A PROJECT ON

Diabetes detection using KNN and SVM

Submitted to Manipal University, Jaipur
Towards the partial fulfillment for the Award of the Degree of

BACHELOR OF TECHNOLOGY
In Information Technology
2020-2021

By

Rajat Goyal (189302034)
Aditya Sankrityayan (189302041)

MANIPAL UNIVERSITY
JAIPUR

Under the guidance of
Dr. Anju Yadav

Department of Information Technology
School of Computing and Information Technology
Manipal University Jaipur
Jaipur, Rajasthan

Introduction (Problem statement)

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict **whether or not a patient has diabetes**, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

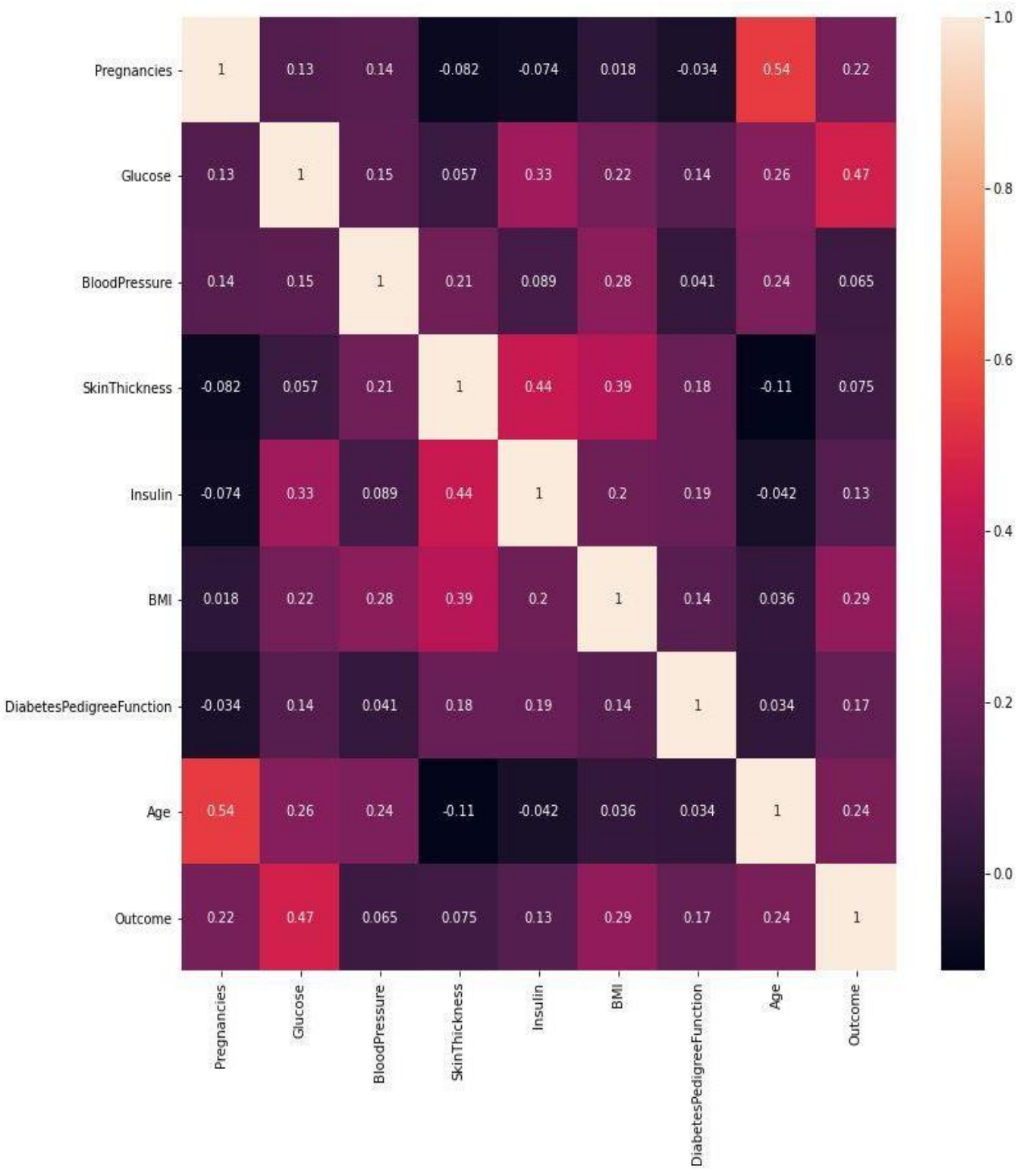
Dataset description

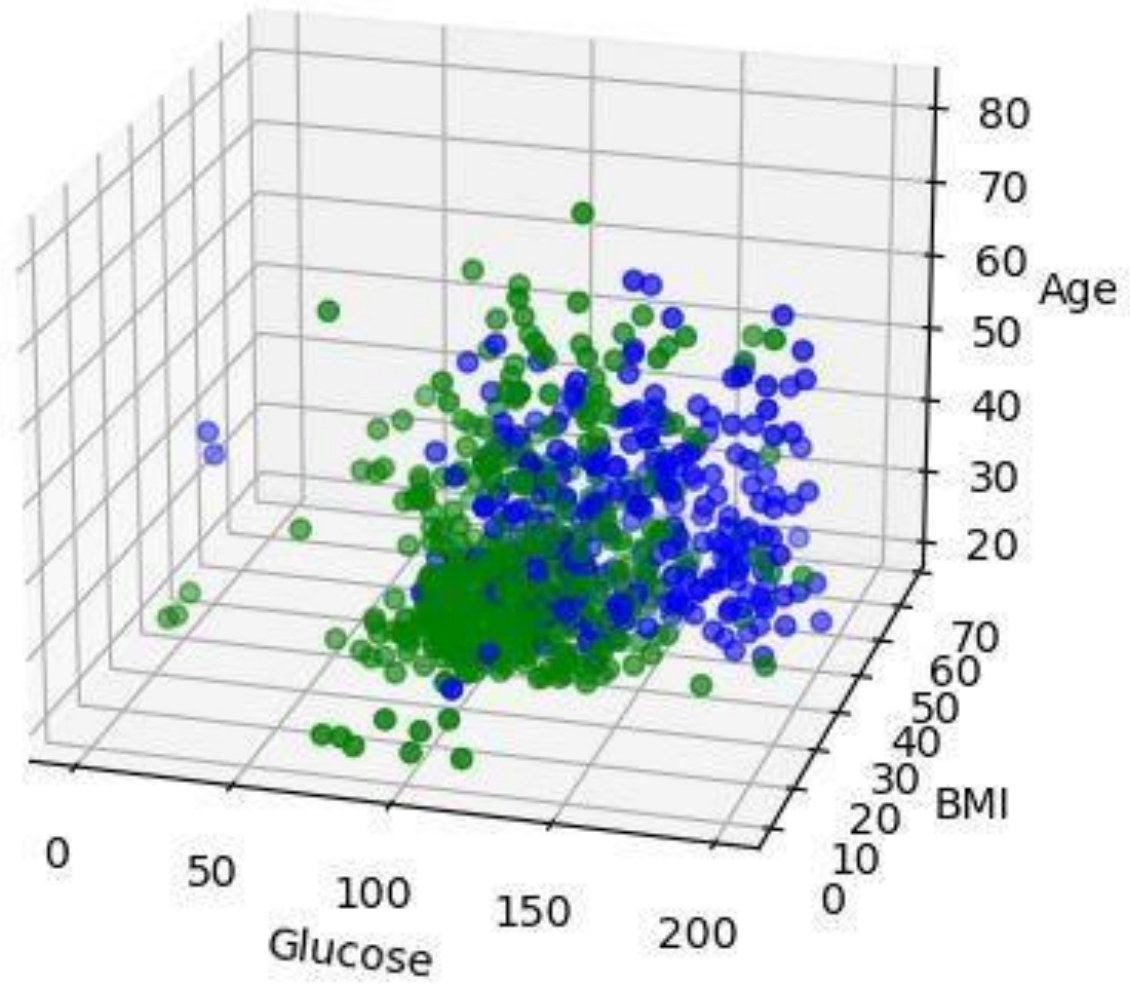
The datasets consist of **eight medical predictor variables** and **one target variable**, `Outcome`. Predictor variables include:

- The number of pregnancies the patient has had
- BMI
- Insulin level
- Glucose
- Age
- Blood pressure
- Skin thickness
- Diabetes pedigree function.
- There are a total of **768** entries.

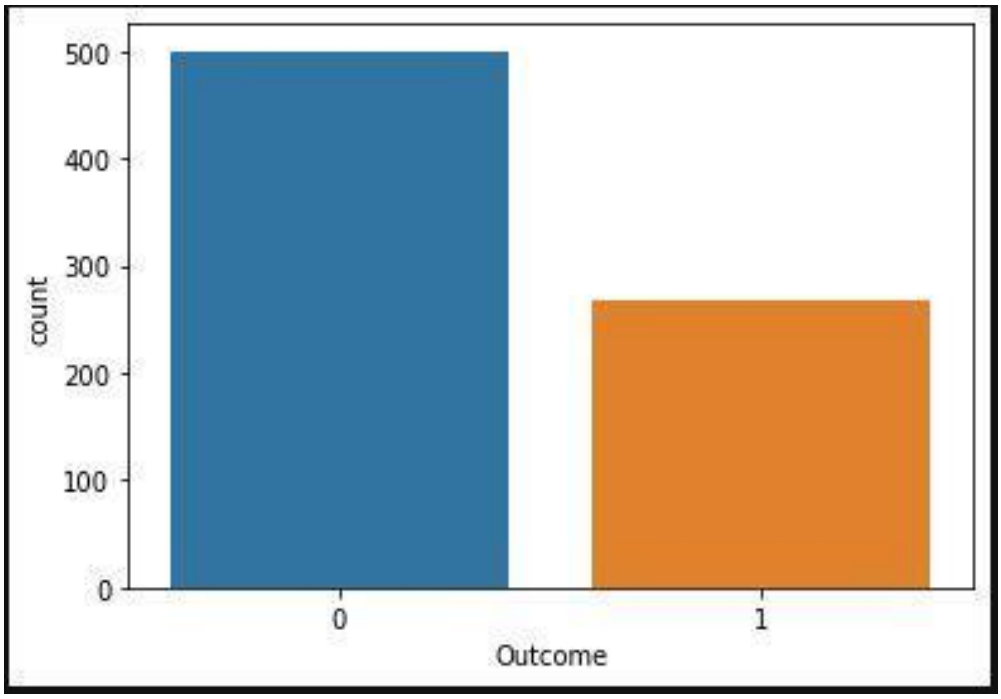
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0

