STATISTICAL AND MACHINE LEARNING APPRAOCHES FOR MARKETING

INDIVIDUAL ASSIGNMENT REPORT

MARCH 31, 2022

HARIKRISHNAN GOPALKRISHNAN

Contents

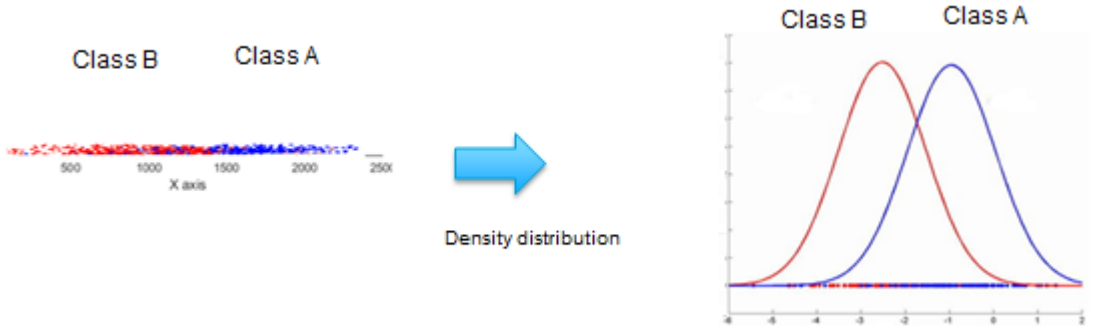
1.Algorithms

1.1. Logistic Regression

-The general idea of the algorithm:

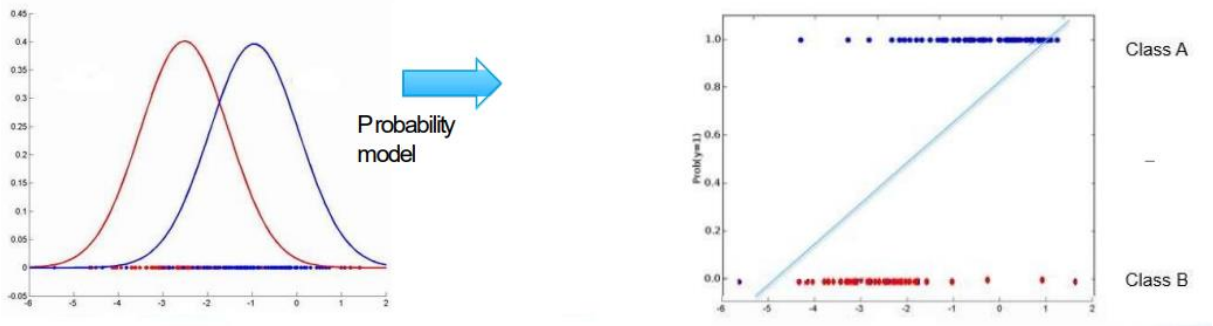
It is a classification algorithm built on the same concept as linear to predict classes. The response or target variable is categorical. In its simplest form, the response variable is binary i.e belongs to one form or the other.

For the given value of predictor (variable x),the model estimates the probability that the new data point belongs to particular class out of 2 classes(For ex:Class 'A' and 'B').The probabilities can range from 0 to 1.So,any new data point should be classified either class A or class B.



As seen in the figure,For the given distribution the data points that are closer to the point in the origin is unlikely to belong to class A and data points away from the origin is likely to belong to class A.

In turn, the linear model is passed onto a logistic function, whose result is the probability of a data point belonging to class "A" or class "B" for the given variable x.



