PREDICTING CUSTOMER CHURN IN A TELECOMMUNICATION COMPANY

Ayush Jain, 12103026, B. Tech. CSE, Lovely Professional University

INTRODUCTION

The churn rate, also known as the rate of attrition or customer churn, is the rate at which customers stop doing business with an entity. It is most commonly expressed as the percentage of service subscribers who discontinue their subscription within a given time-period. It is also the rate at which employees leave their jobs within a certain period. For a company to expand its clientele, its growth rate (measured by the number of new customers) must exceed its churn rate.

- The churn rate measures a company's loss of subscribers for a given period of time.
- Churn rates can be applied to subscription-based businesses as well as to the number of employees that leave a firm.
- Churn rate is the opposite of growth rate, which measures the acquisition of customers.
- For a company to experience growth it must ensure that its new subscriptions are higher than its lost subscriptions in a given period.
- Each industry will have a different average churn rate that companies can compare themselves with to understand their competitiveness.

While the churn rate tracks lost customers, the growth rate tracks new customers. A company can compare its new subscribers to its loss of subscribers to determine both its churn rate and growth rate. The difference between the two shows whether there was overall growth or loss in a specific time period.

If the growth rate was higher than the churn rate, the company experienced growth. If the churn rate was higher than the growth rate, the company experienced a loss in its customer base.

For example, if in one quarter a company added 100 new subscribers but lost 110 subscribers, the net loss would be 10. There was no growth for the company this quarter but rather a loss. This would be a negative growth rate and a positive churn rate.

It is critical for a company to ensure that its growth rate is higher than its churn rate otherwise it will experience declining revenues and profits with the eventual scenario of having to close the business.

DATA COLLECTION & PREPROCESSING

A dataset named “Telco-Customer-Churn.csv” was downloaded from the dataset which consists of 21 Columns & 7043 Rows. It then went through preprocessing techniques which include Checking for Null Values & Duplicates and handling them and Encoding the Categorical Values.

The Column “CustomerID” was dropped as it doesn’t play any significant role in the further process, thereafter 22 Duplicate Rows were dropped.

The Column “TotalCharges” was converted from Object to Float which created 11 Null Values as some values did not get transformed which were then populated with “most-frequent” values through Imputation.

At the last stage of preprocessing, all the Categorical Columns were encoded to into Numerical values through LabelEncoder, after which the preprocessed data was left with only 20 Columns & 7021 Rows in it.

EXPLORATORY DATA ANALYSIS (EDA)

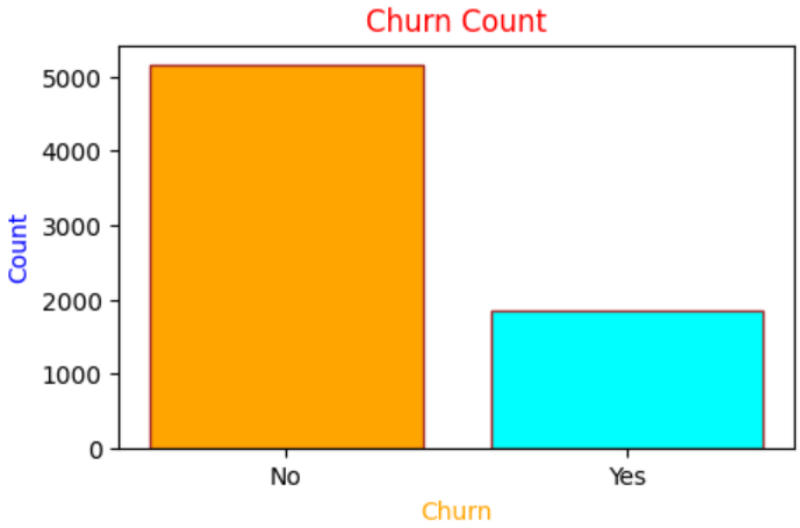

Fig. 1 CHURN COUNT

From the bar plot in Fig. 1, it is inferred that the number of churn counts were 1857 and the ones who did not churn are 5164