Dataset visualization





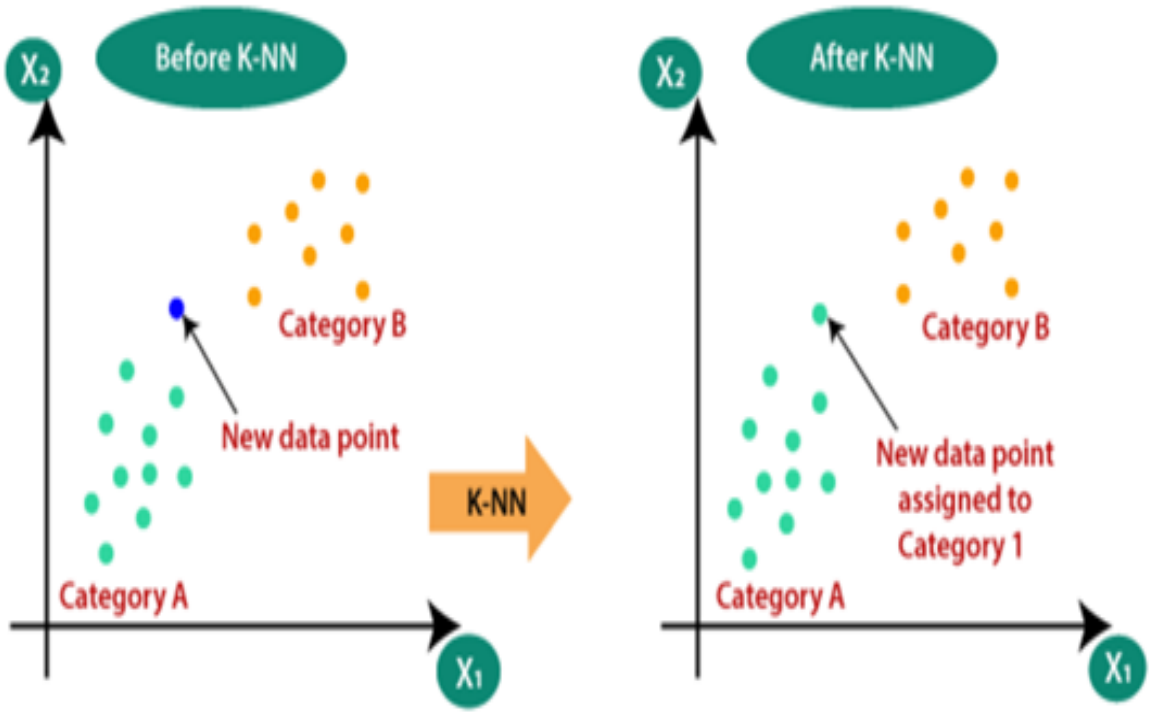
We picked the 3 features (Glucose, Age, BMI) with the highest correlation values with the outcome from heatmap and plotted a 3d scatter graph for them where green points represent the outcome 0 and blue points represent the outcome 1.



There is total 268 patient with the outcome = 1 meaning they are diagnosed with diabetes and the remaining 500 patients with the outcome = 0 aren’t diagnosed with diabetes. It is binary classification.

KNN (K-nearest neighbour)

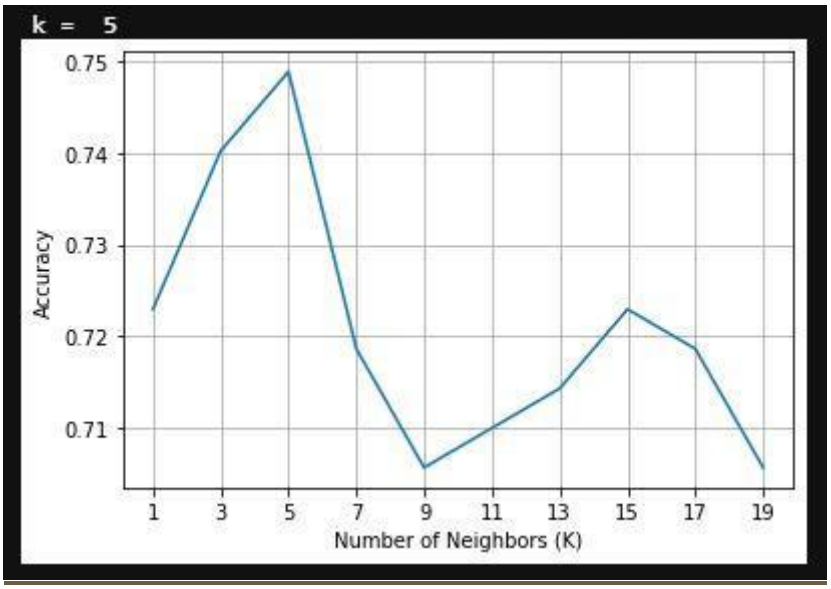
K-nearest neighbours (KNN) algorithm uses ‘feature similarity’ to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set.



