Customer Churn Prediction Using Machine Learning Algorithms

Syed Muzammil Ahmed

Objective: -

Customer churn is a major problem and one of the most important concerns for large companies. Due to the direct effect on the revenues of the companies, especially in the telecom field, companies are seeking to develop means to predict potential customer to churn. Therefore, finding factors that increase customer churn is important to take necessary actions to reduce this churn. The main contribution of our work is to develop a churn prediction model which assists telecom operators to predict customers who are most likely subject to churn.

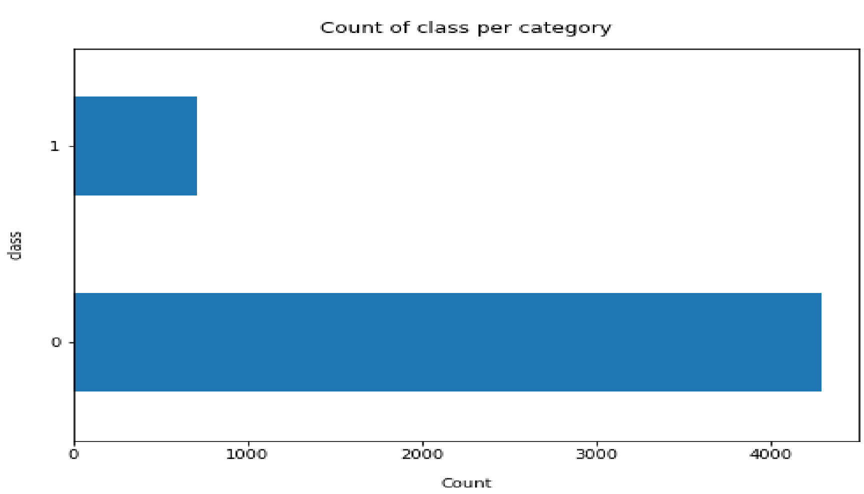
Dataset: -

Dataset we are using in this case study contain 5000 customer records and 21 features. We are using “class” as our target variable. It is a Binary class classification problem where value of class is 0 and 1 (0 means do not churn and 1 means churn). Data set contain other features as well such as state, phone number etc.

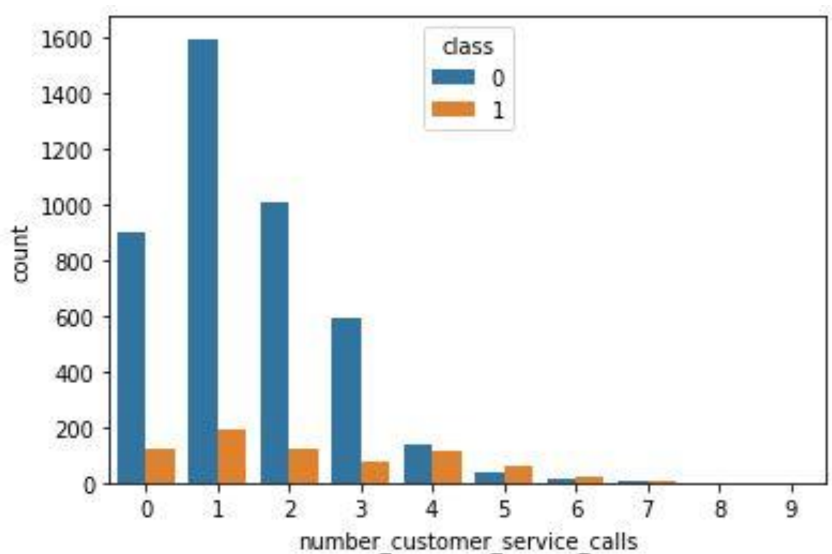
Why are customers churning?

So, after performing exploratory data analysis on our dataset we have got some insights, which will inform us about why customers are churning. So, here are some insights that we have got after doing analysis.

