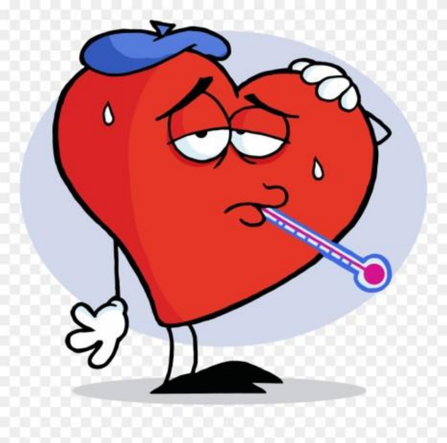
PREDICTING THE RISK OF HEART DISEASE: A CLASSIFICATION PROBLEM



PROJECT WORK

GROUP MEMBERS:

- Saswat Nanda
- Debargha Samanta

SUPERVISORS:

- Mr. Akash Maurya

PROBLEM STATEMENT :-

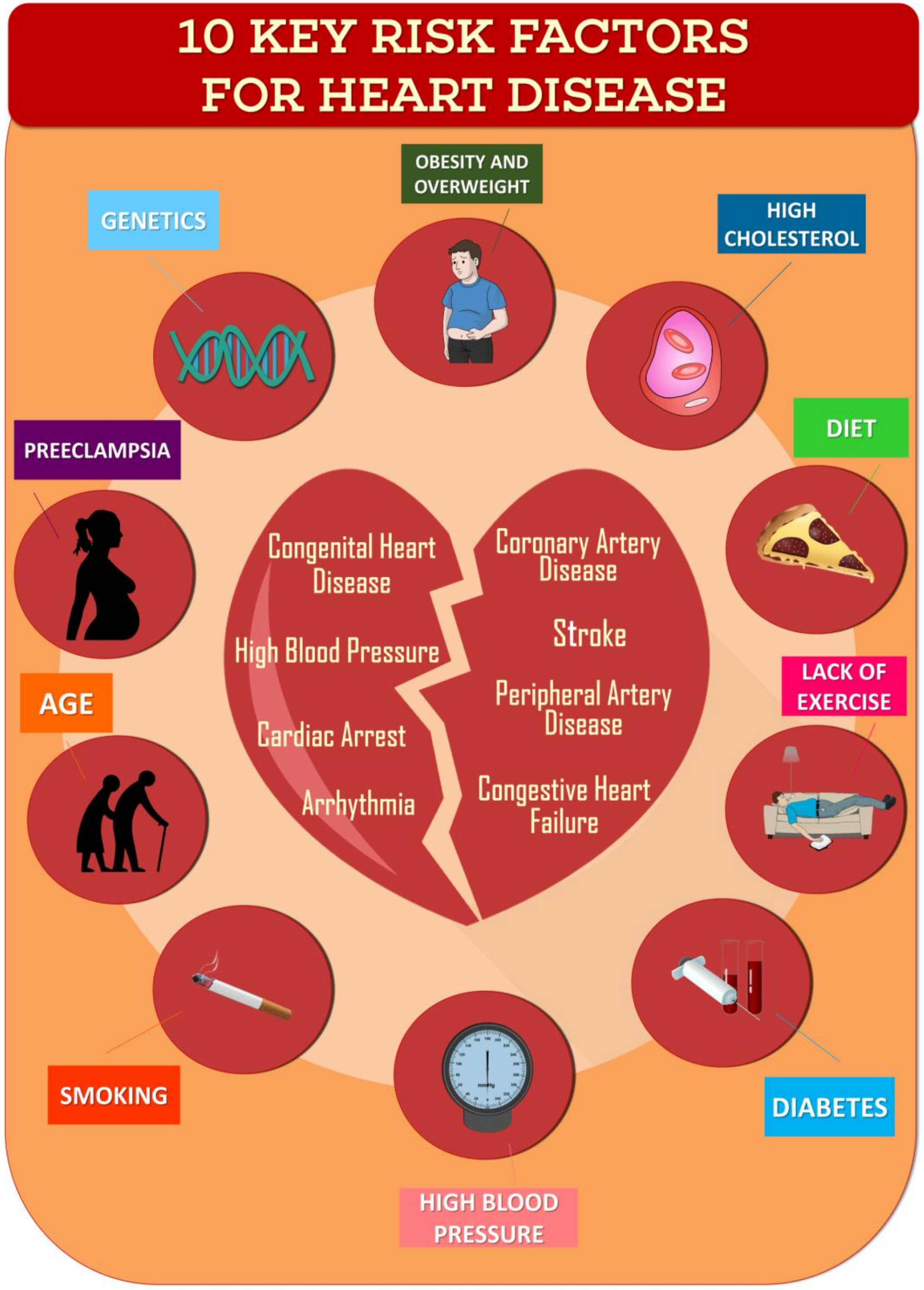
The aim of the project is to classify, whether the patient is going to have a heart disease or not in the upcoming ten years based on his/her habits and current health status. This can be of a great use to the society as it can be possible to prevent or at least reduce the increasing rates of heart diseases among people.

OBJECTIVE :-

We carry out the task with help of some classification techniques as well as some boosting techniques and at the end we are going to find out which classification and boosting technique works well for the given dataset.

SOFTWARES USED :-

- Python
- MS Excel



DATA DESCRIPTION AND SOME CHALLENGES FACED

Variables Used and it’s data types

Covariates:

- Gender – categorical (binary), nominal
- Age – continuous
- Education – categorical, nominal
- Current Smoker – categorical (binary), nominal
- Cigarettes consumed per day – continuous
- Intake of BP Medicines – categorical (binary), nominal
- Prevalent Stroke – categorical (binary), nominal
- Hypertensive – categorical (binary), nominal
- Diabetic - categorical (binary), nominal
- Total Cholesterol Level – continuous
- Systolic Blood Pressure – continuous
- Diastolic Blood Pressure – continuous
- Body Mass Index – continuous
- Heart Rate per minute – continuous
- Glucose Level – continuous

Response:

- Heart Disease - categorical (binary), nominal

Challenges faced

Issue of Missing data

- Replaced with column mean for continuous covariates.
- Omitted the missing values for categorical covariates

Issue of response class imbalance

Resolved the issue using **Oversampling** technique.

