**Introduction**

Churn (loss of customers to competition) is a problem for telecom companies because it is more expensive to acquire a new customer than to keep your existing one from leaving.

Most telecom companies suffer from voluntary churn. Churn rate has strong impact on the life time value of the customer because it affects the length of service and the future revenue of the company. For example if a company has 25% churn rate, then the average customer lifetime is 4 years; similarly a company with a churn rate of 50%, has an average customer lifetime of 2 years. It is estimated that 75 percent of the 17 to 20 million subscribers signing up with a new wireless carrier every year are coming from another wireless provider, which means they are churners. Telecom companies spend hundreds of dollars to acquire a new customer and when that customer leaves, the company not only loses the future revenue from that customer but also the resources spend to acquire that customer. Churn erodes profitability.

**Data Description**

The predictors provided are as follows:

**● account length**

**● international plan**

**● voicemail plan**

**● number of voicemail messages**

**● total day minutes used**

**● day calls made**

**● total day charge**

**● total evening minutes**

**● total evening calls**

**● total evening charge**

**● total night minutes**

**● total night calls**

**● total night charge**

**● total international minutes used**

**● total international calls made**

**● total international charge**

**● number of customer service calls made**

**Target Variable : move: if the customer has moved (1=yes; 0 = no)**

**This is an imbalanced dataset as number of customers who churned are high compared to the number of customers who did not Churn(~14%)**

**Data Cleaning**

* **Checking for Null Values**

The training and the testing set were both tested for null values first. The datasets did not contain any null values.

* **Dropping Redundant Columns**

The phone number column in the training and test set was dropped as it is unique to a customer and would not provide any information for classifying the samples.

* **Encoding categorical columns**

The categorical columns such as Churn, voice-mail plan and international plan contained string values such as True and False or yes and no.

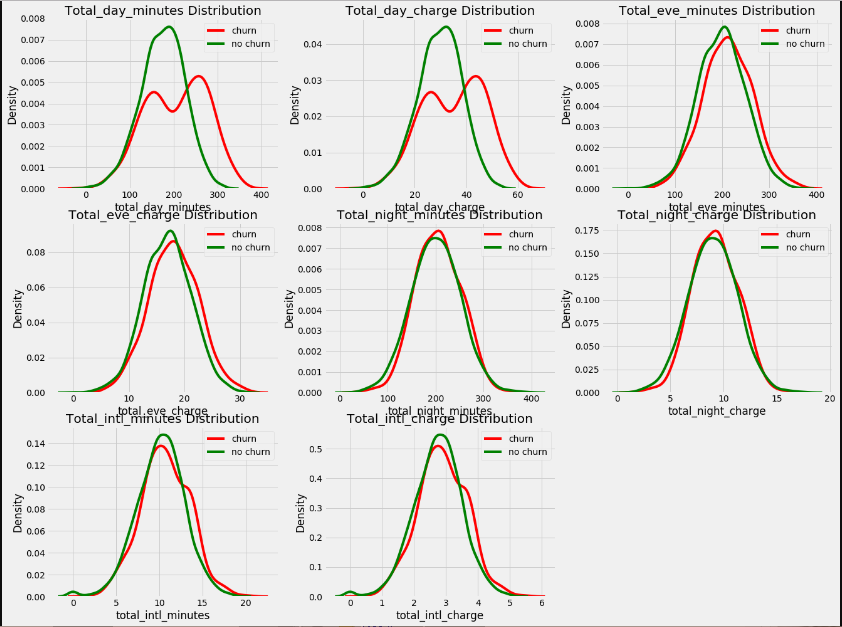
These values where encoded to their numeric representation of 0 and 1 respectively.

**Exploratory Data Analysis**

Extensive exploratory data analysis was performed on the dataset to gather insights from the data and identify potential features that would help us in the next machine learning step which is feature engineering.

And lastly, we also used T-Sne, a data visualization technique to help us get an idea about the separation between the classes.

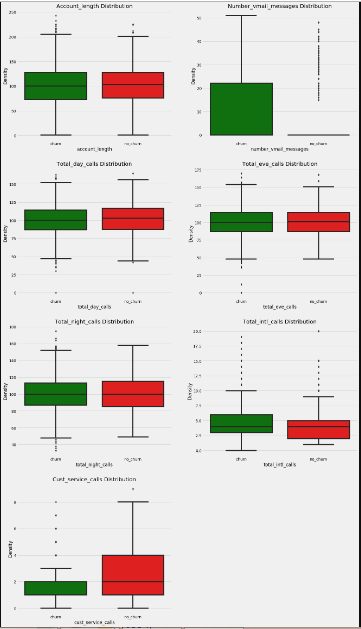
**Analysis of distribution of continuous variables w.r.t to Target**



**Observations:**

1. Total day minutes distribution is different for Churned and not-churned Customers
2. Minutes and charge distributions are very similar for day, evening and night which should be expected, as the charges are calculated based on the number of minutes/talk time.

