A model for early prediction of diabetes

By Anekait Kariya 2017A7PS0031G
Fulfillment of FDS CS F320 Assignment
Professor: Vidyapu Sandeep

Based on paper “A model for early prediction of diabetes” -
https://www.sciencedirect.com/science/article/pii/S2352914819300176

Problem Statement:

Diabetes is one of the deadliest diseases in the world. It is not only a disease but also creators of different kinds of diseases like heart attack, blindness etc. The normal identifying process is that patients need to visit a diagnostic center, consult their doctor, and sit tight for a day or more to get their reports.

So, the objective of this project is to identify whether the patient has diabetes or not based on diagnostic measurements.

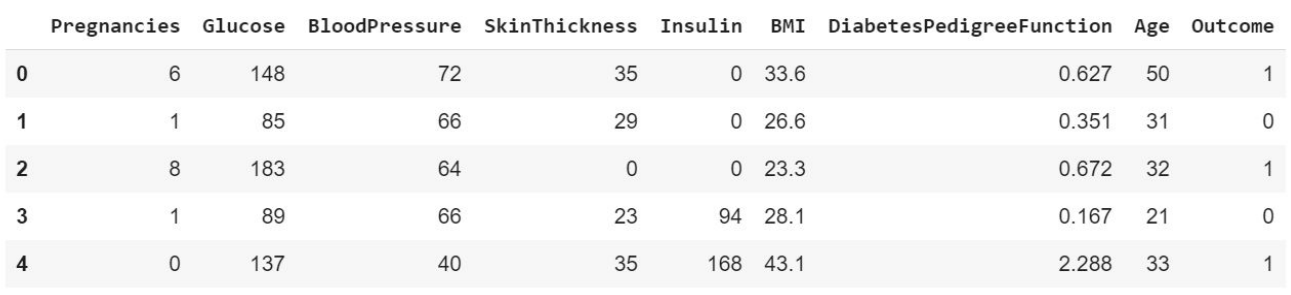
Dataset:

The dataset used in this study is originally taken from the National Institute of Diabetes and Digestive and Kidney Diseases (publicly available at: UCI ML Repository https://data.world/uci/pima-indians-diabetes or https://www.kaggle.com/uciml/pima-indians-diabetes-database).

The Pima Indian Diabetes (PID) dataset having:
9 = 8 + 1 (Class Attribute) attributes
768 records describing female patients of which
there were **500 negative** instances (65.1%) and **268 positive** instances (34.9%))

Data Description:

Sample data

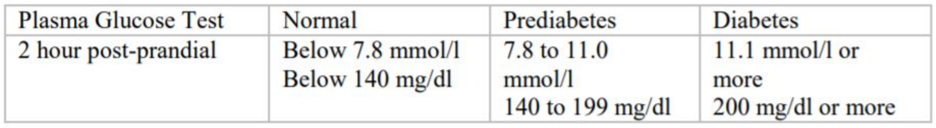


	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

There are **8 independent variables**:

1. Pregnancies: No. of times pregnant

2. Glucose: Plasma Glucose Concentration a 2 hour in an oral glucose tolerance test (mg/dl)



Plasma Glucose Test	Normal	Prediabetes	Diabetes
2 hour post-prandial	Below 7.8 mmol/l Below 140 mg/dl	7.8 to 11.0 mmol/l 140 to 199 mg/dl	11.1 mmol/l or more 200 mg/dl or more

Prediabetes means you are at increased risk of developing diabetes over time.
A glucose level of 200 mg/dL (11.1 mmol/L) or higher is used to diagnose diabetes.

3. Blood Pressure: Diastolic Blood Pressure(mmHg) -
If Diastolic B.P > 90 means High B.P (High Probability of Diabetes)
Diastolic B.P < 60 means low B.P (Less Probability of Diabetes)

4. Skin Thickness: Triceps Skin Fold Thickness (mm) -
A value used to estimate body fat. Normal Triceps SkinFold Thickness in women is 23mm.
Higher thickness leads to obesity and chances of diabetes increases.

5.Insulin: 2-Hour Serum Insulin (mu U/ml) -
Normal Insulin Level 16-166 mIU/L
Values above this range can be alarming.

6. BMI: Body Mass Index (weight in kg/ height in m2) -
Body Mass Index of 18.5 to 25 is within the normal range
BMI between 25 and 30 then it falls within the overweight range.
A BMI of 30 or over falls within the obese range.

7. Diabetes Pedigree Function:
It provides information about diabetes history in relatives and genetic relationship of those relatives with patients.
Higher Pedigree Function means a patient is **more likely** to have diabetes.

8. Age (years)

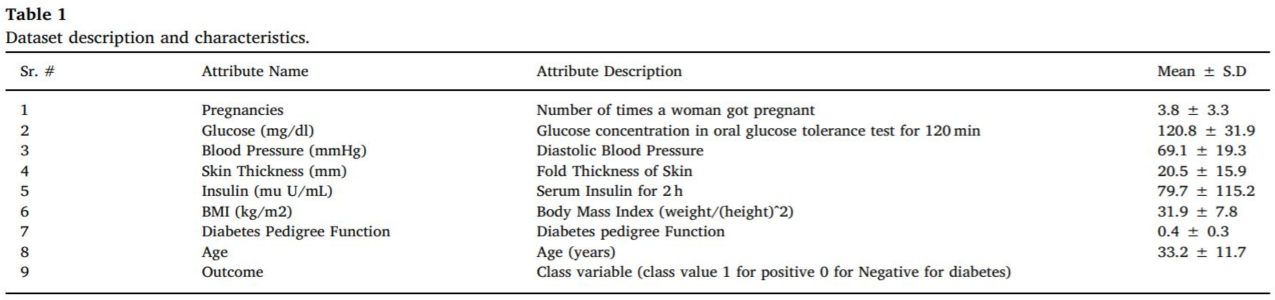


Table 1
Dataset description and characteristics.

