Churn Reduction - Report

Syed Imran

16 October 2018

Contents

1 Introduction

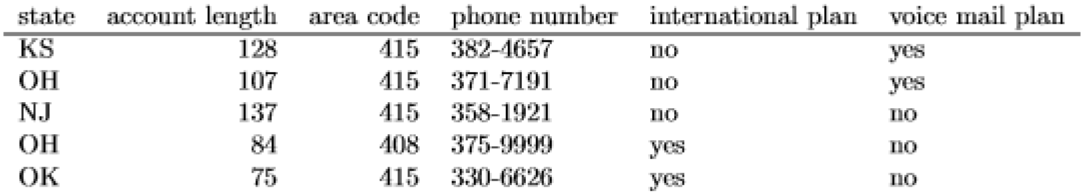
1.1 Problem Statement

Churn (loss of customers to competition) is a problem for companies because it is more expensive to acquire a new customer than to keep your existing one from leaving. The aim of this project is to predict the customers which company might lose to competition using machine learning models based on the usage of the services.

1.2 Data

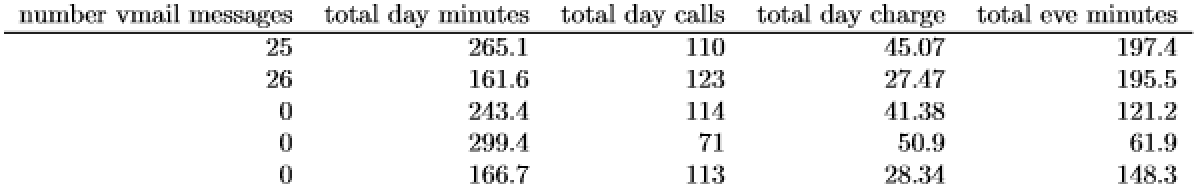
Our task is to build classification models which will classify the customer depending on various service usage factors. Given below is the sample of the data set that we will use to classify the customers.

Table 1: Training Data (Columns 1-6)



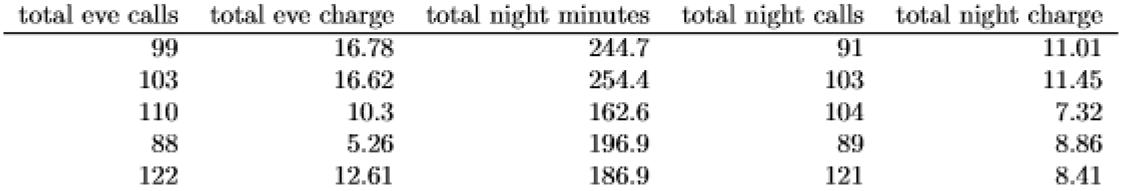
state	account length	area code	phone number	international plan	voice mail plan
KS	128	415	382-4657	no	yes
OH	107	415	371-7191	no	yes
NJ	137	415	358-1921	no	no
OH	84	408	375-9999	yes	no
OK	75	415	330-6626	yes	no

Table 2: Training Data (Columns 7-11)



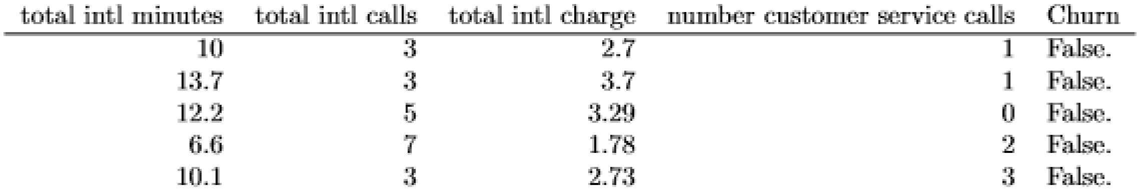
number vmail messages	total day minutes	total day calls	total day charge	total eve minutes
25	265.1	110	45.07	197.4
26	161.6	123	27.47	195.5
0	243.4	114	41.38	121.2
0	299.4	71	50.9	61.9
0	166.7	113	28.34	148.3

Table 3: Training Data (Columns 12-16)



total eve calls	total eve charge	total night minutes	total night calls	total night charge
99	16.78	244.7	91	11.01
103	16.62	254.4	103	11.45
110	10.3	162.6	104	7.32
88	5.26	196.9	89	8.86
122	12.61	186.9	121	8.41

Table 4: Training Data (Columns 17-21)



total intl minutes	total intl calls	total intl charge	number customer service calls	Churn
10	3	2.7	1	False.
13.7	3	3.7	1	False.
12.2	5	3.29	0	False.
6.6	7	1.78	2	False.
10.1	3	2.73	3	False.