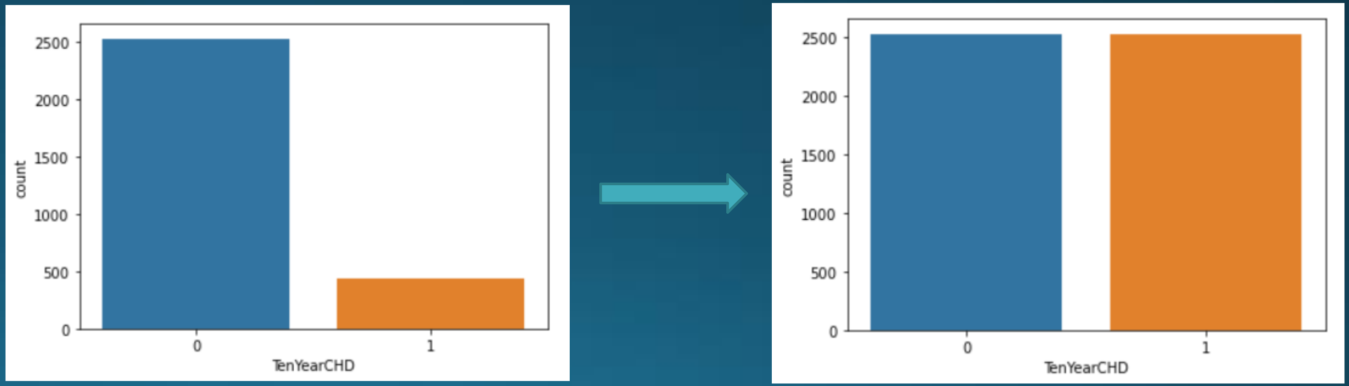
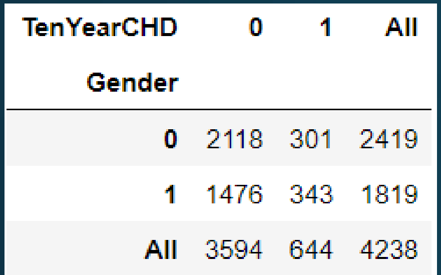


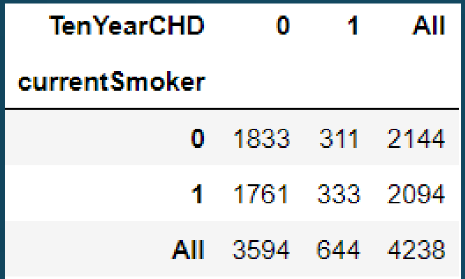
Fig.: Oversampling technique solving the issue of response class imbalance

EXPLORATORY DATA ANALYSIS

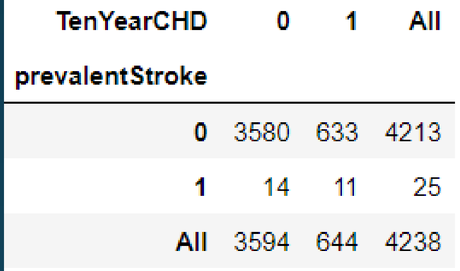
Let’s have a look at some contingency tables which shows the relation of the categorical variables with the response variable which is also categorical in nature.



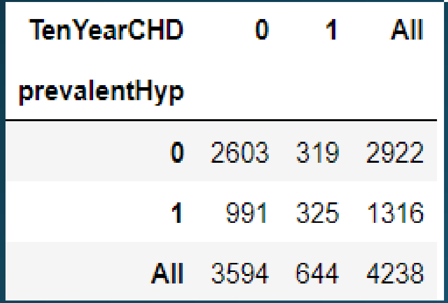
TenYearCHD	0	1	All
Gender			
0	2118	301	2419
1	1476	343	1819
All	3594	644	4238



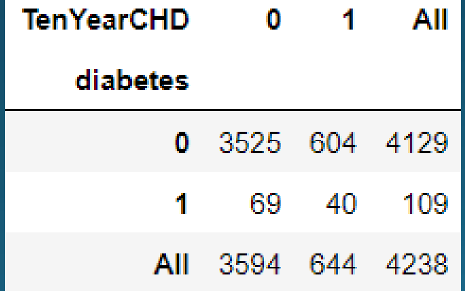
TenYearCHD	0	1	All
currentSmoker			
0	1833	311	2144
1	1761	333	2094
All	3594	644	4238



TenYearCHD	0	1	All
prevalentStroke			
0	3580	633	4213
1	14	11	25
All	3594	644	4238

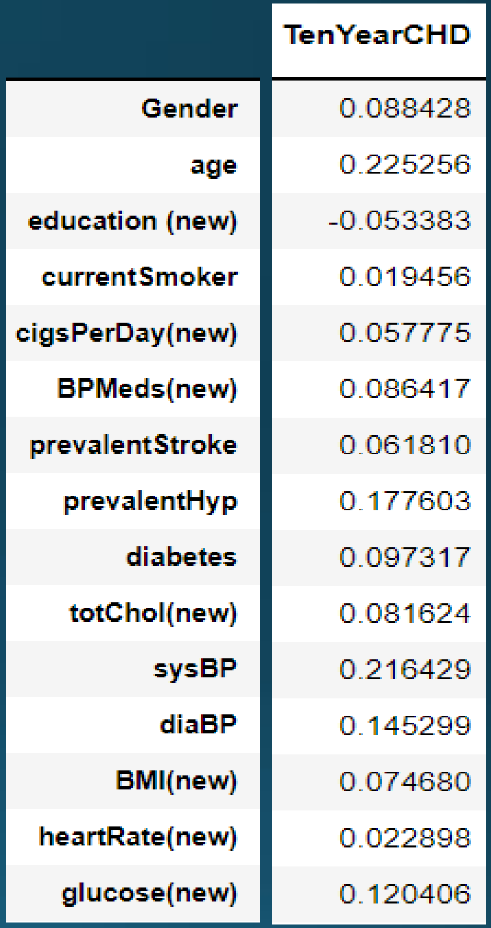


TenYearCHD	0	1	All
prevalentHyp			
0	2603	319	2922
1	991	325	1316
All	3594	644	4238



TenYearCHD	0	1	All
diabetes			
0	3525	604	4129
1	69	40	109
All	3594	644	4238

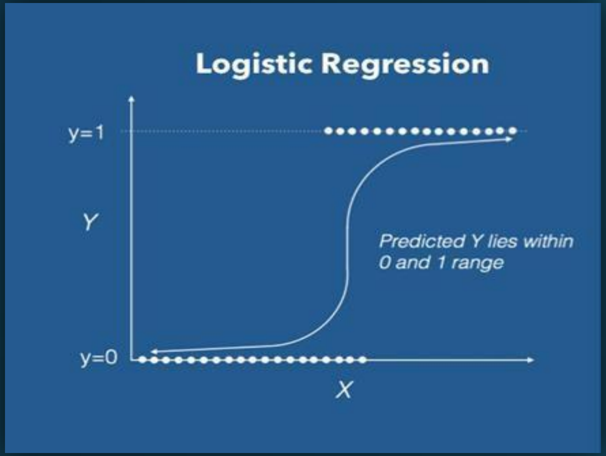
The contingency table gives us an idea about what proportion of a particular category of a feature variable faced a heart disease in the near future.