Sr. #	Attribute Name	Attribute Description	Mean ± S.D
1	Pregnancies	Number of times a woman got pregnant	3.8 ± 3.3
2	Glucose (mg/dl)	Glucose concentration in oral glucose tolerance test for 120 min	120.8 ± 31.9
3	Blood Pressure (mmHg)	Diastolic Blood Pressure	69.1 ± 19.3
4	Skin Thickness (mm)	Fold Thickness of Skin	20.5 ± 15.9
5	Insulin (mu U/mL)	Serum Insulin for 2 h	79.7 ± 115.2
6	BMI (kg/m2)	Body Mass Index (weight/(height)^2)	31.9 ± 7.8
7	Diabetes Pedigree Function	Diabetes pedigree Function	0.4 ± 0.3
8	Age	Age (years)	33.2 ± 11.7
9	Outcome	Class variable (class value 1 for positive 0 for Negative for diabetes)	

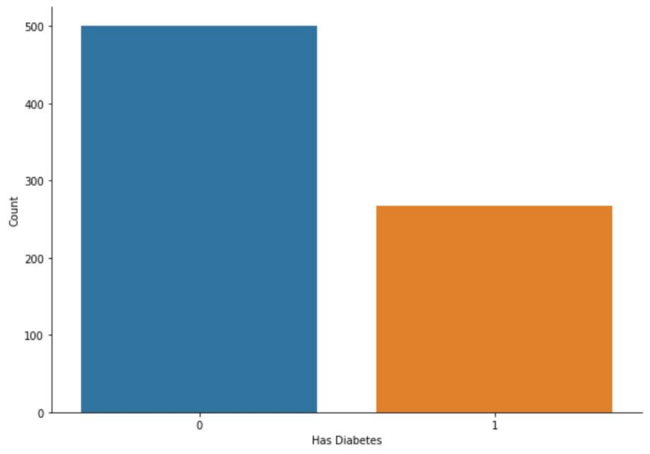
Data Exploration:

Data exploration is the initial step in data analysis, where we will explore the data set in an unstructured way to uncover initial patterns, characteristics, and points of interest. This process isn't meant to reveal every bit of information a dataset holds, but rather to help create a broad picture of important trends and major points to study in greater detail.

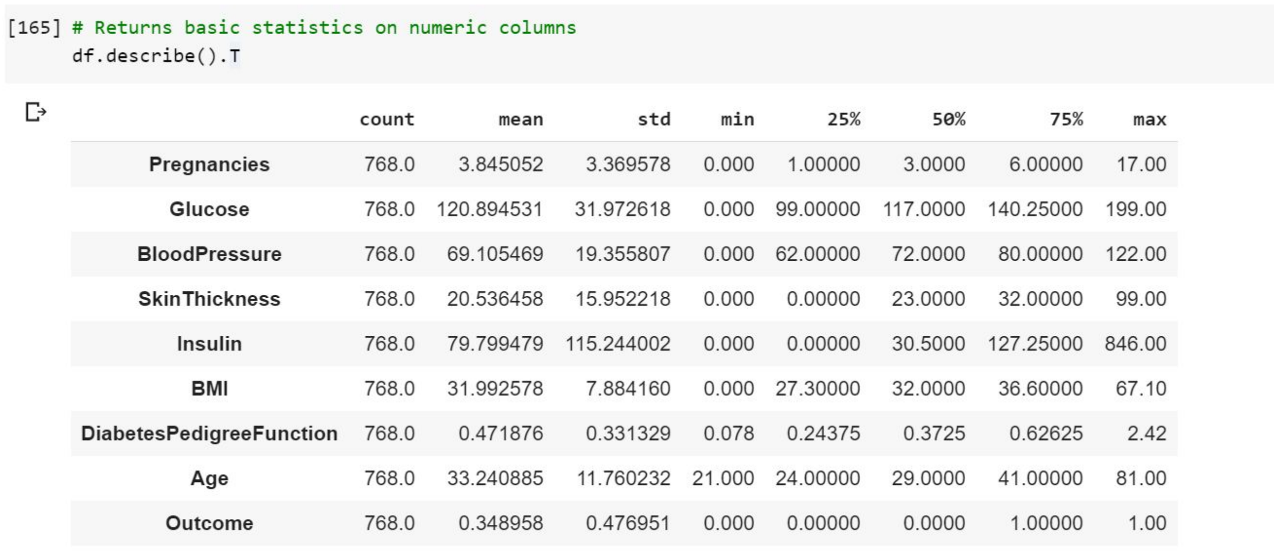
1. **How many people in the dataset are diabetic and how many are not:**

   500 negative instances (65.1%) and **268 positive** instances (34.9%))

   Below is the barplot of the same:



2. **Basic statistics on different features:**



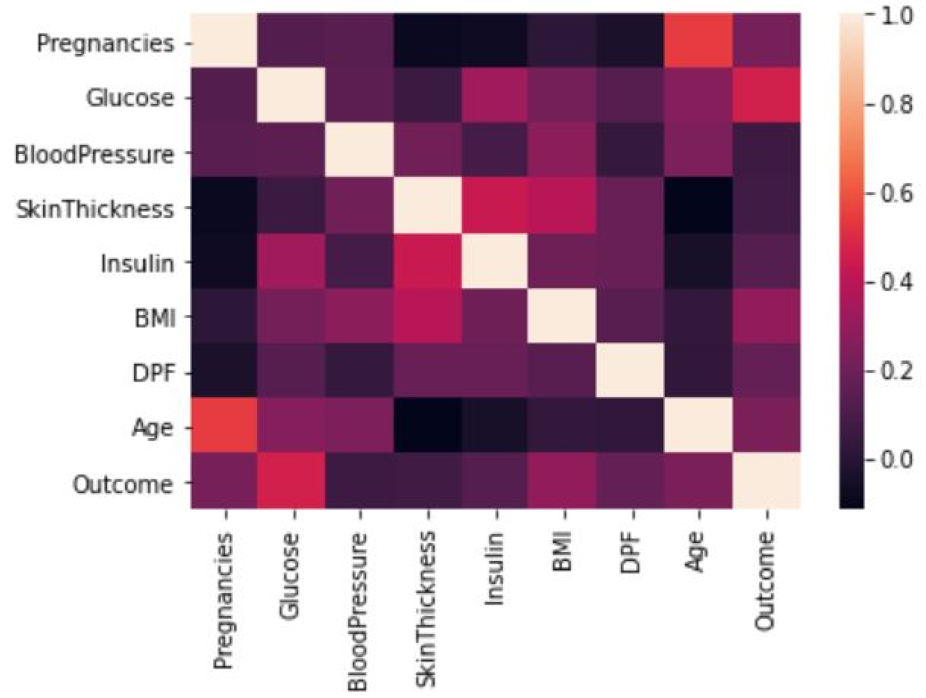
```
[165] # Returns basic statistics on numeric columns
      df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.00000	17.00
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	140.25000	199.00
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	80.00000	122.00
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	32.00000	99.00
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	127.25000	846.00
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	36.60000	67.10
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.62625	2.42
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.00000	81.00
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.00000	1.00

3. **Heatmap for correlation of every pair of features:**

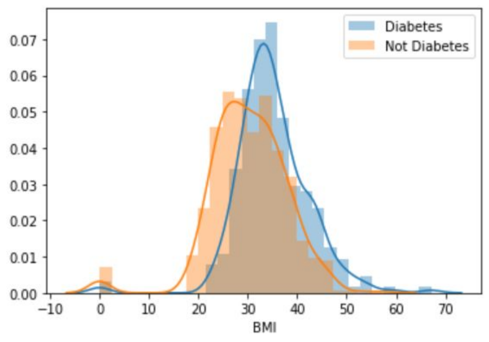
   In the below heatmap, **brighter colors indicate more correlation**. As we can see from the table and the heatmap, glucose levels, age, BMI and number of pregnancies all have significant correlation with the outcome variable. Also notice the correlation between pairs of features, like age and pregnancies,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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f91480c0390>
```