The Features available to predict whether the customer will move or not are:

S.No.	Features
1	state
2	account length
3	area code
4	phone number
5	international plan
6	voice mail plan
7	number vmail messages
8	total day minutes
9	total day calls
10	total day charge
11	total eve minutes
12	total eve calls
13	total eve charge
14	total night minutes
15	total night calls
16	total night charge
17	total intl minutes
18	total intl calls
19	total intl charge
20	number customer service calls

Table 5: Features Available

2 Methodology

2.1 Pre Processing

Any predictive modeling requires that we look at the data before we start modeling. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as **Exploratory Data Analysis.** To start this we will first separate the variables in continous and categorical types. Then we will look at the distribution of the continous variables. Most analysis require the data to be normally distributed. We can visualise this by glancing at the distribution plots of the variables.

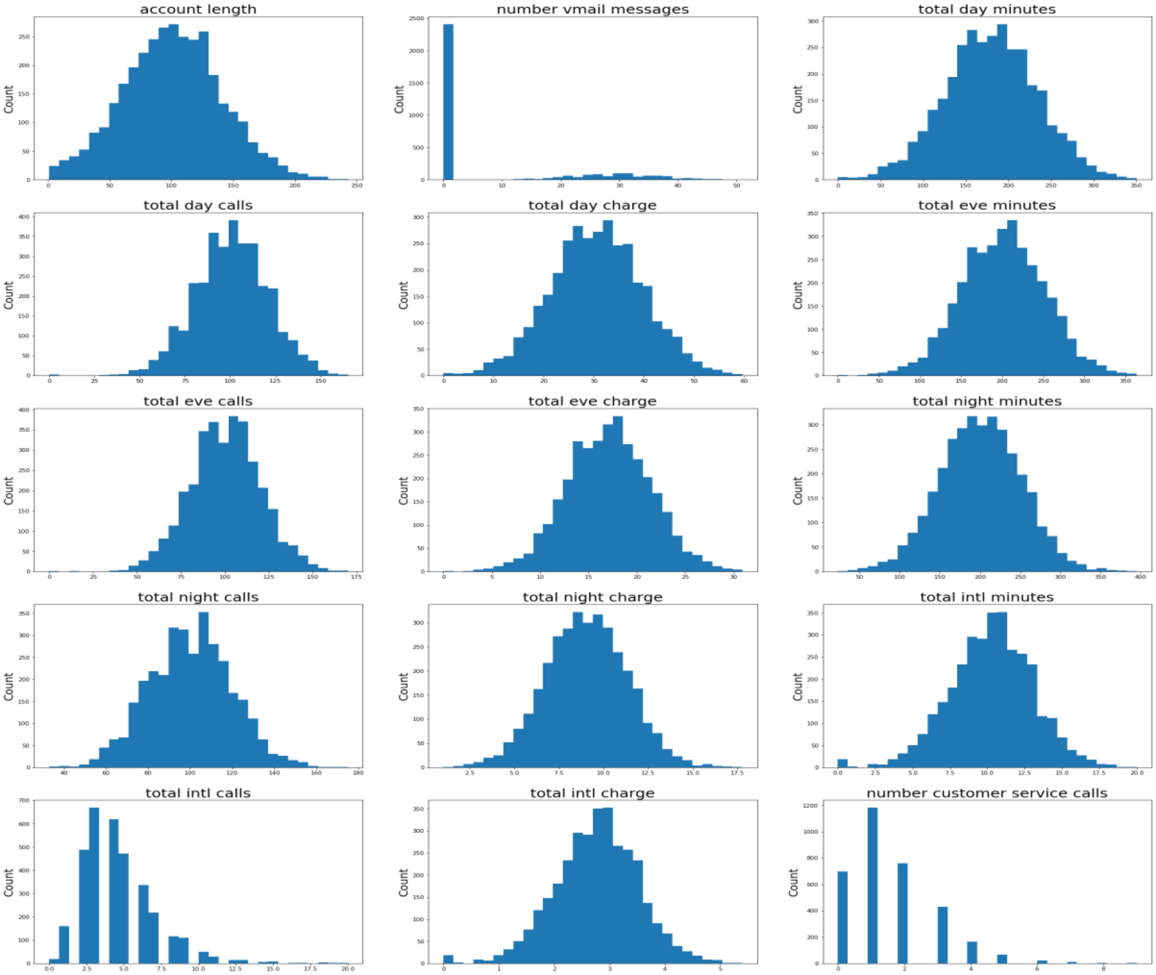


Figure 1: Distribution plots

In Fig.1 we have plotted distributions of the continous variables we have in the data and we can see that all of them are almost uniformly distributed except the *number of voicemail messages, total international calls and number of service calls*.

We also look at the counts of the target variable i.e. *Churn* shown in Fig.2. We can clearly see that there is a target class imbalance.

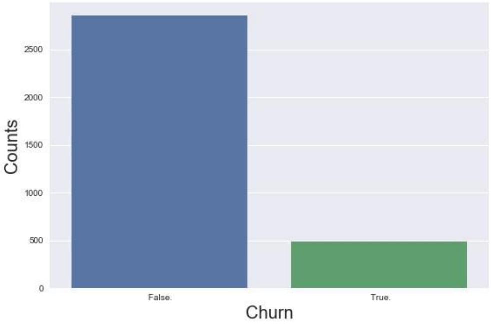


Figure 2: Count plot of Target Class

2.1.1 Outlier Analysis

