Churn Reduction

Gaurav Malik

Contents

Introduction

1.1 Problem Statement

1.1.1What is Churn?

Customer churn occurs when customers or subscribers stop doing business with a company or service. It is also referred as loss of clients or customers. One industry in which churn is very often is telecommunications industry. Churn happens in telecommunications industry because customers have multiple options from which to choose. In this industry churn rates can be very useful.

This report describes various models which can help us in gathering the information about a specific customer whether he/she would churn or not.

1.2 Data Interpretation

The motive is to use the data and classify whether a customer would churn using the data provided.

The data consists of

COLUMN 1-11

state	account length	area code	phone number	international plan	voice mail plan	nber vmail messa	total day minutes	total day calls	total day charge	total eve minutes
KS	128	415	382-4657	no	yes	25	265.1	110	45.07	197.4
OH	107	415	371-7191	no	yes	26	161.6	123	27.47	195.5
NJ	137	415	358-1921	no	no	0	243.4	114	41.38	121.2
OH	84	408	375-9999	yes	no	0	299.4	71	50.9	61.9
OK	75	415	330-6626	yes	no	0	166.7	113	28.34	148.3
AL	118	510	391-8027	yes	no	0	223.4	98	37.98	220.6

COLUMN 12-21

total eve calls	total eve charge	otal night minutes	total night calls	total night charge	total intl minutes	total intl calls	total intl charge	r customer servic	Churn
99	16.78	244.7	91	11.01	10	3	2.7	1	False.
103	16.62	254.4	103	11.45	13.7	3	3.7	1	False.
110	10.3	162.6	104	7.32	12.2	5	3.29	0	False.
88	5.26	196.9	89	8.86	6.6	7	1.78	2	False.
122	12.61	186.9	121	8.41	10.1	3	2.73	3	False.
101	18.75	203.9	118	9.18	6.3	6	1.7	0	False.

There are total 21 columns in the dataset consisting all the information of the customer and even whether the customer churned or not.

The description of the columns are provided below:

```
Data columns (total 21 columns):
state                           3333 non-null object
account length                  3333 non-null int64
area code                       3333 non-null int64
phone number                    3333 non-null object
international plan              3333 non-null object
voice mail plan                 3333 non-null object
number vmail messages           3333 non-null int64
total day minutes               3333 non-null float64
total day calls                 3333 non-null int64
total day charge                3333 non-null float64
total eve minutes               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 night charge              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
number customer service calls   3333 non-null int64
Churn                           3333 non-null object
```

Total 21 columns with no missing data.

Methodology

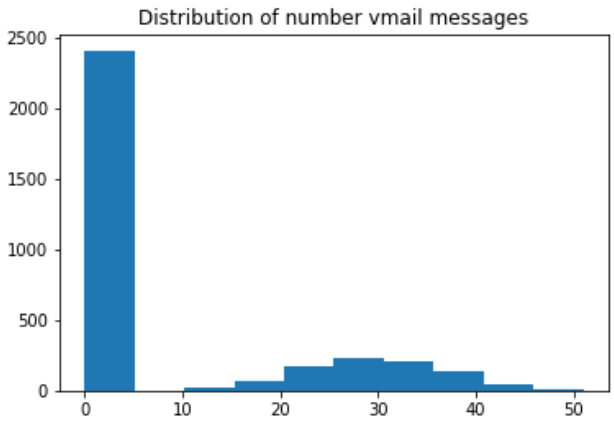
2.1 Pre Processing

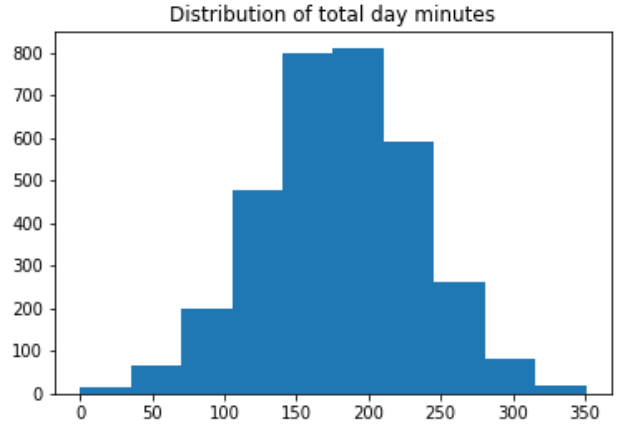
Data Pre Processing is just as important as creating a machine learning model. Before using the models of Machine Learning it is very essential that we first make the data which can be better suited to the algorithms. This process mostly includes cleaning the data, look for missing values, handling categorical data, looking for outliers. During our pre processing we also need to need to plot various bars and plots to look for outliers and the distribution of the data too.