-The objective function:

Log Loss-The cost function used in Logistic Regression

Log Loss is the negative average of the log of corrected predicted probabilities for each instance.To find the best fit line from the infinite possibilities,the log los function is given by :

$$LogLoss = -\frac{1}{n}\sum_{i=0}^{n}[y_i log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Where,

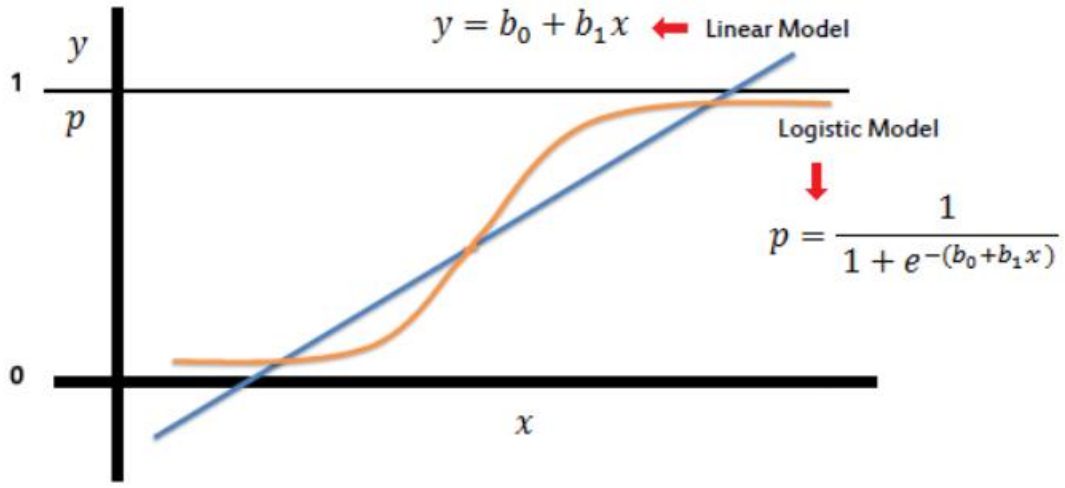
n = number of observations.

yi= observed probabilities

ŷi= predicted probabilities

log = (natural base e) logarithm.

For any classification problem, a lower log loss value means better predictions. So that main objective is to make the log loss as large negative numbers as possible.



-The algorithm fitting process:

- Once we have a logistic regression model we need to fit it to a set of data in order to estimate the parameters β0 and β1.In logistic regression,The method used to fit the data is called maximum likelihood method.Maximum likelihood will provide values of β0 and β1 which maximise the probability of obtaining the data set.
- The likelihood function is used to estimate the probability of observing the data,given the unknown parameters (β0 and β1).
- Based on the observed values of the independent variables, the likelihood represents the probability that the observed values of the dependent variable can be predicted. Similar to the probability, The likelihood ranges from 0 to 1.
- Using the log-likelihood value of a regression model, one can measure the fit of a model to a data set. The higher the log-likelihood value of a given model, the better the model fits the data set.
- The first derivative of the log-likelihood equation is used to estimate the parameters β0 and β1. Iterative computing is then used in any of the coding languages.An arbitrary value for the coefficients is first chosen .Then log-likelihood is computed and variation of coefficients values are observed .Then Reiteration is done until the maximization of log likelihood function. The final obtained results are the maximum likelihood function estimates of β0 and β1.

-Pros and cons of the algorithm:

Advantages:

Very fast at classifying unknown records

Good accuracy for many simple data sets

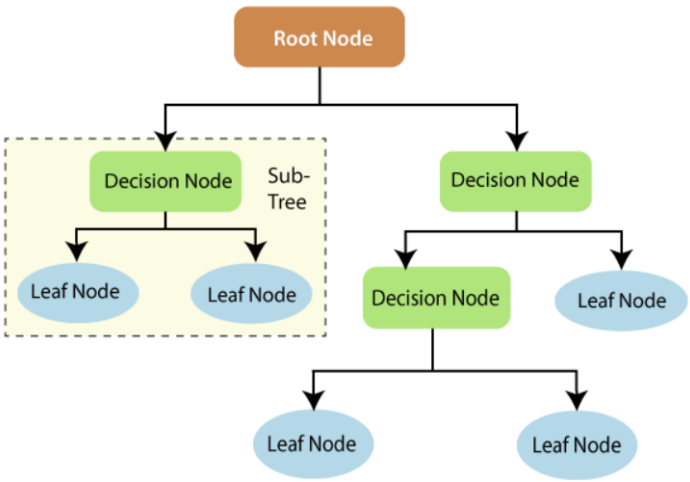
Resistant to overfitting

Disadvantages:

Constructs linear boundaries

1.2. Random Forests

-The general idea of the algorithm:

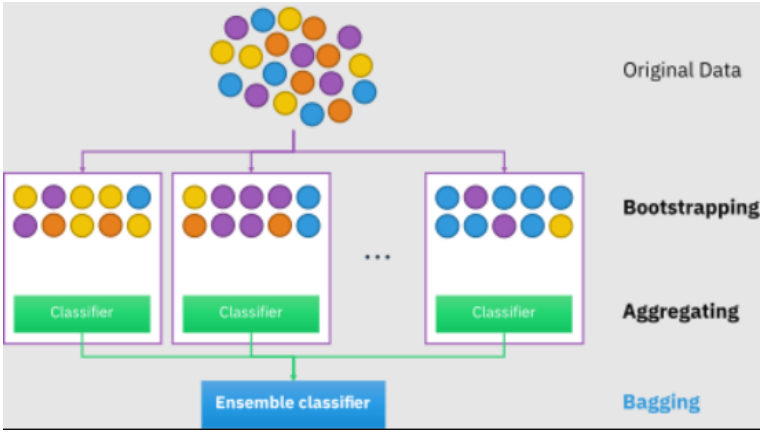


Random forest is a supervised Machine Learning algorithm that is commonly used to analyze classification and regression problems. It is also an ensemble technique that builds decision trees on different datasets and takes the majority vote in classification and average in regression.

This algorithm has the advantage of handling data sets that contain continuous variables, as in regression, or categorical variables, as in classification. This can improve results for classification problems.

-The objective function:

Bootstrap Aggregation



The re-sampling of the population is simulated by re-sampling our training data in order to create random decision forests.This process is called bootstrapping. If we have a dataset of size N, then we create new, random datasets of size M by sampling from the original with replacement.If the similar process of bootstrapping is done to produce a variety of predictors,it is known as **bootstrap aggregation or bagging method.**

Additionally, we use a random subset of input variables for every individual tree, which further decorrelates the trees.

Finally, it is evident that the goal of this method is to reduce variance by using high-variance, low-bias estimators, which we aggregate to the Forest. This method is equivalent to the regularization method.So we use larger trees with more nodes and lower bias as the base estimator for the model.

-The algorithm fitting process:

- The random forest algorithm fitting process counts to selecting hyper-parameters.The hyper-parameters of a random forest are :
  -The number of decision trees to consider in the forest-regularization parameter.
  -The base estimator's hyper parameters.
- Increasing the number of trees decreases variance. As the number of trees goes higher, accuracy does not suffer. Inference time, however, scales linearly with the number of trees.
- Out-of-bag cross validation is used to speed up the hyper-parameter selection. 'Bag ' refers to the subset of the training data selected by the bootstrapping method. There is a high possibility to generate an estimate of the validation error during the fitting process and without using an actual validation dataset. This is typically a good estimate. So using the out-of-bag error approach can save time compared to re-running the fitted tree on a validation dataset.
- The out-of-bag sample errors and test errors are compared and plotted to determine the number of predictors to fit the model.