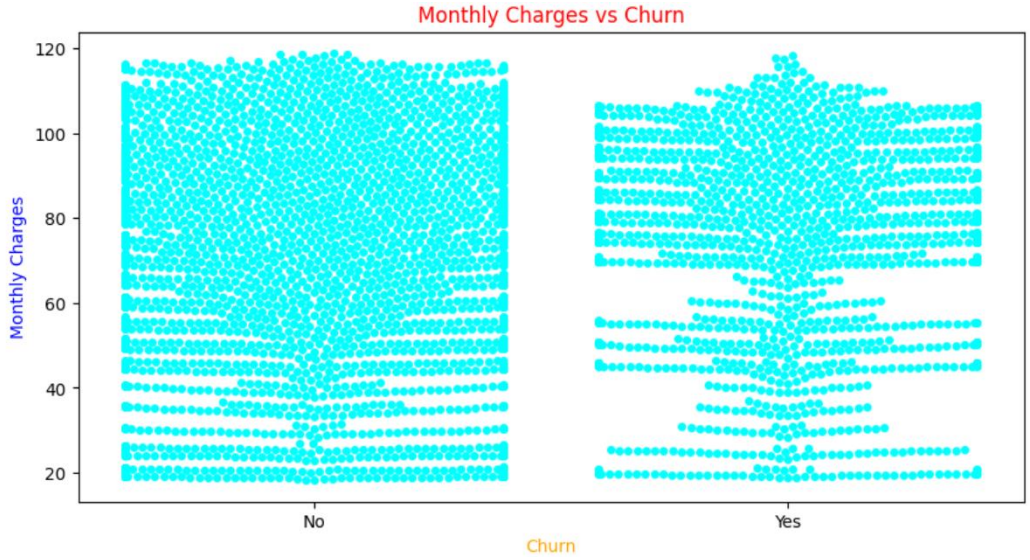

Fig. 2 MONTHLY CHARGES VS CHURNS

From the Swarm Plot in Fig. 2, it is inferred that the customers who churned had more monthly charges in the later stages than the ones who did not churn.

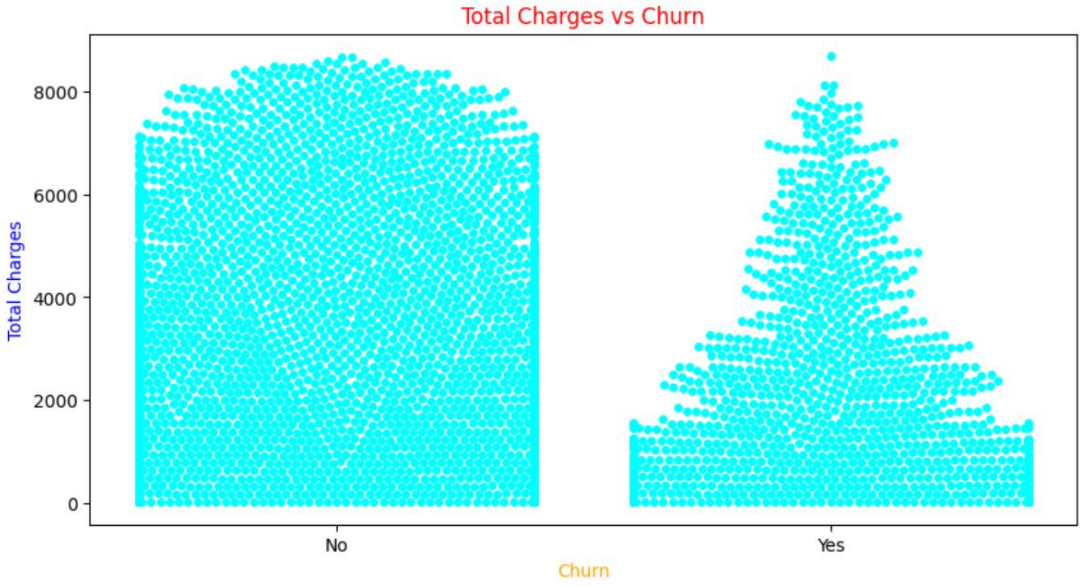

Fig. 3 TOTAL CHARGES VS CHURNS

From the Swarm Plot in Fig. 3, it is inferred that the customers who churned had more yearly charges in the initial stages whereas the ones who did not churn had the same charges throughout process.

