Safety Data Analytics

Customer Churn Prediction

Ajinkya(ajinkya.takawale97@gmail.com), Devjyoti(devjyotichandra1@gmail.com), Manish(manishagrawal.iitkgp@gmail.com)

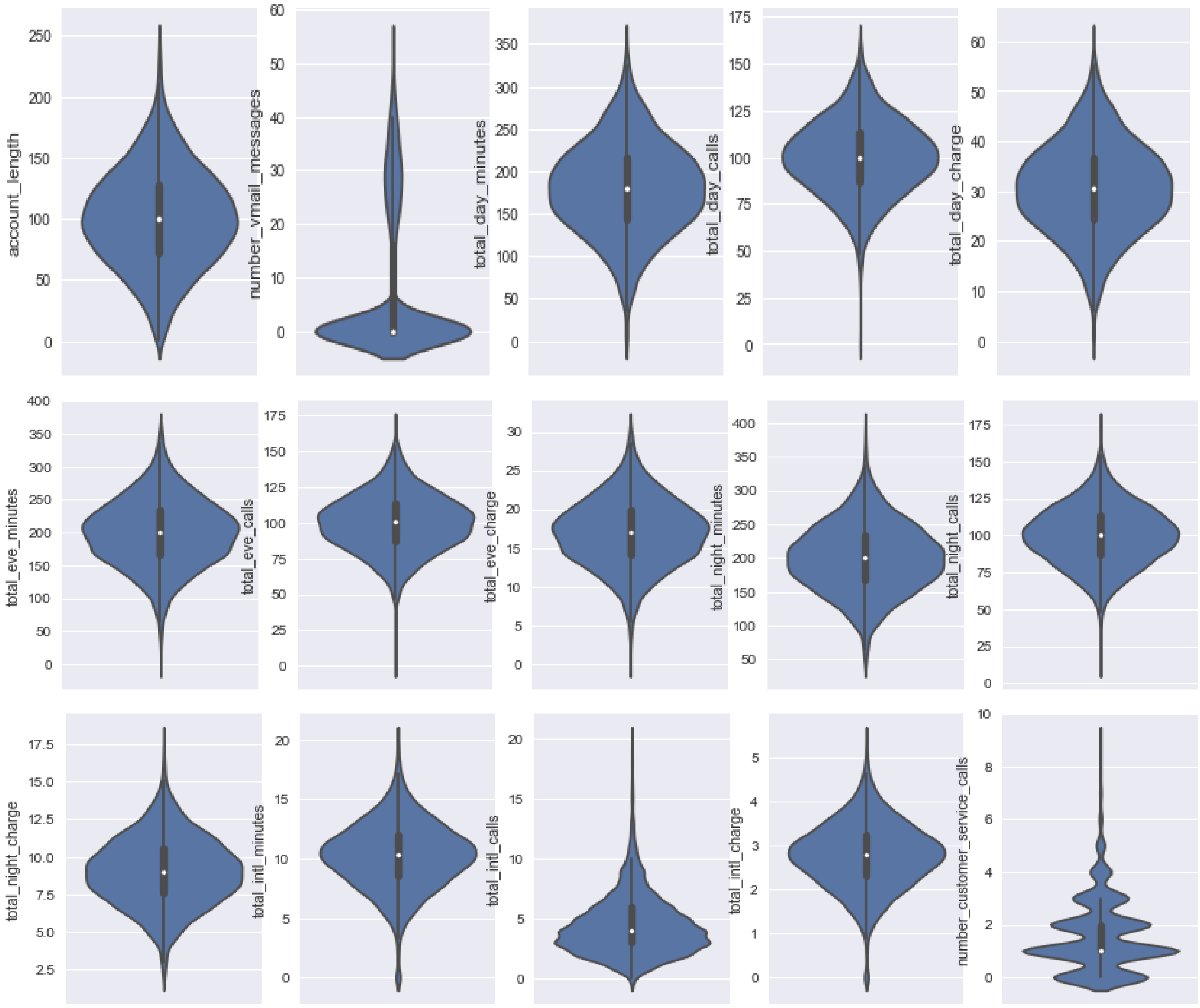
Customer Churn can be simply understood as customers not continuing further with the services that the company is providing. Knowing the insights of the customers is important for the companies as based on those insights they further enhance the company customer relations. The given statement and dataset are for the Telecommunication companies. The dataset contains the valuable information of customers such as their total day calls, charges they pay for calls, total minutes for calls, whether they have opted for international value pack, the area code of their residence, etc.

