CUSTOMER CHURN PREDICTION

Churn Prediction project guided and inspired by Edwisor

BY
DIPU KUMAR
Aspiring Data Scientist

Objective of the Project:

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. This problem statement is targeted at enabling churn reduction using analytics concepts.

Project Outline:

This course is divided into the below sections:

1. Introduction to the problem
2. Exploratory Data Analysis (EDA) and Preprocessing
3. Model building and Feature engineering
4. coding examples

Table of Contents:

8) Model Building- Oversampling, Under sampling and both sampling together
9) Evaluation Matrix

Problem Statement:

The objective of this Case is to predict customer behavior. We are providing you a public dataset that has customer usage pattern and if the customer has moved or not. We expect you to develop an algorithm to predict the churn score based on usage pattern. The predictors provided are as follows:

- account length
- international plan
- voicemail plan
- number of voicemail messages
- 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
- Churn

It is a classification problem where we have to predict whether a customer would be Churn or not.

Hypothesis Generation:

Below are some of the factors which I think can affect the Churn (dependent variable for this problem):

1) **States**- Different states may have different of customer density and Churn also may differ
2) **Voicemail Plan**- Customer who has voice mail plan activated, I consider that customer as an old and regular customer so voice mail plan may have significant impact on churning rate
3) **Call Charge-** Today telecom market is very competitive, service providers try to keep call charges as cheaper as possible to attract customer so high charges may have significant effect on churning.

4) **International plan:** People with international call plan considered as an old and regular customer so they may have less tendency towards churning

Loading the data:

For this practice problem, we have been given two CSV files: train, test

- Train file will be used for training the model, i.e. our model will learn from this file. It contains all the independent variables and the target variable.
- Test file contains all the independent variables, but not the target variable. We will apply the model to predict the target variable for the test data.

Note: I will provide the structured code in R and Python for all these operations separately

Understanding the Data:

In this section, we will look at the structure of the train and test datasets. Firstly, we will check the features present in our data and then we will look at their data types.

Train Data columns (total 21 columns):	
state	3333 non-null object
account_len	3333 non-null int64
area_code	3333 non-null int64
phone_num	3333 non-null object
international_plan	3333 non-null object
voice_mail_plan	3333 non-null object
number_vmail_msg	3333 non-null int64
total_day_minutes	333 non-null float64
total_day_calls	3333 non-null int64
total_day_charge	3333 non-null float64
total_eve_mnts	3333 non-null float64
total_eve_calls	3333 non-null int64
total_eve_charge	3333 non-null float64
total_night_minutes	3333 non-null float64
total_night_calls	3333 non-null int64
total_nightcharge	3333 non-null float64
total_intl_minutes	3333 non-null float64
total_intl_calls	3333 non-null int64
total_intl_charge	3333 non-null float64
Num _customer_service_calls	3333 non-null int64
Churn	3333 non-null object

Dtypes: float64 (8), int64 (8), object (5)

Test Data columns (total 21 columns):	
state	1 667 non-null object
account_len	1667 non-null int64
area_code	1667 non-null int64
phone_num	1667 non-null object
international_plan	1667 non-null object
voice_mail_plan	1667 non-null object
number_vmail_msg	1667 non-null int64
total_day_minutes	1667 non-null float64
total_day_calls	1667 non-null int64
total_day_charge	1667 non-null float64
total_eve_mnts	1667 non-null float64
total_eve_calls	1667 non-null int64
total_eve_charge	1667 non-null float64
total_night_minutes	1667 non-null float64
total_night_calls	1667 non-null int64
total_nightcharge	1667 non-null float64
total_intl_minutes	1667 non-null float64
total_intl_calls	1667 non-null int64
total_intl_charge	1667 non-null float64
Num _customer_service_calls	1667 non-null int64
Churn	1667 non-null object
Dtypes: float64 (8), int64 (8), object (5)	

We can see there are three format of data types:

- **Object:** Object format means variables are categorical. Categorical variables in our dataset. ex: state, international plan, voicemail plan, account length, area code
- **Int64**: It represents the integer variables- Num _customer_service_calls , total_eve_calls, total_day_calls , number_vmail_msg , area_code, account_len
- **float64**: It represents the variable which have some decimal values involved. They are also numerical variables - total_day_minutes, total_day_charge, total_eve_mnts , total_eve_charge, total_night_minutes , total_nightcharge, total_intl_minutes, total_intl_charge

Let's look at the shape of the train dataset.

Train shape is: (3333, 21)

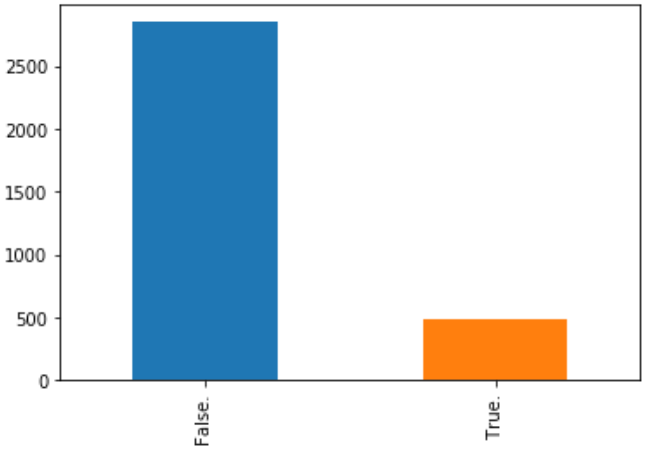
Univariate Analysis

Target Variable

We will first look at the target variable, i.e., Churn. As it is a categorical variable, let us look at its frequency table, percentage distribution and bar plot

Churn(class)	Counts	Proportion
False.	2850	85%
True.	483	14%
Name:Churn,dtype:int64		

Bar Chart of target variable Churn



Now let's visualize each variable separately. Different types of variables are Categorical, ordinal and numerical.