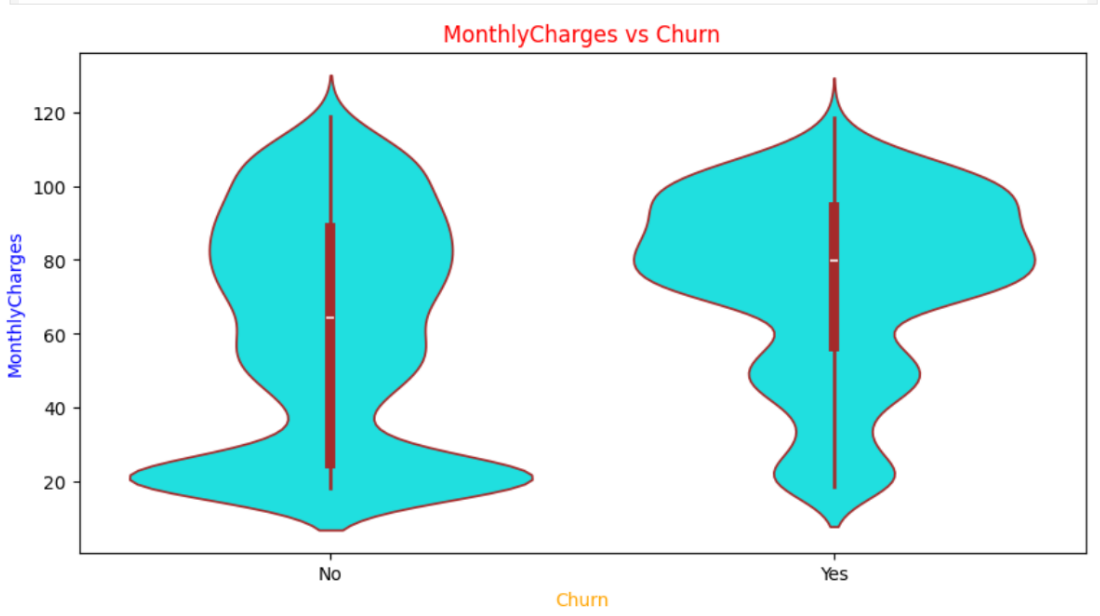

Fig. 4 MONTHLY CHARGES VS CHURNS

From the Violin Plot in Fig. 4, it is inferred that the customers who churned had more monthly charges in the later stages than the ones who did not churn.