-Pros and cons of the algorithm:

Advantages:

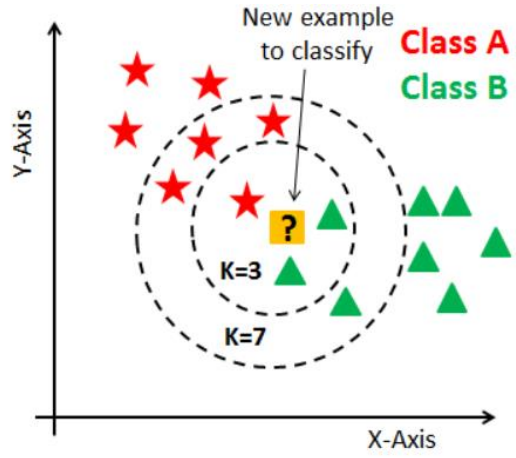
- Random forests can handle large datasets efficiently.
- It can perform both regression and classification tasks.
- A random forest produces good predictions that can be understood easily.

Disadvantages:

- Random forest is highly complex when compared to decision trees where decisions can be made by following the path of the tree
- Training time is more compared to other models due to its complexity.

1.3.K-Nearest Neighbors

-The general idea of the algorithm:



In K-NN, the similarity between the new case/data and the available cases is assumed, and the new case is put into the category that is most similar to existing categories.

K-NN algorithm stores all available data and classifies that data based on its similarity, so that if new data arises, it can be easily categorized into the well-suited category using K-NN algorithm.

As a non-parametric algorithm, K-NN makes no assumptions about underlying data.It is also called as a lazy learner algorithm which means it does not learn from the training data immediately instead it stores the dataset and at the time of classification ,it performs an action on the dataset.

-The objective function:

The distance metrics and K value are the most important factors for implementing the model.

Distance metrics :

The distance metric is the effective hyper-paramter through which we measure the distance between data feature values and new test inputs.For ex:Minkowski distance,Euclidean distance and Manhattan distance.

Eucliean distance approach is the most commonly used to measure the distance between the test samples and trained data values. We measure the distance along a straight line from point (x1, y1) to point (x2, y2).

$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^{n} \left(x_i - y_i\right)^2}$$

Choosing the optimal value of K:

Initiate by inputting a random K value and start computing. A low value of K will result in unstable decision boundaries. A higher K value can smooth out decision boundaries.

Derive a plot between error rate and K value giving the values in a defined range.Then the select the K value as having a minimum error rate.This K value can be used to implement the model.

-The algorithm fitting process:

Steps:

- At first the implementation of algorithm process starts by loading the training and test data.
- Choose the nearest data points (the value of K).It can be any integer.
- The above steps are repeated for test data.
- Use distance metrics to calculate the distance between the test data and each row of training data.
- Then sort that data in ascending order based on the distance value.
- From the sorted array,choose the top K rows.
- Based on the appearing class of these rows,it will assign a class to the data points.

-Pros and cons of the algorithm:

Advantages:

- Quick computation time
- Doesn’t require any additional assumptions about data.(No tuning parameters involved)
- Simple algorithm to interpret.

Disadvantages:

- The prediction process may be slow for large datasets.
- Requires high memory to store all of the training data.