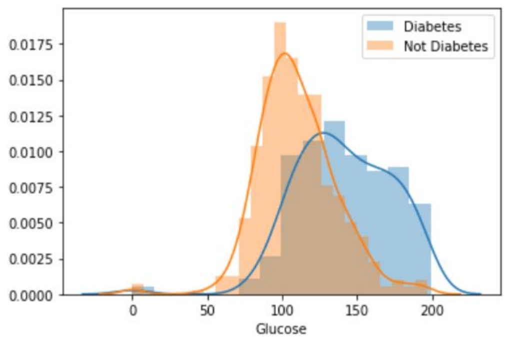


4. Using the following graphs we can predict that
   - **Higher** value of **Glucose** and **higher** value of **BMI** could imply Diabetes.
   - Using similar analysis we can't comment on other features .

```
<matplotlib.legend.Legend at 0x7f9148b656d8>
```



```
<matplotlib.legend.Legend at 0x7f91481152e8>
```

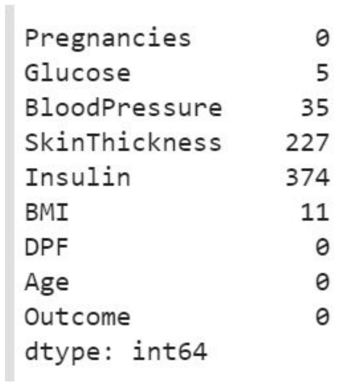


Data cleaning:

Data cleaning is the process of preparing data for analysis by removing or modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted. This data is usually not necessary or helpful when it comes to analyzing data because it may hinder the process or provide inaccurate results. There are several methods for cleaning data depending on how it is stored along with the answers being sought.

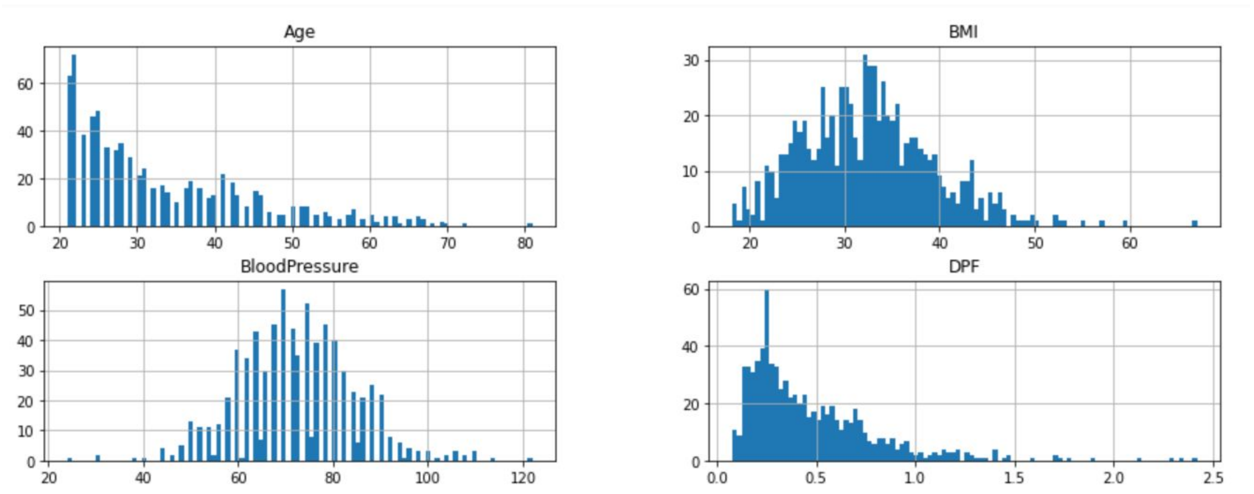
- We find that following number of records of each feature are zero hence we need to replace it by NaN

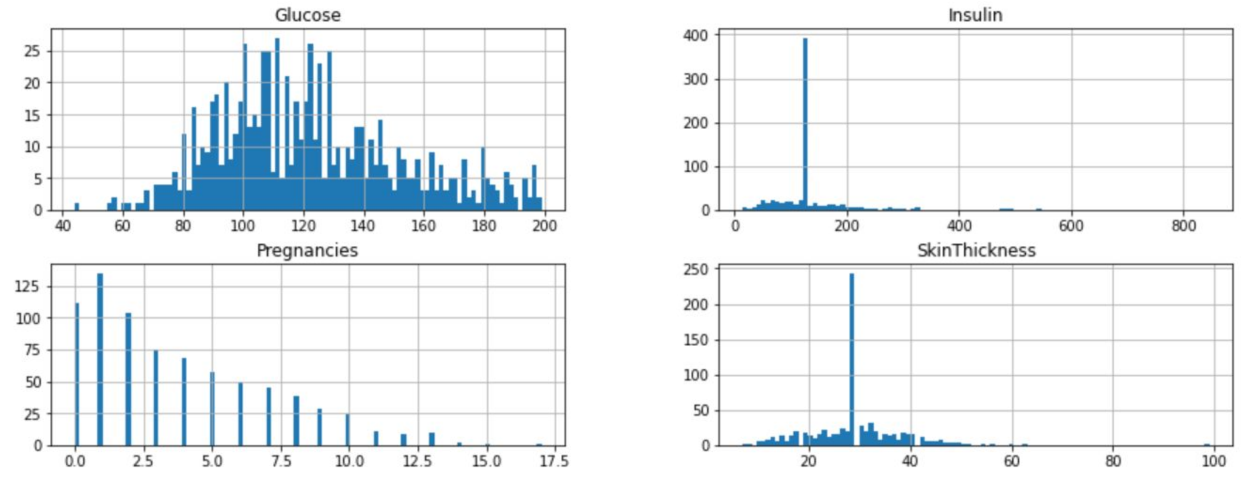
- Replacing the 0 values from ['Glucose','BloodPressure','SkinThickness','Insulin','BMI'] by NaN



```
Pregnancies        0
Glucose            5
BloodPressure     35
SkinThickness    227
Insulin          374
BMI               11
DPF                0
Age                0
Outcome            0
dtype: int64
```

Histogram of dataset after replacing NaN values:





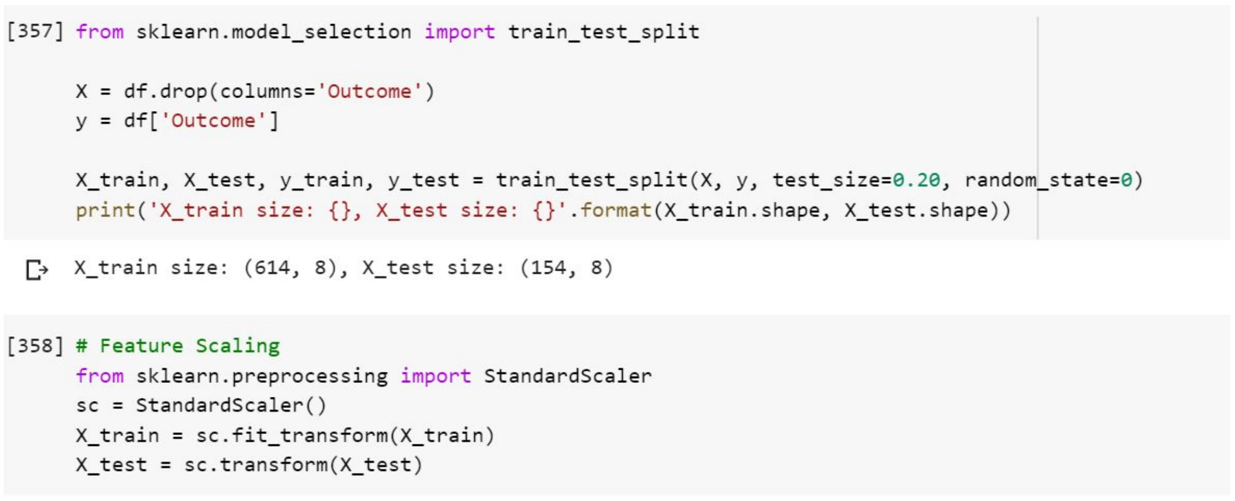
Dataset Preparation (splitting and normalization):

- When using machine learning algorithms we should always split our data into a training set and test set. In our case, we will also separate out some data for train and test.

- The data set consists of a record of **768** patients in total.
  To train our model we will be using **614 records** (80% of data set) . We will be using **154 records** (20% of data set) records for testing.

- As the final step before using machine learning, we 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, when we'll be looking at the model weights. We'll normalize the data such that each variable has 0 mean and standard deviation of 1.

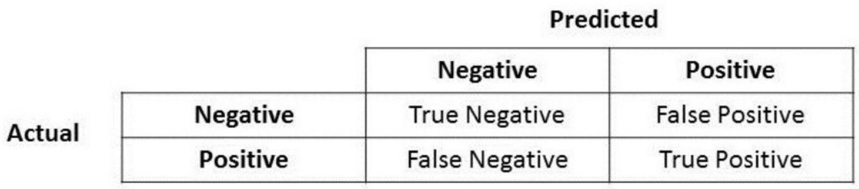
- Code achieving the above-



```
[357] from sklearn.model_selection import train_test_split

      X = df.drop(columns='Outcome')
      y = df['Outcome']

      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)
      print('X_train size: {}, X_test size: {}'.format(X_train.shape, X_test.shape))

      X_train size: (614, 8), X_test size: (154, 8)

[358] # Feature Scaling
      from sklearn.preprocessing import StandardScaler
      sc = StandardScaler()
      X_train = sc.fit_transform(X_train)
      X_test = sc.transform(X_test)
```

Important terms to understand before moving ahead to models

Confusion Matrix



		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

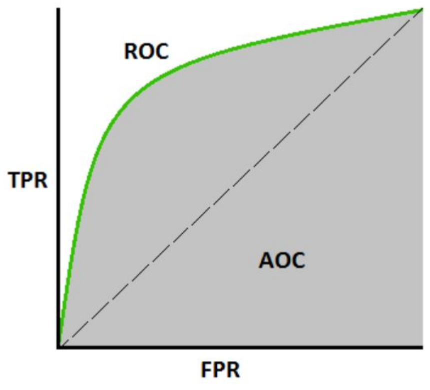
Precision vs recall vs F1- score

- Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.
- Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to all observations in actual class - positive.
- F1 score - F1 Score is the weighted average of Precision and Recall.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad \text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \qquad \text{F1} = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Understanding AUROC (Area Under the Receiver Operating Characteristics) -

- AUC - ROC curve is a performance **measurement for classification problems** at various thresholds settings.
- ROC is a probability curve and AUC represents **degree or measure of separability**. It tells **how much a 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 patients with disease and no disease.
- The ROC curve is plotted with **TPR against the FPR** where TPR is on the y-axis and FPR is on the x-axis.
- A sample ROC curve-



Artificial Neural Network Model

- Artificial neural networks (ANN) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules.

- An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it.

- A simple example of ANN

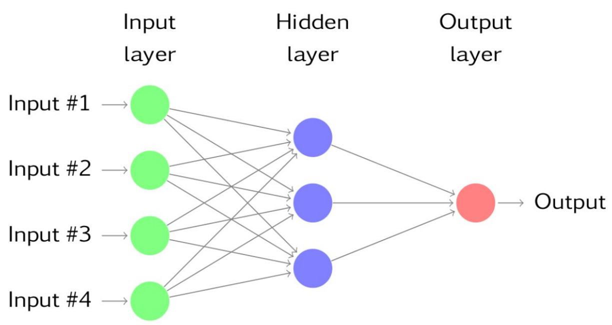


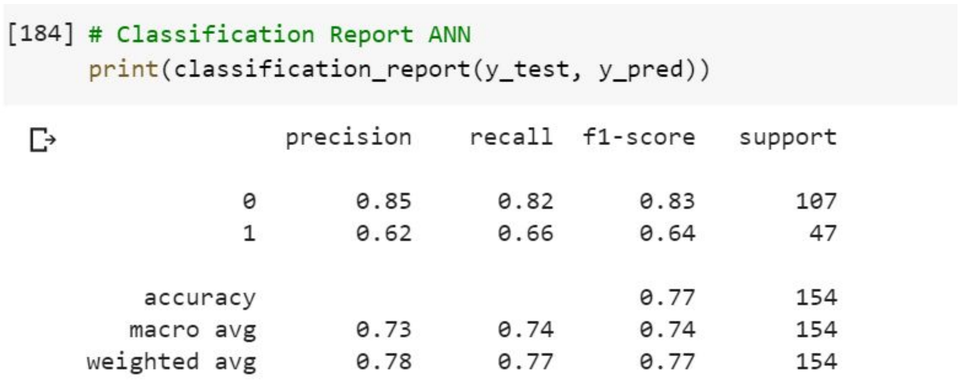
Figure 11.12: A neural network with four inputs and one hidden layer with three hidden neurons.

- The objective of ANN is to convert input into significant output. Input is the combination of a set of input values that are associated with the weight vector, where the weight can be negative or positive. There is a function that sums the weight and maps the result to the output, such as y = w1x1 + w2x2.

- After processing, the actual output is compared with required outputs. Errors are then back propagated to the system for adjustment. During training, the data is processed many times, so that the network can adjust the weights and refine them.