Categorical features: State, voicemail Plan, International Plan

A. Let's check which state having maximum number of customers

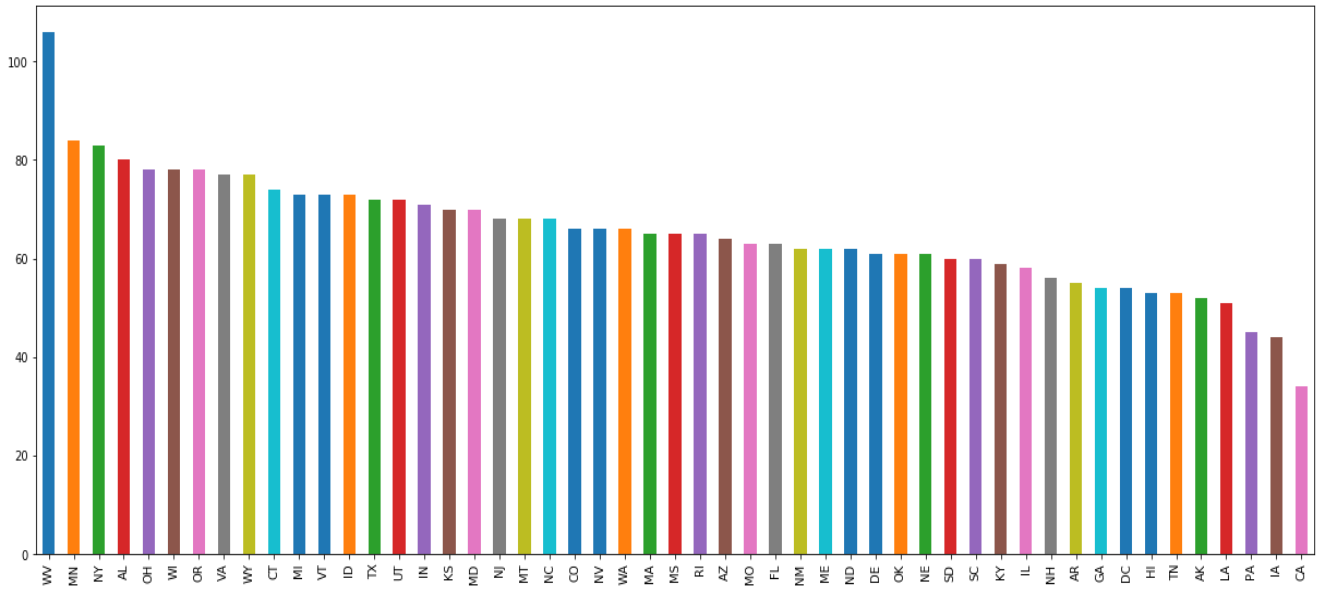


Fig: Number of customers in each state

Above bar graph shows WV (West Virginia) having maximum number of customer and CA (California) having minimum number of customer

B. Let's Check how categorical variable voice mail plan is distributed , frequency table and bar plot

Labels	Counts	Proportion
No	2411	72%
Yes	922	27%
Name: voice_mail_plan, dtype: int64		

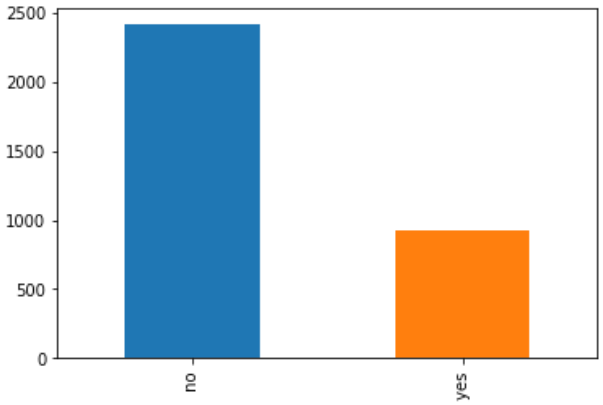


fig3: Voice_mail plan Bar Chart

C. Let's Check how categorical international plan is distributed , frequency table and bar plot

Labels	Counts	Proportion
No	3010	90%
Yes	323	9%
Name: international_plan, dtype: int64		