The K-NN working can be explained based on the below algorithm:

- **Step-1:** We run a loop of odd values ranging from 1 to K as even values could result in equal number of points belonging to two classes in binary classification and end up in no result.
- **Step-2:** Compute the accuracy list for each value of K and sort it according to its index to obtain the value of K for which the accuracy is maximum and train the model for that K.
- **Step-3:** Calculate the Euclidean distance of all points to the point we are predicting.
- **Step-4:** Take the K nearest neighbours as per the calculated Euclidean distance.
- **Step-5:** Among these k neighbours, count the number of the data points in each category.
- **Step-6:** Assign the new data points to that category for which the number of the neighbour is maximum.

How to select the value of K



Formula used for Distance calculation:

$$Euclidean = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

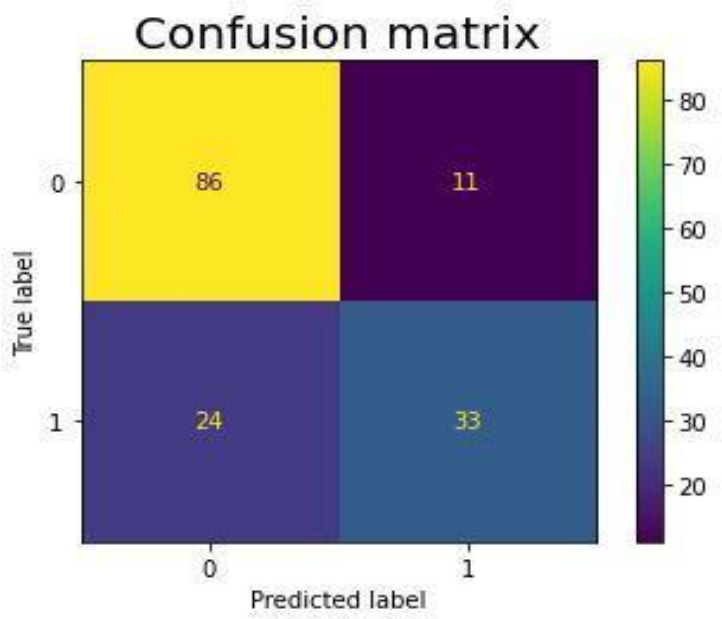
$$Manhattan = \sum_{i=1}^{k} |x_i - y_i|$$

