**Churn Reduction**

Aditya Kanungo  
28th May 2018

**Contents**

**1 Introduction**  
 1.1 Problem Statement   
 1.2 Data

**2 Methodology**

2.1 Pre-Processing

2.1.1 Missing value-analysis  
 2.1.2 Dropping un-necessary variables (Non-predictors)

2.1.3 Replacing some variables and converting into categorical  
 2.1.4 Visualization of dataset  
 2.1.5 Chi-Square test for correlation analysis of categorical variables  
 2.1.6 Outlier Analysis  
 2.1.7 Feature Scaling

2.2 Modeling  
 2.2.1 Model Selection  
 2.2.2 Binary Classification  
 2.2.3 Logistic regression  
 2.2.4 Gaussian Naive Bayes Classifier  
 2.2.5 k-Nearest Neighbors

**3 Conclusion**

3.1 Model Evaluation

**Appendix A – Python Code  
Appendix B – R Code  
References**

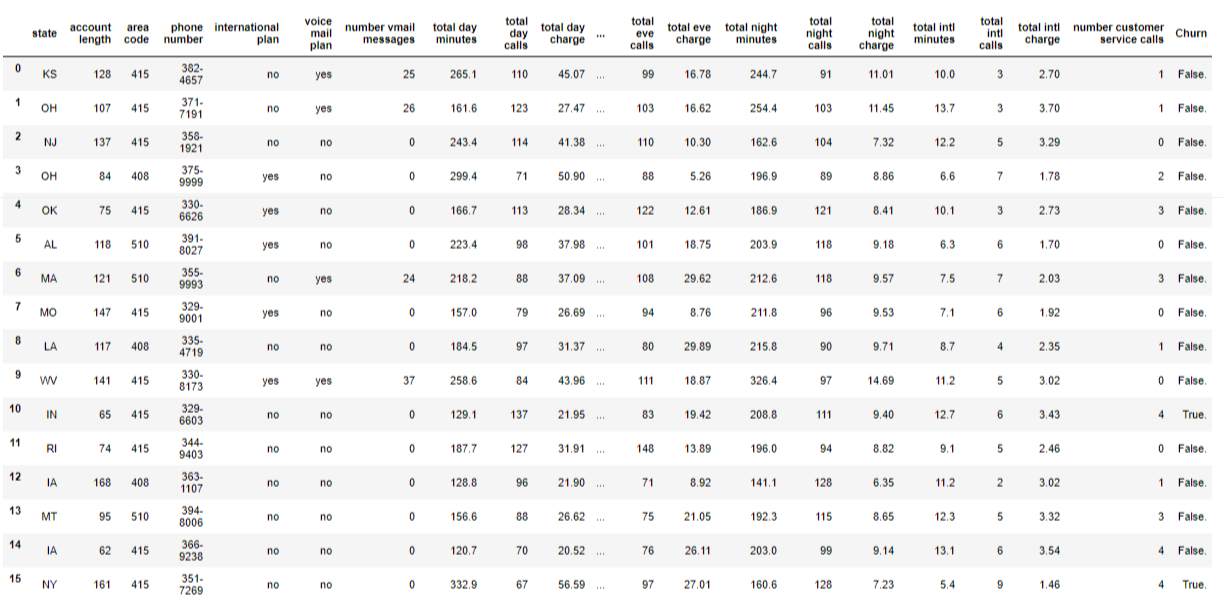
**Chapter 1**

**Introduction**

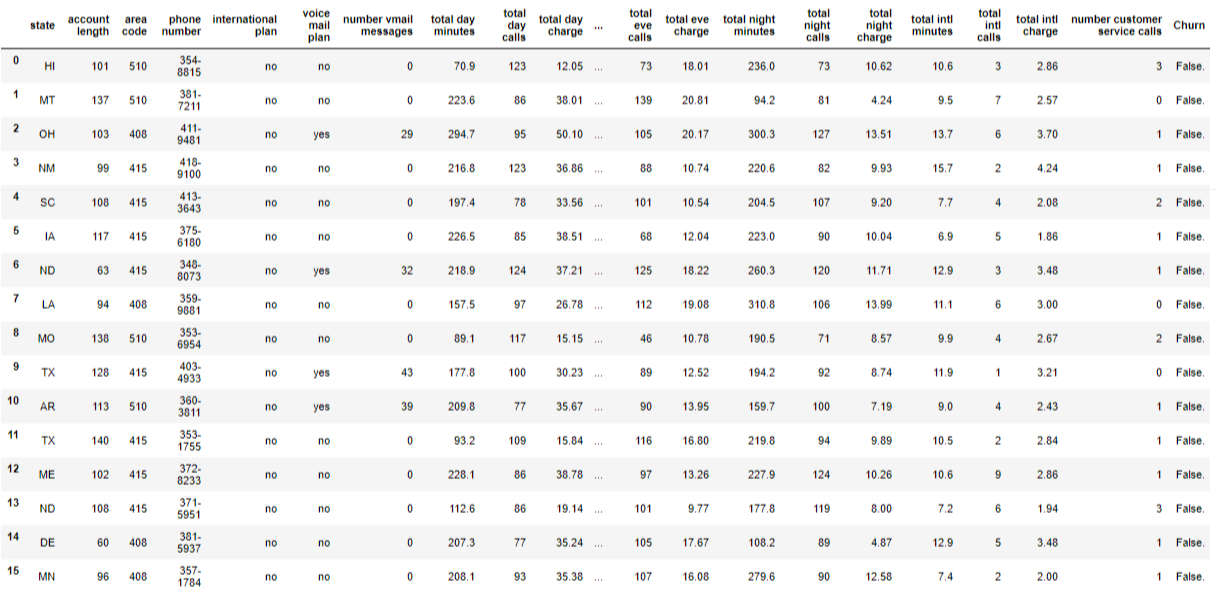
* 1. **Problem Statement**

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. This problem statement is targeted at enabling churn reduction using analytics concepts. Our objective is to predict customer behavior. We are provided with a public dataset that has customer usage pattern and if the customer has moved or not. We are expected to develop an algorithm to predict the churn score based on usage pattern.

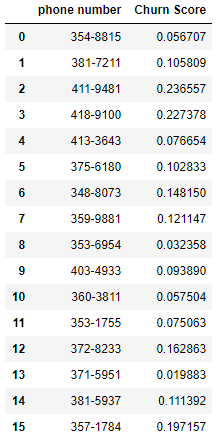
* 1. **Data**
     1. **Train Dataset:**
* (state**,** account length, area code, phone number, international plan, number vmail messages, total day minutes, total day calls, total day charge, total eve minutes, total eve calls, total eve charge, total night minutes, total night calls, total night charge, total intl minutes, total intl calls, total intl charge, number customer service calls, Churn)
* **(**3333 rows × 21 columns)



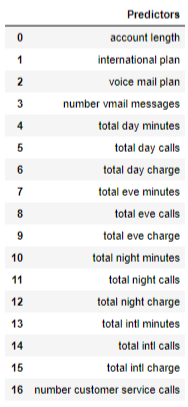
* + 1. **Test Dataset:**