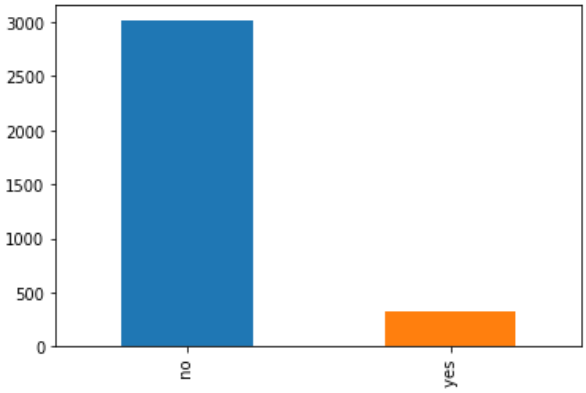
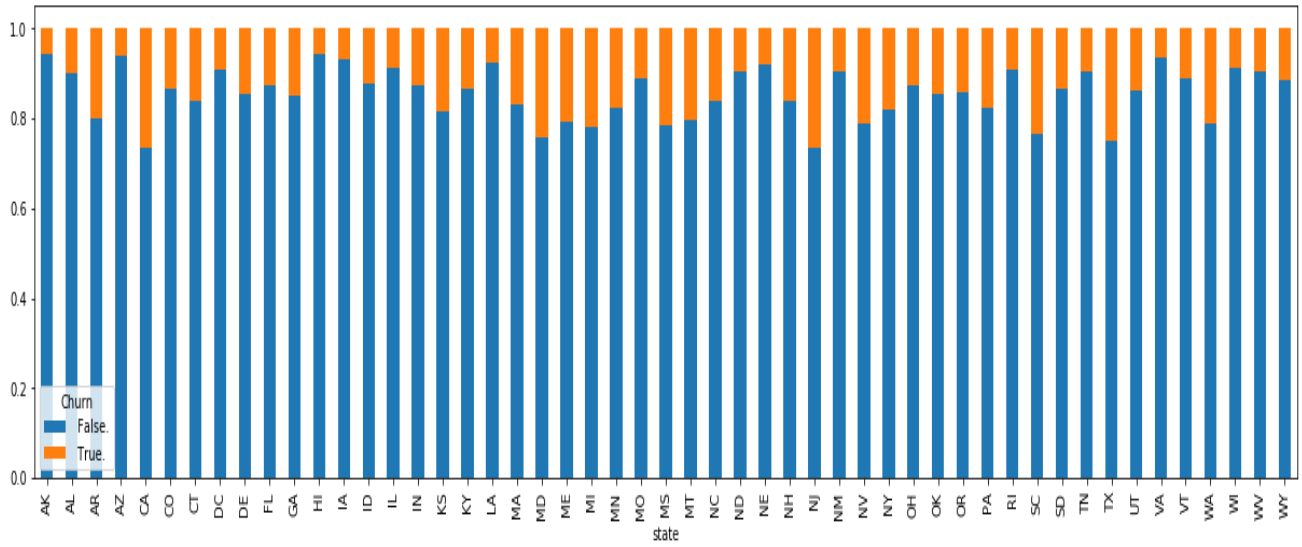


Fig4: Bar Chart for international plan

Bivariate Analysis:

Let's check which State having higher Churning rate



- Highest Churn rate states are: CA, MD, NJ, UT
- Lowest Churn rate states are: AK, AL, WV, WY

Churning in CA is concern because there is already very few users and company loosing most of them.

Now Check Churning with and without voicemail Plan:

Voice_mail_plan	Churn	Churn Number
No	0	2008
	1	403
Yes	0	838
	1	79

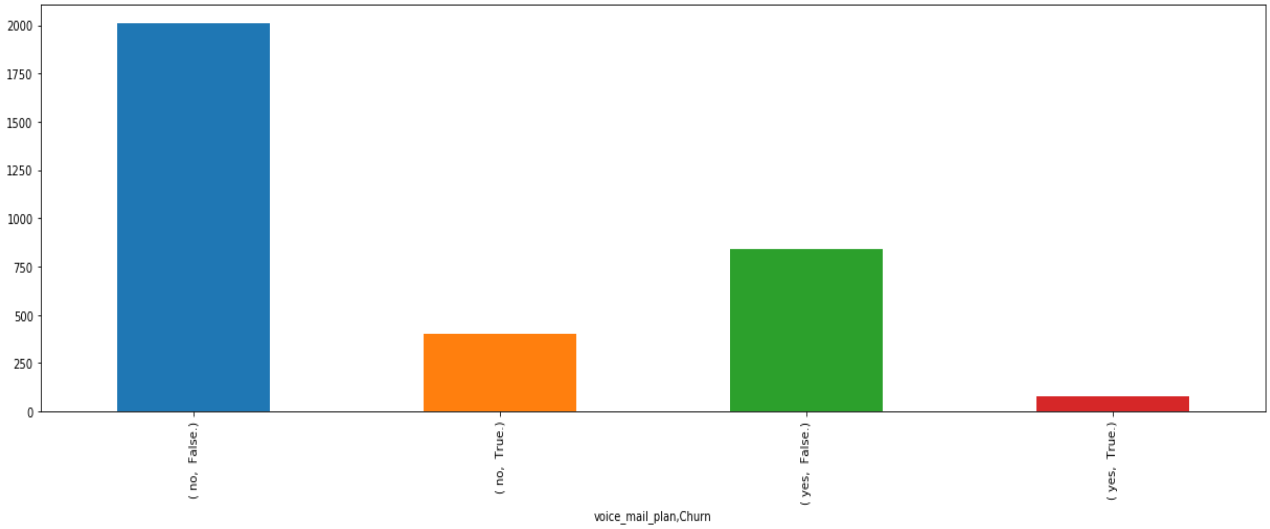


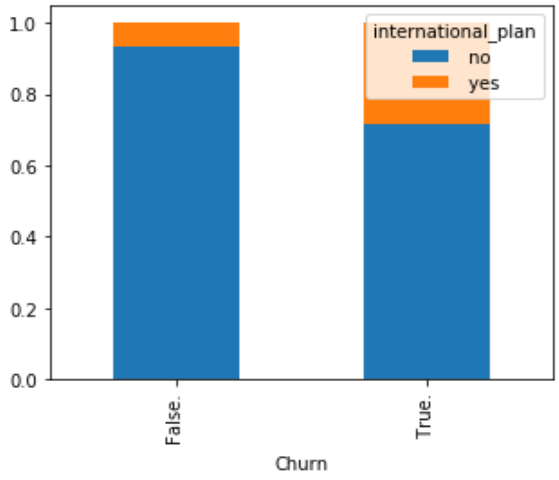
fig 5:VoiceMail Plan VS Churning

Customer with no voice mail plan active having high churning rate

Let's check effect of international plan on churning rate:

International_plan	Churn	
No	False.	2664
	True.	346
Yes	False.	186
	True.	137
Dtype: int64		