$$Minkowski = \sqrt[q]{\sum_{i=1}^{k} (|x_i - y_i|)^q}$$

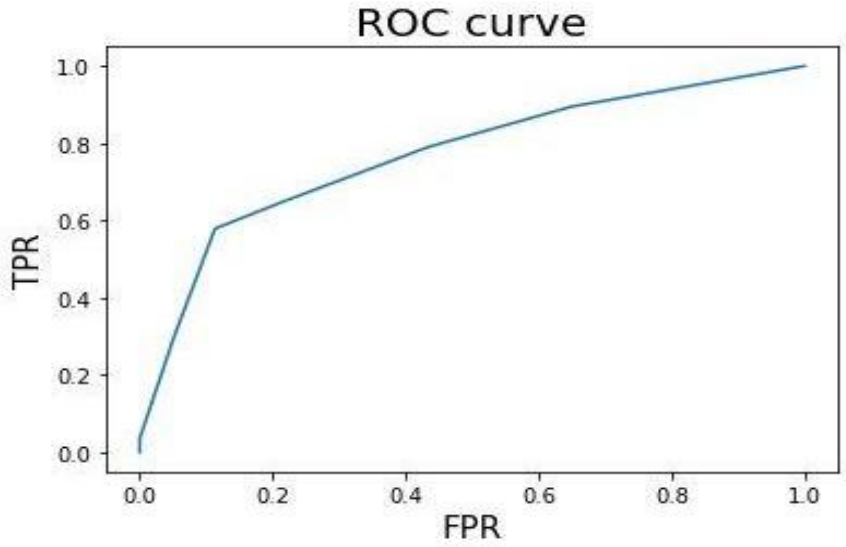
Result analysis

- **80% Train- 20% Test split**

➔ Confusion matrix

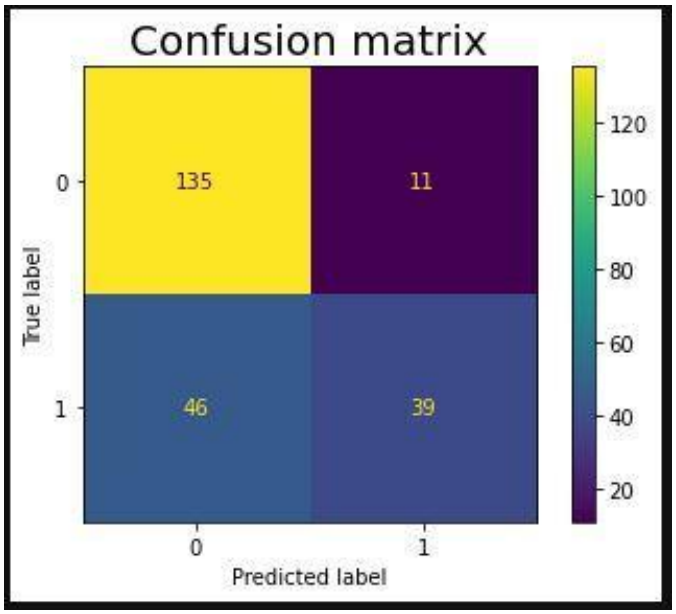


➔ ROC Curve

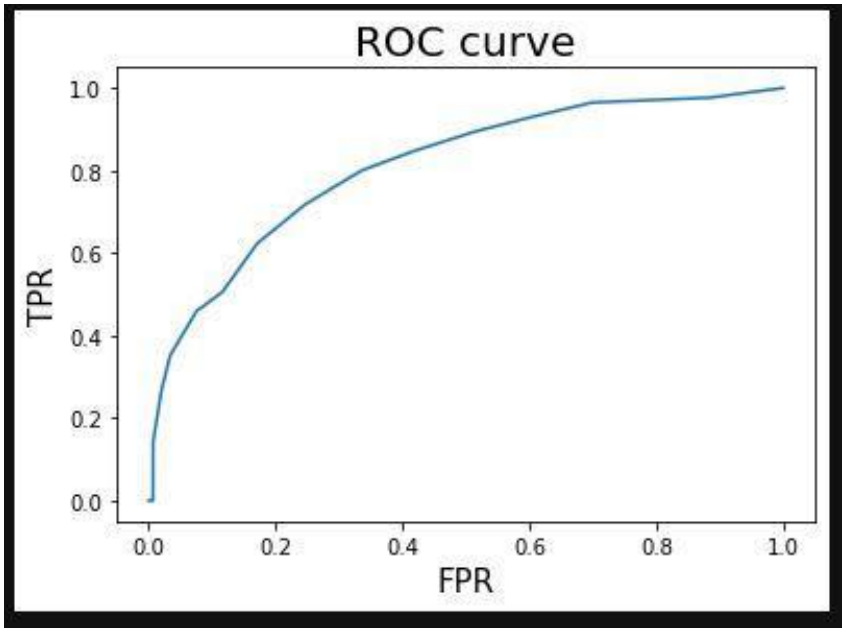


- **70% Train- 30% Test split**

➔ Confusion matrix

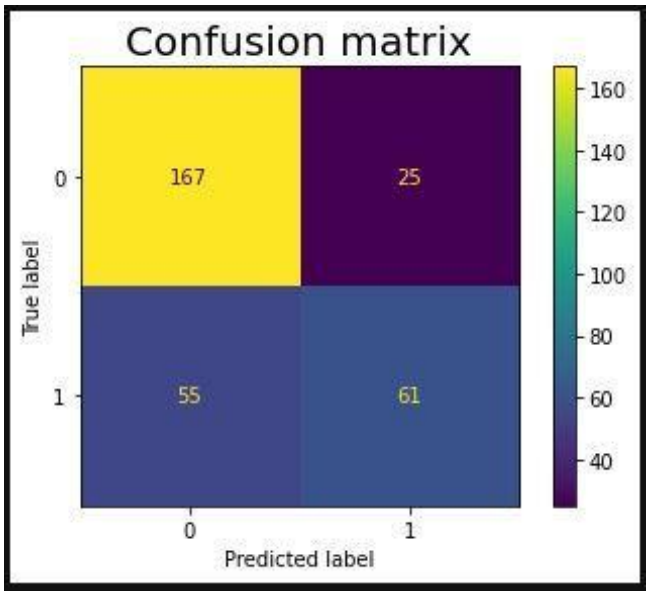


➔ ROC Curve

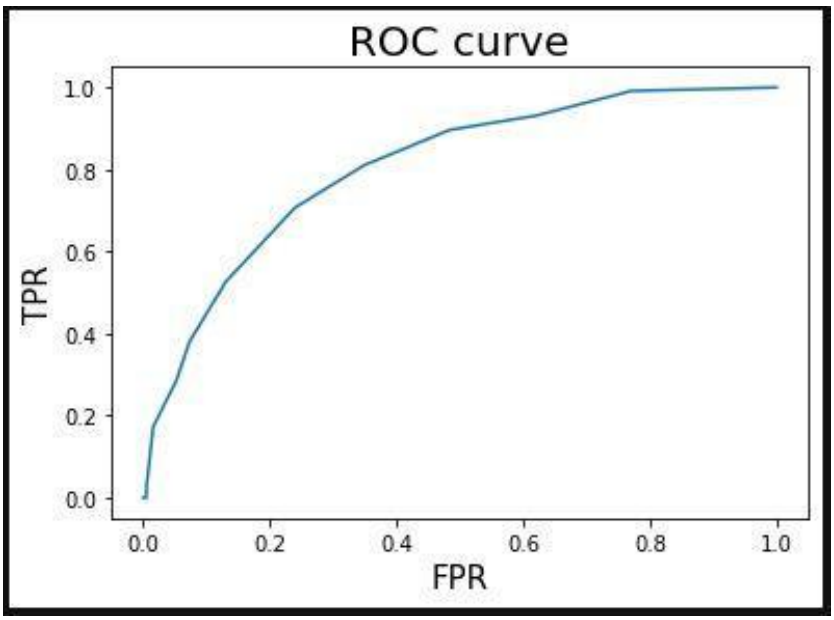


- **60% Train- 40% Test split**

➔ Confusion matrix



➔ ROC Curve



Other measures

Measures	80:20	70:30	60:40
Specificity	0.89	0.87	0.87
Sensitivity	0.58	0.51	0.53
Accuracy	0.77	0.75	0.74
Precision	0.75	0.68	0.71
FPR	0.11	0.13	0.13
FNR	0.42	0.49	0.47
NPV	0.78	0.77	0.75
FDR	0.25	0.32	0.29
F1-Score	0.65	0.59	0.60
MCC	0.50	0.42	0.43

SVM (Support Vector Machine)

Support Vector Machine (SVM) is a supervised machine learning algorithm which is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well. Support Vectors are simply the coordinates of individual observation. Maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called Margin.