As we have seen the variables *number of voicemail messages, total international calls and number of service calls* are skewed, these could be due to outliers, but looking at the *number of voicemail messages* without the zero value which has clearly the most count, the rest of the data is uniformly distributed as shown in Fig.3 also the zero value is not specific to any of the target class as seen in Fig.4, so we will leave the zero value as it is otherwise if we drop those observations we will lose too much of data and the zero value of this variable has a significance. The variables *total international calls and number of service calls* also have similar distributions for both classes as shown in appendix in fig.8 and 9 and the range of these variables aren't too big as to remove the outliers so we will also leave these variables as it is.

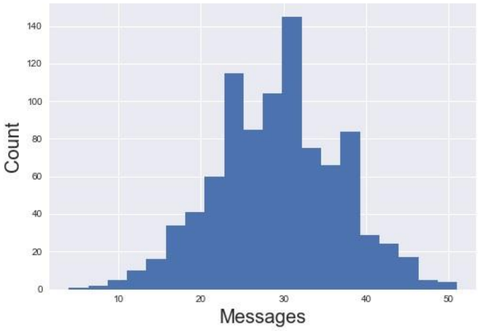


Figure 3: Number of voice mail messages distribution without zero

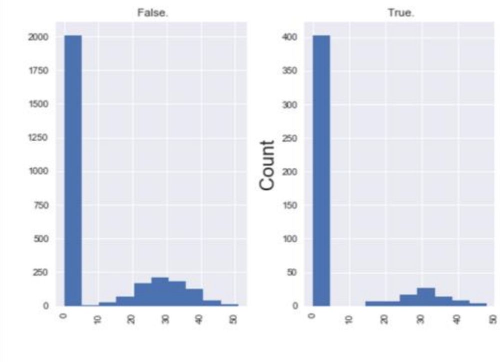


Figure 4: Number of voice mail messages distribution by Target Class

2.1.2 Feature Selection

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. There are several methods of doing that but we have used Chi Square test to asses the significance of the categorical variables. We can see from the p values of each variable that except the *area code* variable all are significant.

We also have to check whether if any of the variables are corelated as to avoid multicollinearity. We can do so by making a heatmap of the corelation matrix of the continous variables as shown in Fig.5. We can clearly see that all the *charge* and *minutes* variable pairs are strongly corelated which is obvious, so we should drop one of these from the training data.

All the other variables should have different significance as we can say from practical knowledge that the variables like *total day calls, total eve calls, total night calls* should have different importance in deciding whether the customer will keep using the service or not.