Fig6: International Plan Vs Churn

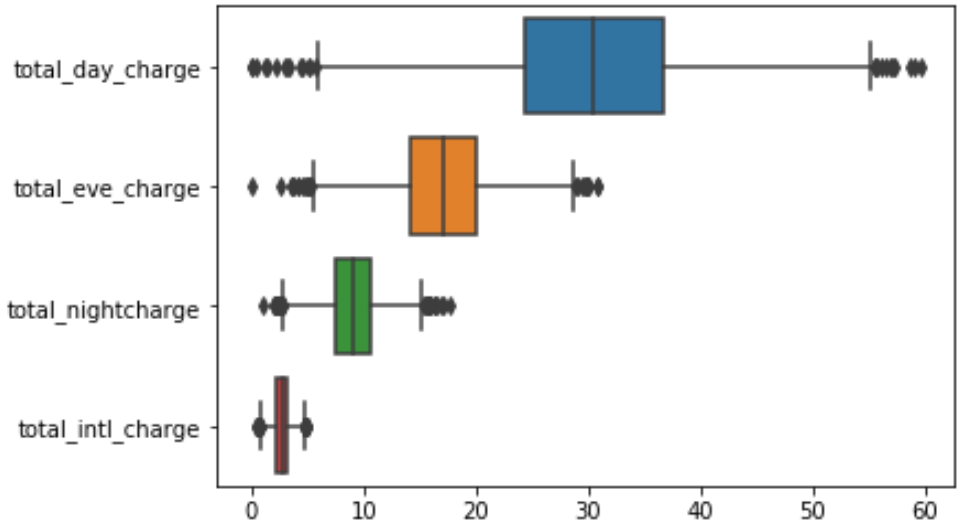


As we can see above customer with international plan having high churning rate

EFFECT OF CHARGE ON CHURNING RATE:Four different charge rate have mentioned in the dataset

- Day Charge
- Evening Charge
- Night Charge
- Intl Charge

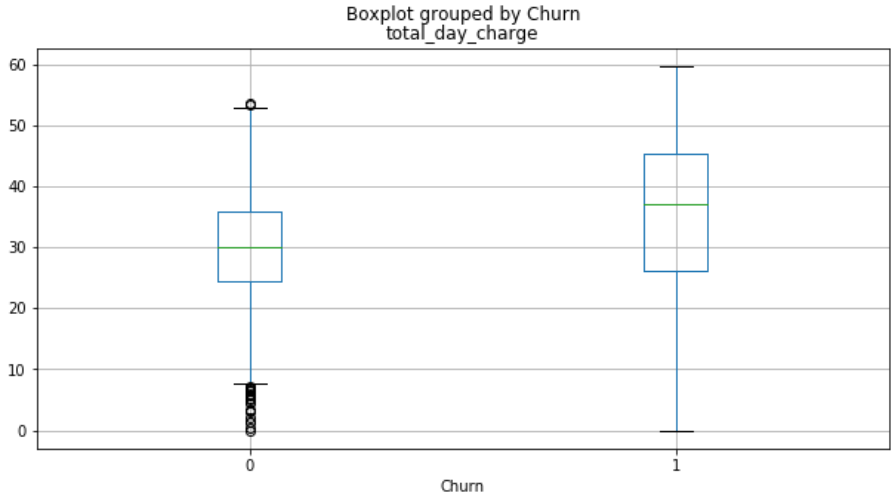
Lets Check the distribution of each separately



Box Plot: Distribution of Charges

As we can see above customer charge highly in a day time and intl charge is low as compare to other charges.

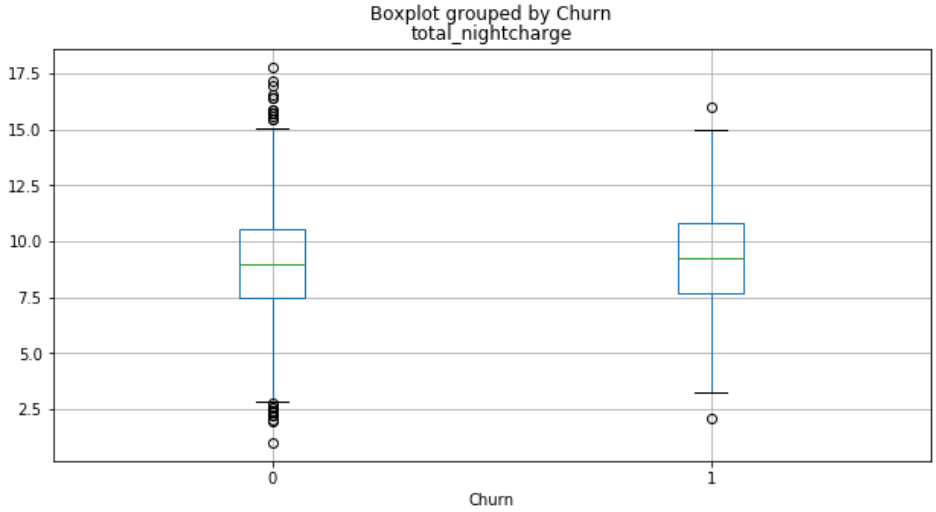
Lets sudy the effect of day charge on churning:



BoxPlot: DayCharge Vs Churning

High day charge having significant effect on churning

Night Charge Vs Churning:



BoxPlot NightCharge Vs Churning

Night charge rate not significantly affecting Churning

Number of Customer service call Vs Churning:

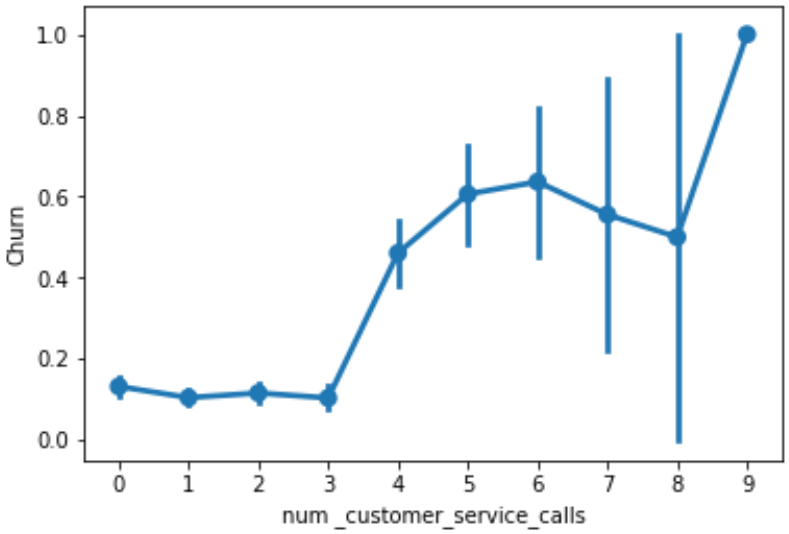


Fig:Customer service call Vs Churning

Churning rate increasing drastically with increase in number of customer service call

Outlier Treatment

After exploring all the variables in our data, we can now treat the outliers because outliers can have adverse effect on the model performance

	count	mean	std	min	25%	50%	75%	max
account_len	3333.0	101.064806	39.822106	1.00	74.00	101.00	127.00	243.00
total_day_minutes	3333.0	179.775098	54.467389	0.00	143.70	179.40	216.40	350.80
total_day_calls	3333.0	100.435644	20.069084	0.00	87.00	101.00	114.00	165.00
total_day_charge	3333.0	30.562307	9.259435	0.00	24.43	30.50	36.79	59.64
total_eve_mnts	3333.0	200.980348	50.713844	0.00	166.60	201.40	235.30	363.70
total_eve_calls	3333.0	100.114311	19.922625	0.00	87.00	100.00	114.00	170.00
total_eve_charge	3333.0	17.083540	4.310668	0.00	14.16	17.12	20.00	30.91
total_night_minutes	3333.0	200.872037	50.573847	23.20	167.00	201.20	235.30	395.00
total_night_calls	3333.0	100.107711	19.568609	33.00	87.00	100.00	113.00	175.00
total_nightcharge	3333.0	9.039325	2.275873	1.04	7.52	9.05	10.59	17.77
total_intl_minutes	3333.0	10.237294	2.791840	0.00	8.50	10.30	12.10	20.00
total_intl_calls	3333.0	4.479448	2.461214	0.00	3.00	4.00	6.00	20.00
total_intl_charge	3333.0	2.764581	0.753773	0.00	2.30	2.78	3.27	5.40

Fig: Numerical data discription

As we can see in above image no much difference between mean and median so we can consider as data is normaly distributed so we will use Z-score to calcuate outlier since if data is normally disributed 99.97% of the data contained by 3 standered deviation(std).i will keep data upto 3 std and remove data beyond that considering as an outlier.R and Python Code for the same has been shared in R file and jupyternotebook

z-score Short discription: z-score calculates how many std away the data point is from the mean

FEATURE SELECTION:

Feature selection is the crucial part of data analysis. Many features are correlated with each other, carries same information. Feeding all the features to the model not only takes unnecessary memory space but also impact the model performance. So we need to identify the highly correlated feature variables and remove which is not important from the business point of view.

There are many techniques available for analyzing correlation between the target variables like PCA, correlation calculation for numeric variables, chi-square test, ANOVA Test etc.

I have used correlation plot using heat map where dark red shows numeric features are highly positively correlated and deep blue colors shows features are highly negatively correlated.

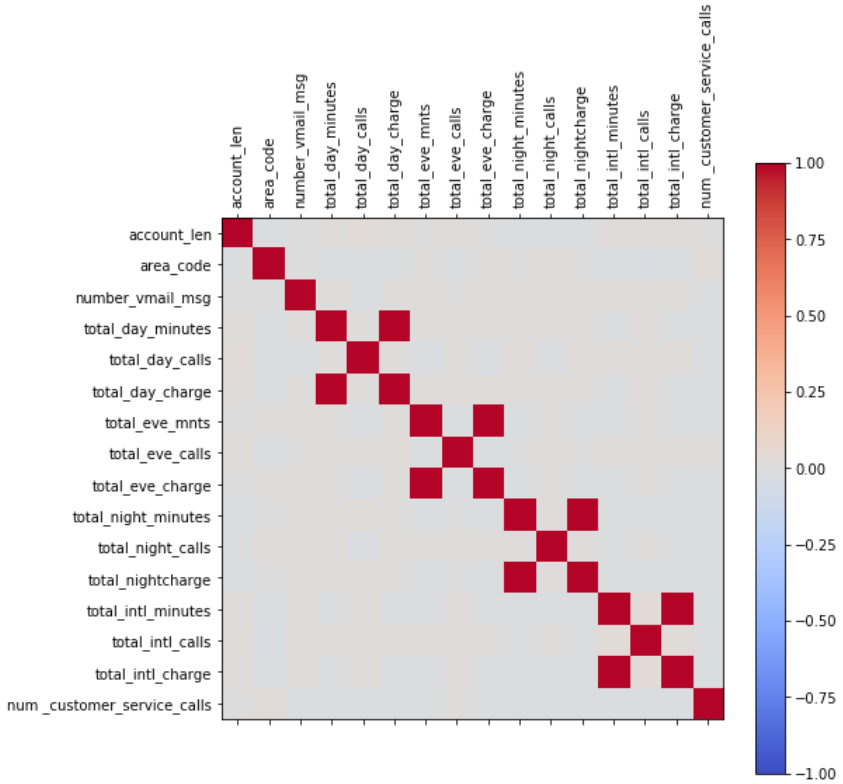


Fig: Heatmap correlation analysis

Significance analysis for categorical var:

Chi-square Test:

Correlation cacuation can only be peformed with numeric variable.So for calculating the importance of categorical variable we need to perform chi-square test which shows the significance of categorical features with respect to the target variable.