The given dataset required some initial analysis to draw out the insights. Based on these insights we draw out further modelation is carried out. **Violin Plots** were drawn which are histogram (combination of Density plot and Box plot) turned on its side and mirrored. Violin plots shows the distribution along with the quantiles.

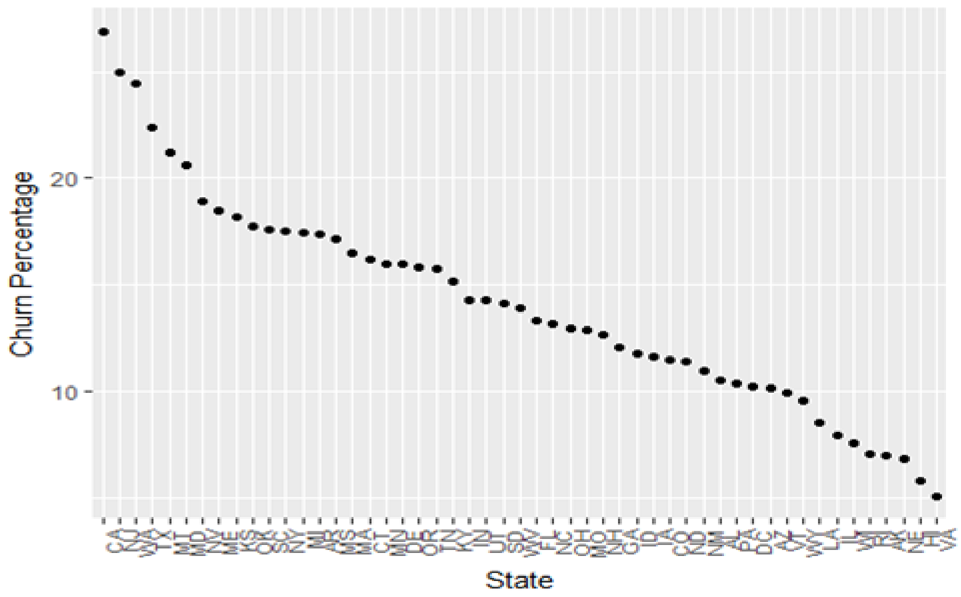
Heatmap : The variables were tested if there are some correlation among themselves. For this purpose correlation matrix was calculated and then graphically represented in Heatmap. Following is the Heatmap graph which was produced:

Given dataset was split into training(80%) and testing set(20%) using createDataPartition() of R Caret package. createDataPartition() does a stratified split of the data.

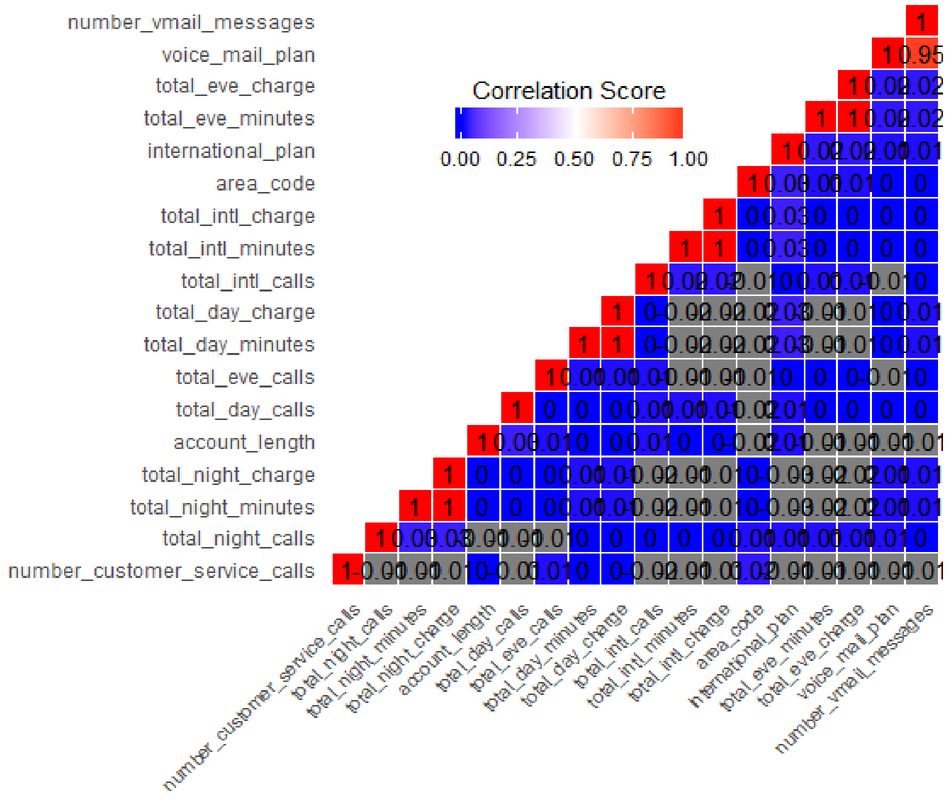
Different plots were drawn to get some insights about the data. Sometimes the median and mean aren’t enough to understand a dataset. Therefore, violin plots were made for each variable which show the probability density of the data at different values. The white dot represents the median, the thick black bar in the center represents the interquartile range and the thin grey line represents the 95% confidence interval.