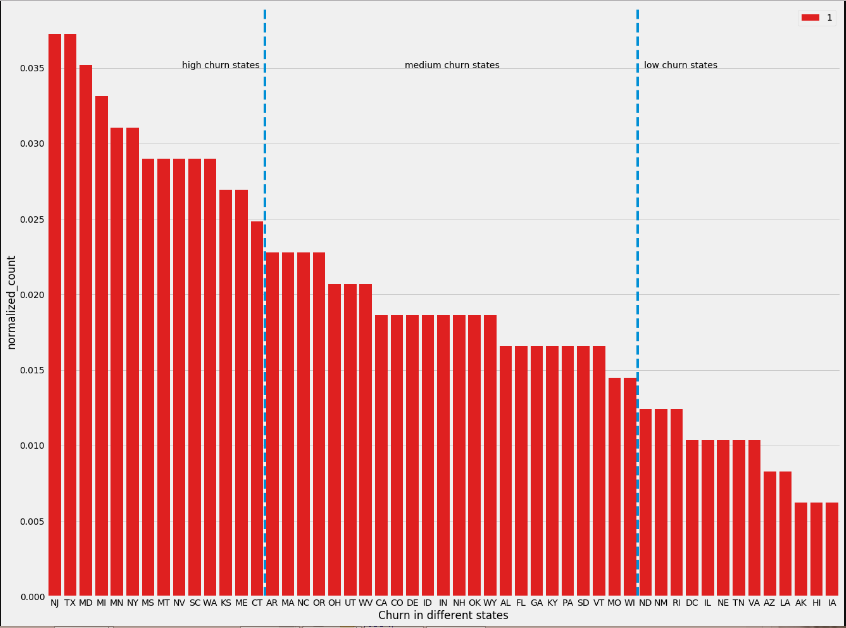
**Analysis of distribution of discrete variables w.r.t to Target**



**Observations:**

1. Total day calls, total international calls, number of voice mail messages, number of customer service calls have a significant difference in their box-plot distributions.
2. It appears that the customers who churn make a lot of customer service calls indicating that they are dissatisfied with the company’s service or their complaints are not being resolved effectively.
3. The customers who churn have almost 0 voice mail messages as most of them did not opt for a voice mail plan.

**Churn by State**



**Observations:**

1. The churn rates are different in each state and therefore a customer from a high churn state might have a higher probability of churning.
2. Therefore, we should take this into account in feature engineering stage and consider binning the states into segments of high, medium and low churn.

**Feature Engineering**

We engineered a total of 18 new features from our existing features to give our ML models more scope to differentiate between the two classes.

Some of the engineered features are:

**1. Voice and international plan**

This feature tells us if the customer has opted for both a voice mail and international plan indicating a customer being invested in the telecom company

**2. Minutes / call**

A continuous feature obtained by dividing total day, evening , night and international calls with their respective minutes giving us an idea about minutes spent talking on call at different times of day and internationally.

**3. Revenue and Revenue per day**

These features tell us the total revenue from the customer and revenue per day with respect to the account length.

**4. Difference from state aggregate**

We calculated the median statistic of various features such as number of customer service calls, revenue, day, evening and night calls for different states and obtained a difference of the values from features of customers belonging to their state.

**5. Statewise churn segments**

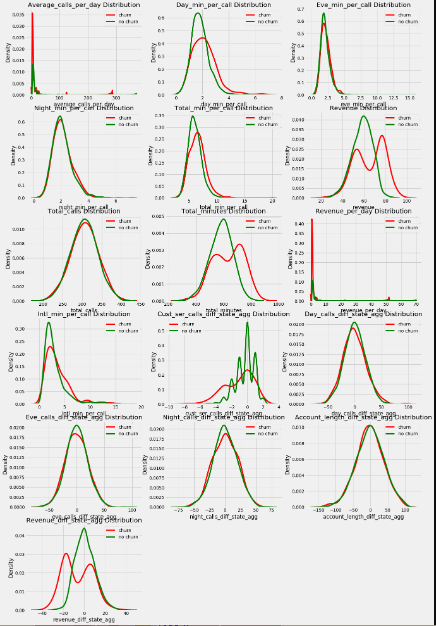
We engineered a new feature where we segregated each state with respect to the number of customers churned and assigned a number indicating the state belonging to a particular segment.