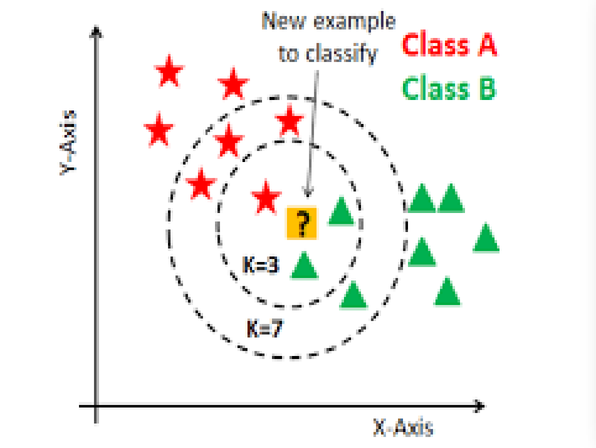
	TenYearCHD
Gender	0.088428
age	0.225256
education (new)	-0.053383
currentSmoker	0.019456
cigsPerDay(new)	0.057775
BPMeds(new)	0.086417
prevalentStroke	0.061810
prevalentHyp	0.177603
diabetes	0.097317
totChol(new)	0.081624
sysBP	0.216429
diaBP	0.145299
BMI(new)	0.074680
heartRate(new)	0.022898
glucose(new)	0.120406

As we can observe from the correlation matrix, the feature variables Age and Systolic BP has relatively higher correlation with the response variable.

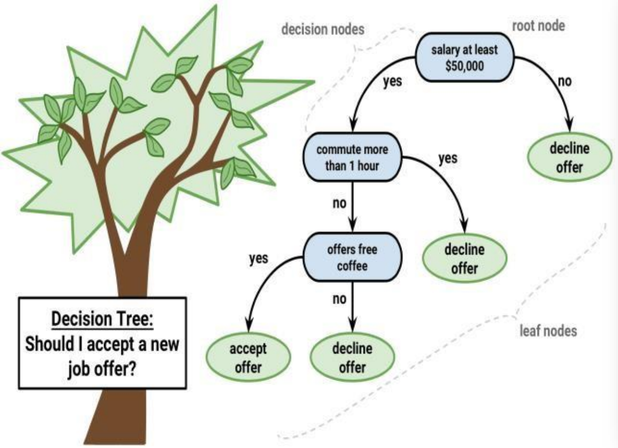
METHODS USED FOR CLASSIFICATION



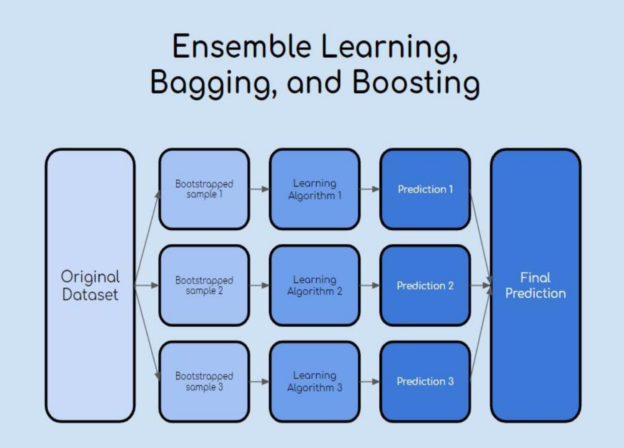
A. LOGISTIC REGRESSION



B. K NEAREST NEIGHBORS

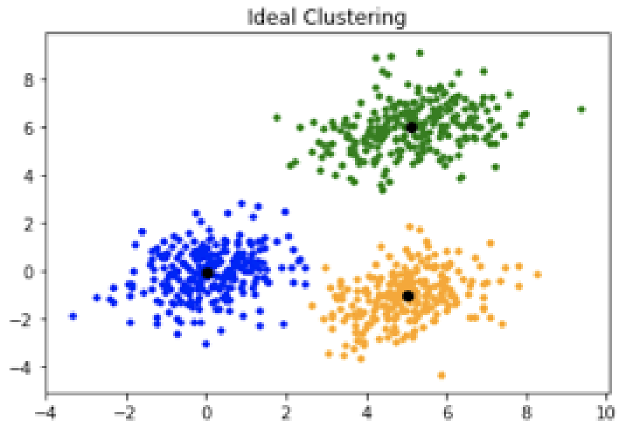


C. DECISION TREE



D. BOOSTINGTECHNIQUES

The classification techniques will be compared based on certain performance measures, namely, ACCURACY, PRECISION, RECALL, SPECIFICITY and F1 SCORE generated from the Confusion Matrix. We will find out which classification technique works well for this dataset.