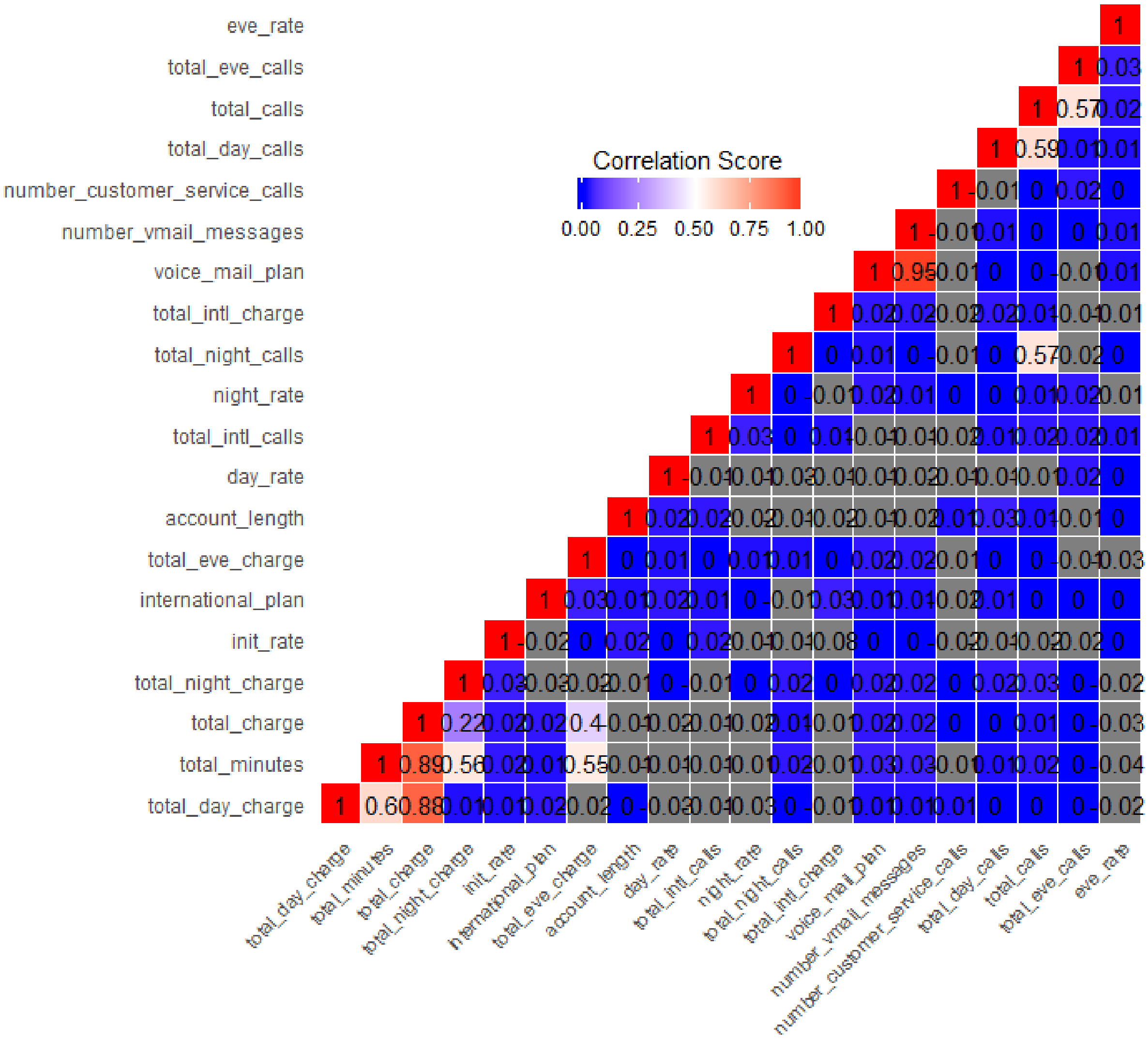


This plot shows state-wise churn percentage.