**6. Target encoding state and area code**

We used target encoding to encode our categorical features state and area code rather than label encoding them with arbitrary numbers. Encoding them this way preserves the distributions the categories share with the Target and provides us with useful values to replace these categories with.

**Data Analysis post Feature Engineering**

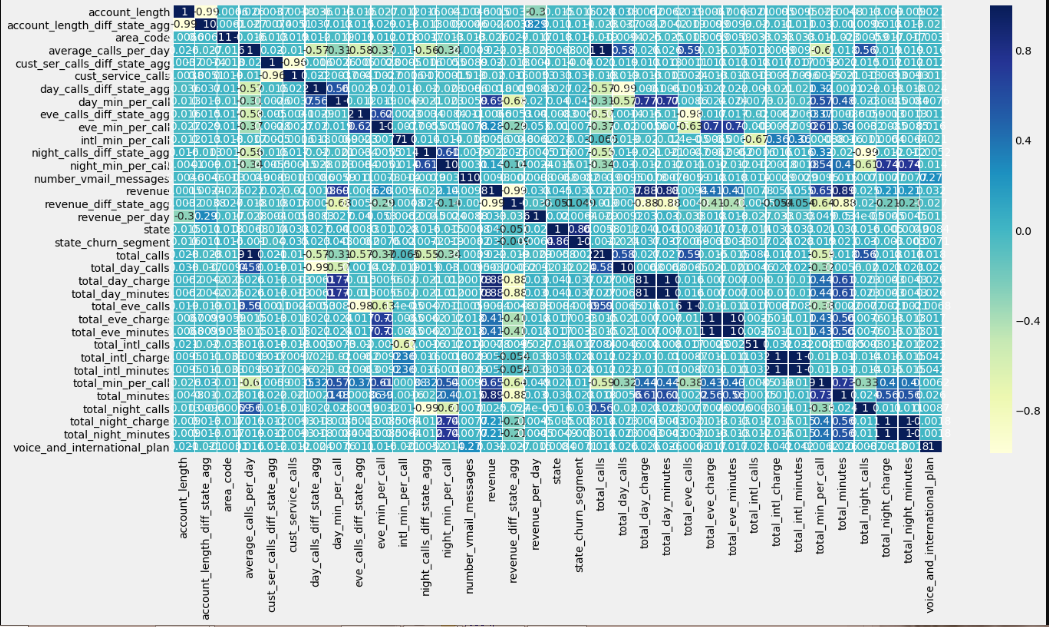
We performed data analysis to assess if our engineered features could be useful features for classification and can help our machine learning models discriminate between the two classes.



**Observations:**

1. Day\_min\_per\_call, total\_min\_per\_call, Revenue, International minutes per call, Customer Service\_calls\_diff\_state\_agg appear to be useful for classifying our target classes.

**Correlation Plot of all features**



**Observations:**

We can see that a lot of the features are highly correlated with each other. This is expected as we have engineered features included here as well.

We would be scaling our features by removing the mean and dividing by std deviation to remove the Data collinearity.

We would be performing feature selection using variance inflation factor to remove highly collinear variables.

**Feature Selection using Variance Inflation Factor**

Variance Inflation factor detects multicollinearity in our dataset by regressing each predictor variable with the rest of the predictors and produces a VIF value which is the inverse of R-squared value.

This value tells us how much of variance of a predictor is explained by other variables.