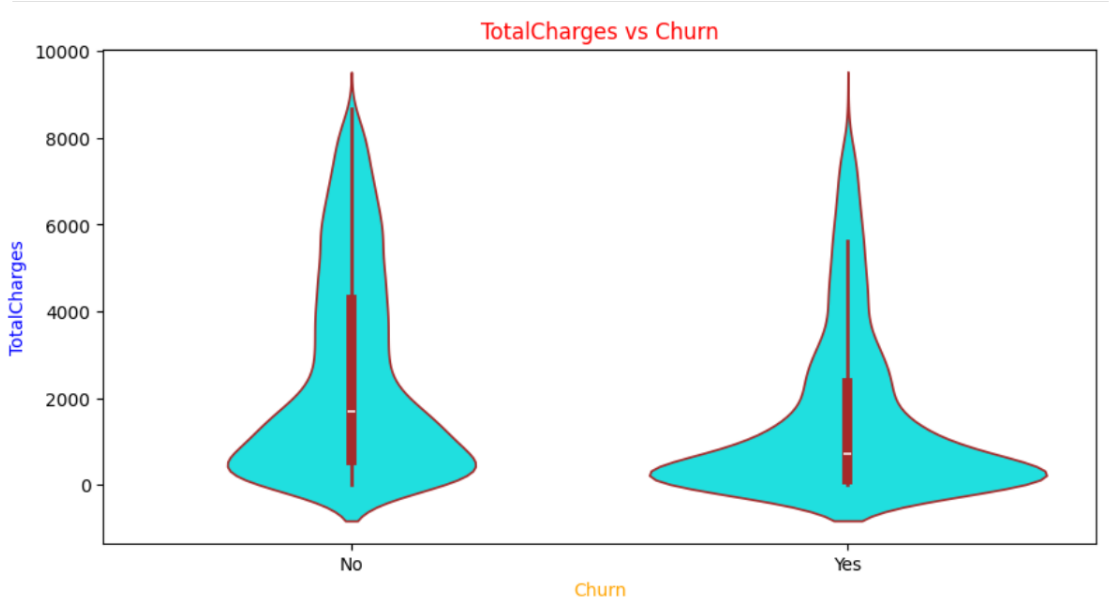

Fig. 5 TOTAL CHARGES VS CHURNS

From the Violin Plot in Fig. 5, it is inferred that the customers who churned had more yearly charges in the initial stages whereas the ones who did not churn had the same charges throughout process.