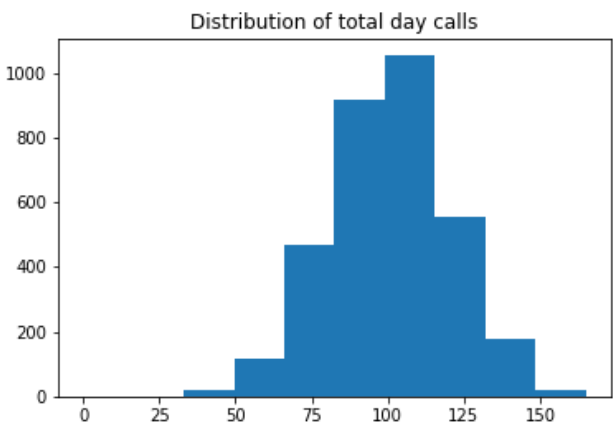
2.1.1 Distribution Histograms

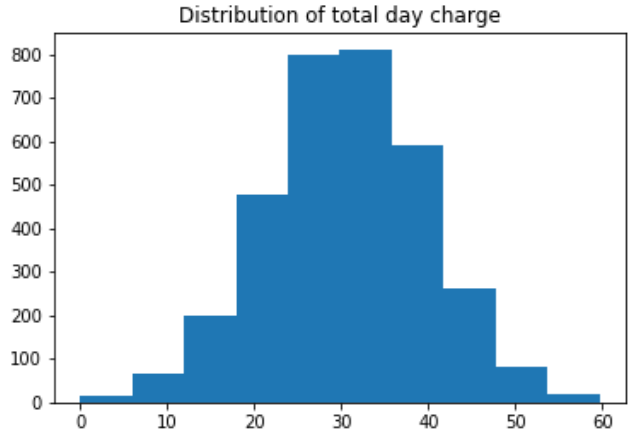
While we do our pre processing it becomes very important to observe the distribution of data. To neutralise the effect of skewed data first we need to take a quick peek into the distribution of data. The best way to do this is to plot histograms and analyse the data skewness.

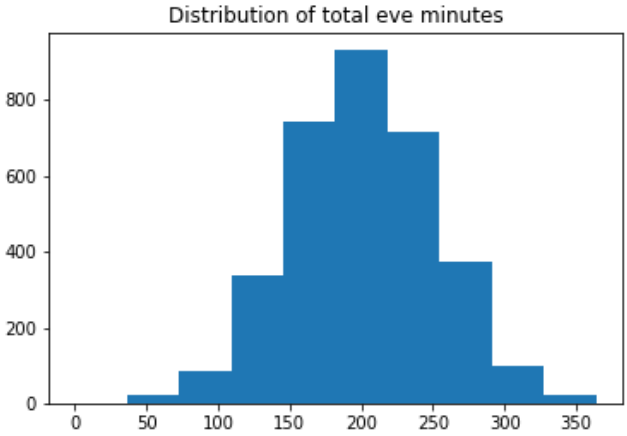
Following are the histograms of columns indicating their distribution

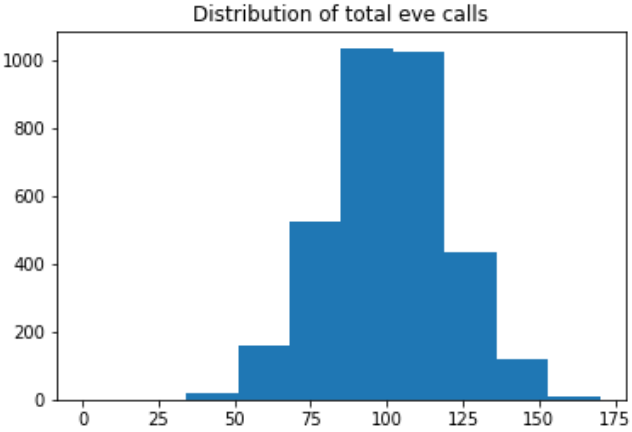


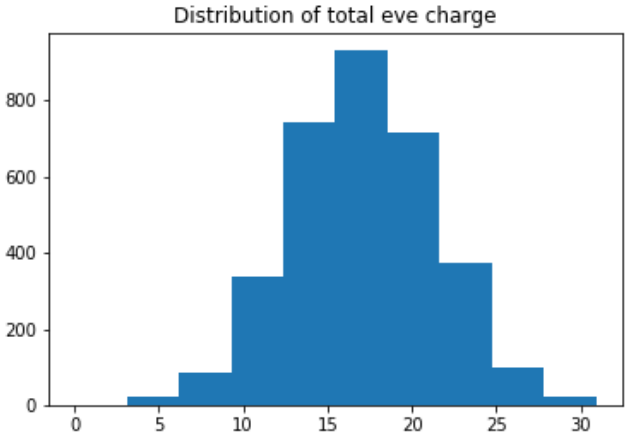


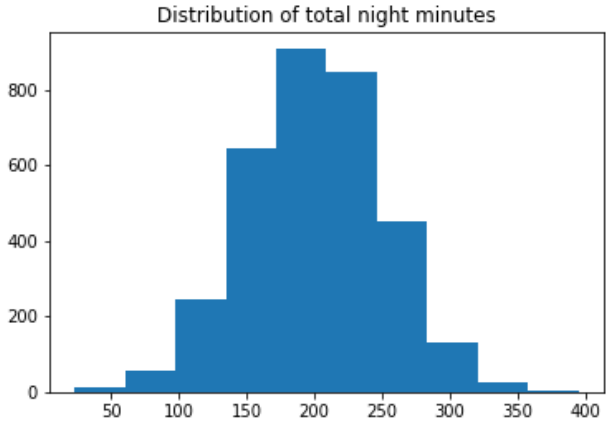

Distribution of total day calls
1000
800
600
400
200
0
0
25
50
75
100
125
150

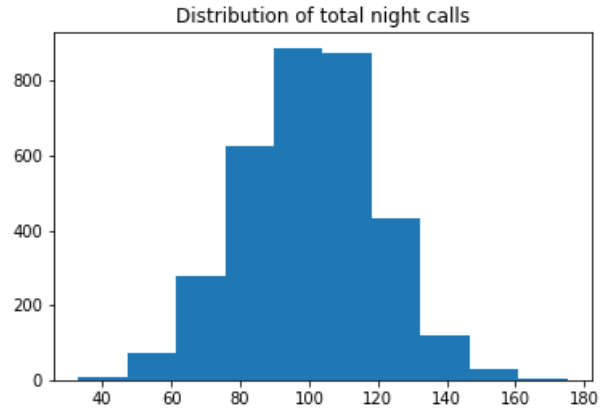

Distribution of total day charge
800
700
600
500
400
300
200
100
0
0
10
20
30
40
50
60