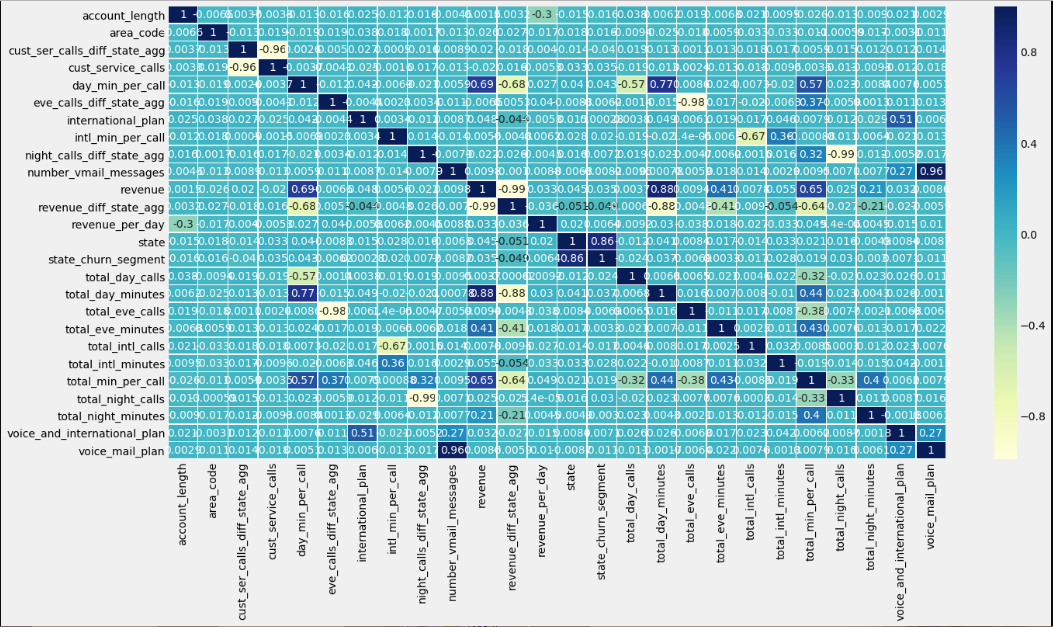
Generally, values above 5 to 8 means that there is a strong indication of collinearity in the variables and values above 8 indicate problematic collinearity which needs to be dealt with.

After performing VIF analysis for two iterations, we decided to drop some of these features from our dataset:

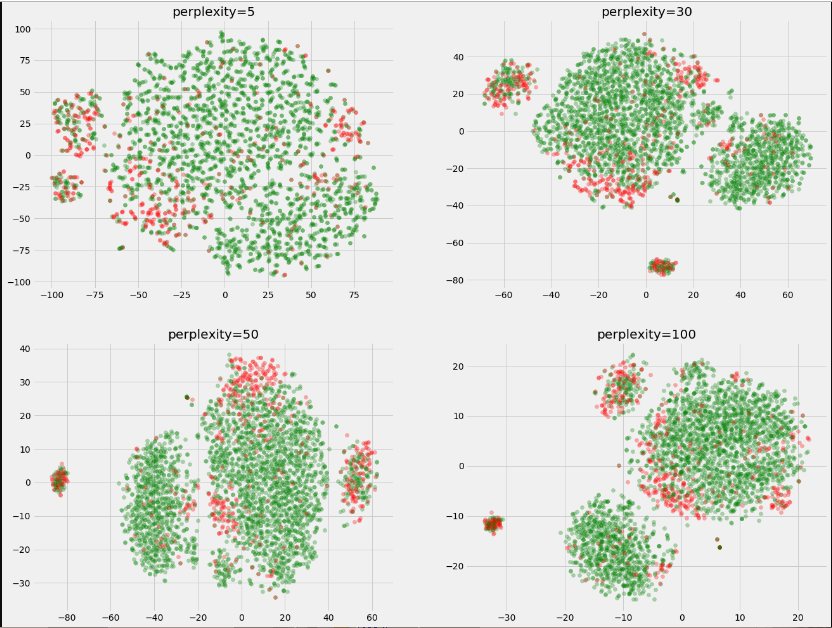
1. Day, night, evening and international charges.

2. Night\_min\_per\_call, evening\_min\_per\_call, average calls per day etc.

**Correlation plot after dropping some features**



**Data Visualization using TSne**



**Observations:**

1. We see some degree of separation in our dataset as observed in two dimensions.

Using this visualization technique gives us an approximate idea about the class separability in our data.

**Dataset Preparation**

This is an imbalanced data set where the number of samples in each class is not balanced. This would cause a problem to ML models which are classifying our data the models would tend to classify the samples as belonging to the majority class and undermine the minority class samples.

We could address this issue by oversampling our minority class to match the total number of samples in the majority class by randomly sampling from the minority class samples with replacement.

We should also be careful during cross validation as the number of samples in a validation fold should have both the classes represented. This can be addressed by using stratified k-fold during cross validation.

**Evaluation Metric**

There are quite a few metrics that need to be considered for our classification task apart from accuracy.

Therefore, we would be analyzing the precision-recall curve, roc curve, auc and f-measure to assess the performance of our classification models

Since, we are oversampling our training samples before fitting our models with the training set or folds during cross-validation; we would be using accuracy as our evaluation metric.

