing

SELECTING THE ALGORITHM:

NAIVE BAYES :

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in *text classification* that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

IMPLEMENTATION:

Here implemented the naïve bayes algorithm to predict the person has a diabetic or not. Test the model's accuracy with training data with Accuracy: 0.7542. Test the model's accuracy with testing data with Accuracy:0.7359

Accuracy is `0.7542` for training model, and `0.7359` for testing model

Confusion Matrix:

```
Confusion Matrix
 [[118  33]
 [ 28  52]]
```

As we can see in the above confusion matrix output, there are 28+33=61 incorrect predictions, and 118+52=170 correct predictions.

Classification Report for Naive Bayes:

```
Classification Report
             precision    recall  f1-score   support

          0       0.81      0.78      0.79       151
          1       0.61      0.65      0.63        80

avg / total       0.74      0.74      0.74       231
```

- **recall** = true positive rate/ sensitivity = measures how well the model is predicting diabetes when the result is diabetes

INFERENCE:

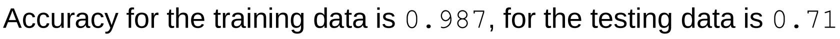
Our aim is to classify the diabetic patient with >70% accuracy but by using this naïve bayes classification we get accuracy lower than 70% which Recall is 0.65, and precision is 0.61, lower than the objective (>70%). So we use another model to classify the diabetic and non-diabetic person.

Random Forest:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ***ensemble learning,*** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

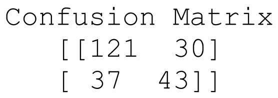
IMPLEMENTATION:

Here implemented the Random forest classification algorithm to predict the person has a diabetic or not. Then we Check performance on the training data using Random Forest model which gives the accuracy of 100% and Check performance on the testing data using Random Forest model which has the accuracy of 71%.



Accuracy for the training data is `0.987`, for the testing data is `0.71`

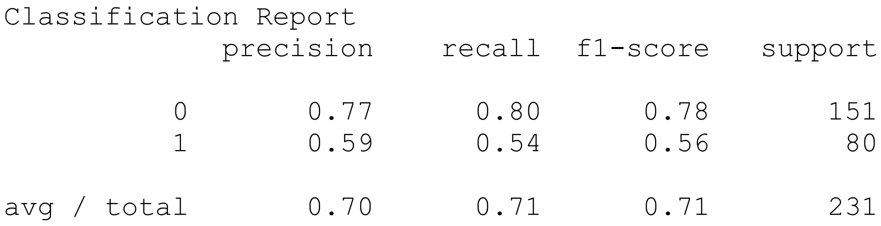
Confusion matrix for Random Forest:



```
Confusion Matrix
   [[121  30]
   [ 37  43]]
```

As we can see in the above confusion matrix output, there are 37+30=67 incorrect predictions, and 121+43=164 correct predictions.

Classification report for Random Forest:



```
Classification Report
             precision    recall  f1-score   support

          0       0.77      0.80      0.78       151
          1       0.59      0.54      0.56        80

avg / total       0.70      0.71      0.71       231
```

RANDOM FOREST RESULT:

Recall is 0.54, and precision is 0.59, both are lower than the Naive Bayes model. Looks like we have an overfitting problem for the Random Forest model.

GRADIENT DESCENT:

Gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function (f) that minimizes a cost function (cost).Gradient descent is best used when the parameters cannot be calculated analytically (e.g. using linear algebra) and must be searched for by an optimization algorithm.

ACCURACY AFTER GRADIENT DESCENT:

```
Accuracy on training set: 0.802
  Accuracy on test set: 0.776
```

The accuracy of the Random Forest model is 80%

LOGISTIC REGRESSION:

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

IMPLEMENTATION:

```
Accuracy:0.7446

Confusion Matrix:
```

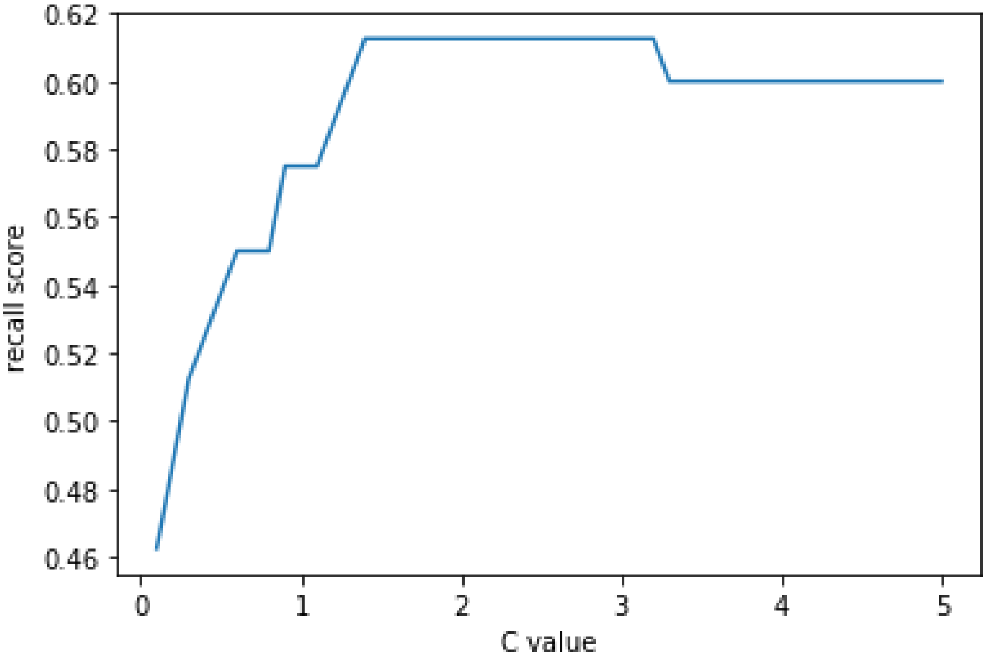
```
Confusion Matrix
   [[128  23]
   [ 36  44]]
```

Classification Report:

```
              precision    recall  f1-score   support

           0       0.78      0.85      0.81       151
           1       0.66      0.55      0.60        80

avg / total       0.74      0.74      0.74       231
```

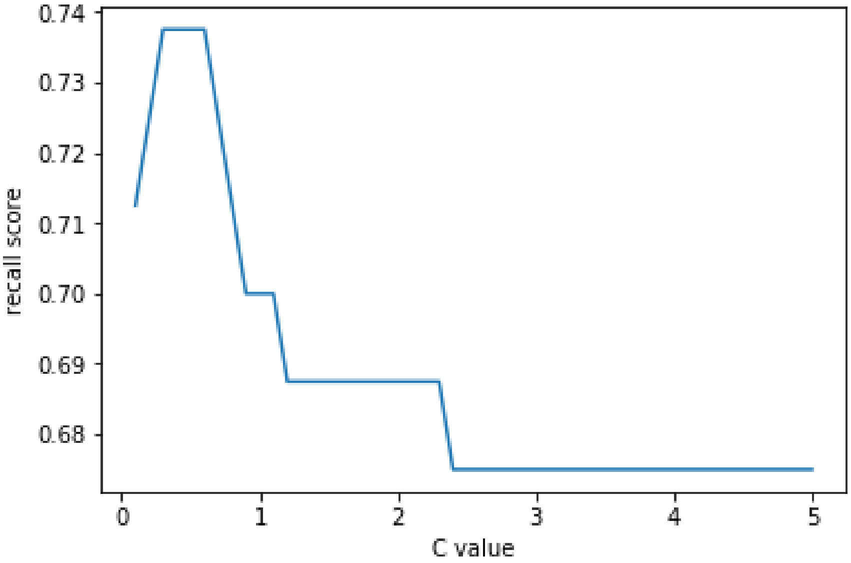
Here implemented the logistic regression classification algorithm to predict the person has a diabetic or not. We got a accuracy of 74% which is a quite good model. As we can see in the above confusion matrix output, there are 36+23=59 incorrect predictions, and 128+23=151 correct predictions. Recall is 0.55, and precision is 0.66, both are lower than the objective (>70%).

Regularization parameter for logistic regression model:



logistic regression with class_weight="balanced":

This is to solve the fact that the classes are not balanced (i.e. there are 35% Diabetes vs. 65% No Diabetes in this dataset).Because it's not 50/50, unbalanced classes may yield poor prediction results. Implementing **balanced weight** will cause a change in the predicted class boundary.



```
Accuracy: 0.7143
```

```
[[106  45]
 [ 21  59]]
```

Classification Report:

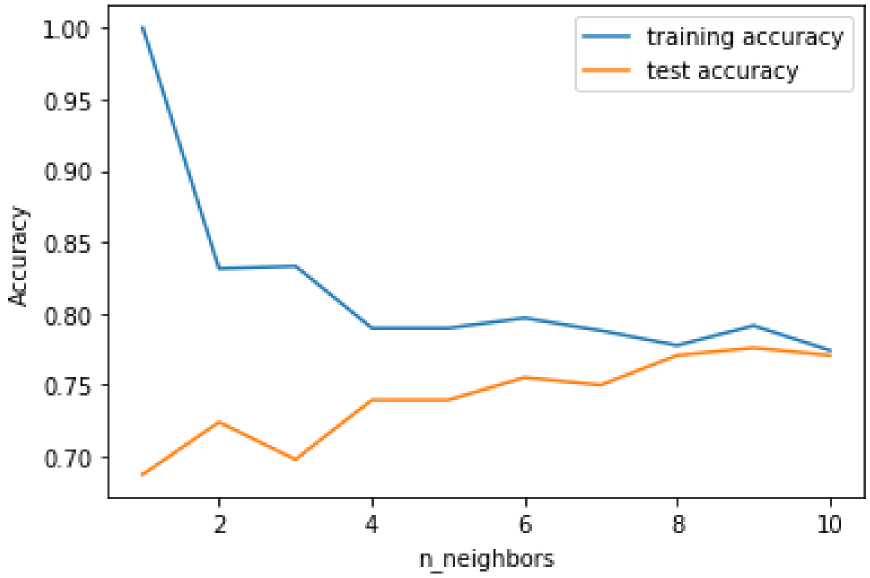
```
             precision    recall  f1-score   support

          0       0.83      0.70      0.76       151
          1       0.57      0.74      0.64        80

avg / total       0.74      0.71      0.72       231

0.7375
```

Logistic Regression with balanced weights:

Recall is 0.74, and precision is 0.57. Recall > 70% means that we've achieved the objective.

K-NEAREST NEIGHBORS:

The k-NN algorithm is arguably the simplest machine learning algorithm. Building the model consists only of storing the training data set. To make a prediction for a new data point, the algorithm finds the closest data points in the training data set—its “nearest neighbors.”



The above plot shows the training and test set accuracy on the y-axis against the setting of n_neighbors on the x-axis. Considering if we choose one single nearest neighbor, the prediction on the training set is perfect. But when more neighbors are considered, the training accuracy drops, indicating that using the single nearest neighbor leads to a model that is too complex. The best performance is somewhere around 9 neighbors.

ACCURACY OF KNN MODEL:

```
Accuracy of K-NN classifier on training set: 0.79
 Accuracy of K-NN classifier on test set: 0.78
```

DECISION TREE:

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

ACCURACY:

```
Accuracy on training set: 1.000
 Accuracy on test set: 0.714
```

The accuracy on the training set is 100%, while the test set accuracy is much worse. This is an indicative that the tree is overfitting and not generalizing well to new data. Therefore, we need to apply pre-pruning to the tree. We set max_depth=3, limiting the depth of the tree decreases overfitting. This leads to a lower accuracy on the training set, but an improvement on the test set.

```
Accuracy on training set: 0.773
 Accuracy on test set: 0.740
```

After improvement the accuracy of the model.

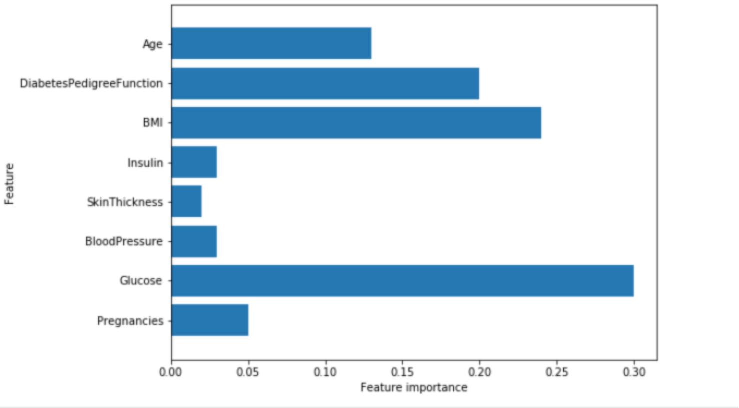
SUPPORT VECTOR MACHINE:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

ACCURACY:

```
Accuracy on training set: 0.944
 Accuracy on test set: 0.724
```

CONCLUSION:



Glucose level, BMI, pregnancies and diabetes pedigree function have significant influence on the model, specially glucose level and BMI. It is good to see our machine learning model match what we have been hearing from doctors our entire lives. Blood pressure has a negative influence on the prediction, i.e. higher blood pressure is correlated with a person not being diabetic. (also, note that blood pressure is more important as a feature than age, because the *magnitude* is higher for blood pressure).Although age was more correlated than BMI to the output variables (as we saw during data exploration), the model relies more on BMI. This can happen for several reasons, including the fact that the correlation captured by age is also captured by some other variable, whereas the information captured by BMI is not captured by other variables.

Result:

Among the models we evaluated:
*** Naive Bayes**
*** Random Forest**
*** Logistic Regression**
*** Logistic Regression with balanced classes**
*** Decision Tree**
***Support Vector Machine**
***K Nearest Neighbour**

Random Forest seems to provide the best recall value. 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, **SEVEN** machine learning classification algorithms are studied and evaluated on various measures. Experiments are performed on Diabetes Database. Experimental results determine the adequacy of the designed system with an achieved accuracy of 80 % using the Random Forest Machine 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.