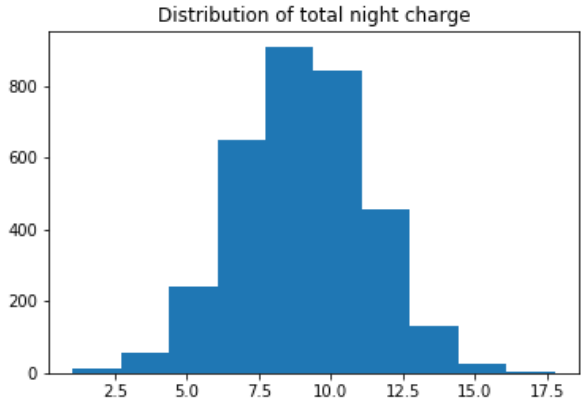

Distribution of total eve minutes
800
600
400
200
0
0
50
100
150
200
250
300
350

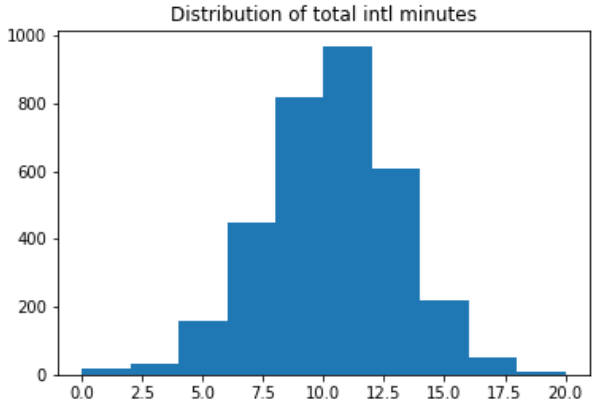

Distribution of total eve calls
1000
800
600
400
200
0
0
25
50
75
100
125
150
175

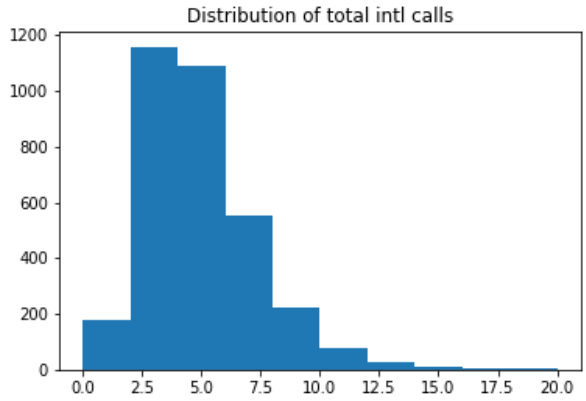

Distribution of total eve charge
800
600
400
200
0
0
5
10
15
20
25
30

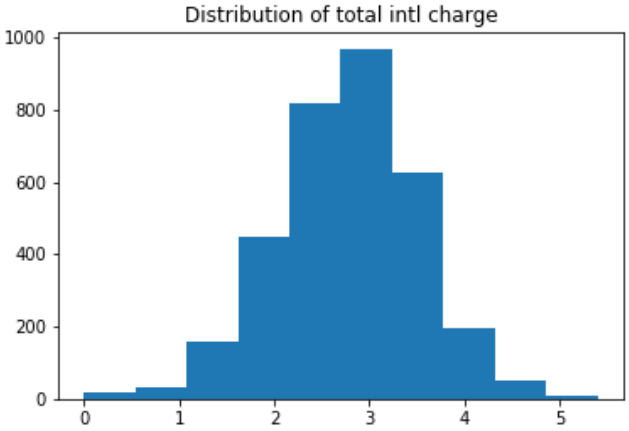

Distribution of total night minutes
800
600
400
200
0
50
100
150
200
250
300
350
400

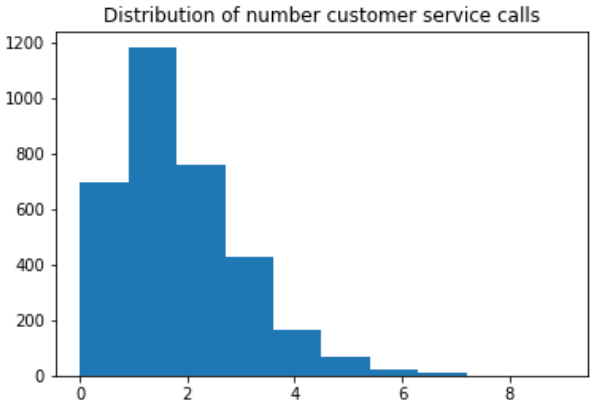












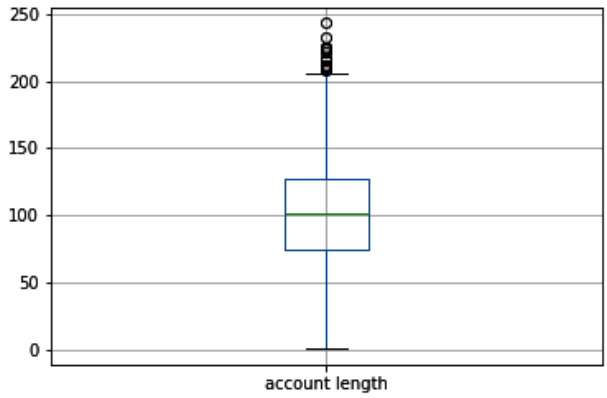
With the above data distribution plots we can say that the data is normally distributed with few columns like Number of Vmail Messages, Total international calls, Number of customers service calls which are skewed. These patterns of data in general disturb the result but in our case this is really important as customers are

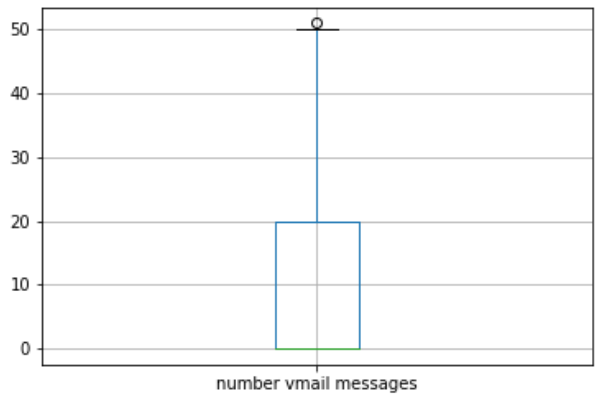
human beings who show unique behaviour, in every case a new pattern is observed by the algorithm.