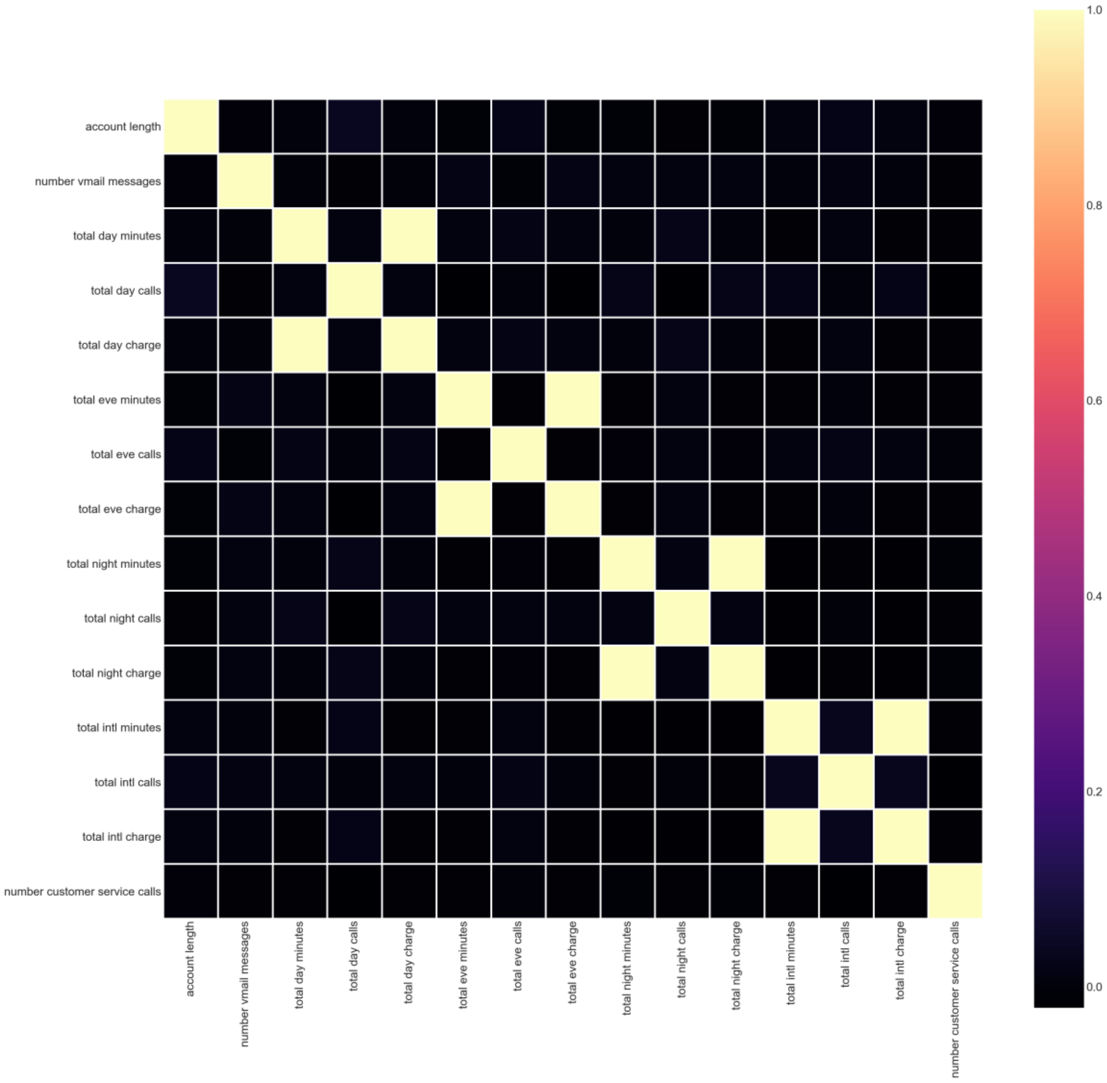


Figure 5: Correlation plot

2.2 Modelling

2.2.1 Preparing Data for Models

Now we have to edit the data so as the model could understand it. We could create dummy variables of the categorical variables but as we can observe except the *state* variable all the categorical variables are binary even the target variable so we could simply subsititue the values with *0 or 1* for *No or Yes* or *Fales or True* respectively as to avoid increasing the dimensionality of the data. Also we can see in Fig.6 that the state variable is also uniformly distributed so either we can create dummy variables for each state or we could assign a code value to each state, we have assigned code values as we have tried both ways, but models worked better for the latter.