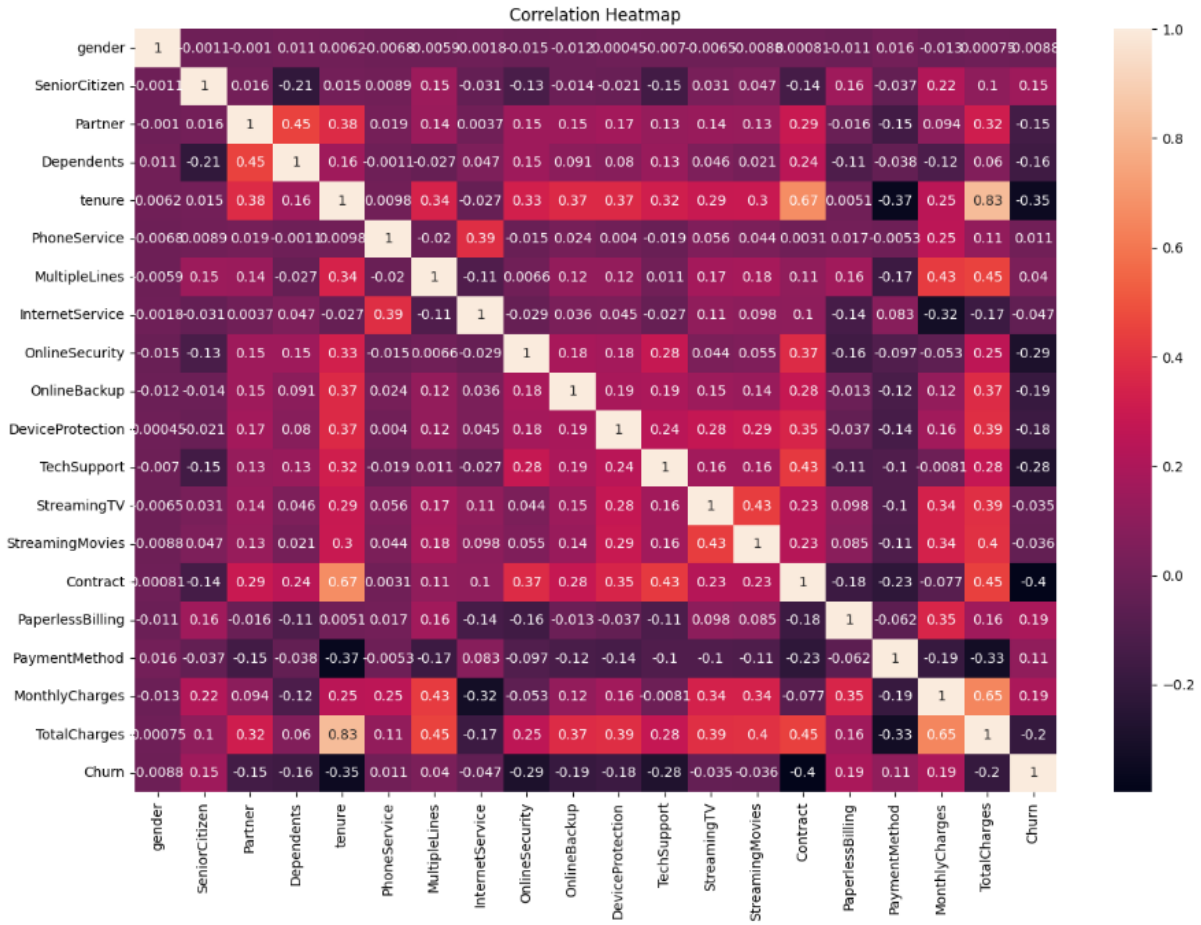

Fig. 6 CORRELATION HEATMAP

Fig. 6 displays the correlation between each and every column or the features of the dataset

FEATURE ENGINEERING & FEATURE SCALING

After getting insights from the EDA, feature engineering was performed in which the target column "Churn" was separated and deleted from the dataset for predictions in the later stages.

Then, the dataset was then partitioned into Training & Testing Data with the ratio of 70:30

As the maximum values of all the columns were varying hugely, it was required to scale it, hence it was scaled using "Standard Scaler", after which the maximum values became came nearby to each other, and Mean & Variance got shifted to 0 & 1 respectively.

Now, the dataset is ready to apply models on it.

MODEL BUILDING

For the model, 7 different algorithms were implemented individually, namely, Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbor, Random Forest, AdaBoost & MLP Classifier.

Later, 5 different algorithms namely, Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbor & MLP Classifier were implemented using GridSearchCV.

Let us explore all these Algorithms:

LOGISTIC REGRESSION

Logistic regression is a supervised machine learning algorithm used for classification tasks where the goal is to predict the probability that an instance belongs to a given class or not. Logistic regression is a statistical algorithm which analyse the relationship between two data factors.

Logistic regression is used for binary classification where we use sigmoid function, that takes input as independent variables and produces a probability value between 0 and 1.

- 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.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

SUPPORT VECTOR MACHINE

A support vector machine is a popular supervised learning model developed by Vladimir Vapnik, used for both data classification and regression. That said, it is typically leveraged for classification problems, constructing a hyperplane where the distance between two classes of data points is at its maximum. This hyperplane is known as the decision boundary, separating the classes of data points (e.g., oranges vs. apples) on either side of the plane.

SVM has 3 Kernels:

- Linear

- Polynomial
- rbf (Default)

NOTE: *As stated, "rbf" is the default kernel, in the model created, "poly" kernel is used as, in rbf, no true samples were created because of which the model was not able to fit properly.*

DECISION TREE

Decision tree learning is one of the most widely used and practical methods for inductive inference. It is a method for approximating discrete-valued functions, in which the learned function is represented by a decision tree.

It is robust to noisy data and capable of learning disjunctive expressions. Decision tree learning algorithms that includes widely used algorithms such as ID3, CART, and C4.5.

Learned trees can also be re-represented as sets of if-then rules to improve human readability. Decision tree algorithms transform raw data to rule based decision-making trees.