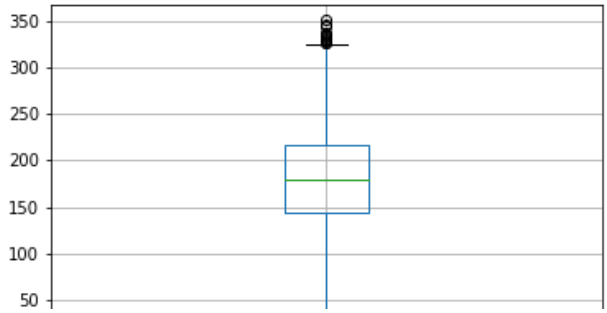
2.1.2 Outliers Analysis

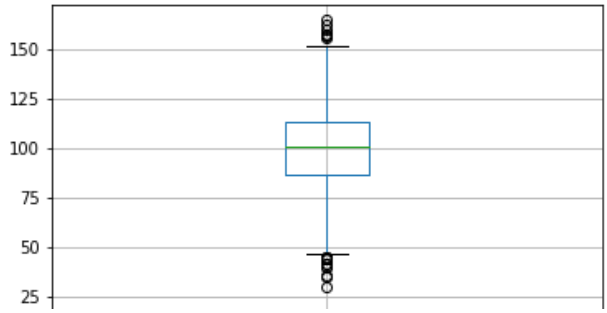
When we are cleaning our data, analysing outliers becomes among essential things to do. With the help of outliers box plot we can see whether the data which is too far from the rest of the data points will affect the prediction and remove them replacing with better values like mean of the data.

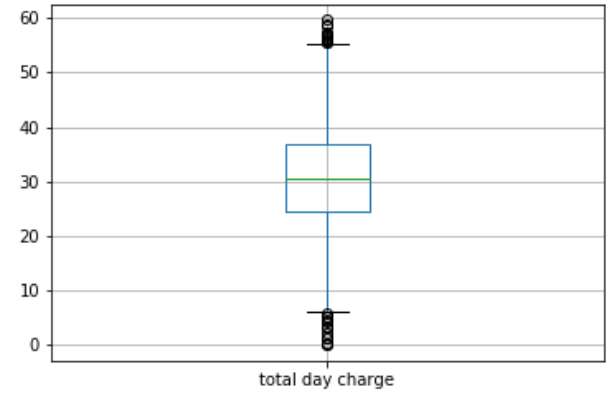
Following are the box plots for the data we have on customers:

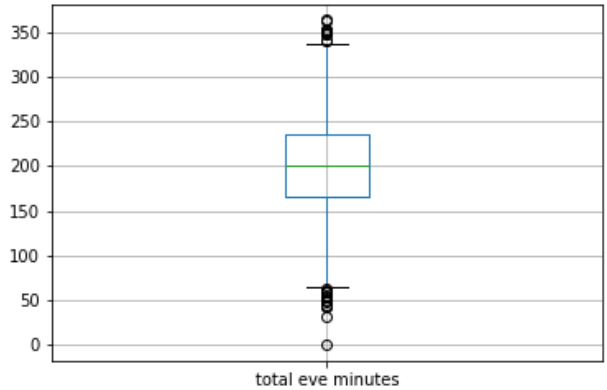


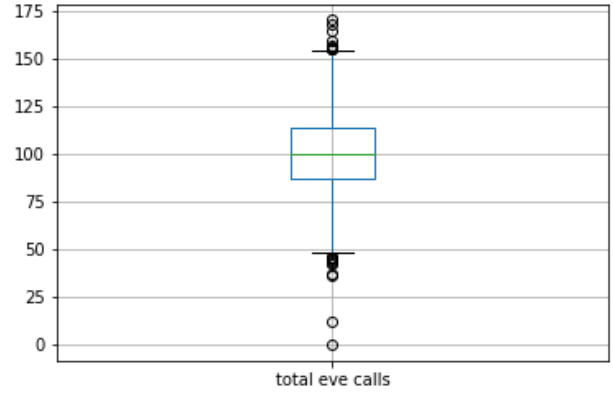


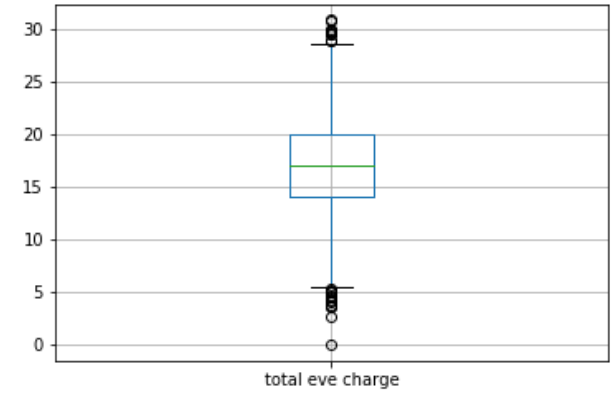


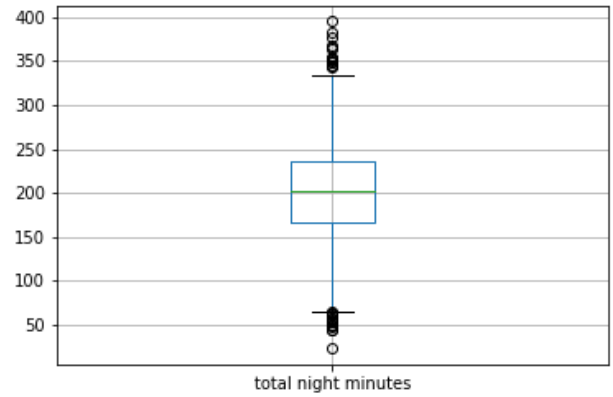


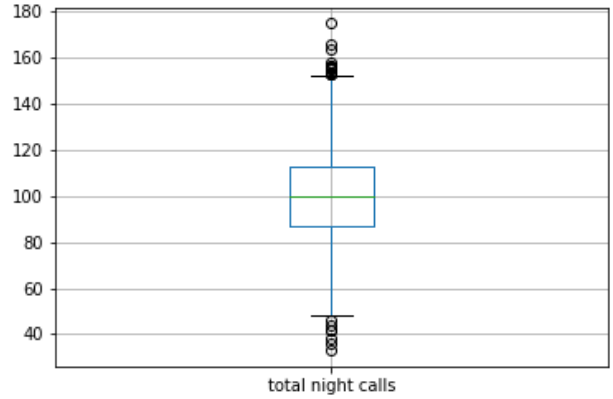


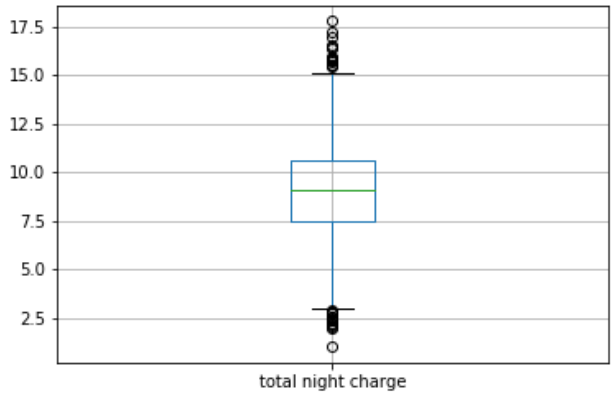


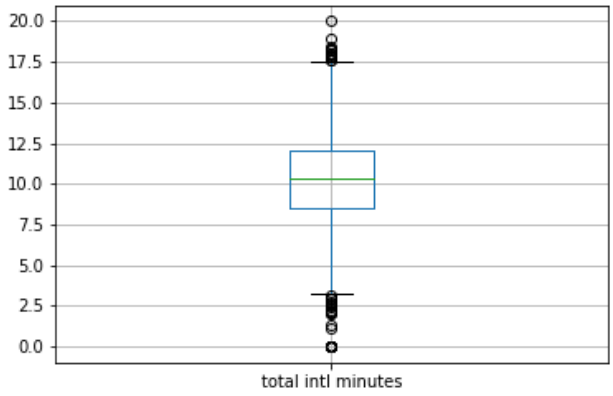

175
150
125
100
75
50
25
0
total eve calls

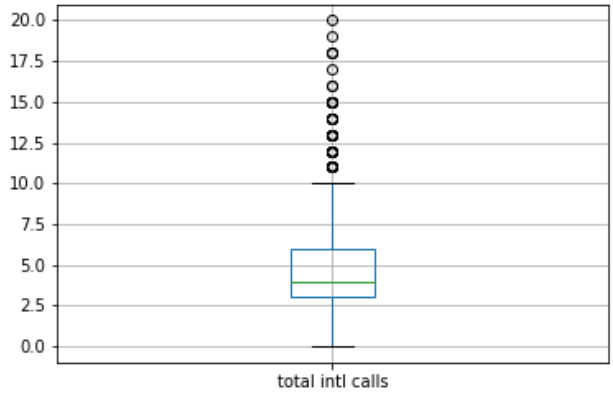

30
25
20
15
10
5
0
total eve charge

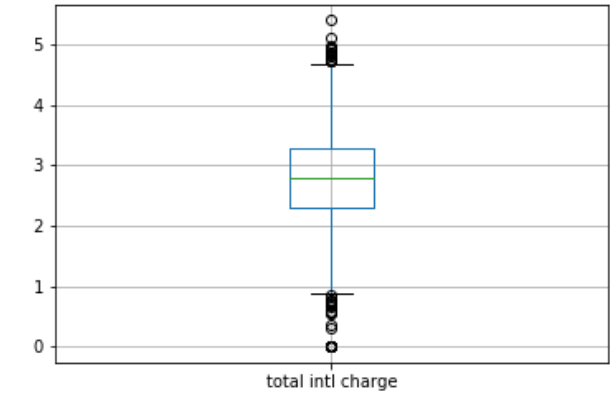

400
350
300
250
200
150
100
50
total night minutes

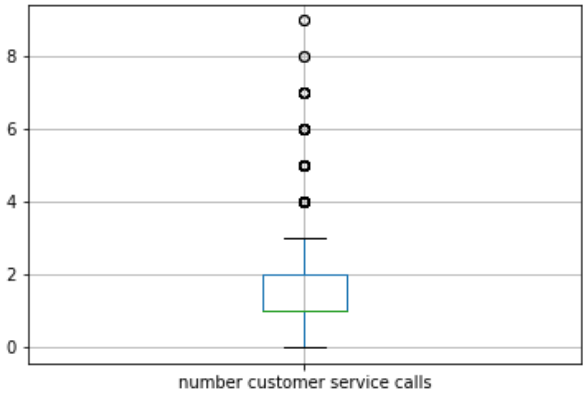

180
160
140
120
100
80
60
40
total night calls


17.5
15.0
12.5
10.0
7.5
5.0
2.5
total night charge


20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
total intl minutes


20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
total intl calls


5
4
3
2
1
0
total intl charge



Looking at the outliers, which are in every data features but as we can observe there are no outliers that are way beyond the limit except for columns like Total international calls and Number Customer Service Calls. These data points we can reject to make the distribution normalised but as we are going to classify the data in yes and no of customers leaving the company's service these all data points hold an important place in our analysis.