The remaining metrics would be used later on after predicting on our testing set.

**ML Models and Hyper-parameter tuning**

We would be using three types of classification models that belong to different families of ML models and therefore, would help us understand how different models perform on our dataset.

**Logistic Regression(Generalized Linear Models)**

**SVM(Non-parametric models)**

**Gradient-Boosting Trees(Ensemble Models)**

We would be performing hyper-parameter tuning of these models using Bayesian optimization which uses the Bayes’ Theorem in the background to assess which parameters would yield a greater score or minimize the loss of an ML model based on past history of parameter values.

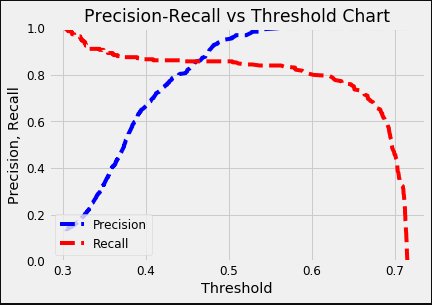
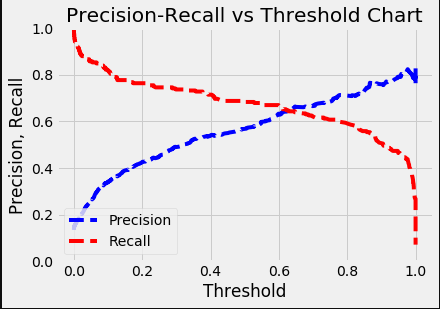
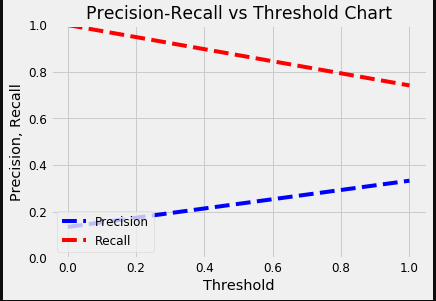
**Results**

**Models Logistic SVM GBM**

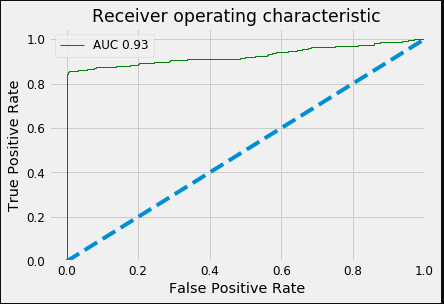
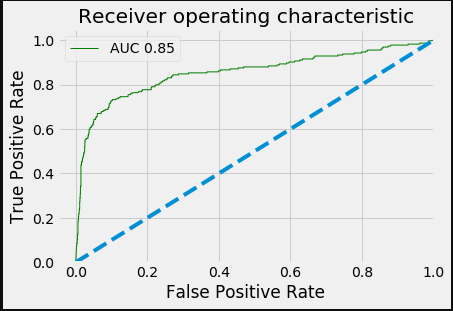
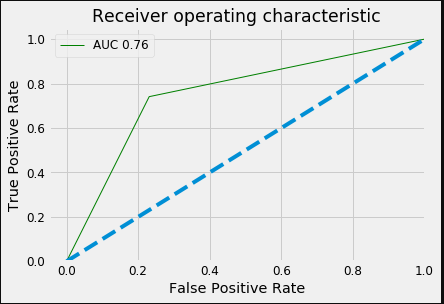
**Accuracy: 0.7654 0.8836 0.9754**

**F\_score: 0.4591 0.6135 0.9035**

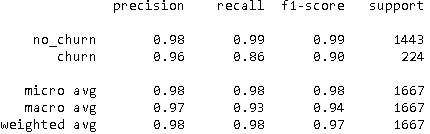
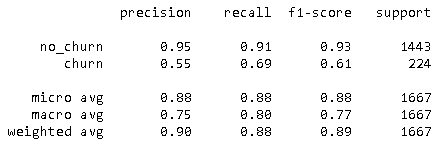
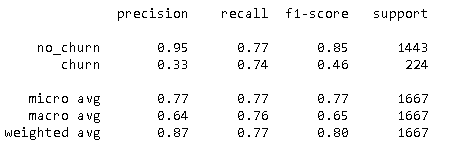
**Precision Recall Curve**