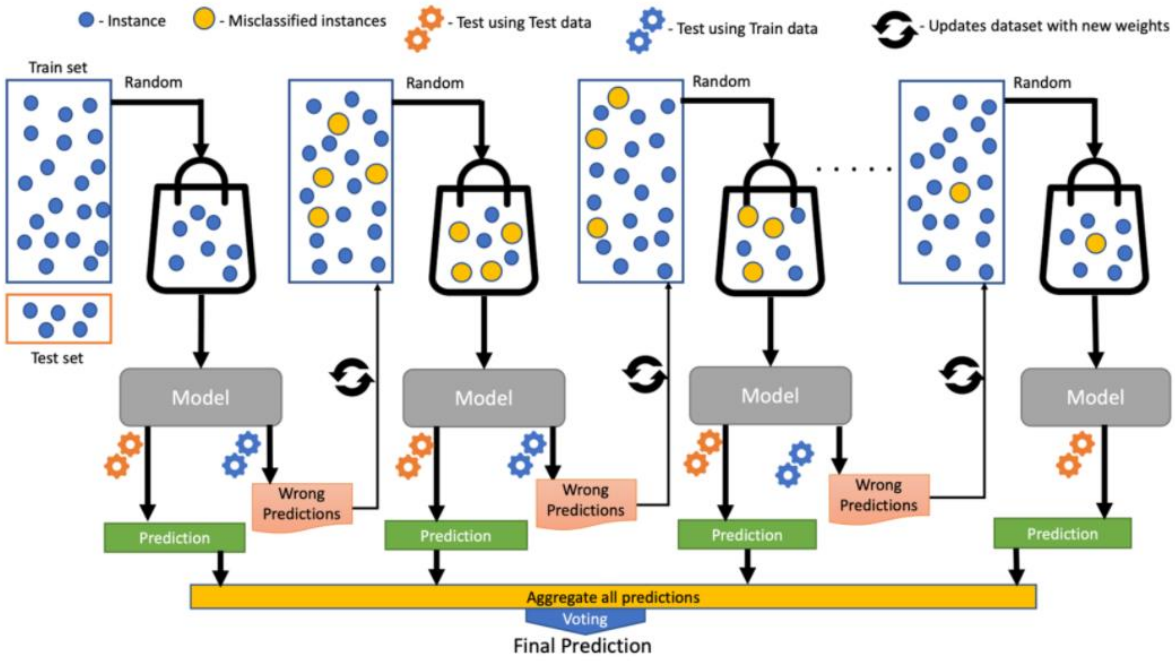
1.4. Gradient Boosting

-The general idea of the algorithm:

In this algorithm, the main idea is to build models sequentially and to reduce errors in the previous models through subsequent models.This is done by building a new model on the errors or residuals of the previous model.

In the case of target column is continuous,we use Gradient Boosting regressor and on the other hand in the case of Classification problem,we use Gradient Boosting Classifier.

The only difference between the two types is the “Loss function”.The objective is to minimize the loss function adding weak learners using gradient descent.So for regression problems, we’ll have different loss functions like Mean squared error (MSE) and for classification, we will have different functions like Log likelihood.



Internal working of boosting algorithm

-The Objective function:

Loss Function

The loss function basically tells how my algorithm, models the data set. It is calculated as the difference between actual values and predicted values.

Regression Loss functions:

L1 loss or Mean Absolute Errors (MAE)

L2 Loss or Mean Square Error(MSE)

Quadratic Loss

Binary Classification Loss Functions:

Binary Cross Entropy Loss

Hinge Loss

A gradient descent procedure is used to minimize the loss when adding trees.

Weak Learner

In gradient boosting algorithms, weak learners are used sequentially to reduce the errors generated from previous models and return a strong model in the end. Decision trees are most commonly used as weak learners.

-The algorithm fitting process:

Steps:

- Calculate the average of the target or response variable.
- Calculate the residuals for each sample of the data.It can be calculated difference between actual and predicted value.
- Construct a decision tree with the objective to predict the residuals.
- Predict the target label using all the decision trees within the ensemble.
- Repeat all the above steps until the number of iterations matches the number specified by the numbers of estimator.
- Once the model is trained, we use all of the decision trees in the ensemble to make a final prediction as to value of the target variable.

-Pros and cons of the algorithm:

Advantages:

- Handles missing data - imputation not required.
- No data pre-processing required – It often works great with categorical and numerical values as is.
- Flexibility – It can optimize on different loss functions and provides several hyper parameter tuning options

Disadvantages:

- Computationally expensive – It often require many decision trees (>1000) which can be time and memory consuming.
- Gradient Boosting Models will continue improving to minimize all errors. This can overemphasize outliers and cause overfitting.