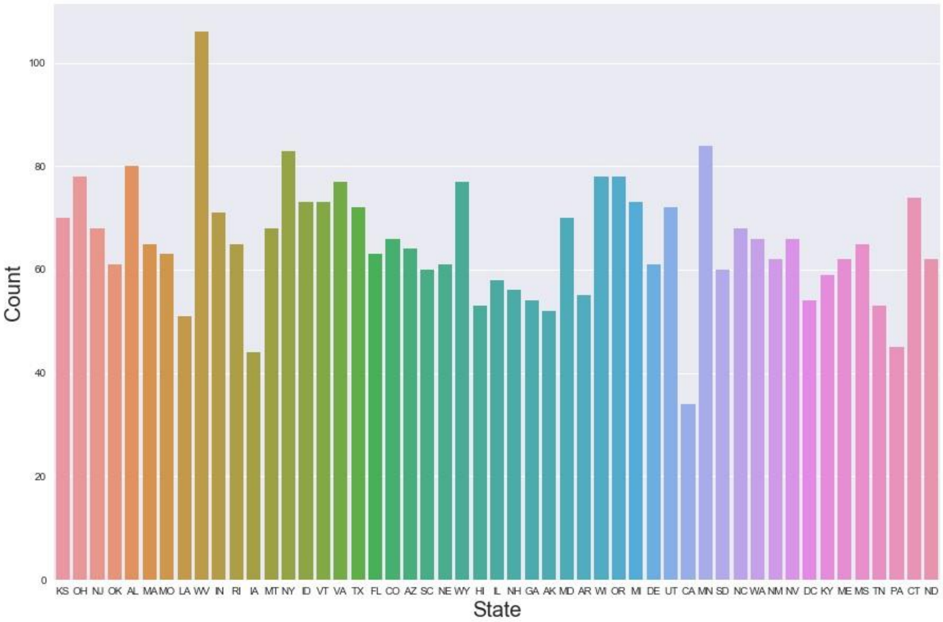


Figure 6: States distribution plot

Also we have seen in Fig.2 that there is a target class imbalance, we can deal with it by oversampling the data using Informed over sampling by *Synthetic minority over-sampling Technique (SMOTE)* to balance the classes.

2.2.2 Model Selection

The dependent variable, in our case *Churn* is binary. So the only predictive analytics we can use is **Classification**. Now we will try building various classifiers to predict our target variable and then select whichever will work best.

2.2.3 Decision Tree Classifier

Now while building a Decision Tree Classifier we can tune some parameters like *maximum depth of tree* which specifies how big tree to build. So for finding on which parameter will our model will work best we have done a grid search for parameters based on which we found of that *maximum depth* of **10** works best.

Max Depth	0
6	0.8682
8	0.8698
10	0.8722
12	0.8645
15	0.8547
18	0.8495
20	0.8435

Table 6: Test Set AUC for Decision Tree

2.2.4 Random Forest

We can also tune the parameters of the Random Forest like *number of estimators, maximum depth of each tree*. For finding the best parameters we have again used Grid search which gives us number of estimators to use as **80** and maximum depth to use as **6.**

Number of estimators\Max Depth	6	8	10	12	15	18	20
40	0.8415	0.8567	0.8534	0.8286	0.8579	0.857	0.851
60	0.852	0.8506	0.8524	0.8561	0.857	0.857	0.8544
80	0.8571	0.8627	0.8507	0.8707	0.8691	0.8647	0.8716
100	0.852	0.867	0.8507	0.8638	0.8639	0.8638	0.8535
200	0.8597	0.8498	0.8568	0.8612	0.8672	0.8647	0.8707

Table 7: Test AUC with different parameters for Random Forest

2.2.5 Logistic regression

Our main importance in the prediction is to keep the False negative rate to a minimum so we can select a custom threshold probability instead of default 0.5. We can find out what threshold to use by looking at the receiver operating characteristics **(ROC) Curve**. We would want a threshold value for which we have a high True positive rate and less false positive rate. We can see in Fig.7 that for a value of **0.4** we have a high true positive rate while a low false positive rate.