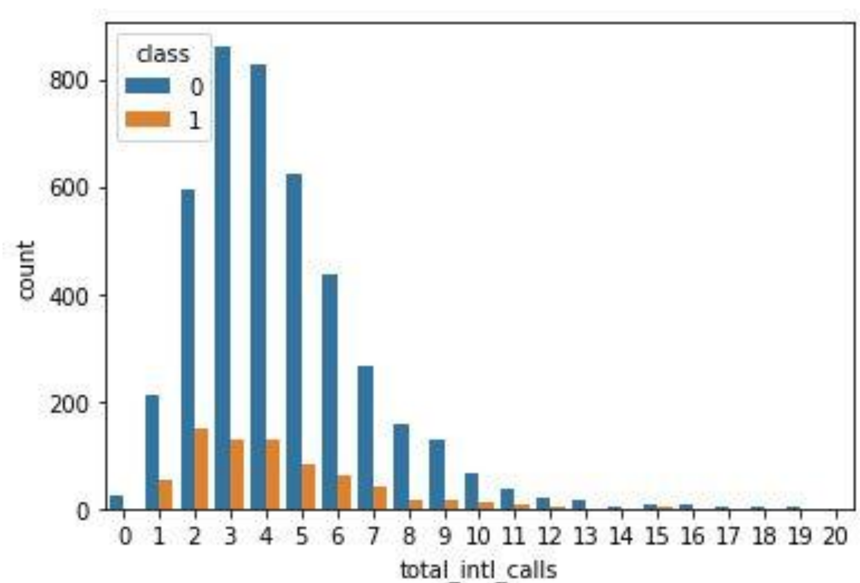
1) So, in this dataset we have around 87% of records which are not churning and 13% of records which churned to another company.



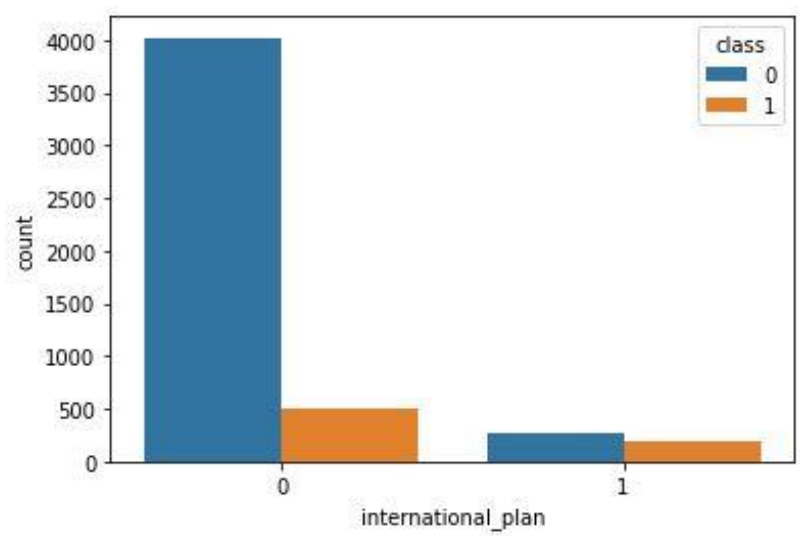
2) When we analyze our data, we got to know that churn rate is high among the customers who are receiving high number of customer service calls. Specifically, customers receiving more the 2 or 3 customer service calls are churning to other networks.