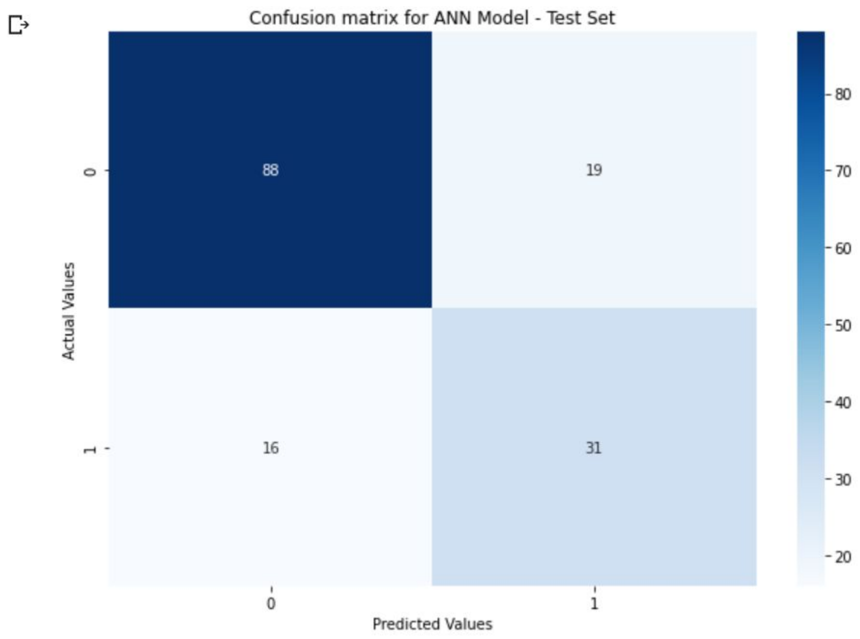
ANN Model Results -

Accuracy on test set ANN: 77.27%

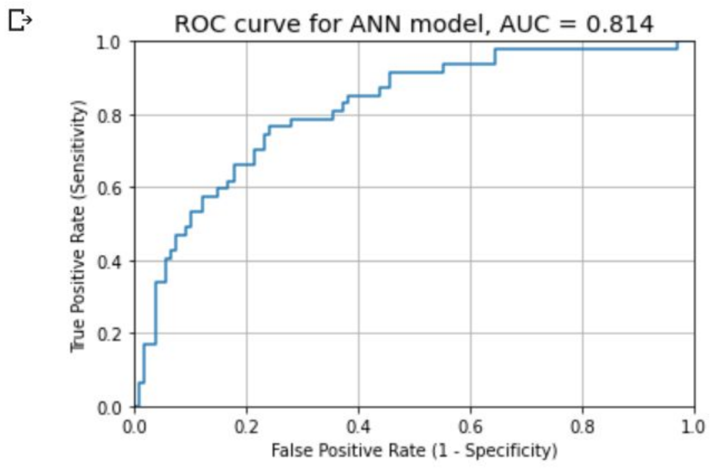


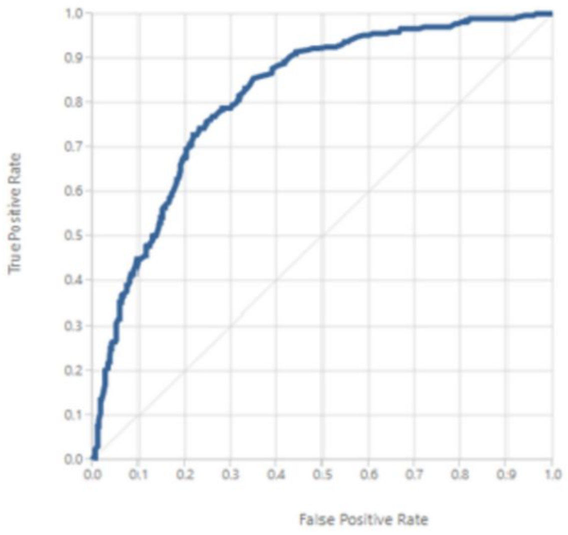
```
[184] # Classification Report ANN
      print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

           0       0.85      0.82      0.83       107
           1       0.62      0.66      0.64        47

    accuracy                           0.77       154
   macro avg       0.73      0.74      0.74       154
weighted avg       0.78      0.77      0.77       154
```

Confusion matrix for ANN Model - Test Set



ROC curve obtained by my ANN model vs Paper



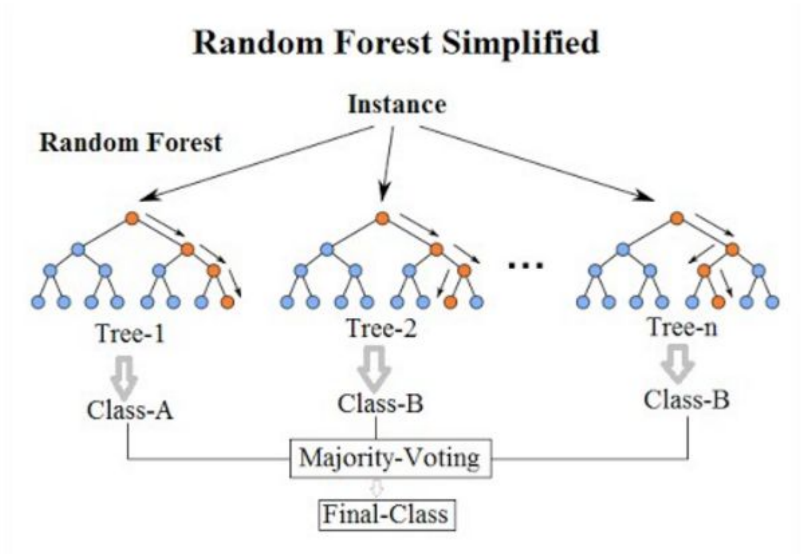


Random forest Model

- Random forest builds **multiple decision trees and aggregates them** to achieve more suitable and accurate results.In a random forest, a random subset of attributes gives more accurate results on large datasets, and more random trees can be generated by fixing a random threshold for all attributes, instead of finding the most accurate threshold. This **algorithm also solves the overfitting issue.**

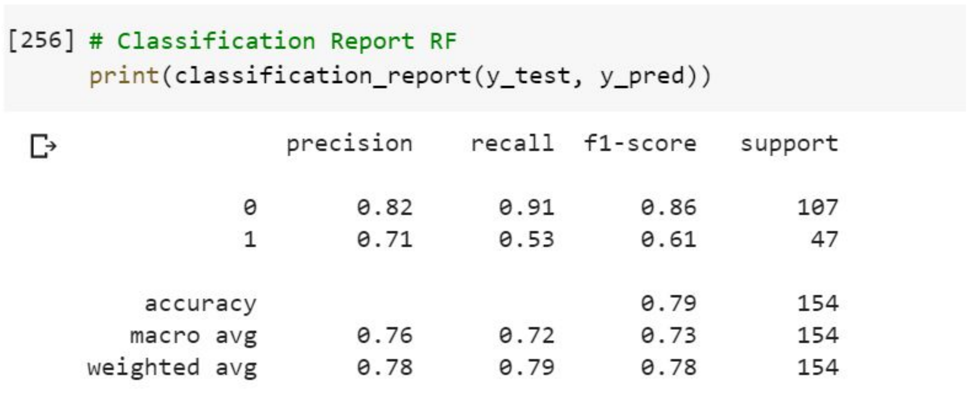
- The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don't constantly all error in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to **move in the correct direction.**

- Each decision tree in the forest considers a **random subset of features** when forming questions and only has access to a random set of the training data points. This increases diversity in the forest leading to more robust overall predictions and the name 'random forest.'