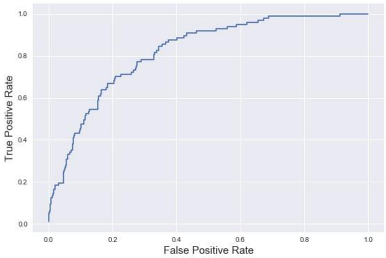
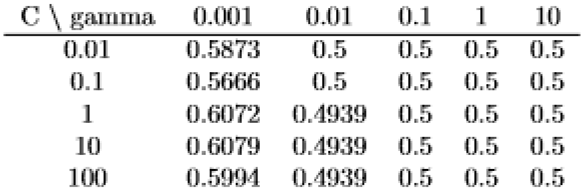


Figure 7: ROC Curve

2.2.6 Support Vector Classifier

We can tune the parameter *C and gamma* for a Support vector Classifier, where **C** is a regularization parameter, the higher it is higher the model will penalise the wrong prediction and **gamma** is a complexity parameter, which decides how complex will our decision boundary will be. Running a Grid search over **C and gamma** we get C as **10** and gamma as **0.001.**

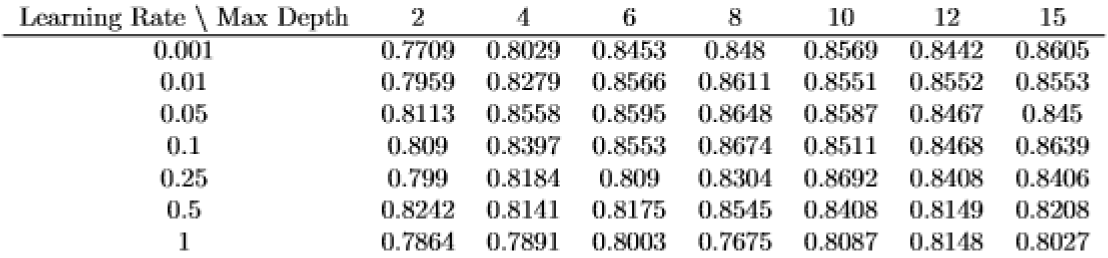


C \ gamma	0.001	0.01	0.1	1	10
0.01	0.5873	0.5	0.5	0.5	0.5
0.1	0.5666	0.5	0.5	0.5	0.5
1	0.6072	0.4939	0.5	0.5	0.5
10	0.6079	0.4939	0.5	0.5	0.5
100	0.5994	0.4939	0.5	0.5	0.5

Table 8: Test set AUC with different parameters for Support Vector Classifier

2.2.7 Gradient Boosted Classifier

Gradient Boosted classifier will build a series of trees while giving more weightage to the wrongly classified samples in the consecutive step. We can tune the *learning rate* and the *maximum depth of tree*. Running a Grid search over these parameters we get a learning rate of **0.01** and maximum depth as **6**.



Learning Rate \ Max Depth	2	4	6	8	10	12	15
0.001	0.7709	0.8029	0.8453	0.848	0.8569	0.8442	0.8605
0.01	0.7959	0.8279	0.8566	0.8611	0.8551	0.8552	0.8553
0.05	0.8113	0.8558	0.8595	0.8648	0.8587	0.8467	0.845
0.1	0.809	0.8397	0.8553	0.8674	0.8511	0.8468	0.8639
0.25	0.799	0.8184	0.809	0.8304	0.8692	0.8408	0.8406
0.5	0.8242	0.8141	0.8175	0.8545	0.8408	0.8149	0.8208
1	0.7864	0.7891	0.8003	0.7675	0.8087	0.8148	0.8027

Table 9: Test set AUC with different parameters for Gradient Boosted Classifier

3 Conclusion

3.1 Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive performance
2. Interpretability
3. Computational Efficiency

In our case, the latter two, Interpretability and Computation Efficiency, do not hold much significance. Therefore we will use Predictive performance as the criteria to compare and evaluate models.