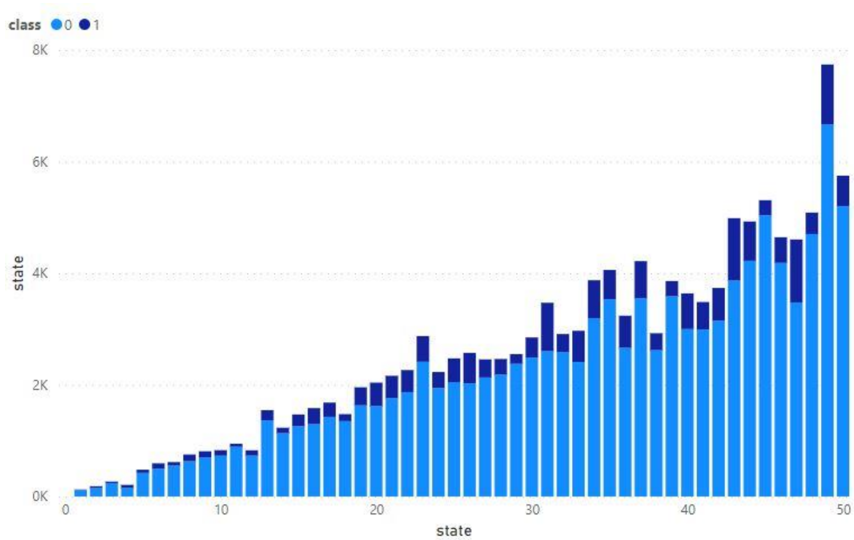
3) Churn rate is high among the customers, who have high number of total international calls. It means customers who are doing high number of international calls are churning more.



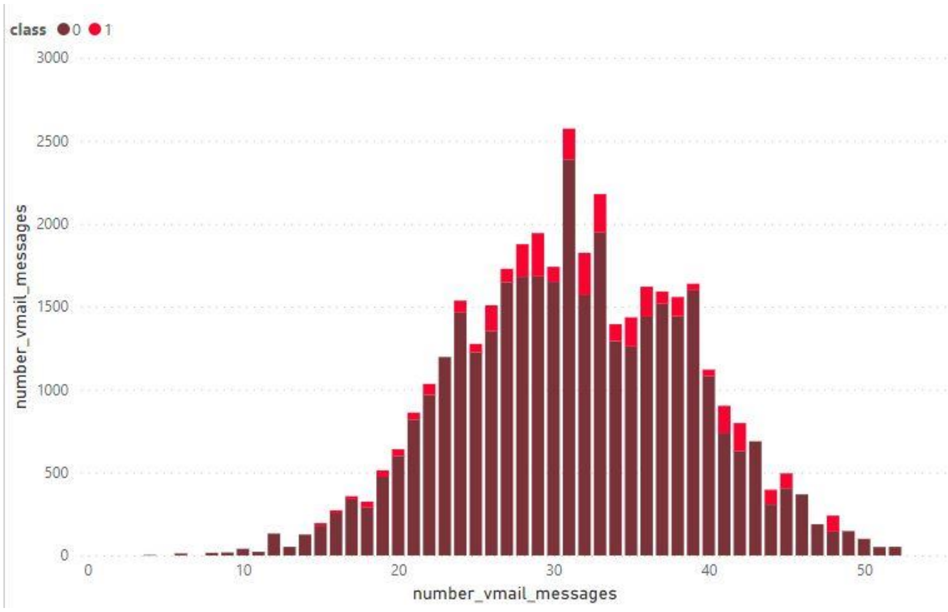
4) Churn rate is high among customers, who are subscribers of international plan. Customers using international plan subscription is more likely to churn.



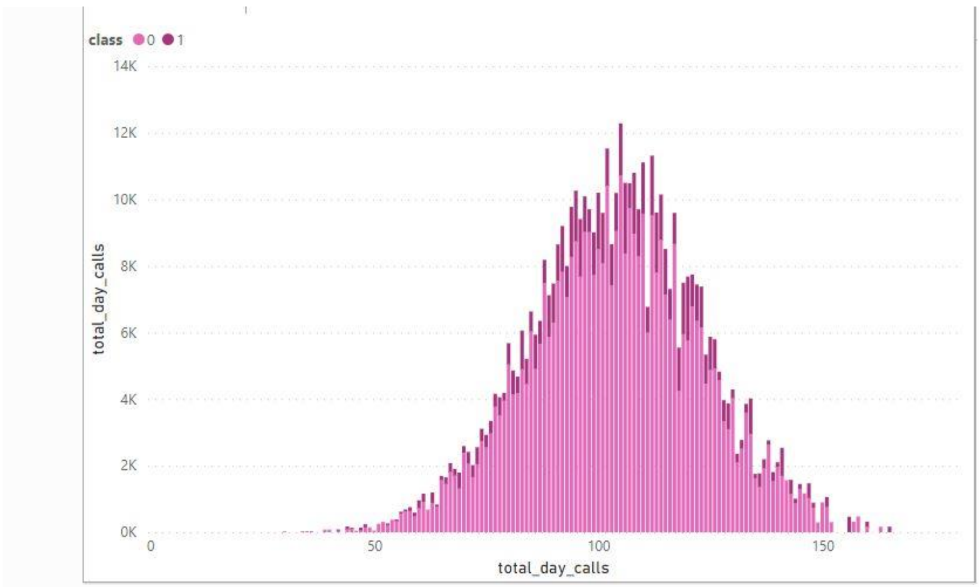
5) So company has customers from about 50 states but customers from states 31,43 and 47 are churning more as comapre to the other states.