Discription: Chi-sqare can be calculted using below formula

$$\sum(obsrved - expecte)2/expected$$

After calculating the chi-square value we need to match with critical value from the critical value table for the p-value 0.05. If observed value is greater than critical we consider that there is no significance relationship of the feature variable to the target variable and we can remove those variables before modeling.

Based on that we can remove following variables before modeling: Phone number, Area-code

Data Type conversion:

Many models does not accept variables which contains string values so before modeling we need to convert string into numbers by using different coding method.(R, Python code has been shared)

	state	account_len	area_code	phone_num	international_plan	voice_mail_plan	number_vmail_msg	total_day_minutes	total_day_calls	total_day_charge
0	KS	128	415	382-4657	no	yes	25	265.1	110	45.07
1	OH	107	415	371-7191	no	yes	26	161.6	123	27.47
2	NJ	137	415	358-1921	no	no	0	243.4	114	41.38
3	OH	84	408	375-9999	yes	no	0	299.4	71	50.90
4	OK	75	415	330-6626	yes	no	0	166.7	113	28.34

fig:Before Encoding

	international_plan	voice_mail_plan	number_vmail_msg	total_day_calls	total_day_charge	total_eve_calls	total_eve_charge	total_night_calls	total_nightcharg
0	0	1	19	0.477386	1.566978	-0.056757	-0.070475	-0.463812	0.86650
1	0	1	20	1.124491	-0.335740	0.143907	-0.107579	0.149337	1.05966
2	0	0	0	0.676495	1.168056	0.495069	-1.573202	0.200432	-0.75373
3	1	0	0	-1.463926	2.197254	-0.608583	-2.741990	-0.566003	-0.07753
4	1	0	0	0.626718	-0.241685	1.097061	-1.037508	1.069059	-0.27512

Fig:After Encoding

Now after cleaning and preprocessing dataset is ready for modeling.

MODELING PART-1(WITHOUT SAMPLING TARGET VARIABLE)

I Have devided modeling part into two stage.Since data set target variable contains Churn==False very high in numbers as copared to Churn==True,Sometimes modeling algorithm completely ignore the minority class and provide very high level accuracy which is completely mislead the result.

Let us make our first model to predict the target variable. We will start with decision tree which is used for predicting the categorical class.

Some important points regarding Tree based methods

- Decision Tree is a classification algorithm. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables.
- When building a classification tree, either the Gini index or the entropy are typically used to evaluate the quality of a particular split

Train, Test Split:

I have used stratified sampling technique for train and test split which divided 80% train and 20% test.
Let's check the size of each:
X_train.shape, Y_train.shape, X_test.shape, Y_test.shape

```
(2648, 63) (2648,) (663, 63) (663,)
```

Decision Tree Model Evaluation:
We will use confusion matrix, AUC-ROC curve for evaluating the model performance
Let's look at the confusion matrix below after applying decision tree

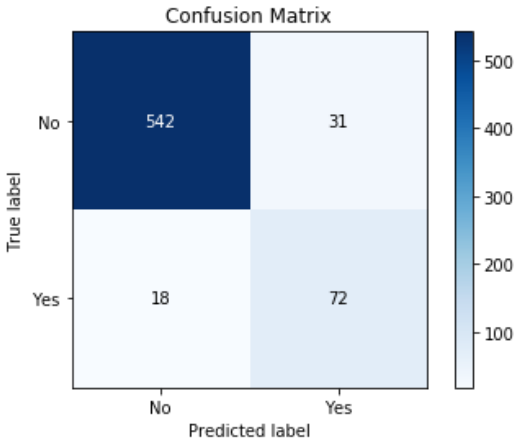


Fig: Decision tree confusion Matrix

TP=	72
TN=	542
FP=	31
FN=	18
((TP+TN)*100)/(TP+TN+FP+FN) = 92.60	

Accuracy score is 92% which is quite good

Now let's check the false negative rate:

(FN*100)/(FN+TP)= 20%

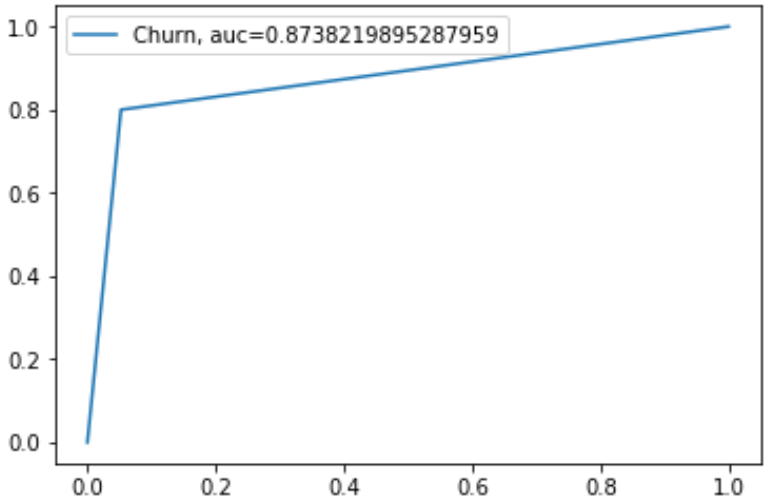


Fig:AUC Curve(Descion Tree)

Conclusion:

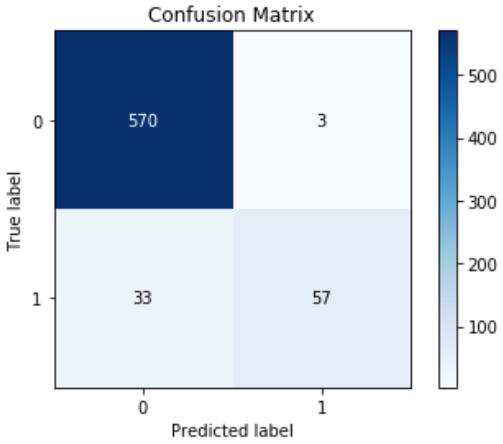
- Accuracy Score is 92% which is quite good
- False Negative rate is 20%.
- Auc Curve is 87% which is also good may be imporved by using another algorithm