1.5. Support Vector Machines

-The general idea of the algorithm:

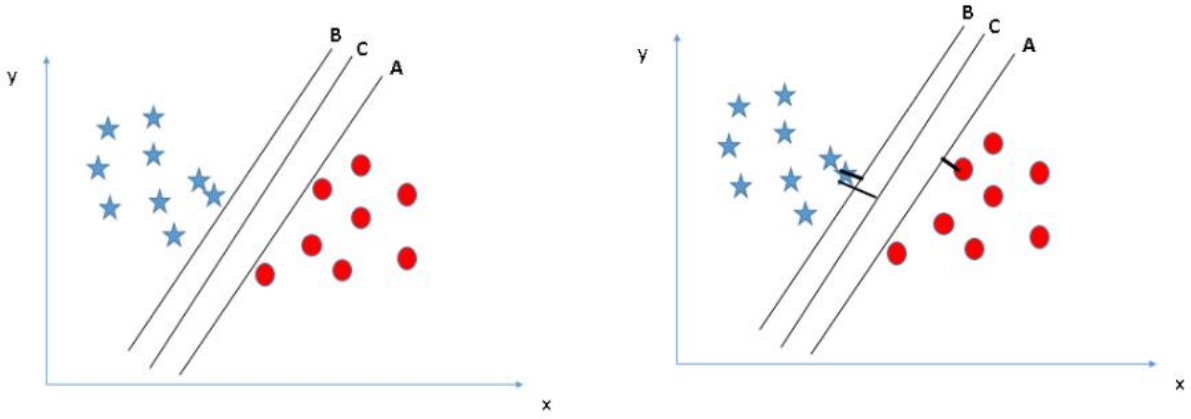
Support Vector Machine (SVM) is a supervised machine learning algorithm used for binary classification problems, although it can also be used for regression as well. SVM's objective is to find an N-dimensional hyperplane that accurately classifies data points.

The idea behind support vector machines (SVM) is that the vectors (i.e. cases) are transferred to a higher dimension. The optimal linear hyperplane is determined by the greatest distance between the categories or classes.

-The Objective function:

Identify the right hyper-plane

Th real challenge in separating the two classes is to find the right hyperplane.There are many options to do this for different scenarios.Let us see on particular way to determine the right hyperplane.



From the above left side figure, we have three hyper-planes (A, B, and C) and all are separating the classes well.We can identify the right hyper-plane by maximizing the distance between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin.**

From the above right side figure, we can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we can conclude that the right hyper-plane as C. Robustness is one specific reason for selecting the hyper-plane with higher margin. There is high possibility of miss-classification of data points in case of selecting the hyper-plane with low margin.

Kernel trick: The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem. It is mostly useful in non-linear separation problem.

-The algorithm fitting process:

Steps:

- At first, the two classes or categories can be separated with a curve for the given dataset.
- The boundary between the two categories is added.
- The SVM kernel function is used for the transformation and fit the model,kernel types are Linear,Polynomial,Sigmoid etc.

  A linear kernel function is mostly recommended when linear separation of the data is straightforward.

-Pros and cons of the algorithm:

Advantages:

- It is effective in cases where the number of dimensions is greater than the number of samples.
- It works really well with a clear margin of separation
- It is effective in high dimensional spaces.

Disadvantages:

- It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping

2.Benchmark Experiment

Description

The Data set is from a Taiwan banking institution to predict the default of credit card of clients . Specifically, the classification goal is to predict if the client will be able to pay the credit in the upcoming month.

2.1. Variable Selection

Method:Univariate selection

In this variable selection method ,statistical tests can be used to select those features that have the strongest relationship with the output variable.

In scikit-learn, a class named **SelectKBest** can be used with a variety of statistical tests to select a particular set of features.

Features	Score
PAY_1	1619.032218
PAY_2	1120.300108
PAY_3	952.294120
PAY_4	810.424574
PAY_5	737.271665
PAY_6	599.951833
LIMIT_BAL	423.755436
PAY_AMT1	109.358577
PAY_AMT4	80.749450
PAY_AMT5	72.286806
PAY_AMT2	68.716598
PAY_AMT3	58.782693
PAY_AMT6	57.133830
Graduate_school	44.963815
male	23.859334

F_classif score function is used to as a statistical test measure in the credit card default dataset to get the top 15 features.