6) Customer receiving offers from mail messages are less churning as compare to people receiving promotional calls.



7) Customers having high number of day calls are more likely to churn as compared to customers having normal day call routine.



8) People having more night calls are also churning as compared to people having normal numbers of night call.

9) Most of the customers belongs to area code 415, but churn rate is also high in this area.

These are some insights we got to find out why customers are churning. There are numerous reasons because which customers are moving to other networks such as:

1- Most of the customers churning because they are receiving high number of promotional and customer service maybe these calls are irritating most of them that’s why they decide to move on to another network.

2- Customers with subscription of international plan are churning more as compared to customers with no international plan. So, maybe companies international service or plan is not feasible and profitable enough.

3- Customers with high number of international calls are not churning more maybe they find rates of international calls costly and uneconomical.

What offers could be made to retain them?

To avoid customer churn in company, these steps should be considered:

1- Improve quality of service and efficient communication.

2- Better offer and more proficient and economical international plan for users.

3- Special discount and offers for customers with high minute counts.

4- Minimize promotional calls and customer service calls, instead communicate through email service.

5- Special focus on states where churn rate is high such as state 31, 47, and 43.

6- Provide better and economical rates than the competitors.

7- Very few people are using international plan and the one who is using is mostly churning. So, improve numbers of international plan and promote among the people to increase users.

8- Provide data monetization and data security to the customers.

Churn Prediction Model

In this prediction model, we are using different supervised learning algorithms to design a model which predict customer churning. So, we are using 5 different machine learning classification algorithms which are Logistic regression, Decision tree, Random Forest, Gradient boosting, and K nearest neighbors to predict the best result. In last we are using an assemble method which is voting classifier which is combination of best models.

We are using AUC_ROC score to compare performance of an algorithm although, we are also calculating accuracy, but our focus is on AUC_ROC score.

Data Pre - Processing: -

Data preprocessing is an important stage for handling the data before using it in the Machine learning algorithms. Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format. For our dataset, we also must perform some data pre-processing but first, we load our dataset using pandas and check shape of our dataset which is 5,000 and 21 columns.

```
print('Length of Dataset: ', data.shape[0])
print('Number of Attributes: ', data.shape[1])
```

[194] 0.1s Python

```
Length of Dataset:  5000
Number of Attributes:  16
```

```
data.head()
```

[5] 0.1s Python

	state	account_length	area_code	phone_number	international_plan	voice_mail_plan	number_vmail_messages	total_day_minutes	total_day_calls	total_day_charge	...	tota
0	16	128	415	2845	0	1	25	265.1	110	45.07	...	
1	35	107	415	2301	0	1	26	161.6	123	27.47	...	
2	31	137	415	1616	0	0	0	243.4	114	41.38	...	
3	35	84	408	2510	1	0	0	299.4	71	50.90	...	
4	36	75	415	155	1	0	0	166.7	113	28.34	...	