Random Forest:

This Algorithm basically used when variation in data set is high

- A random sample of m predictors is chosen as split candidates from the full set of p predictors
- A Fresh sample of each predictor is taken at each split

Random Forest Model Evaluation:



TP=	**57**
TN=	**570**
FN=	**33**
FP=	**3**

((TP+TN)*100)/(TP+TN+FP+FN)	**#94.57**
#False Negative rate	
(FN*100)/(FN+TP)	**#36.66**

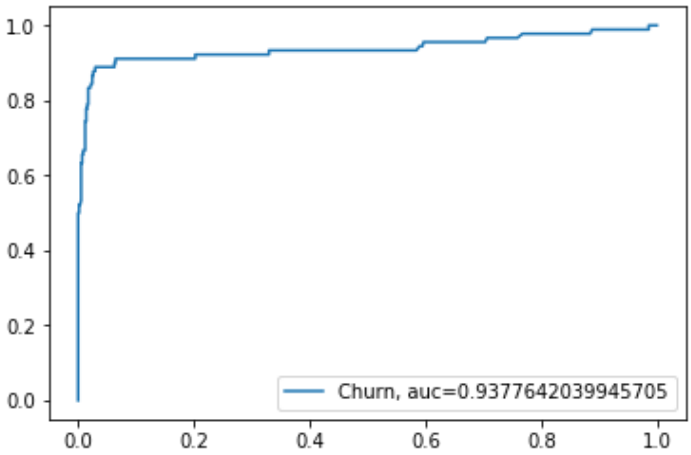


Fig: AUC Curve for Random Forest

Conclusion:

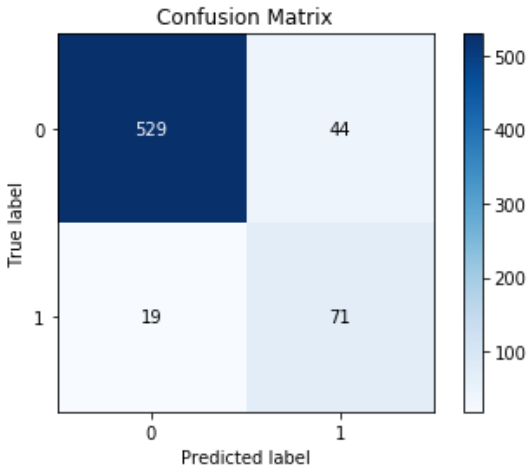
- Model accuracy is 94% which is better than decision tree
- But False negative rate is higher than DT
- AUC curve performance is far better than DT

I have tried many more algorithms but only RF and DT shows significant result so to keep presentation simple I will not mention all of Algorithms.

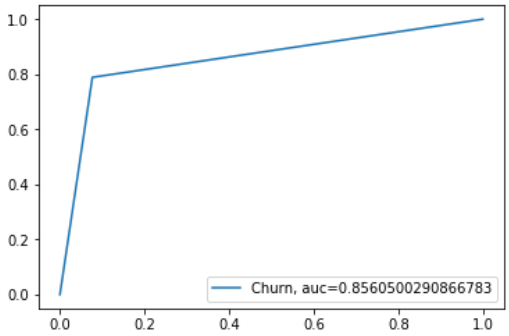
OVERSAMPLING:

Since as we have seen earlier, Target variable Churn contain majority of one class False as compare to class True let's see how model performance gets affected by sampling the classes of Churn Variable.

Decision Tree:



TP= 71
TN= 529
FN= 19
FP= 44
((TP+TN)*100)/(TP+TN+FP+FN)= 90.49
False Negative rate = (FN*100)/(FN+TP) 21.11



Conclusion:

- Model Accuracy is 90.49 %(decreased as compared to without sampling)
- False negative rate is 21.11% higher as compared to without sampling
- AUC curve = 85% also lower than without sampling
- Based on above measurement I can say DT not performing well on oversampling

Random Forest (Oversampling):

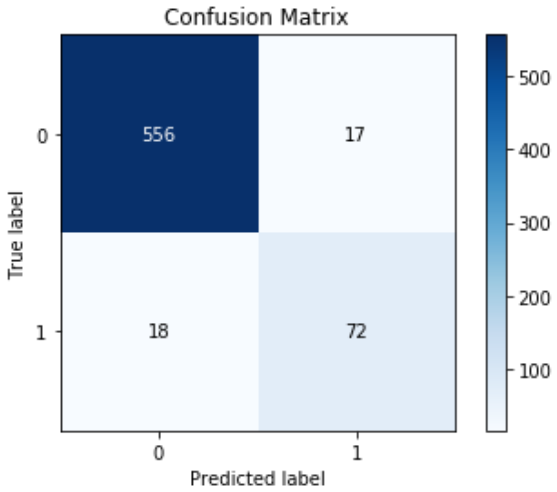


Fig: Random Forest CM

TP= 72	
TN= 556	
FP= 17	
FN= 18	
((TP+TN)*100)/ (TP+TN+FP+FN)	#94.41
False Negative rate	
(FN*100)/ (FN+TP)	#20