As we are detecting anomalies and pattern to which our customer are leaving the service of the company these outliers should also be included. Outliers in our case are those hints which if we lose our model will never be able to find those patterns due to which this outliers leave.

2.1.3 Feature Selection

We have looked upon our data, got a general hint of what it includes, how it is distributed. During data pre processing there comes a problem of whether to select all the features of the data or not. At times the features are all independent of each other but sometimes they are not. So, we have to filter out those highly correlated features.

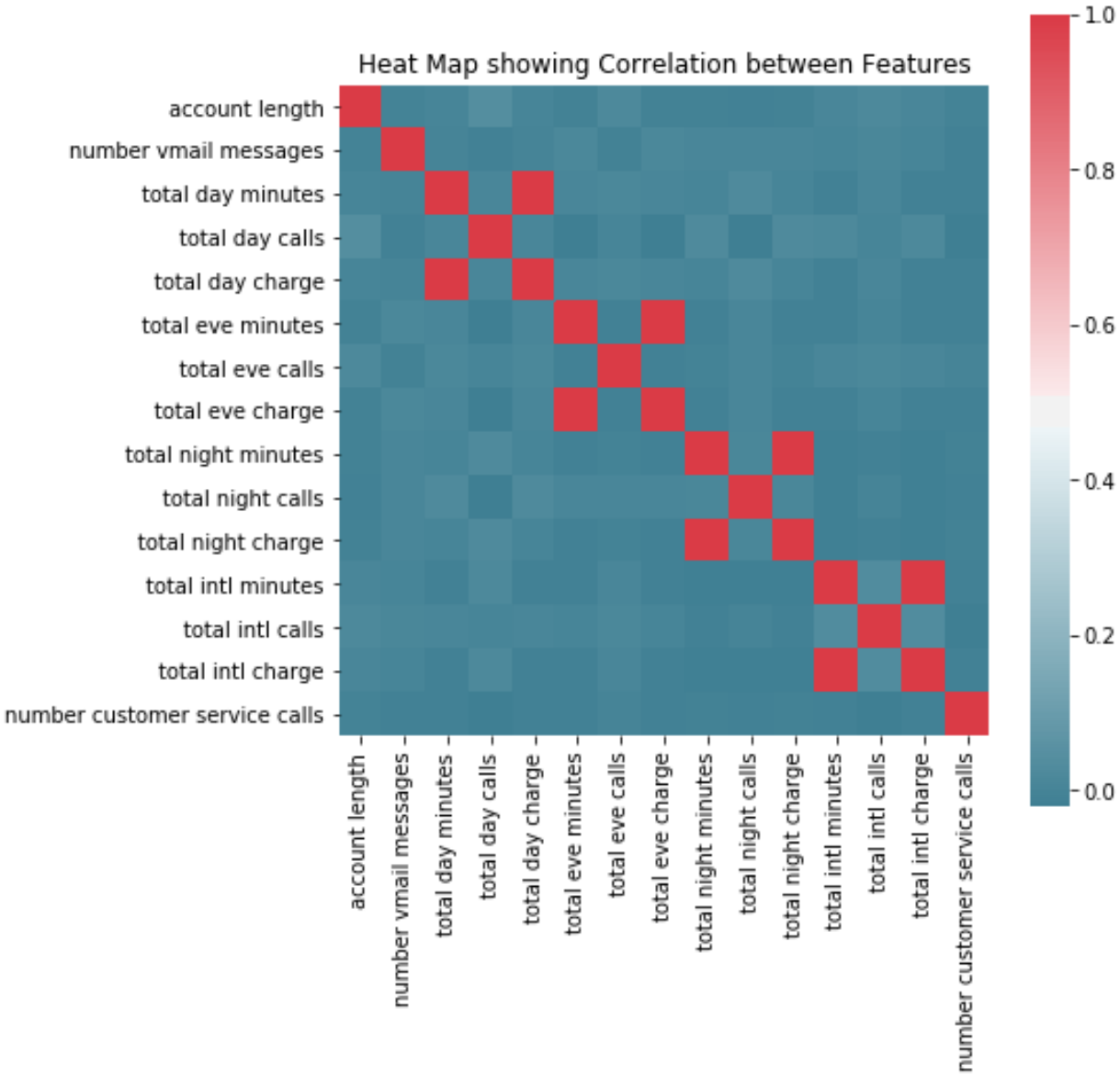
Correlated features in general do not improve the models. There are various benefits to removing the correlated features-

- Making the learning of algorithms faster
- Decreases the biasing in data
- Make the model much simpler and interpretable

For linear models like linear regression, logistic regression multicollinearity can make the prediction highly unstable. For models like Naive Bayes actually benefits from “positive” correlated data and for random forest regression the benefits are indirect as random forest is good at detecting interactions between different features.

So, removing these features can at times be necessary to speed up the learning. As the aim of the data scientist is to make the data interpretable it becomes essential to make the data simple.

Following shows the heat map of our data:



From the heat map we can observe that there are some highly correlated features which needs to be removed:

- Total day minutes & total day charge
- Total eve minutes & total eve charge
- Total night minutes & total night charge
- Total intl minutes & total intl charge

To remove the correlation of features we removed columns: total day minutes, total eve minutes, total night minutes & total intl minutes.

With this our dataset has features which are not highly dependent and this would make our model fitting to the data perfect and simple.

2.1.4 Encode Categorical Data

What is Categorical Data?

Categorical data are variables that contains labels as their value instead of having numbers.

Problem With Categorical Data

Many machine learning algorithms requires input and output variables to be numeric. They cannot work on categorical data directly, unlike decision tree which can work on categorical data without any problem.