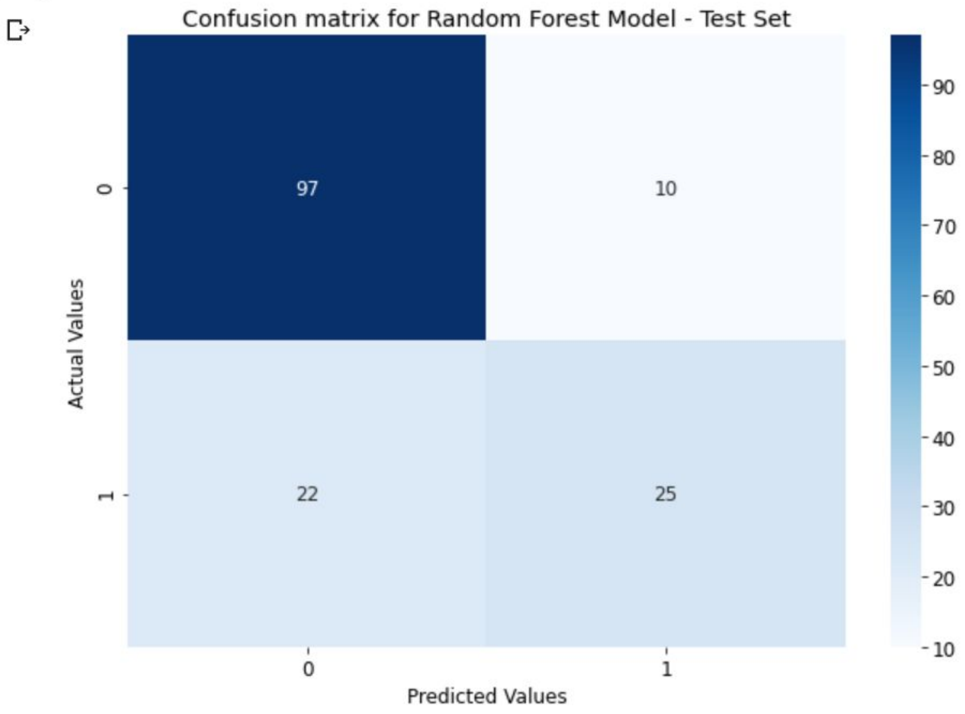
Random Forest Model Results -

Accuracy on test set RF: 79.22%

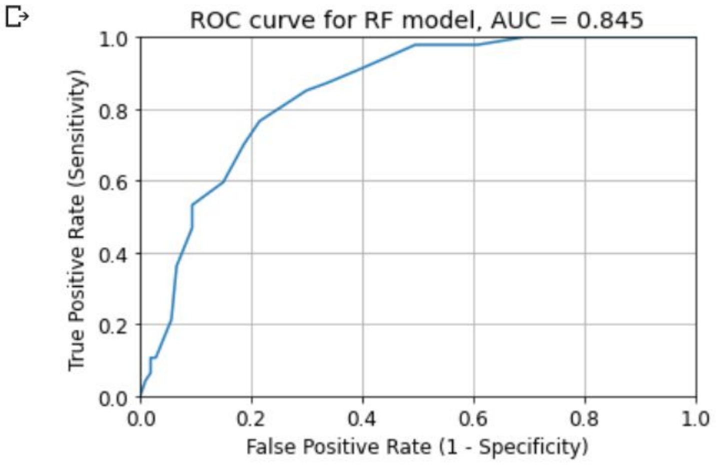


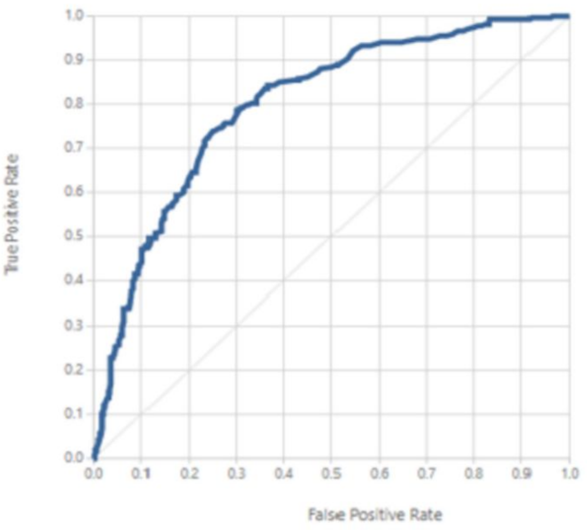
```
[256] # Classification Report RF
      print(classification_report(y_test, y_pred))

                   precision    recall  f1-score   support

               0       0.82      0.91      0.86       107
               1       0.71      0.53      0.61        47

        accuracy                           0.79       154
       macro avg       0.76      0.72      0.73       154
    weighted avg       0.78      0.79      0.78       154
```