5 rows × 21 columns

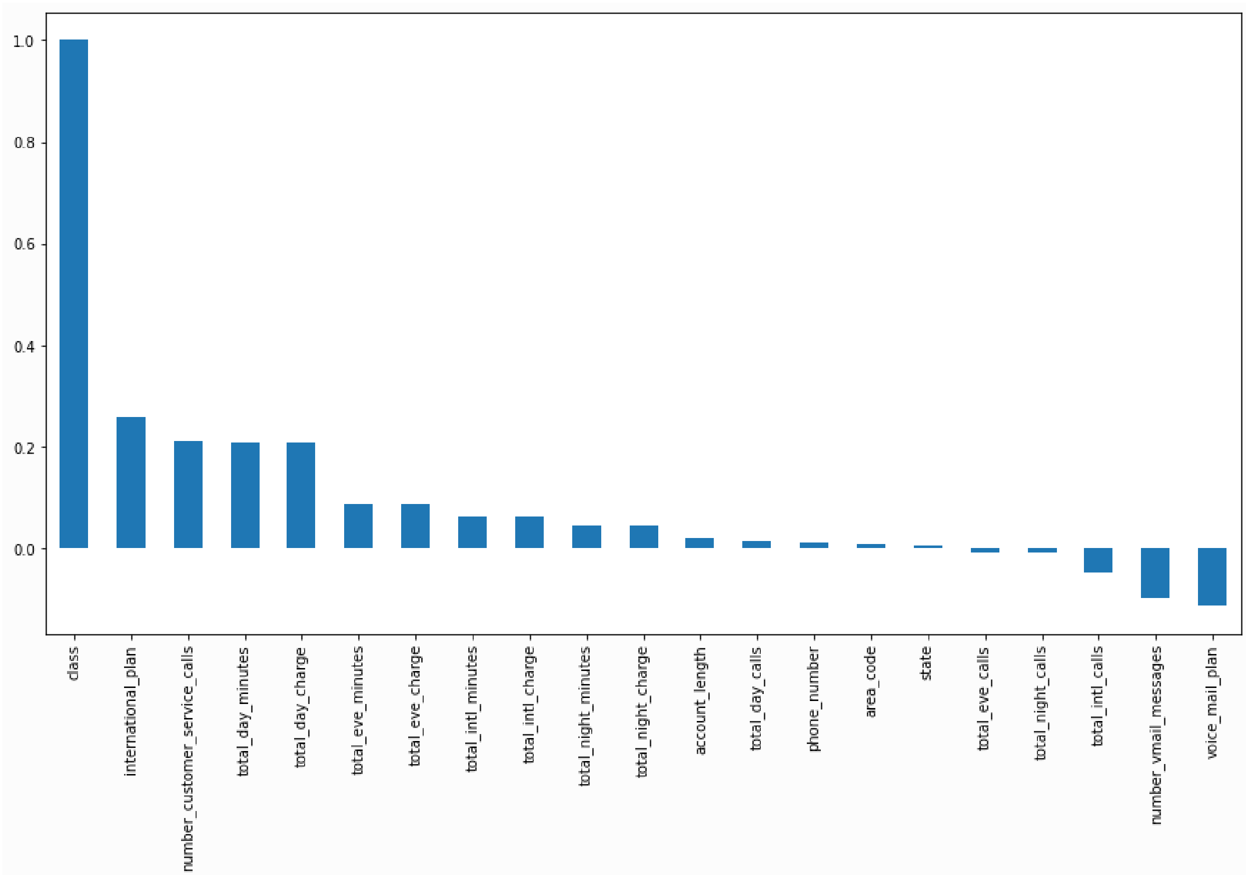
Now, we must check unique values columns, if there is any column with unique values then we must drop it. In our data set, phone number column is the one containing all the unique values.

```
[6]  data.nunique()
     ✓ 0.9s

state                              51
account_length                    218
area_code                           3
phone_number                     5000
international_plan                  2
voice_mail_plan                     2
number_vmail_messages              48
total_day_minutes                1961
total_day_calls                   123
total_day_charge                 1961
total_eve_minutes                1879
total_eve_calls                   126
total_eve_charge                 1659
total_night_minutes              1853
total_night_calls                 131
total_night_charge               1028
total_intl_minutes                170
total_intl_calls                   21
total_intl_charge                 170
number_customer_service_calls      10
class                               2
dtype: int64
```