Predictive performance can be measured by comparing Predictions of the models with real values of the target variables, and calculating some average error measure.

3.1.1 False Negative Rate (FNR)

False negative rate is the percentage of misclassified positives. It can be calculated by creating a confusion matrix.

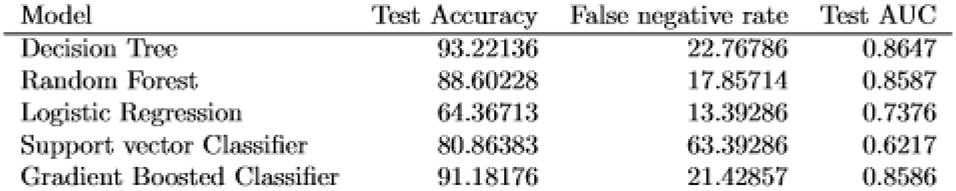
3.1.2 Area under Curve (AUC)

Area under Curve is the area under the ROC curve. It can be evaluated by building a ROC curve.

3.2 Model Selection

Now we have created many models for our data, we will select the one which will minimise the False negative rate as for our problem statement accuracy of model is not that important as to predict almost all the Churn customers. so we will select the model which gives us the best False negative rate.

Comparing the FNR and AUC of all the models we see that the Random Forest seems to be working best on our test data. So we will select the Random Forest as our predictive model. Although the FNR of Logistic Regression is less but it also have a less accuracy, if from the business aspect the high false positive rate is not an issue we can also select Logistic Regression as our model. Here we have chosen Random Forest as to be our classifier.



Model	Test Accuracy	False negative rate	Test AUC
Decision Tree	93.22136	22.76786	0.8647
Random Forest	88.60228	17.85714	0.8587
Logistic Regression	64.36713	13.39286	0.7376
Support vector Classifier	80.86383	63.39286	0.6217
Gradient Boosted Classifier	91.18176	21.42857	0.8586

Table 10: Final results

A Extra Figures

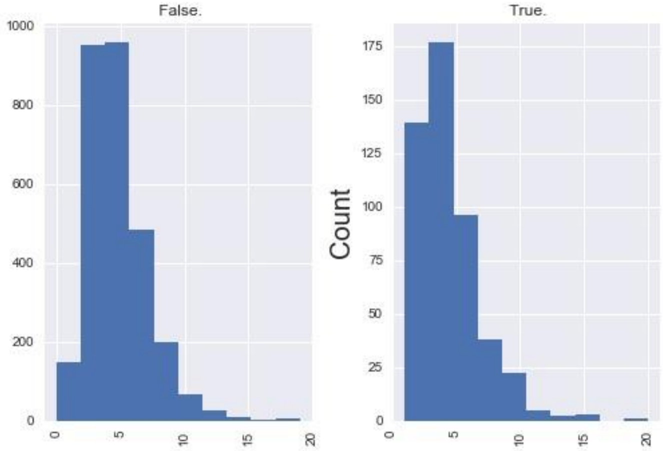


Figure 8: International Calls by Class

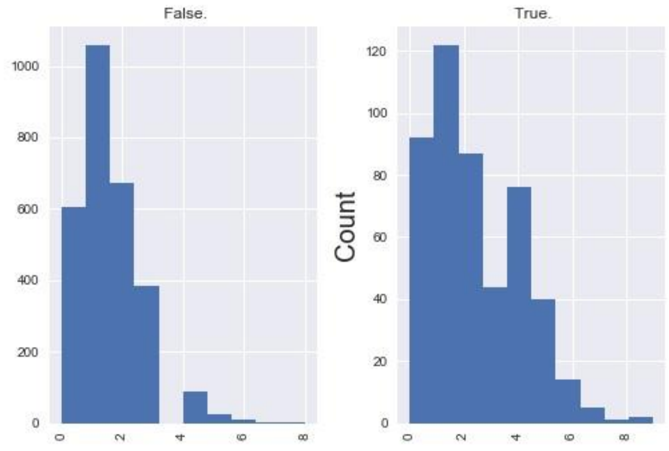


Figure 9: Customer Service Calls by Class

B Pyton Code

churn
reduction_Solution.