Confusion matrix for Random Forest Model - Test Set

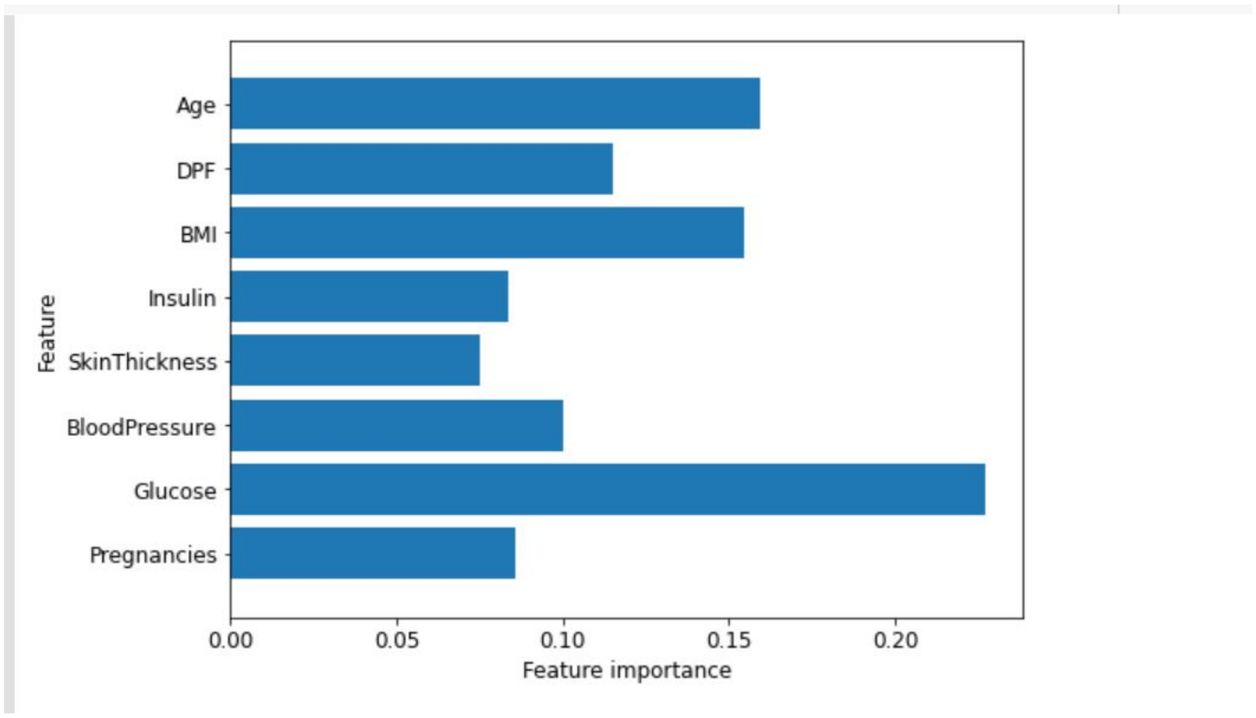


ROC curve obtained by my RF model vs Paper-





Feature importance in Random Forest -



- The random forest gives a lot of importance to the **“Glucose”** feature, but it chooses **“Age”** to be the 2nd most informative feature overall, closely followed by **“BMI”** at 3rd.

- The randomness in building the random forest forces the algorithm to consider many possible explanations, the result being that the random forest captures a much broader picture of the data than a single tree.

- **Note-** Additional implementation compare to paper

Other additional models implemented

1. Logistic Regression

- Logistic Regression is used when the **dependent variable** (target) is **categorical**. A logistic model is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick.

- The hypothesis of logistic regression tends it to limit the **cost function between 0 and 1**. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

- In order to map predicted values to probabilities, we use the **Sigmoid function**. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

- We learnt about the **cost function** J(θ) in the Linear regression, the cost function represents optimization objective i.e. we create a cost function and minimize it so that we can develop an accurate model with minimum error.

$$Sigmoid\ Function: \quad g(z) = \frac{1}{1 + e^{(-z)}}$$

$$Hypothesis: \quad h_\theta(x) = \frac{1}{1 + e^{(-\theta^T x)}}$$

Sigmoid Function

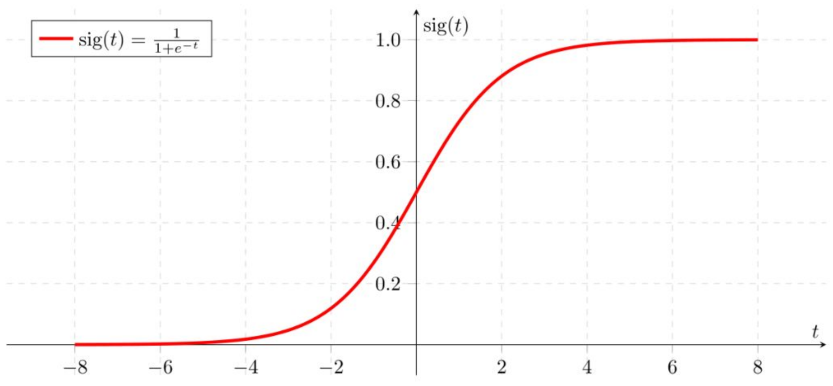


Figure 2: Sigmoid Activation Function

Cost function of the model is the summation from all training data samples:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

$$J(\theta) = \frac{1}{m}[\sum_{i=1}^{m} -y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))]$$

$$m = number\ of\ samples$$

2. Decision Tree

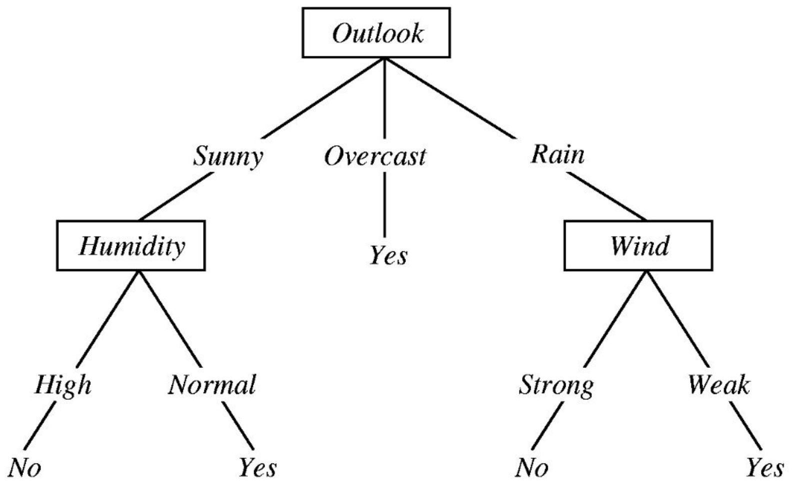
- Decision tree learning is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree (as a predictive model) to go from **observations** about an item (represented in the **branches**) to conclusions about the item's **target value** (represented in the **leaves**).

- Tree models where the target variable can take a discrete set of values are called **classification trees**; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

- Decision trees where the target variable can take continuous values (typically real numbers) are called **regression trees.**

- Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks.

- A example of Decision tree -



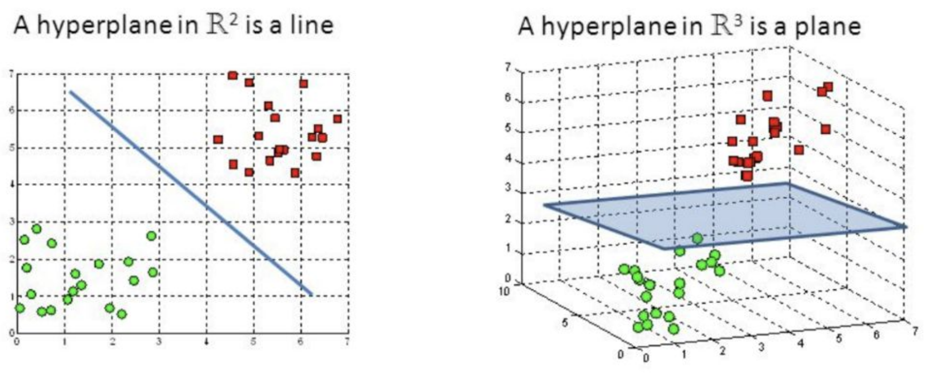
Decision Tree for Rain Forecasting

3. Support Vector Machine

- The objective of the support vector machine algorithm is to find a **hyperplane in an N-dimensional space**(N — the number of features) that distinctly classifies the data points.