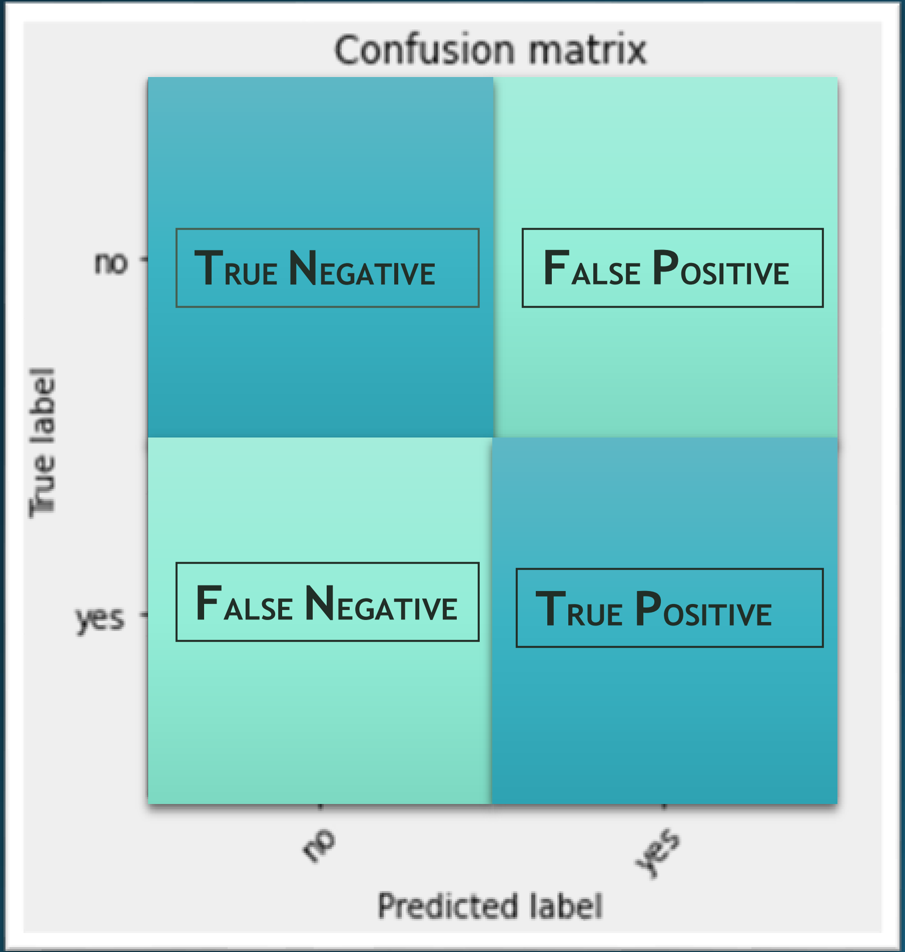


CLUSTER ANALYSIS FOR DATA SEGMENTATION:

(using unsupervised machine learning technique K-means clustering)

K-Means is a clustering algorithm which divides observations into k clusters. Since we can dictate the amount of clusters, it can be easily used in classification where we divide data into clusters which can be equal to or more than the number of classes.

CLASSIFICATION REPORT



Confusion matrix

True label \ Predicted label	no	yes
no	TRUE NEGATIVE	FALSE POSITIVE
yes	FALSE NEGATIVE	TRUE POSITIVE

This is a **Heart disease** data so here our priority is to predict the possibility of heart disease within ten years . Thus we have taken ‘YES’ as our most sensitive class. Also here our motive is to reduce **FN** (False Negative) because it’s more fatal in this case**.** Hence our aim is to

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{F1 score} = \frac{2*((\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}$$

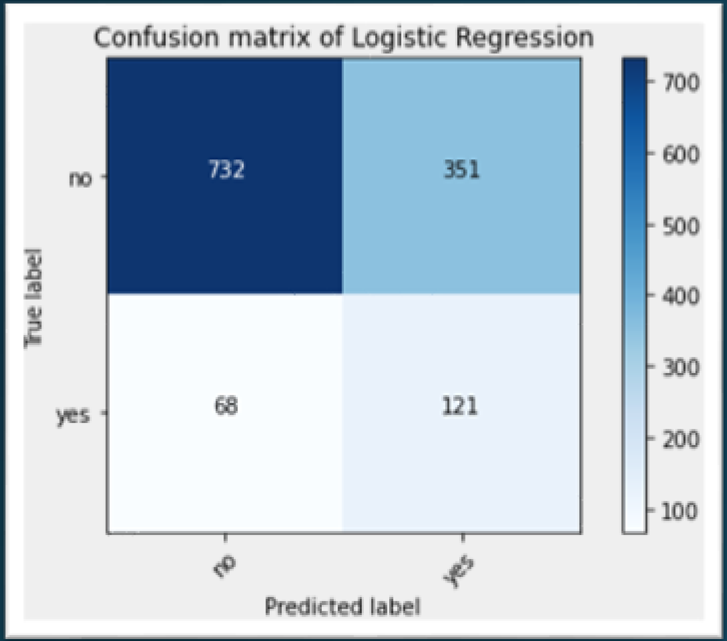
$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$\text{Misclassification} = 1 - \text{Accuracy}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

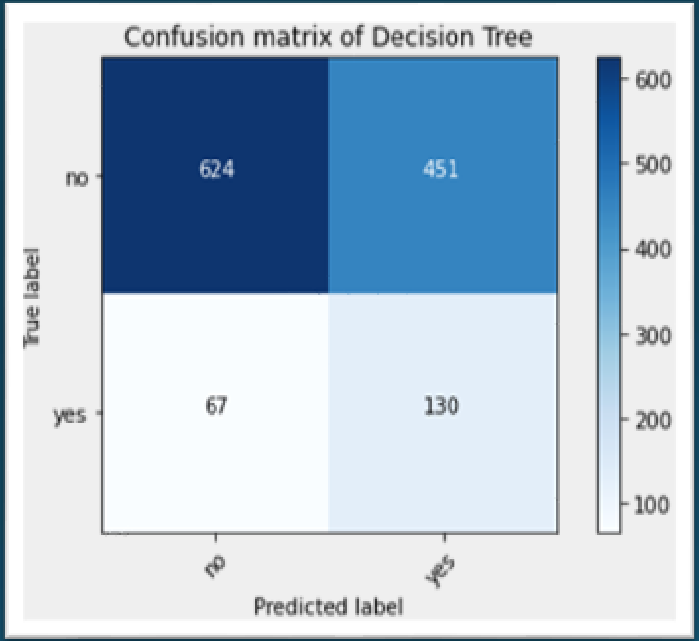
EVALUATION OF CLASSIFICATION TECHNIQUES

Logistic Regression



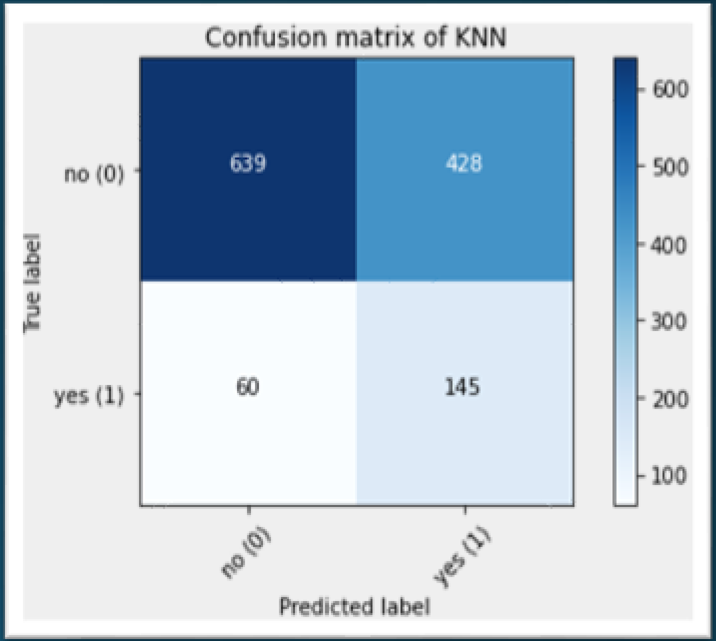
Recall :	64%
Precision:	26%
Accuracy:	67%
F1 score:	37%
Misclassification:	33%
Specificity:	68%