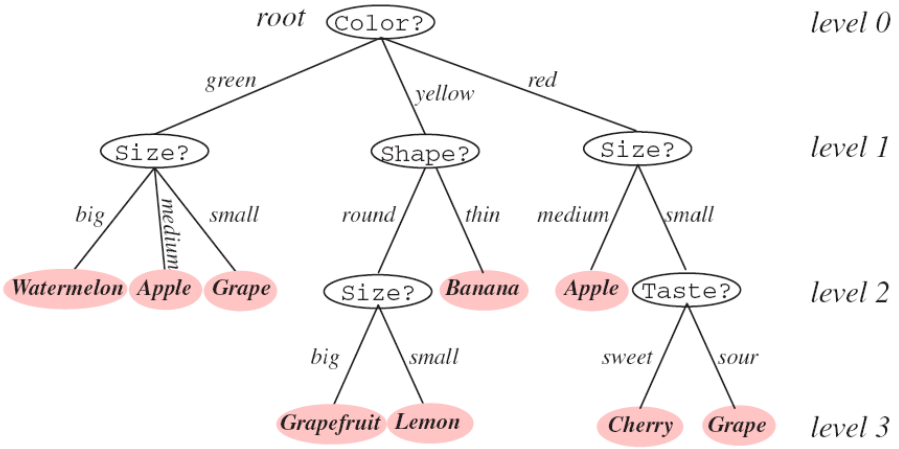

Fig. 7 DECISION TREE

K-NEAREST NEIGHBOR

K-nearest neighbor, also known as the KNN algorithm, is a non-parametric algorithm that classifies data points based on their proximity and association to other available data. This algorithm assumes that similar data points can be found near each other. As a result, it seeks to calculate the distance between data points, usually through Euclidean distance, and then it assigns a category based on the most frequent category or average. Its ease of use and low

calculation time make it a preferred algorithm by data scientists, but as the test dataset grows, the processing time lengthens, making it less appealing for classification tasks. KNN is typically used for recommendation engines and image recognition.

ADVANTAGES

- No assumption about data
- Simple Algorithm
- High Accuracy
- Versatile

DISADVANTAGES

- Computationally expensive
- High Memory Requirement
- Stores all the training data
- Sensitive to irrelevant features & the scale of the data

In the model, a variation of both Training & Testing Accuracy with different neighbors is made as shown in the figure below:

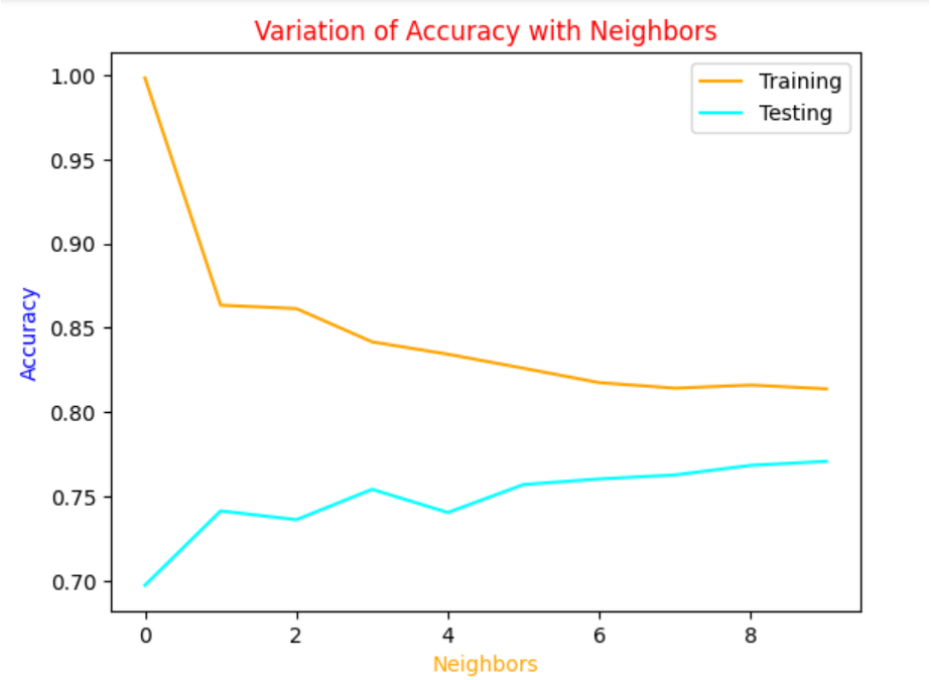

Fig. 8 VARIATION OF ACCURACY WITH NEIGHBORS

This variation suggests that as the number of neighbors increases, the training accuracy decreases and the testing accuracy increases.

RANDOM FOREST CLASSIFIER

The Random Forest is a supervised Machine learning algorithm used for classification, regression, and other tasks using decision trees. Random Forests are particularly well-suited for handling large and complex datasets, dealing with high-dimensional feature spaces, and providing insights into feature importance. This algorithm's ability to maintain high predictive accuracy while minimizing overfitting makes it a popular choice across various domains, including finance, healthcare, and image analysis, among others.

The Random Forest classifier creates a set of Decision Trees from a randomly selected subset of the training set. It is a set of decision trees (DT) from a randomly selected subset of the training set and then it collects the votes from different decision trees to decide the final prediction.

Random Forest Classification is an ensemble learning technique designed to enhance the accuracy and robustness of classification tasks. The algorithm builds a multitude of decision trees during training and outputs the class that is the mode of the classification classes. Each decision tree in the random forest is constructed using a subset of the training data and a random subset of features introducing diversity among the trees, making the model more robust and less prone to overfitting.

ADABOOST CLASSIFIER

AdaBoost short for Adaptive Boosting is an ensemble learning used in machine learning for classification and regression problems. The main idea behind AdaBoost is to iteratively train the weak classifier on the training dataset with each successive classifier giving more weightage to the data points that are misclassified. The final AdaBoost model is decided by combining all the weak classifier that has been used for training with the weightage given to the models according to their accuracies. The weak model which has the highest accuracy is given the highest weightage while the model which has the lowest accuracy is given a lower weightage.

MULTILAYER PERCEPTRON (MLP) CLASSIFIER

A key machine learning method that belongs to the class of artificial neural networks is classification using Multi-Layer Perceptrons (MLP). It is a flexible and effective method for tackling a variety of classification problems, including text classification and picture recognition. Traditional linear classifiers might

not be up to the challenge, but MLPs are known for their capacity to model complicated, non-linear relationships in data.

Although MLPs are well renowned for their capacity to represent complicated relationships in data, they can be sensitive to certain hyperparameters, including the number of hidden layers and neurons, the choice of activation functions, and regularization strategies. For MLPs to operate well, proper hyperparameter adjustment is crucial.

Note: *MLP have maximum iterations of 200 as default, but it was changed in the model to 1100 as 200 iterations were not able to reach the convergence.*

GRIDSEARCH CV

GridSearchCV is the process of performing hyperparameter tuning in order to determine the optimal values for a given model. The performance of a model significantly depends on the value of hyperparameters. As there is no way to know in advance the best values for hyperparameters so ideally, we need to try all possible values to know the optimal values. Doing this manually could take a considerable amount of time and resources and thus GridSearchCV is used to automate the tuning of hyperparameters.

MODEL EVALUATION & VISUALIZATION

ALGORITHMS	TRAINING ACCURACY	TESTING ACCURACY	PRECISION	RECALL	F1-SCORE
LOGISTIC REGRESSION	0.8030	0.8044	0.5465	0.6519	0.5946
SVM	0.8262	0.7887	0.5227	0.7437	0.6139
DECISION TREE	0.9983	0.7261	0.9938	1.0	0.9969
KNN	0.8343	0.7403	0.6243	0.7132	0.6658
RANDOM FOREST	0.9983	0.7897	0.9969	0.9969	0.9969
ADABOOST	0.8109	0.8049	0.5404	0.6789	0.6018
MLP	0.9513	0.7351	0.8922	0.9213	0.9065