* (state**,** account length, area code, phone number, international plan, number vmail messages, total day minutes, total day calls, total day charge, total eve minutes, total eve calls, total eve charge, total night minutes, total night calls, total night charge, total intl minutes, total intl calls, total intl charge, number customer service calls, Churn)
* **(**1667 rows × 21 columns)



* + 1. **Final submission:**
* (phone number, Churn Score)
* (1667 rows × 3 columns)



* + 1. **Predictors:**
* Using these we have to correctly predict the Churn Score.



**Chapter 2**

**Methodology**

**2.1 Pre-Processing**

Before implementing any Machine Learning model on our training data, we are required to look at the data before we start modeling. However, in data science terms looking at data refers to so much more than just looking. 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.

In our case the raw data that we will be working on is public dataset that has customer usage pattern and if the customer has moved or not.

**2.1.1 Missing value-analysis**

We will first check for any missing values in the comments to check if any variable is null or empty. During which we found that there are no missing values in the train and test dataset.

**2.1.2 Dropping un-necessary variables (Non-predictors)**

Dropping those variables from the train and test dataset which are not predictors and won’t be of any significance in the analysis to predict the Churn Score.

* Drop ‘state’, ‘area code’, ‘phone number’ from train and test dataset.

**2.1.3 Replacing some variables and converting into categorical**

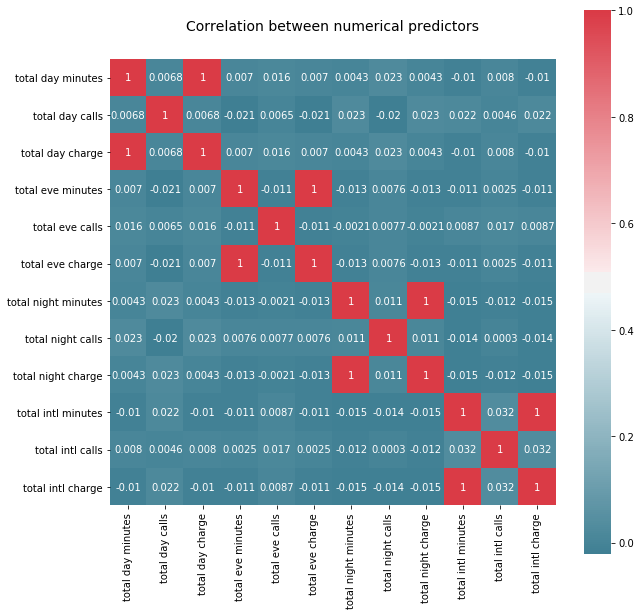
In our dataset we have variables with only 2 unique values such as true/false or yes/no, we will replace such values with 1/0 and convert those variables into categorial/factor