Decision Tree



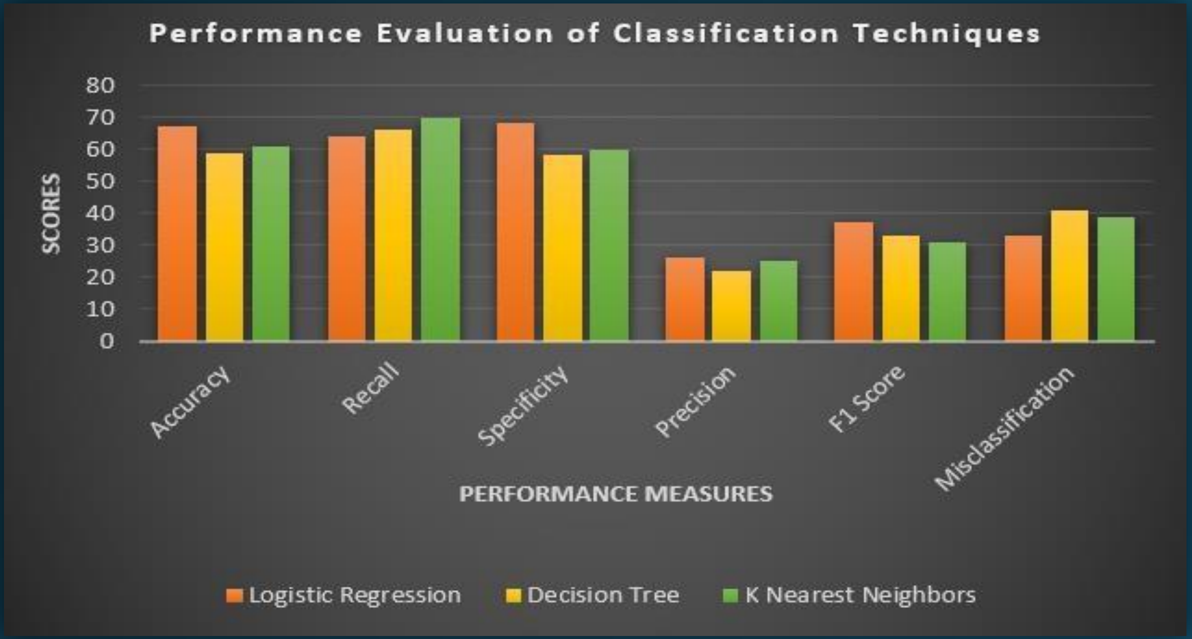
Recall :	66%
Precision:	22%
Accuracy:	59%
F1 score:	33%
Misclassification:	41%
Specificity:	58%

K-nearest neighbors



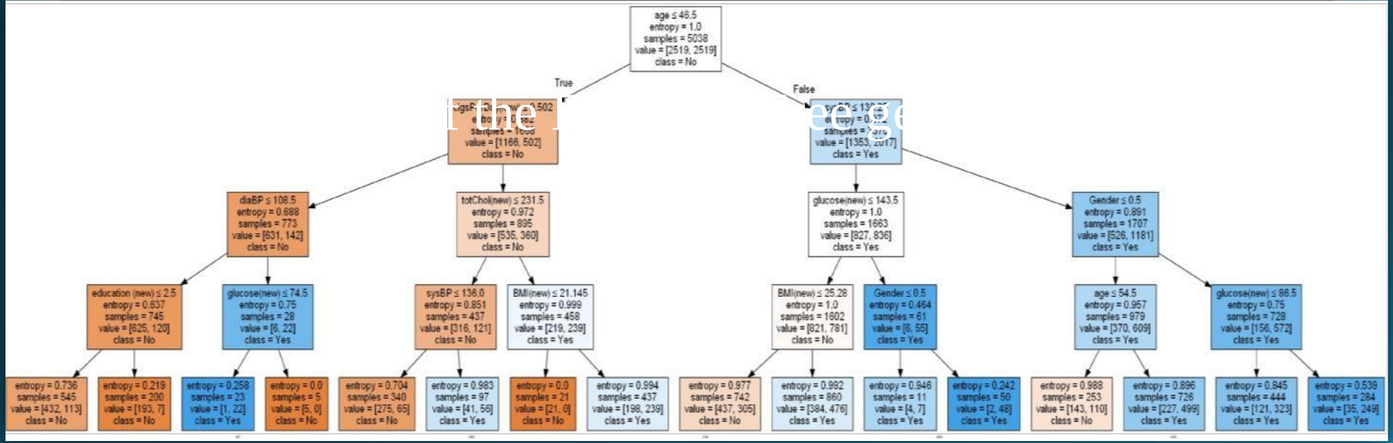
Recall :	70%
Precision:	25%
Accuracy:	61%
F1 score:	31%
Misclassification:	39%
Specificity:	60%

CONCLUSION OF THE CLASSIFICATION TECHNIQUES USED:

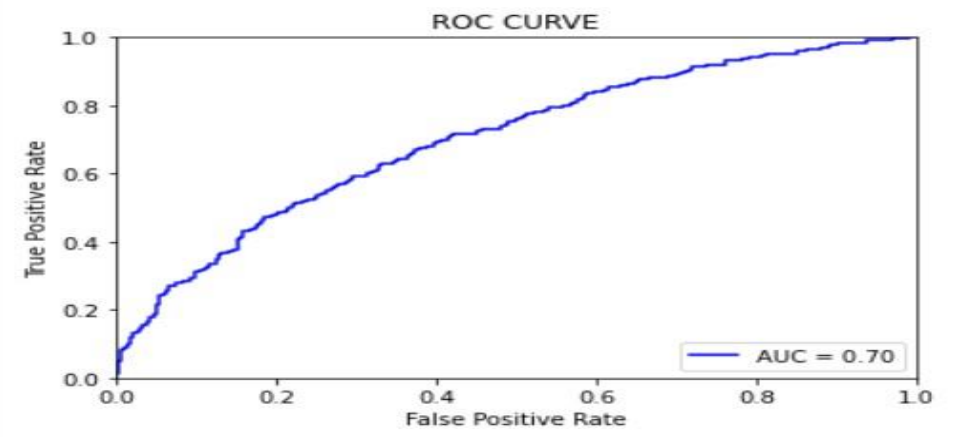


We can observe that Logistic Regression has the highest accuracy, precision, specificity and F1 score as compared to the other classification techniques used and it also has a very decent score for Sensitivity. So, we can easily conclude that **Logistic Regression** provided us the best results for this dataset of binary classification.