TABLE – 1 INDIVIDUAL ALGORITHMS

ALGORITHMS	TRAINING ACCURACY	TESTING ACCURACY	BEST SCORE
LOGISTIC REGRESSION	0.8030	0.8044	0.7979
SVM	0.8042	0.8001	0.7997
DECISION TREE	0.9983	0.7437	0.7252
KNN	0.8341	0.7403	0.7431
MLP	0.8034	0.8044	0.8001

TABLE – 2 INDIVIDUAL ALGORITHMS WITH GRIDSEARCH CV

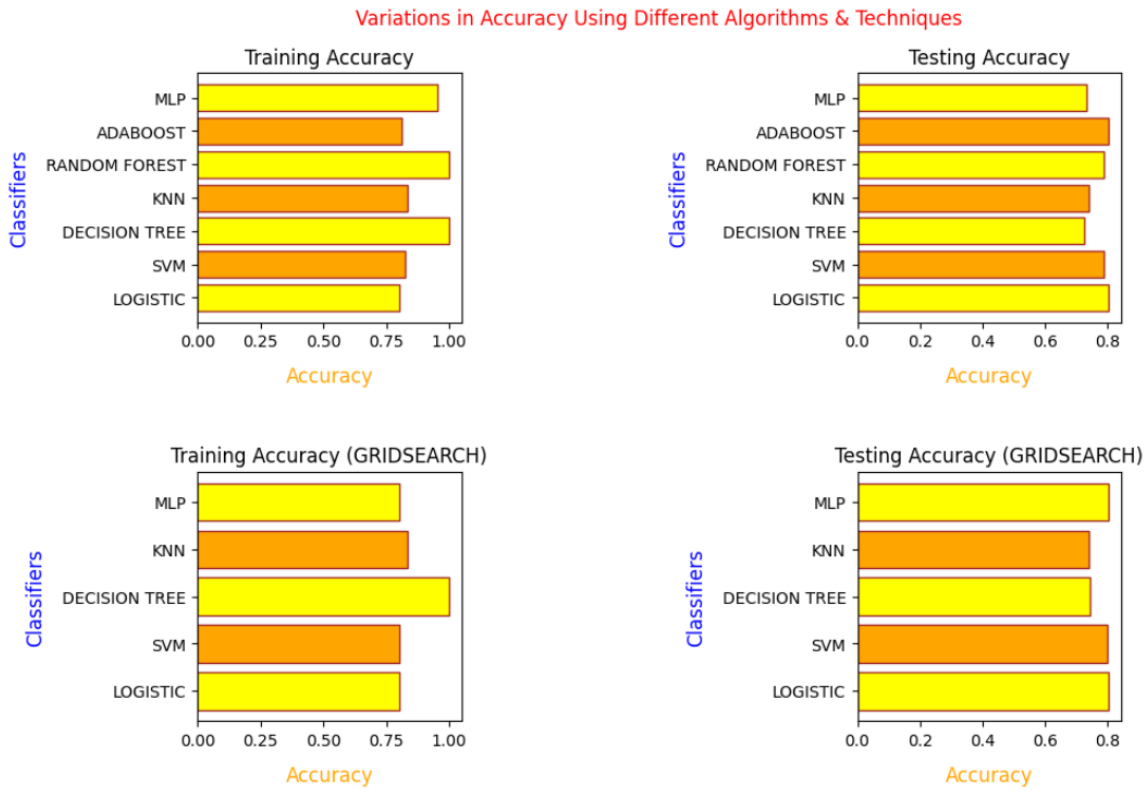

Fig. 9 VARIATIONS IN ACCURACY USING DIFFERENT ALGORITHMS & TECHNIQUES

After Visualizing & Evaluating all the models, it is thus stated that Random Forest Classifier individually instead of having less Testing Accuracy outperformed all the other models.