Now we are checking null values in our data and surprisingly there is no null value present inside out dataset.

```
[91]  data.isnull().sum()
      ✓ 0.3s

state                            0
account_length                   0
area_code                        0
phone_number                     0
international_plan               0
voice_mail_plan                  0
number_vmail_messages            0
total_day_minutes                0
total_day_calls                  0
total_day_charge                 0
total_eve_minutes                0
total_eve_calls                  0
total_eve_charge                 0
total_night_minutes              0
total_night_calls                0
total_night_charge               0
total_intl_minutes               0
total_intl_calls                 0
total_intl_charge                0
number_customer_service_calls    0
class                            0
dtype: int64
```

Now, we are looking for correlation among the features of our data with respect to our target variable class. So, we found out that features such "voice_mail_plan", total night calls, total eve calls etc. are negatively correlated with class.



So now we are moving forward to machine learning algorithms. First thing we need to do is to train test split out data.

Train – Test Split: -

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. Here we are splitting our dataset in train dataset and test dataset with the ratio of 70-30% where 70% is our train dataset and 30 is our test dataset.

METHODOLOGY

We are using different machine learning algorithms in this case study for churn prediction and comparing results of each algorithm to check out which model is fitting performing well on training data set. We are also using some model enhancement techniques to improve ROC score of the model. These techniques are K-fold cross validation, Grid Search CV, and in the end voting classifier. Our sole purpose is to improve ROC score with fine accuracy. In the end, we are implementing our best model and checking its performance on new data set.

Hence, it's a binary-class classification problem. So, we are using ROC_AUC score for measuring our performance although we are also computing accuracy.

1) Logistic Regression: -

The first model we use is Linear regression model, which is a classification model and commonly use to predict probability that an instance belongs to class or not. It uses sigmoid function to predict probability of a particular class and this function return value between 0 and 1. In our case, Logistic regression is not performing very well on our train data set. It is giving us ROC score of only 0.810. which is quite fine. We use different random states to check if score increase but its result was not up to the mark.

2) Decision Tree Classifier: -

The second model we are implementing on our train dataset is Decision Tree Classifier which classify result by creating decision tree. It is quite famous supervised machine learning algorithm use for both regression and classification problems. In our case study, we implement decision tree on different parameters but the result of ROC score prediction was just fine. Although, Decision tree was giving us very good accuracy of 99%. These are some ROC score prediction which we get after implementing decision of different depths.

Decision Tree	Max_depth	ROC Score	Accuracy
1	50	0.756	100%
2	400	0.7527	100%
3	600	0.756	100%
4	5	0.831	92%

We also apply Grid search for getting best parameters and it gives us max_depth = 5 and max_features = "sqrt". So, we implement these results which give us our best roc score among decision trees results which is 0.83105.

3) K- Nearest Neighbor: -

The k-nearest neighbors (KNN) model is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. KNN check for its neighbors and give us result according to them. It calculates distance from different neighbors and find its nearest neighbors. In KNN, data point is classified based on similarity in the specific group of neighboring data points.

We are also using KNN-model in this case study; we implement model with different number of neighbors, but it is giving us result quite like Logistic regression. It is not giving us best result because maybe KNN algorithms are not good and precise with this dataset.

KNN-models	Neighbors	ROC Score	Accuracy
1	5	0.6031	88.58%
2	50	0.6091	86.28%
3	10	0.5766	84.50%
4	100	0.5788	84.70%

4) Random Forest Classifier: -

Random forest is one of the famous supervised learning algorithms. It's an ensemble model which combines the findings of numerous decision trees to create a more powerful learner. RF has a high noise resistance and is resistant to overfitting. Bagging and random selection are the two main concepts of Random Forest.

In our case study, performance of random forest is quite well. It is giving us ROC score better than previous models maybe because it is good with handling large number of features. It is good with high dimension data as we know that it works with making subsets by replacement. So, maybe this is the reason it is giving us good accuracy and roc score on train dataset.