In order to avoid these problems it is preferred to convert all categorical data to numeric form. In case the output variable is categorical variable too, we have to convert it back to categorical variable to present them or to use in applications.

Different Methods To Encode Categorical Data

We can encode categorical data using either of the two steps:

- Integer Encoding
- One-Hot Encoding

In integer encoding each unique categorical variable gets a unique number. This is also called label encoding.

For example in a data set we have three cities: New York, Mumbai, Moscow

In label encoding each of the variables get a number. Like, “New York” would be 1, “Mumbai” would be 2, and “Moscow” would be 3.

With label encoding there comes an uninvited guest called natural ordering in the data. This means that the machine automatically assumes that there is a natural order in the categories with one being higher than the other. To get rid of this problem we do One-Hot Encoding. One-Hot Encoding makes columns same as the number of categorical variables and that too of boolean type.

For example,

NEW YORK	MUMBAI	MOSCOW
1	0	0
0	1	0
0	0	1

The binary variables are often referred as dummy variables.

2.1.5 Feature Scaling

Why Scaling?

Most of the time the data in different features of dataset have high varying magnitude of data. This is a problem in most of the algorithms as they mostly use Eucledian distance to do their computation. This high magnitude features will weigh more than others features in the distance calculations with low magnitude.

To suppress this problem we need to scale the features at a same level for which we do Feature Scaling.

What Happens while Scaling?

When we use algorithms to scale, all the data is reduced to between -1 to 1 or 0 to 1 depending on algorithms and even the parameters.

When to Scale?

There are various algorithms where scaling makes a difference and some where it does not.

- k-Nearest Neighbours is very sensitive with magnitudes as they depend on Eucledian distance and scaling is needed for this algorithm.
- Models which are tree based like Decision Tree & Random Forest do not depend on distance due to which they can handle ranges of features which vary a lot.

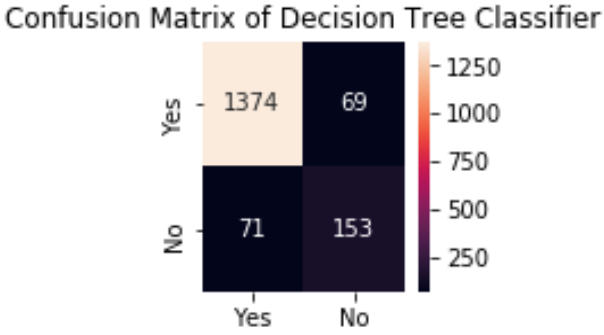
Model Selection

As we know that the motive is to find out whether the customer will churn, to determine the value in values “True” or “False” which is categorical data. From here we know that our problem is of classification type. Classification models include:

- Decision Tree
- Random Forest
- Logistic Regression
- k-Nearest Neighbour
- Naive Bayes
- SVM(linear & non-linear)

Decision Tree Classifier

This classifier organises a series of test questions and conditions in a tree structure. In the decision tree, the roots and internal nodes contains different test conditions to separate records having different characteristic. Applying decision tree classifier produces a confusion matrix shown below:



Following are the results of decision tree:

Accuracy Score of Decision Tree: 91.60167966406718 %

Specificity of Decision Tree: 68 %

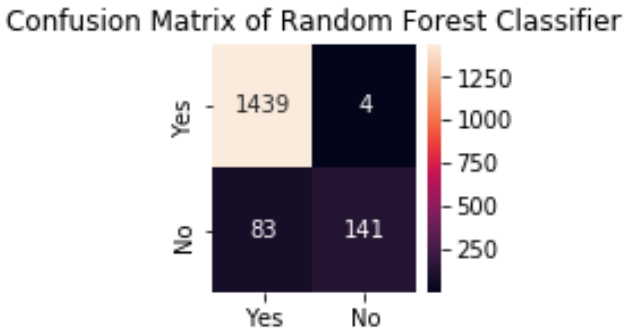
Recall of Decision Tree: 95 %

False Positive Rate of Decision Tree: 31 %

False Negative Rate of Decision Tree: 4 %

Random Forest Classifier

Random Forest algorithm creates the forest with a number of trees. In general, the more the number of trees the more robust the model. In random forest, as we increase the number of trees the accuracy also increases but too many trees can lead to overfitting. In our model, the number of trees is limited to 30. The confusion matrix produced from the random forest:



Following are the results of Random Forest:

Accuracy Score of Random Forest : 94.78104379124174 %

Specificity of Random Forest: 62 %

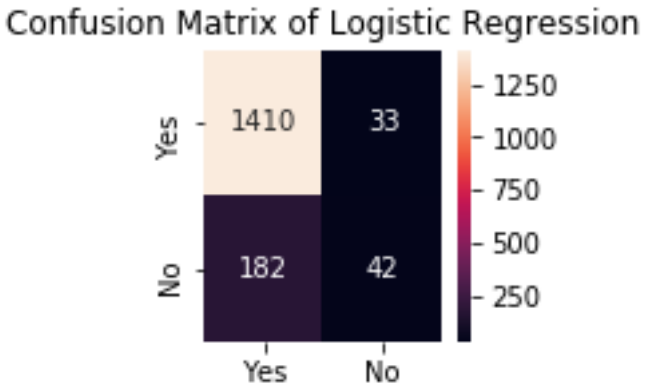
Recall of Random Forest: 99 %

False Positive Rate of Random Forest: 37 %

False Negative Rate of Random Forest: 0 %

Logistic Regression

Logistic Regression is the most famous machine learning algorithm after linear regression. They both are similar in a lot of ways but the biggest difference is how they both are used. Linear regression are used to predict continuous dependent variable whereas logistic regression serves the purpose when we need to do classification. Below is the confusion matrix produced by fitting logistic regression in the data:



Following results are produced by logistic regression:

Accuracy Score of Logistic Regression: 87.10257948410319 %

Specificity of Logistic Regression: 18 %

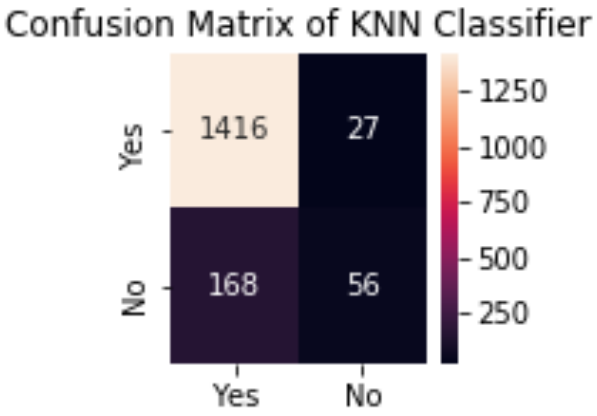
Recall of Logistic Regression: 97 %

False Positive Rate of Logistic Regression: 81 %

False Negative Rate of Logistic Regression: 2 %

k-Nearest Neighbours Classifier

kNN make its predictions using the entire dataset as its model representation. As kNN depends on Eucledian distance for its calculation(generally), the data is required to scaled at a level where there is no biasing for the magnitude of features. After fitting the kNN model in the data the following confusion matrix was produced:



Accuracy Score of KNN Classifier: 88.30233953209358 %

Specificity of KNN Classifier: 25 %

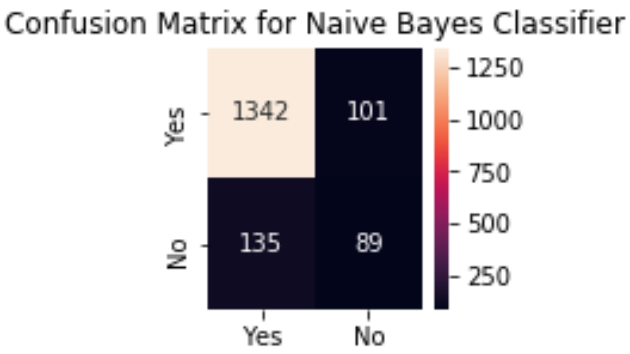
Recall of KNN Classifier: 98 %

False Positive Rate of KNN Classifier: 75 %

False Negative Rate of KNN Classifier: 1 %

Naive Bayes Classifier

Naive Bayes classifiers are a collection of classification algorithms based on Bayes Theorem. The general assumption of this algorithm is that all the features are independent and make equal contribution to the classification model. Bayes theorem finds the probability of the outcome using several algorithms. The confusion matrix for Naive Bayes fitting in the data gives us the results:



Accuracy Score of Naive Bayes Classifier: 85.84283143371326 %

Specificity of Naive Bayes Classifier: 39 %

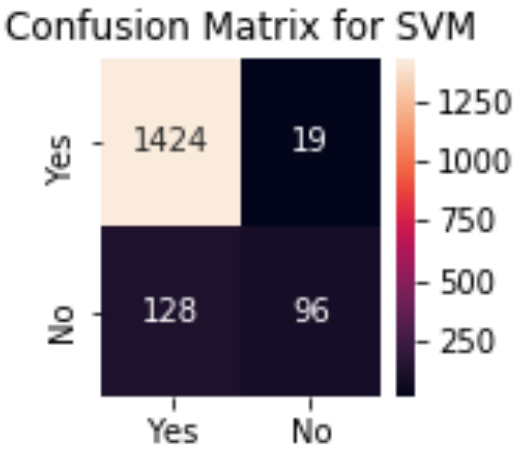
Recall of Naive Bayes Classifier: 93 %

False Positive Rate of Naive Bayes Classifier: 60 %

False Negative Rate of Naive Bayes Classifier: 6 %

Support Vector Machine

Support Vector Machine is highly preferred by many as it produces very good accuracy with minimum computational power. The goal of SVM is to build a hyperplane in an N-dimensional data to classify the data points. For our data SVM classifier produced the following results:



Accuracy Score of SVM: 91.18176364727054 %

Specificity of SVM: 42 %

Recall of SVM: 98 %

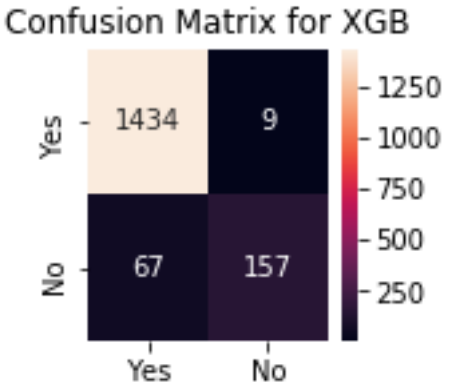
False Positive Rate of SVM: 57 %

False Negative Rate of SVM: 1 %

We have looked into many of the models which provided us with great insights on the data and we also saw their performance. Now we are going to look at one of the most popular and powerful machine learning model, XGBoost.

XGBoost

XGBoost is designed to highly efficient, flexible and portable machine learning model solving the problems of data science in a fast and accurate way. It has become on of the most popular models in machine learning which falls in the Gradient Boosting framework. The following result will put more light on the XGBoost model after it was fitted in the data:



Accuracy Score of XGB: 95.44091181763648 %

Specificity of XGB: 70 %

Recall of XGB: 99 %

False Positive Rate of XGB: 29 %

False Negative Rate of XGB: 0 %

Conclusion

To select the best model for Churn Prediction we will not only look on accuracy, but we also need to check the recall of the models and the false negative rate as this give us the insight to how much wrong was our model in predicting the customer that won't leave the company but still left and according to the observations we find that **XGBOOST** is the best model. With the accuracy of almost 95.4% (the highest) it suites the best to our requirement.

References

- Medium
- www.towardsdatascience.com
- www.stackoverflow.com
- https://stats.stackexchange.com/