Equation of hyperplane:

Maximum margin/Optimal hyperplane: w.x+b=0

Positive hyperplane: w.x+b=+1

Negative hyperplane: w.x+b=-1

$$Hyperplane\ equation = w.x + b = 0$$

$$let\ nearest\ point = x^p$$

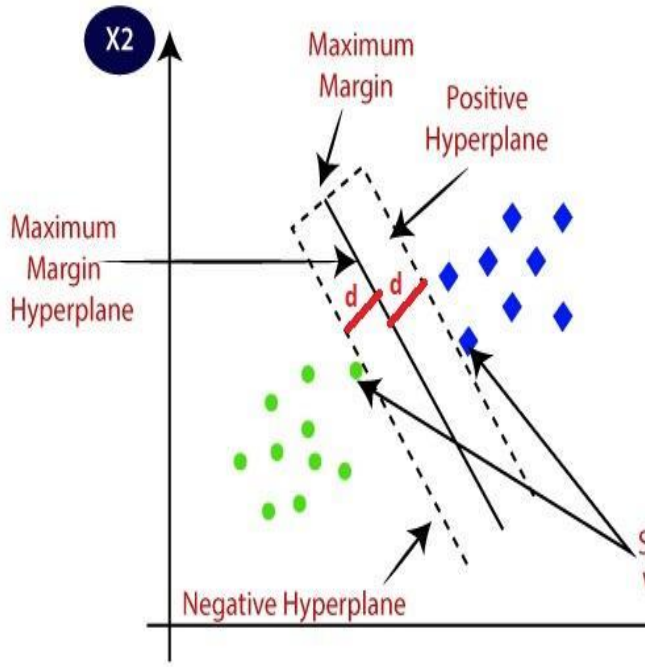
$$distance\ between\ plane\ and\ point = \frac{w.x^p + b}{||w||}$$

$$Let\ w.x^p + b = \gamma$$

For SVM we maximize distance,

$$\Rightarrow = \frac{\gamma}{||w||}$$

$$\frac{w.x^i + b}{||w||} \geq \frac{w.x^p + b}{||w||} \ \forall\ i$$



$$ax + b = 0$$

$$ax + b_2 = 0$$

$$\frac{|b - b_2|}{||a||} = d$$

$$b_2 = b \pm d||a||$$

+ve and -ve hyperplane

$$w.x + b \pm d||w|| = 0$$

$$divide\ by\ d||w||$$

$$\frac{w}{d||w||} + \frac{b}{d||w||} = 1$$

$$w.x + b = 1$$

$$w.x + b = -1$$

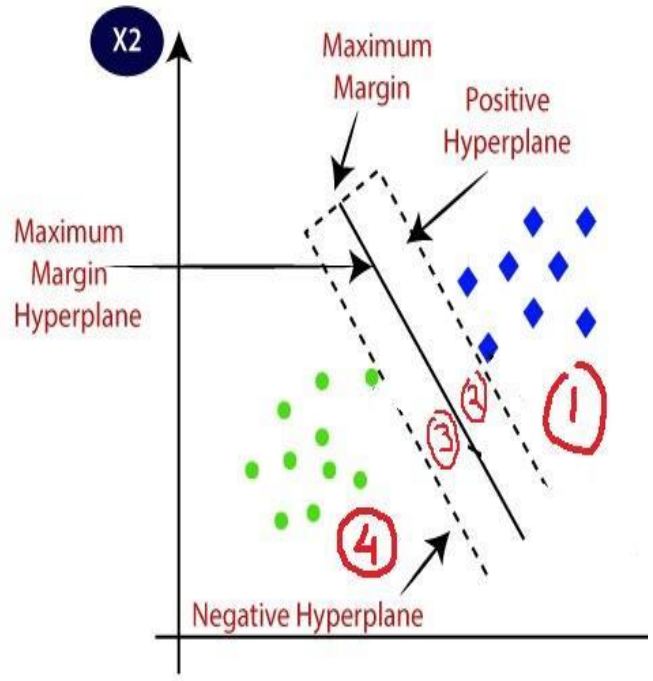
Distance between hyperplane and +ve plane =

$$= \frac{|b - (b-1)|}{||w||}$$

$$= \frac{1}{||w||}$$

$$|w.x^i + b| \geq 1$$

$$y^i(w.x^i + b) \geq 1$$



(i) Region-1:

$If\ y^i = -1 \Rightarrow y^i(w.x^i + b) \leq -1\ but\ it\ satisfies\ y^i = 1$

(ii) Region-4:

$Vice\ versa\ of\ this\ for\ region - 4$

Since we change constraint, we make changes in cost function as well

$$\frac{1}{||w||}$$

$$\frac{||w||^2}{2} + c\sum_{i=1}^{m} \in^i$$

$Note: For\ y^i = 1\ in\ region - 1\ and\ y^i = -1\ in\ region - 4, \in^i = 0$

From the constraint:

$$\in^i = 1 - y^i(w.x^i + b)$$

$$\in^i = max\ ((1 - y^i(w.x^i + b)),0)$$

Now simply apply gradient descent

$$\frac{\partial}{\partial w} = \frac{\partial}{\partial w}\left|\left|w\right|\right|^2 + \frac{\partial}{\partial w} c \sum \quad \in^i$$

$$= \frac{\partial}{\partial w} w^T w + \frac{\partial}{\partial w} \sum \quad \begin{cases} 0 & y^i(w.x^i + b) \\ \geq 1\ 1 - y^i(w.x^i + b) & y^i(w.x^i + b) < 1 \end{cases}$$

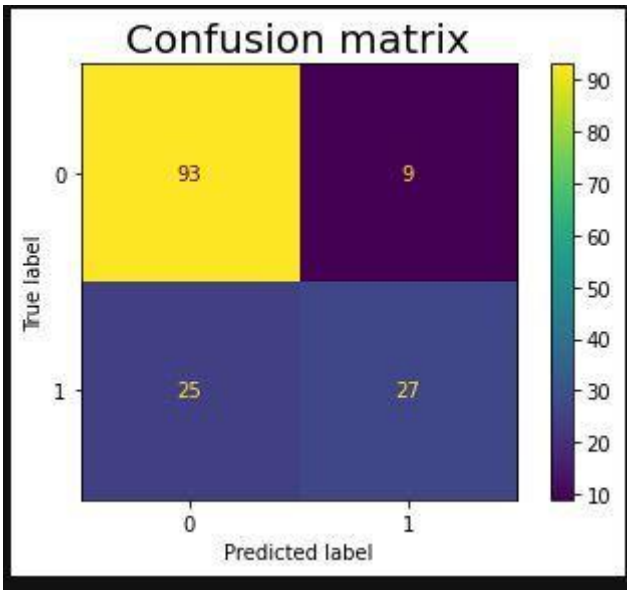
$$= w + c \sum \quad \begin{cases} 0 & y^i(w.x^i + b) \\ \geq 1 - y^i.x^i & y^i(w.x^i + b) < 1 \end{cases}$$

$$\frac{\partial}{\partial b} = c \sum \quad \begin{cases} 0 & y^i(w.x^i + b) \\ \geq 1 - y^i & y^i(w.x^i + b) < 1 \end{cases}$$