Visualization of the Decision Tree generated from the model



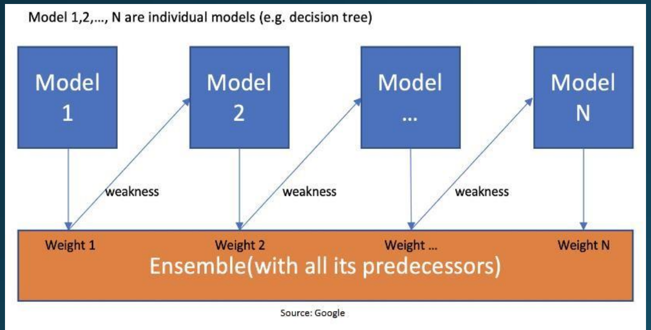
ROC Curve generated from Logistic Regression



BOOSTING TECHNIQUES

Boosting is an ensemble learning method that combines a set of weak learners into a strong learner to minimize training error s. In boosting, a random sample of data is selected, fitted with a model and then trained sequentially—that is, each model tries to compensate for the weaknesses of its predecessor. With each iteration, the weak rules from each individual classifier are combined to form one, strong prediction rule.

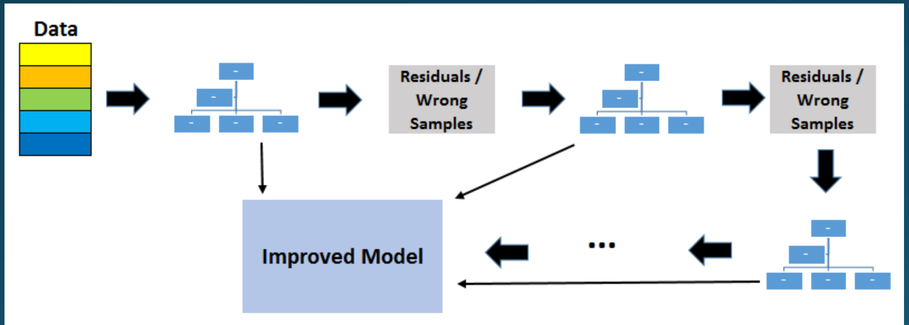
ADABOOST


Model 1,2,..., N are individual models (e.g. decision tree)
Model 1
Model 2
Model ...
Model N
weakness
weakness
weakness
Weight 1
Weight 2
Weight ...
Weight N
Ensemble(with all its predecessors)

Source: Google

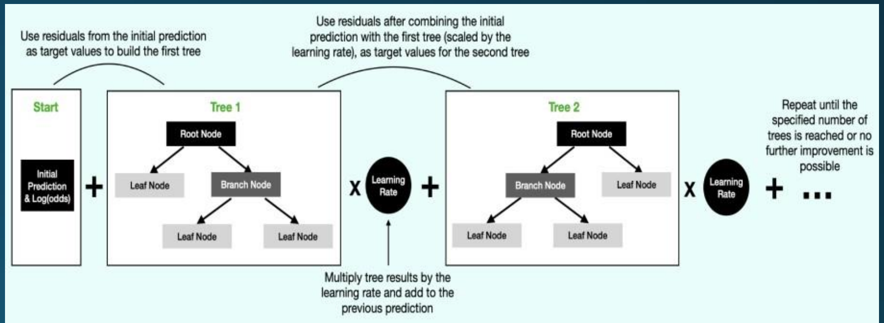
This method operates iteratively, identifying misclassified data points and adjusting their weights to minimize the training error. The model continues optimize in a sequential fashion until it yields the strongest predictor.

GRADIENT BOOST



This method works by sequentially adding predictors to an ensemble with each one correcting for the errors of its predecessor. The gradient boosting trains on the gradients of the previous predictor.

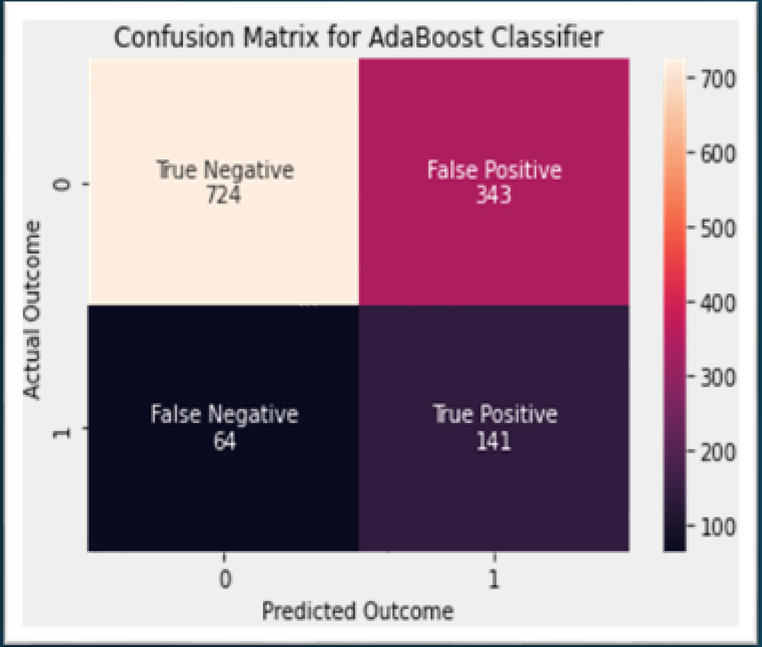
EXTREME GRADIENT BOOST


Use residuals from the initial prediction as target values to build the first tree
Use residuals after combining the initial prediction with the first tree (scaled by the learning rate), as target values for the second tree
Start
Initial Prediction & Log(odds)
+
Tree 1
Root Node
Leaf Node
Branch Node
Leaf Node
Leaf Node
X
Learning Rate
+
Tree 2
Root Node
Branch Node
Leaf Node
Leaf Node
Leaf Node
X
Learning Rate
+ ...
Repeat until the specified number of trees is reached or no further improvement is possible
Multiply tree results by the learning rate and add to the previous prediction

XGBoost is an implementation of gradient boosting that’s designed for computational speed and scale. XGBoost leverages multiple cores on the CPU, allowing for learning to occur in parallel during training.