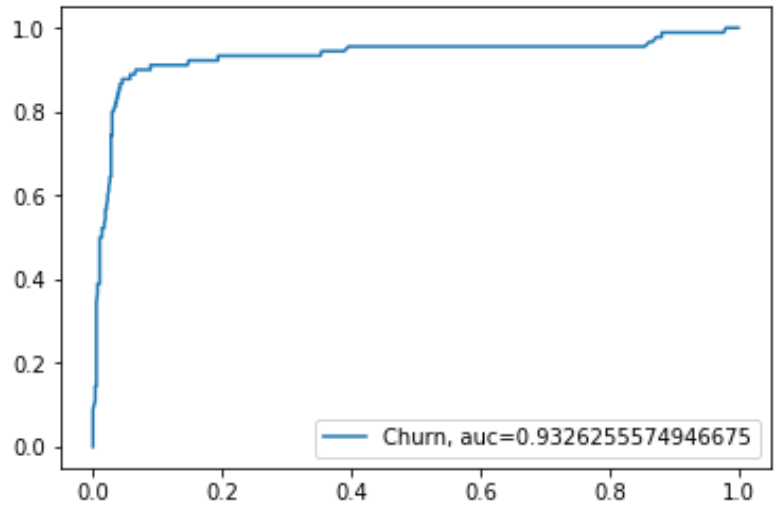
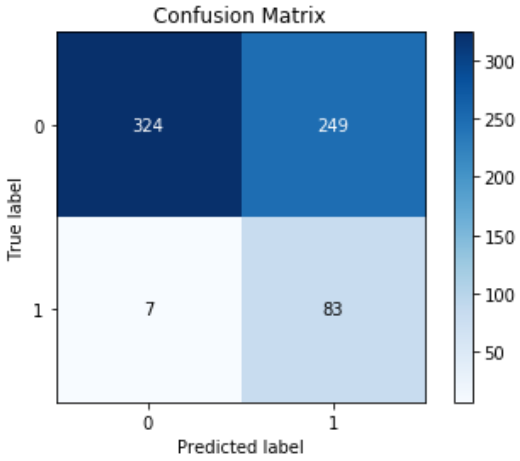


Fig: AUC curve Random Forest

Conclusion:

- Accuracy is 94.41% which is bit higher than without sampling
- False Negtive rate get reduced from 36.66% to 20% which is pretty good as compared to without sampling
- No significant effect on AUC curve

Undersampling: Now lets check performance on undersampling



TP= 83
TN= 324
FN= 7
FP= 249
Accuracy(((TP+TN)*100)/(TP+TN+FP+FN))=61.38
FNR((FN*100)/(FN+TP)) =7.77%

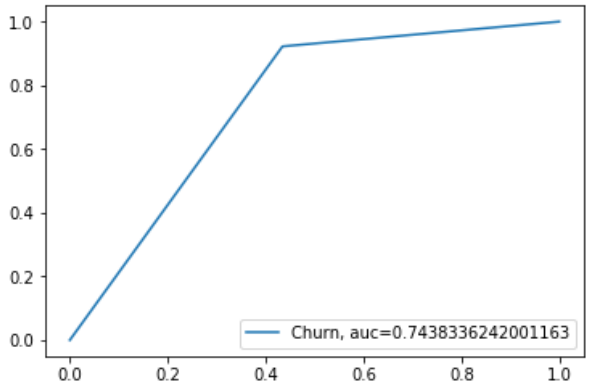
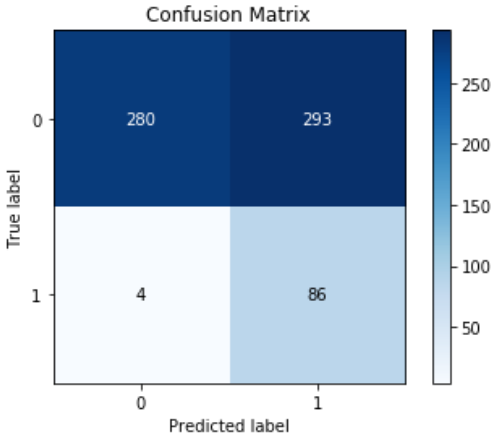


Fig: AUC curve(undersampling)

Conclusion:

- Accuracy score is very low (61%)
- FNR reduced significantly (7.7%)
- AUC performance is very low(74%)

Random Forest:



TP= 86	
TN= 280	
FP= 293	
FN= 4	
((TP+TN)*100)/(TP+TN+FP+FN)	**#55.20**
False Negative rate (FN*100)/(FN+TP)	**#4.44**

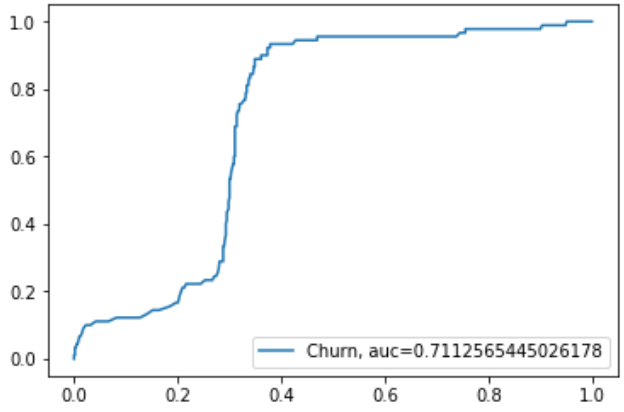


Fig: AUC curve

Conclusion:

- Accuracy score is very low (55.20%)

- FNR reduced significantly (4.44)
- AUC performance is very low(71%)

FINALE CONCLUSION:

Without sampling:			
Algorithm	Accuracy	FNR	AUC
DT	92%	20%	87%
RF	94%	36.66	93%
Oversampling:			
DT	90.4%	21.11%	93%
RF	94.41%	20%	93%
Undersampling:			
DT	61.38%	7.7%	74%
RF	55.2%	4.4%	71%

Based on the Accuracy, FNR and AUC I will go with the oversampled (highlighted above) Random Forest model

This is fairly a decent model. We are able to engage with 94% of the customers who will churn. We are missing 20% for sure. If the goal is to engage and talk to the customers to prevent them from churning, it's ok to engage with those who are mistakenly tagged as 'not churned, as it does not cause any negative problem