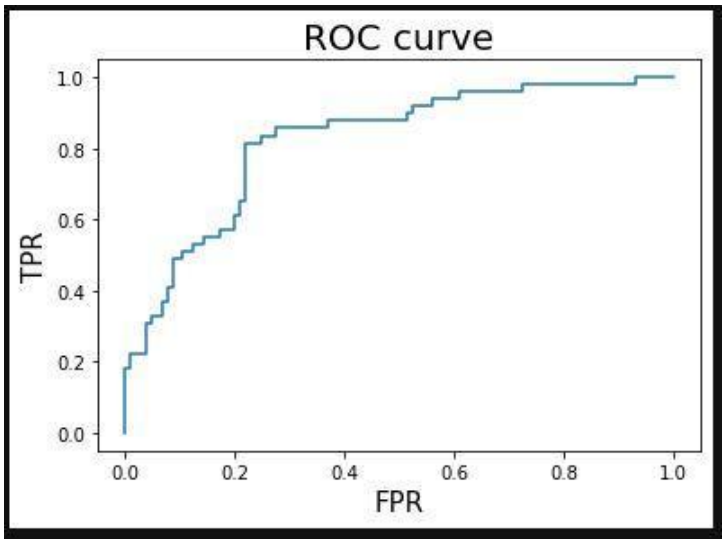
Result analysis

- **80% Train- 20% Test split**

➔ Confusion matrix

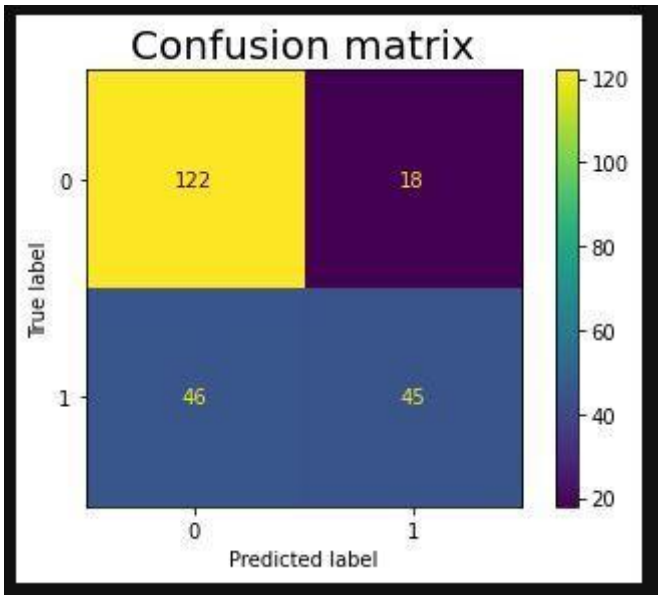


➔ ROC Curve

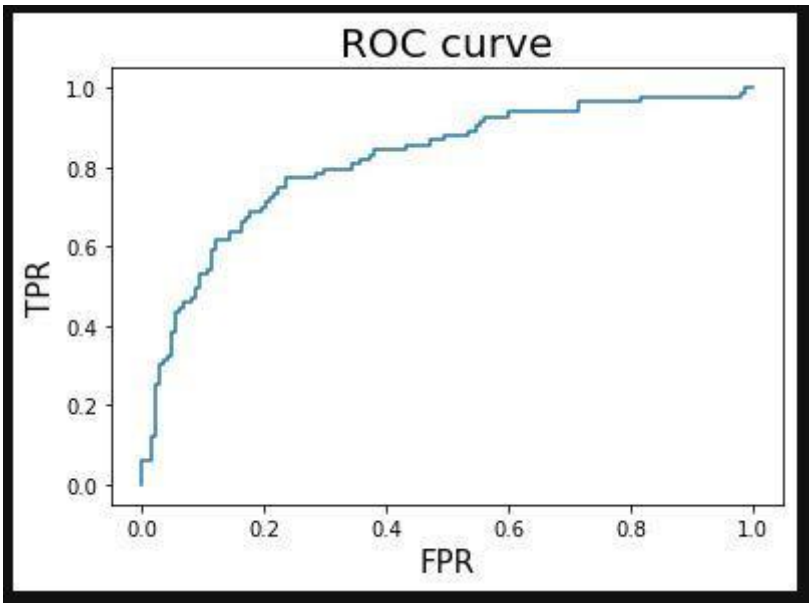


- **70% Train- 30% Test split**

➔ Confusion matrix

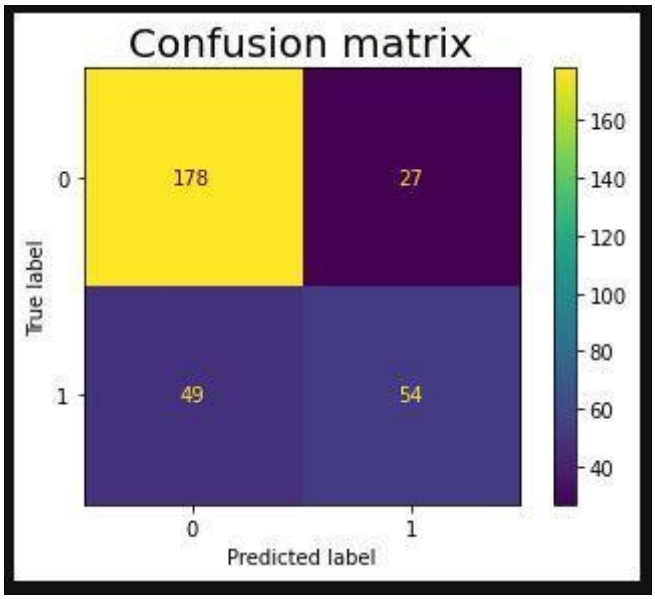


➔ ROC Curve

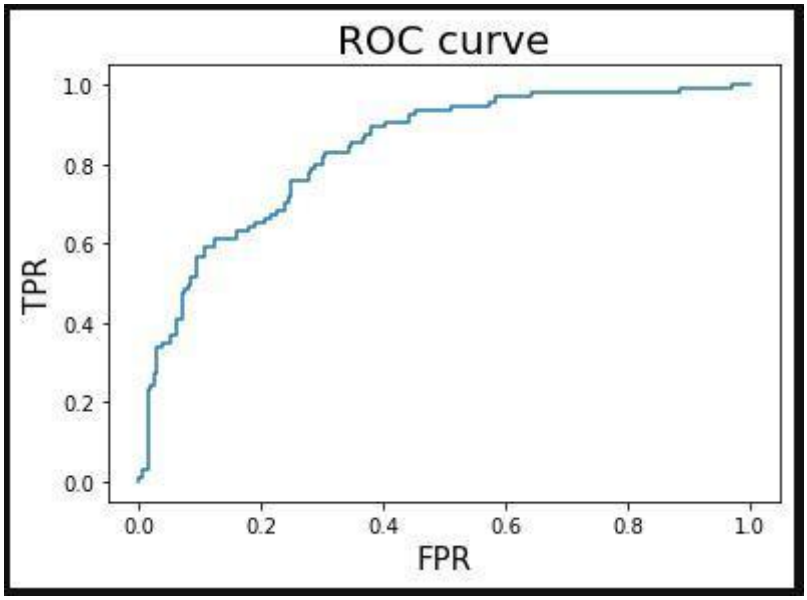


- **60% Train- 40% Test split**

➔ Confusion matrix



➔ ROC Curve



Other measures

Measures	80:20	70:30	60:40
Specificity	0.91	0.87	0.868
Sensitivity	0.52	0.49	0.52
Accuracy	0.78	0.72	0.74
Precision	0.75	0.71	0.67
FPR	0.09	0.13	0.132
FNR	0.48	0.51	0.47
NPV	0.79	0.73	0.78
FDR	0.25	0.29	0.33
F1-Score	0.61	0.58	0.59
MCC	0.48	0.40	0.42