Correlation among the features were calculated and visualized below using heatmaps. It is found that total_day_minutes, total_eve_minutes and total_night_minutes are highly correlated with total_day_charge, total_eve_charge and total_night_charge. So we kept only charge variables. Also we removed number of voicemail messages after observing feature importance and high correlation.

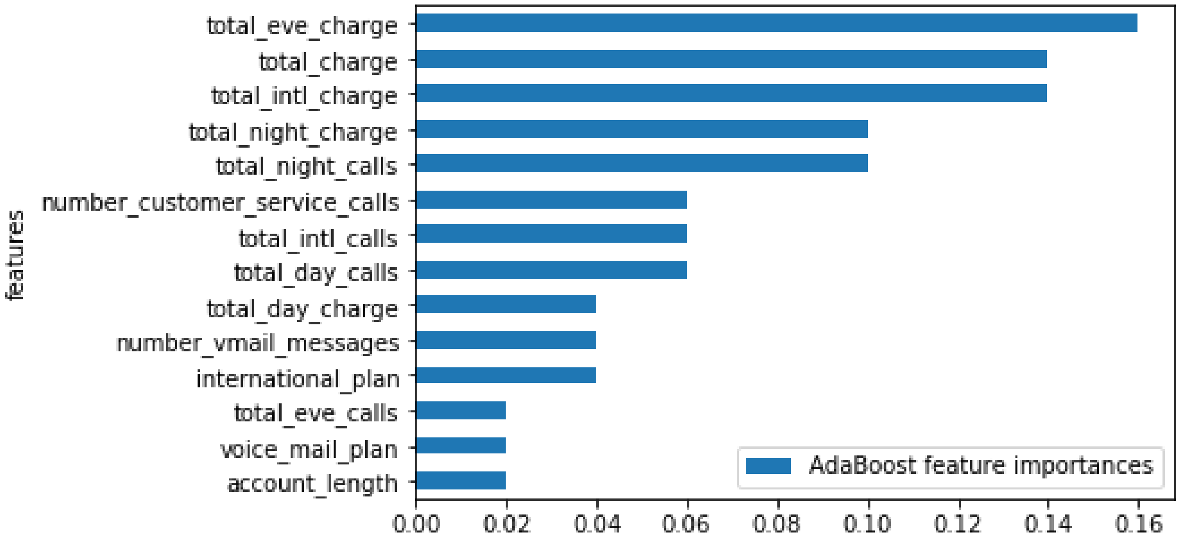


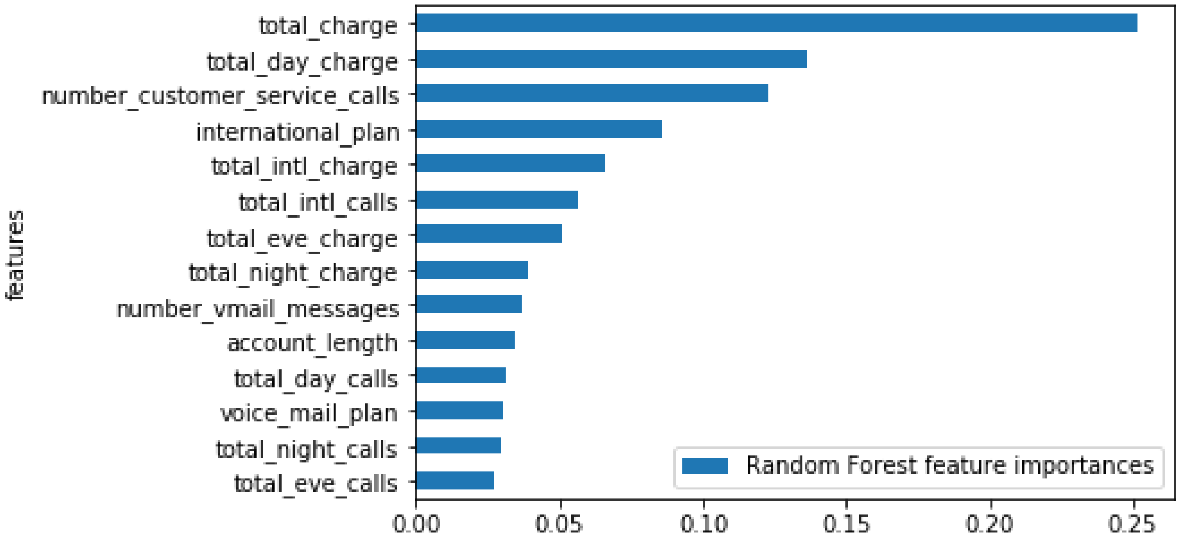


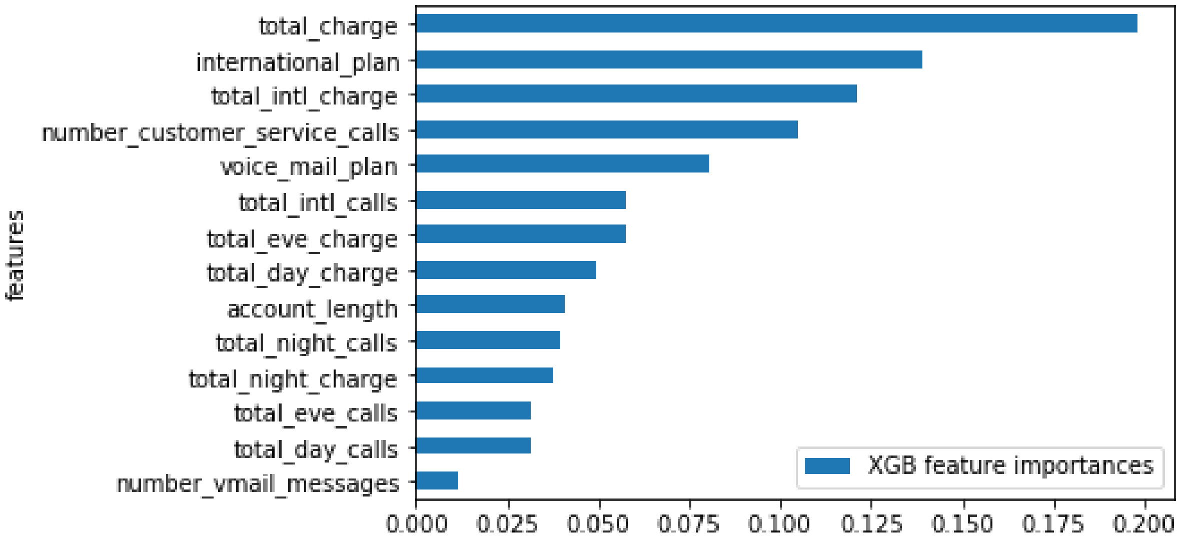
The following models were build on training dataset using scikit learn library in python.

1. Random Forest
2. AdaBoost Classifier
3. K Nearest Neighbors
4. Logistic Regression
5. Gaussian Naive Bayes
6. Support Vector Machine
7. XGBoost

We first calculated feature importance of original training set after removing highly correlated feature and also created new feature total_charges which is sum of all total_carges by time of the day variables which later was found to be very important variable. We used Random Forest, AdaBoost, and XGBoost to calculate feature importance of which plots are shown below:


features
total_eve_charge
total_charge
total_intl_charge
total_night_charge
total_night_calls
number_customer_service_calls
total_intl_calls
total_day_calls
total_day_charge
number_vmail_messages
international_plan
total_eve_calls
voice_mail_plan
account_length
AdaBoost feature importances
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16


features
total_charge
total_day_charge
number_customer_service_calls
international_plan
total_intl_charge
total_intl_calls
total_eve_charge
total_night_charge
number_vmail_messages
account_length
total_day_calls
voice_mail_plan
total_night_calls
total_eve_calls
Random Forest feature importances
0.00
0.05
0.10
0.15
0.20
0.25