2.2. Model Building

Since it is a classification problem,I have chosen 5 classification algorithms to fit the model.The algorithms are as follows:

1. Logistic Regression
2. Random Forests
3. Gradient Boosting
4. K-Nearest Neighbour
5. Support Vector Machines

These 5 algorithms are fitted on the training set based on the 15 selected features.

The classification algorithms are fitted using their default parameters at first and evaluated their performance. Comparison of the algorithms are made based on their ‘accuracy’ and ‘auc’ values.

	Logistic_Regression	Random_Forest	Gradient_Boosting	Support_Vector_Machines	K-Nearest_Neighbour
Accuracy	0.777680	0.806762	0.810967	0.777680	0.753679
AUC	0.638575	0.738830	0.750411	0.444312	0.599313

2.3.Cross Validation

The simplest way to use cross-validation is to call the **cross_val_score** helper function on the estimator and the dataset.It is imported using sklearn.model_selection package.

cross_val_score function is used to estimate the 'auc' of the ML algorithms on the credit card default dataset by splitting the data, fitting a model and computing the auc score 5 consecutive times (with different splits each time):

```
#Gradient Boosting
from sklearn.model_selection import cross_val_score


crossval_scores1 = cross_val_score(boostedTree, trainingSet[top_features],y_train , scoring='roc_auc', cv=5)
crossval_scores1
```

```
array([0.77055112, 0.7690305 , 0.7421821 , 0.77396675, 0.76028848])
```

The above is an example of how the cross_val_score function is used to estimate auc scores of 5 different splits each time.Similary process is carried out for other ML models and following 'auc' scores are obtained.

SVM	KNN	Random_forests	Logistic Regression	Gradient Boosting
0.454753	0.592176	0.757380	0.615329	0.770551
0.545250	0.609580	0.737445	0.640964	0.769030
0.487678	0.588179	0.734766	0.644729	0.742182
0.450927	0.589646	0.763847	0.626863	0.773967
0.408849	0.601993	0.738248	0.651200	0.760288

We can observe that the above auc scores using CV method are similar to the scores that obtained from Model building .

From the above metrics we can observe that Gradient boosting is the best model to fit and classify the class for target :Default for the credit card dataset.

2.4. Hyper Paramater Tuning

Hyperparameters is done in order to improve the performance of the model by changing the model architecture.

GridSearchCV is used as a library function from sklearn's model_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set.

When GridsearchCV is run through the Gradient boosting model we get the following parameter values.

{'learning_rate': 0.03, 'n_estimators': 300}

Then the selected model Gradient Bossting is once again fitted in the training dataset using above best parameter values.Then the model is predicted the probabilities for train and test dataset.

The predictions are evaluated and following metrics are obtained for the Gradient boosting model:

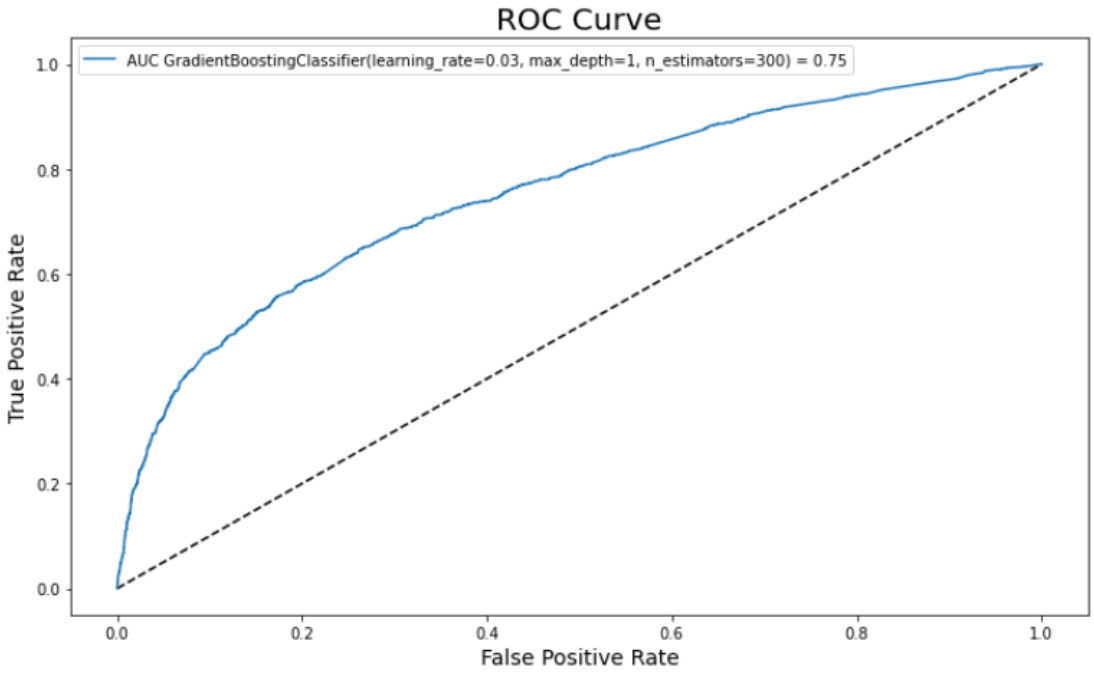
Accuracy Train: ACC=0.8159
Accuracy Test: ACC=0.8120
AUC Train: ACC=0.7748
AUC Test: ACC=0.7524

2.5. Evaluation Metrics

AUC-ROC Curve

AUC(Area under the curve) is also chosen as a metric because it is better measure of classifier performance than accuracy because it does not bias on size of test or evaluation data.

ROC(Receiver Operator characteristic) is a probability curve that plots TPR(true positive rate) versus FPR(False positive rate) for a range of threshold values. It essentially separates the signal from the noise in binary classification problems.AUC is used as the summary of the ROC curve.The higher the AUC ,the better the performance of the model at distinguishing between the classes.



From the above AUC-ROC curve ,we can see the relationship between true positive rate and false postive rate in predicting the classes

True positive is an outcome where the model correctly predicts the positive class.

False positive is an outcome where the model incorrectly predicts the positive class.

Inference: So the 0.75 AUC signifies that Gradient Boosting model has 75% probable chance of separating the classes (Default : Yes or no) in the dataset.

3.REFERENCES:

Logistic Regression

https://medium.com/@vijaya.beeravalli/comparison-of-machine-learning-classification-models-for-credit-card-default-data-c3cf805c9a5a

https://www.analyticsvidhya.com/blog/2020/11/binary-cross-entropy-aka-log-loss-the-cost-function-used-in-logistic-regression/

https://machinelearningmastery.com/logistic-regression-with-maximum-likelihood-estimation/

Random forests

https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/#:~:text=Random%20forest%20is%20a%20Supervised,average%20in%20case%20of%20regression.

https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3

KNN

https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

https://www.mygreatlearning.com/blog/knn-algorithm-introduction/

fig - https://test.basel.in/product/knn-naive-bayes-classifier-using-excel/

Gradient boosting

https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/

https://medium.com/analytics-vidhya/introduction-to-the-gradient-boosting-algorithm-c25c653f826b

fig - https://dzone.com/articles/xgboost-a-deep-dive-into-boosting

SVM

https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

Benchmark:

https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

https://scikit-learn.org/stable/modules/cross_validation.html#computing-cross-validated-metrics

https://github.com/robertofranceschi/default-credit-card-prediction