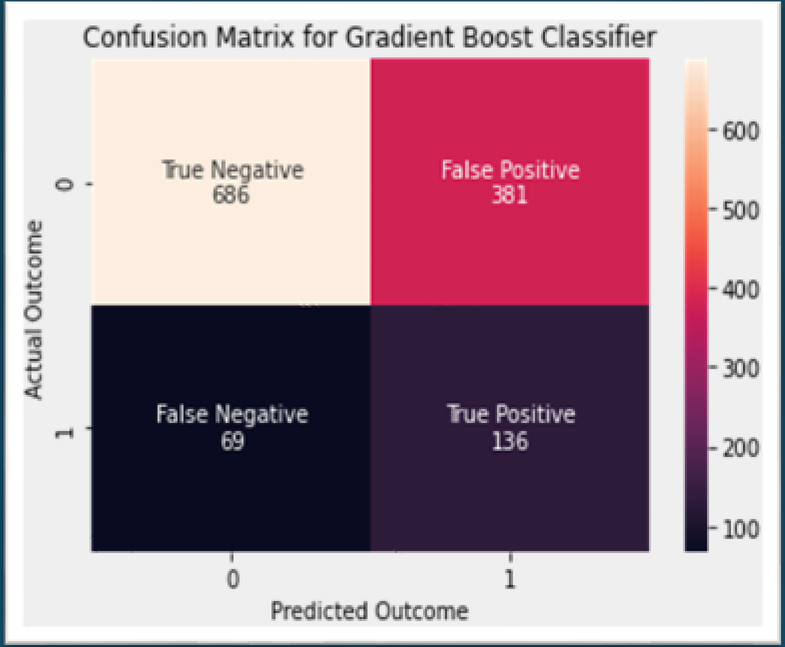
EVALUATION OF BOOSTING TECHNIQUES

ADA Boost



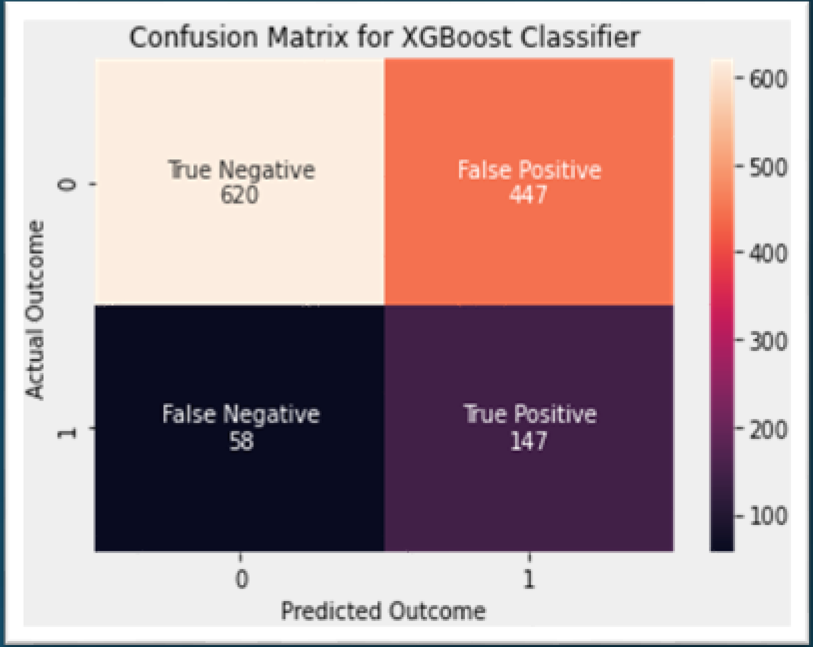
Recall :	69%
Precision:	29%
Accuracy:	68%
F1 score:	41%
Misclassification:	32%
Specificity:	68%

Gradient Boost



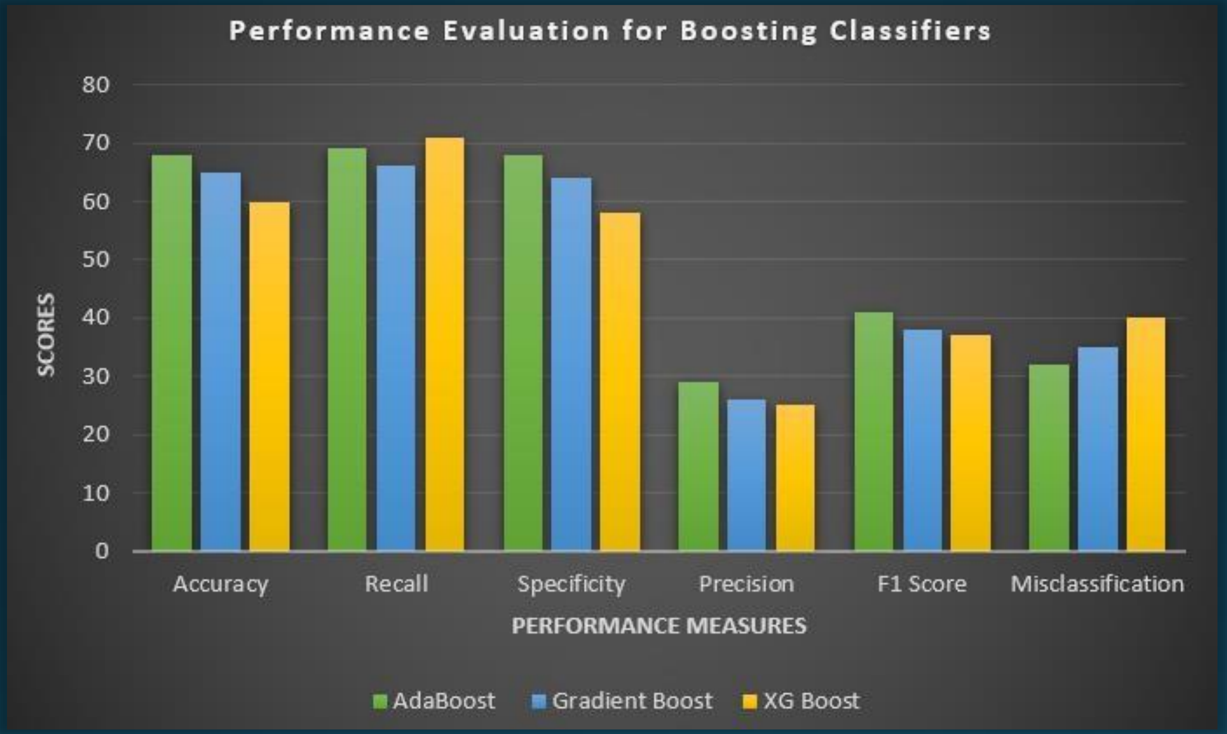
Recall :	66%
Precision:	26%
Accuracy:	65%
F1 score:	38%
Misclassification:	35%
Specificity:	64%