- To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a **plane that has the maximum margin**, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

- **Support vectors are data points** that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.



Hyperplanes in 2D and 3D feature space

- In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.

- Hinge loss function for SVM

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

- Loss function for SVM

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Loss function for SVM

Model Results for Logistic Regression, Decision Tree, SVM

	model	best_parameters	score
0	logistic_regression	{'C': 1}	0.769106
1	decision_tree	{'criterion': 'entropy', 'max_depth': 10}	0.723577
2	svm	{'C': 1, 'kernel': 'linear'}	0.773984

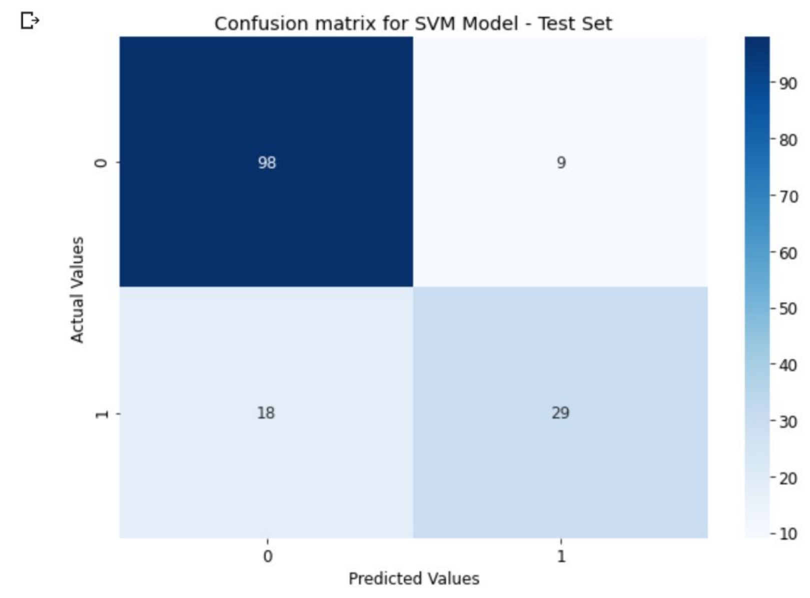
As SVM gave best results more analysis for SVM

Accuracy on test set SVM: 82.47%

```
[265] # Classification Report SVM
      print(classification_report(y_test, y_pred))
```

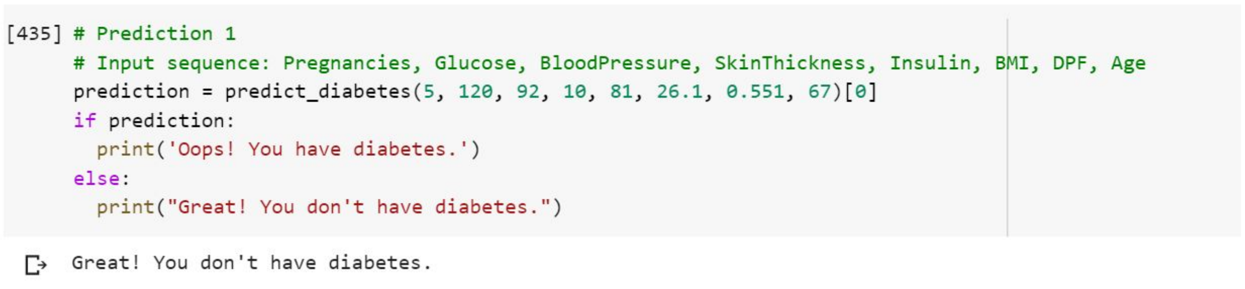
```
              precision    recall  f1-score   support

           0       0.84      0.92      0.88       107
           1       0.76      0.62      0.68        47

    accuracy                           0.82       154
   macro avg       0.80      0.77      0.78       154
weighted avg       0.82      0.82      0.82       154
```

Confusion matrix for SVM Model - Test Set



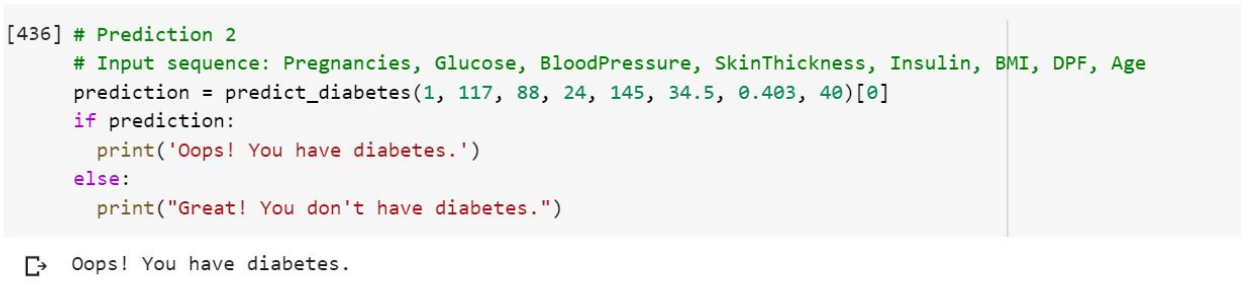
Predictions:

Sample predictions using model created by me-



```
[435] # Prediction 1
      # Input sequence: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DPF, Age
      prediction = predict_diabetes(5, 120, 92, 10, 81, 26.1, 0.551, 67)[0]
      if prediction:
        print('Oops! You have diabetes.')
      else:
        print("Great! You don't have diabetes.")
```

```
Great! You don't have diabetes.
```



```
[436] # Prediction 2
      # Input sequence: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DPF, Age
      prediction = predict_diabetes(1, 117, 88, 24, 145, 34.5, 0.403, 40)[0]
      if prediction:
        print('Oops! You have diabetes.')
      else:
        print("Great! You don't have diabetes.")
```

```
Oops! You have diabetes.
```

Conclusions:

Models

1. **Artificial Neural Network** (paper)
   - Accuracy on test set ANN: **77.27% vs Expected 75.7%** according to paper

2. **Random forest** (paper)
   - Accuracy on test set RF: **79.22% vs Expected 74.7%** according to paper

3. **Logistic Regression** (additional)
   - Score of **0.7691** on training set

4. **Decision Tree** (additional)
   - Score of **0.7317** on training set

5. **Support Vector Machine** (additional)
   - Score of **0.7739** on training set
   - Accuracy on test set SVM: **82.47%**

   Note- We can't directly compare the accuracy as during implementation I have split the dataset in 80% and 20% train and test respectively where as in paper they seem to have used the complete dataset for train and there isn't clarity on the test dataset used by them. Still a comparison made to give the report a closure.

By Anekait Kariya
2017A7PS0031G