N estimators	Max depth	ROC Score	Accuracy
50	5	0.8908	91.69%
150	5	0.8930	92.21%
150	15	0.8840	99.2%
200	8	0.8854	96.8%

We are also using Grid Search cross validation on Random Forest, which is suggesting us parameters of 550 n estimators and 40 max depths. These parameters are also giving us result of 0.8834 ROC with random forest as we can see above.

5) Gradient Boosting Classifier: -

Gradient boosting is supervised machine learning algorithm used for classification and regression problem. It is an ensemble technique in which we convert weak learners into strong learners for improving the model prediction of any given algorithm. the weak learner is sequentially corrected by predecessors and in the process, they are converted into strong learners. gradient boosting classifier is usually fast and contain less storage is compared to other algorithms.

In this case study, we also use Gradient boosting classifier and it’s giving us optimum results as compared to other algorithms. Gradient boosting is better than random forest because it converts and enhance weak learner by reducing its error, but if the data is noisy than accuracy of gradient may affect sometime. We implement gradient boosting with different max depth, learning rate and n estimators and it’s fitting quite well. We are also using Grid search on Gradient boosting for better parameter selection.

N estimators	Max depth	Learning Rate	ROC Score
50	2	0.1	0.895
50	5	0.1	0.891
350	6	0.1	0.876
200	5	0.05	0.889

Here we are getting our best result through using hyper optimization tunning of gradient booting classifier on depth 5 with 0.05 learning rate and 200 estimators. Whereas we are getting slightly less ROC score value with random forest.

Below we are comparing some of the best results obtain from different models.

Models	ROC Values
Logistic Regression	0.810
Decision Tree	0.831
KNN – Algorithm	0.601
Random Forest	0.893
Gradient Boosting	0.889

VOTING CLASSIFIER (Hybrid Model): -

Voting classifier is basically an ensemble technique which unite different base classifiers and based on result of these classifiers it computes new result. It is a technique that may be used to improve model performance, ideally achieving better performance than any single model used in the ensemble.

We are using soft voting technique here which it computes probability based on probabilities and weights associated with different classifiers. In voting classifier, we are using our best models. We are using 6 random forest models and 2 gradient boosting models which were giving us best results. So, after combining all our best performing model we create and voting classifier model, which is giving us good result but not better than Random Forest.

Models	N estimators	Max depth	Learning Rate	ROC Score
Grad Boosting 1	200	5	0.05	0.889
Grad Boosting 2	350	6	0.01	0.876
Random Forest 1	550	40	-	0.8834
Random Forest 2	200	8	-	0.884
Random Forest 3	150	5	-	0.8930
Random Forest 4	250	15	-	0.884
Random Forest 5	450	40	-	0.890
Random Forest 6	350	8	-	0.88
Voting Classifier	-	-	-	0.8876

RESULTS

This section presents the result after executing voting classifier model (GB + RF) and other base models such as GB, RF, LR, DT, and KNN.

Models	ROC Values
Logistic Regression	0.810
Decision Tree	0.831
KNN – Algorithm	0.601
Random Forest	0.893
Gradient Boosting	0.889
Voting Classifier	0.8876

So, here we can see Random Forest classifier is giving us our best ROC Score of 0.899. So, we will implement Random classifier on our validation dataset and consider this as our best model. Among our base models, Random Forest is giving best ROC Score. Decision tree is good in term of accuracy, but its ROC score is not up to the mark. Whereas Gradient boosting is also giving good roc score with 84% accuracy. After considering all the results, Random Forest classifier shows the highest performance results to predict churn.

Performance on Validation Set

After selecting our best model which is Random Forest classifier (depth = 5, estimators = 150), now we are moving forward to fit our best model on validation set. So, after fitting our model on validation set, we got ROC score of 0.946 which is highest and better than roc score on train data which was 0.899 before.

So, lift or our model is around 7% as compared to train dataset score.