XG Boost



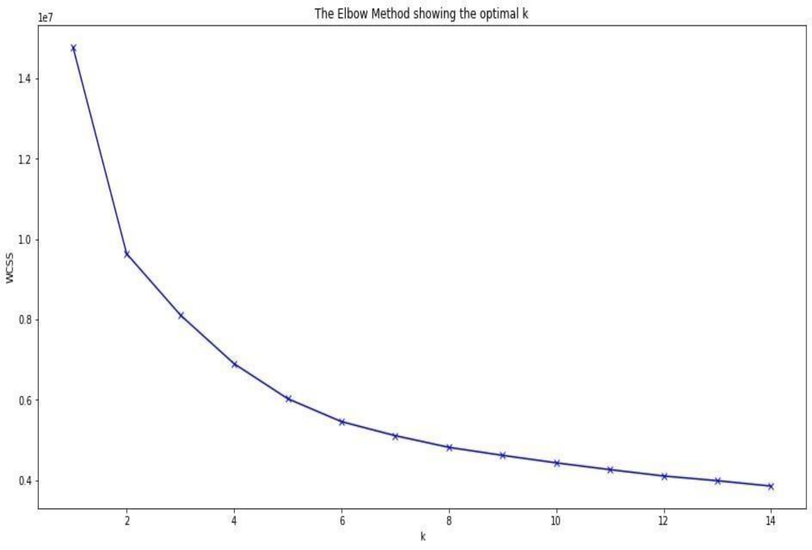
Recall :	71%
Precision:	25%
Accuracy:	60%
F1 score:	37%
Misclassification:	40%
Specificity:	58%

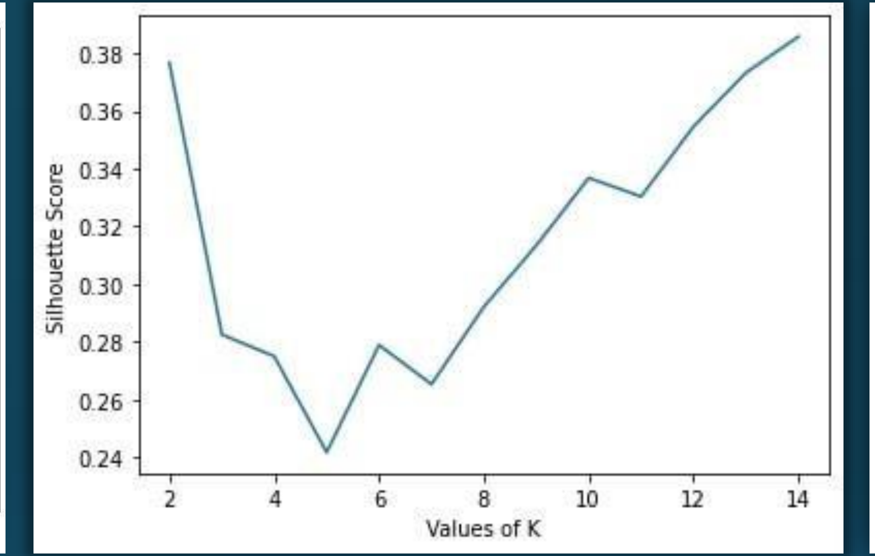
CONCLUSION OF THE BOOSTING TECHNIQUES USED

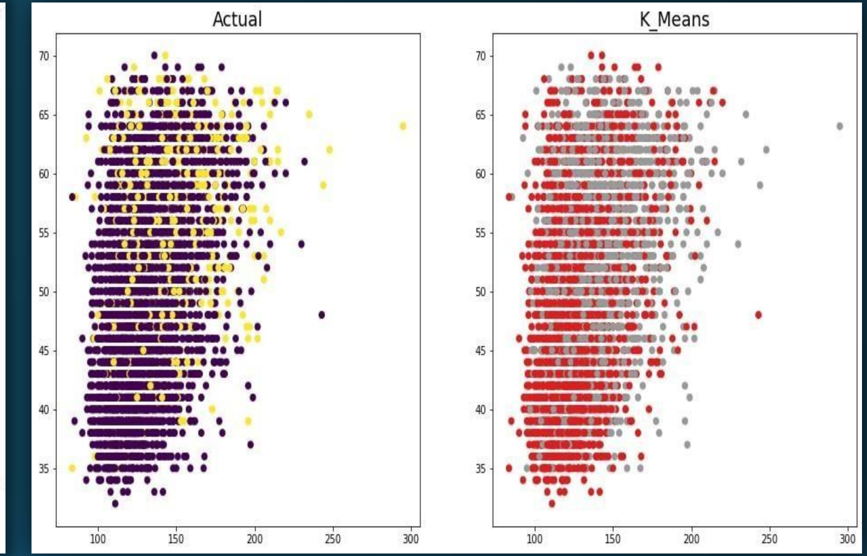


Based on the sensitivity or recall score, we can say that XG Boost provided us the based results as it has a recall score of around 70%. But based on an overall performance, we can say that Adaboost performed pretty well as it has decent scores for all the performance measures. The secret behind Adaboost’s performance is that, it has the previously fitted Logistic Regression model as it’s base estimator, which yielded very good scores for all the performance measures.

K-MEANS CLUSTERING







Elbow Plot:

WCSS is **the sum of squared distance between each point and the centroid in a cluster**. When we plot the WCSS with the K value, the plot looks like an Elbow. As the number of clusters increases, the WCSS value will start to decrease.

Conclusion:

From this Elbow plot we can't conclude the optimal number of clusters for the data. We can only suggest that the number of clusters should be 4, 3 or 2.

Silhouette Coefficient:

Silhouette Coefficient or silhouette score is a metric used to calculate the goodness ofa clustering technique. Its value ranges from -1 to 1.

Silhouette Score = (b-a)/max(a , b)

Where, **a**(Cohesion)= average intra-cluster distance & **b**(Separation)= minimum of average inter-cluster distance.

Conclusion:

For K=4 Silhouette Score = 0.248

For K=3 Silhouette Score = 0.24

For K=2 Silhouette Score = 0.32

So now we can say optimal number of clusters for the data is 2.

Actual Vs K-means Plot:

X axis 'sys_BP' & Y axis 'age' for both diagram. Here ACTUAL plot displays 'TenYearCHD' [Yes, No]

and K_means plot displays two clusters based on 'age'&'sys BP'.

Conclusion:

From correlation matrix we can say that 'age' & 'sys BP' have most influence over 'TenYearCHD'

So we used those factors as Y and X respectively in the plots above. We can conclude that clustering segmented the data well enough if we compare both plots.

Thank You