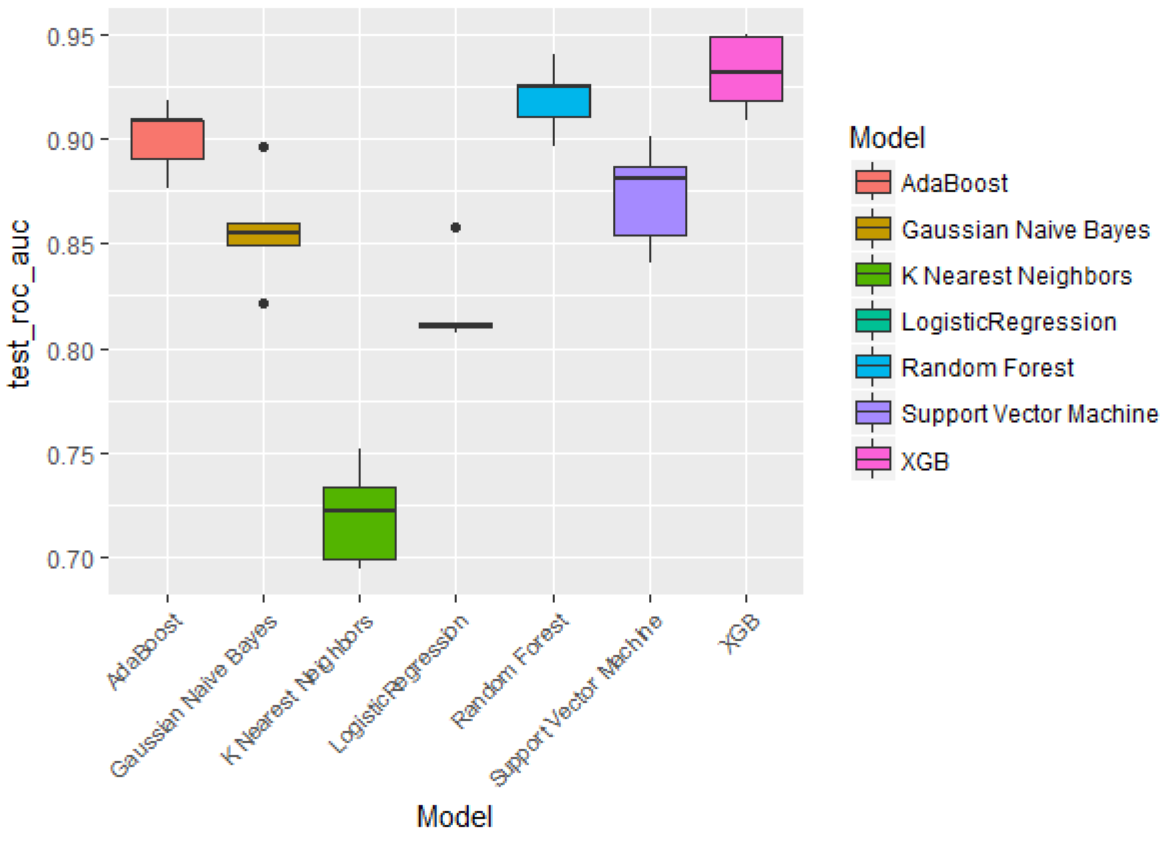

features
total_charge
international_plan
total_intl_charge
number_customer_service_calls
voice_mail_plan
total_intl_calls
total_eve_charge
total_day_charge
account_length
total_night_calls
total_night_charge
total_eve_calls
total_day_calls
number_vmail_messages
XGB feature importances
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200

We found following optimized hyperparameter for our models:

Model	Tunned Hyperparameters	ROC_AUC Score
Random Forest	n_estimators: 100 | max_depth: 4 | max_features: 'sqrt' | min_samples_leaf: 2	0.92
SVM	C: 1000 | gamma: 0.0001	0.87
Logistic Regression	C: 10	0.82
XGBoost	colsample_bylevel: 1 | n_estimators: 100 | colsample_bytree: 0.8 |	0.93
	learning_rate: 0.1 | max_depth: 6 | reg_alpha: 1 | reg_lambda: 0.5	
AdaBoost	learning_rate: 0.1	0.90
KNN	neighbours: 9	0.69

Below is the summary of models we trained:

Model	test_acc	test_f1	test_precision	test_recall	test_roc_auc
AdaBoost	0.91	0.58	0.81	0.46	0.90
Gaussian Naive Bayes	0.87	0.53	0.52	0.53	0.86
K Nearest Neighbors	0.88	0.35	0.74	0.23	0.72
LogisticRegression	0.87	0.30	0.59	0.20	0.82
Random Forest	0.94	0.72	0.97	0.57	0.92
Support Vector Machine	0.91	0.57	0.80	0.44	0.87
XGBoost	0.97	0.90	0.98	0.83	0.93



After comparing different models from box plot and average cross validation result report we finally decided to go with Random Forest, AdaBoost, Gaussian Naive Bayes, Support Vector Machine, and XGBoost. Where while blend ensembling we used all the above models as base learners and XGBoost as super learner on the blended dataset. Final classification report on test dataset is shown below:

Final Classification Report

Churn	Precision	Recall	F1-Score	Support
No	0.98	1	0.99	885
Yes	0.98	0.88	0.92	145