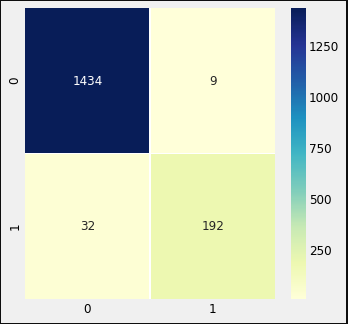
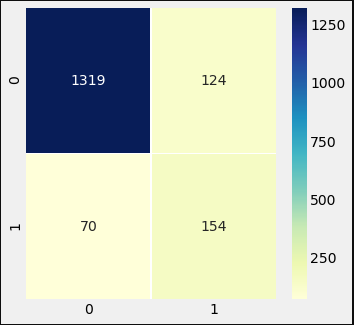
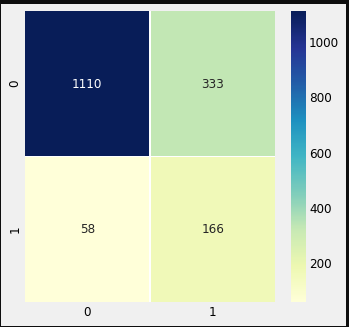
**ROC-Curve with AUC**



**Classification Report**



**Confusion Matrix**



**Conclusion**

We started our modelling stage by using Logistic Regression which assumes a linear relationship between the target variable and the predictors and outputs a class probability of a sample belonging to a particular class.

As per the results, we can see that it does a decent job of classifying our test data in terms of accuracy because of the class imbalance as most of the samples belong to the no-churn class.

We need to look at other metrics to gauge the performance of our classifier.

If we observe the F-measure of our churn class, it is quite low which means that it is doing a poor job in correctly classifying our churn class as we can observe the precision and recall of 0.33 and 0.74 respectively. ROC curve also highlights this issue as the AUC is quite low at 0.76 compared to other classifiers used for prediction.

Therefore, we can conclude that there exist non-linearities in our data which cannot be adequately modelled by the logistic regression classifier.

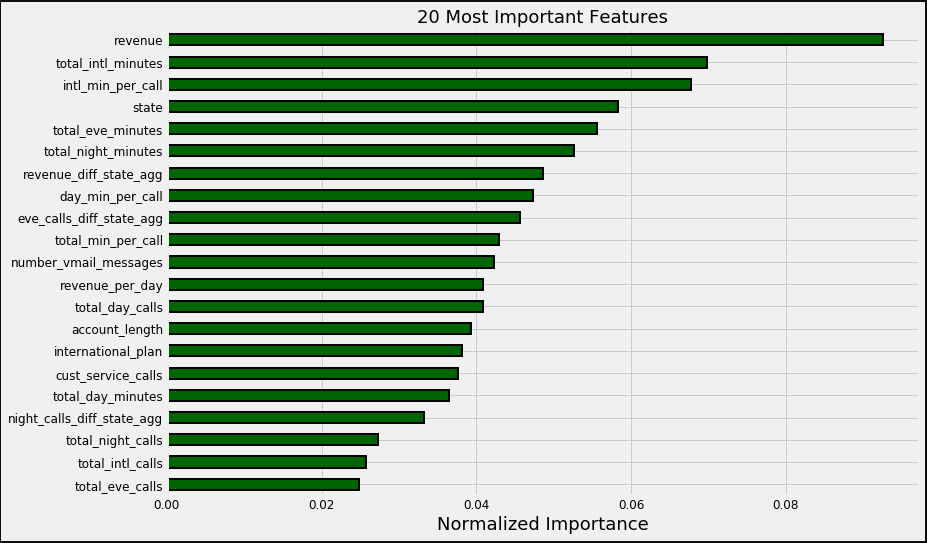
The next classifier we used was the SVM classifier which tries to find a separating plane between the samples in our training data. It also projects our features to a high dimensional space using kernels to find a better separation between classes.

The results show that it was able to capture non linearities in our data going by the metrics calculated from the predictions. It outperforms logistic regression in every evaluation criteria and exhibits a better recall and precision of our churn class.

The final classifier, which is the Gradient Boosting Trees from the family of ensemble models, provides the highest accuracy and the highest precision and recall for our churn class.

High performance of our GBM classifier is attributed to the fact that it combines the predictive power of many weak learners to predict the target class and hence, tends to outperform traditional machine learning models.

**Feature Importances given by the GBM Model**



The engineered features in the feature engineering stage such as revenue, international minutes per call, day min per call etc. all feature in the 20 most important features for prediction by the GBM model which reinstates the fact that feature engineering is a crucial step to improve the performance of ML models.