* Replace yes/no in international plan with 1/0
* Replace yes/no in voice mail plan with 1/0
* Replace True/False in Churn with 1/0

And the convert the datatype of these 2 variables into category or factor.

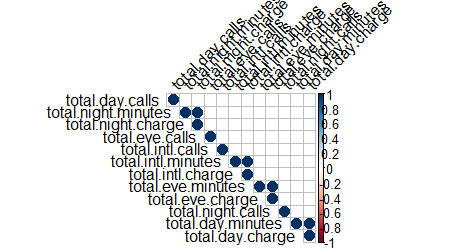
**2.1.4 Visualization of dataset**

2.1.3.3 Correlation analysis using heatmap



**From the above heat-map we made in Python we can infer the following:**

- total day minutes & total day charge are highly positively correlated.  
- total eve minutes & total eve charge are highly positively correlated.  
- total night minutes & total night charge are highly positively correlated.  
- total intl minutes & total intl charge are highly positively correlated.



**From the above Correlation plot we made in R we can infer the following:**

- total day minutes & total day charge are highly positively correlated.  
- total eve minutes & total eve charge are highly positively correlated.  
- total night minutes & total night charge are highly positively correlated.  
- total intl minutes & total intl charge are highly positively correlated.

Therefore, we will drop the total day charge, total eve charge, total night charge i.e. highly positively correlated variables carrying redundant information.

* Drop ‘total day charges’, ‘total eve charge’, ‘total night charge’, ‘total intl charge’ from train and test dataset.

**2.1.2 Chi-Square test for correlation analysis of categorical variables**

Using Chi-square contingency test on international plan and voice mail plan we got p-values of both the categorical variables < 0.05 therefor we will reject the null hypothesis and will consider both variables significant for further analysis.

**2.1.4 Outlier Analysis**

In [statistics](https://en.wikipedia.org/wiki/Statistics), an outlier is an observation point that is distant from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the [data set](https://en.wikipedia.org/wiki/Data_set). An outlier can cause serious problems in statistical analyses.

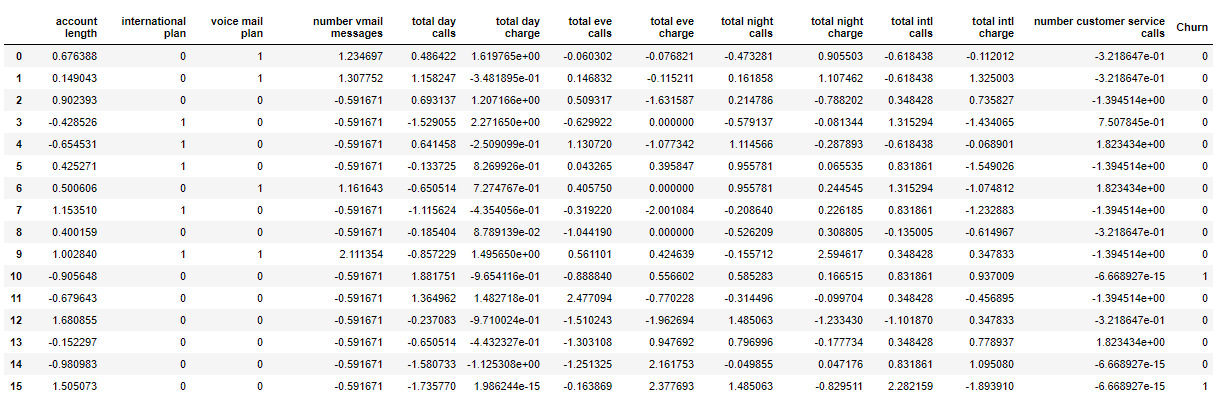
Initial Box-plot analysis confirms significant outliers in the numeric data. So, we will use boxplot method and inter quartile range (IQR) to find outliers in our dataset and replace it with null value or NA. Then we can impute the NA or the replaced outlier values using mean, median or knn-imputation method.

**2.1.4 Feature Scaling**

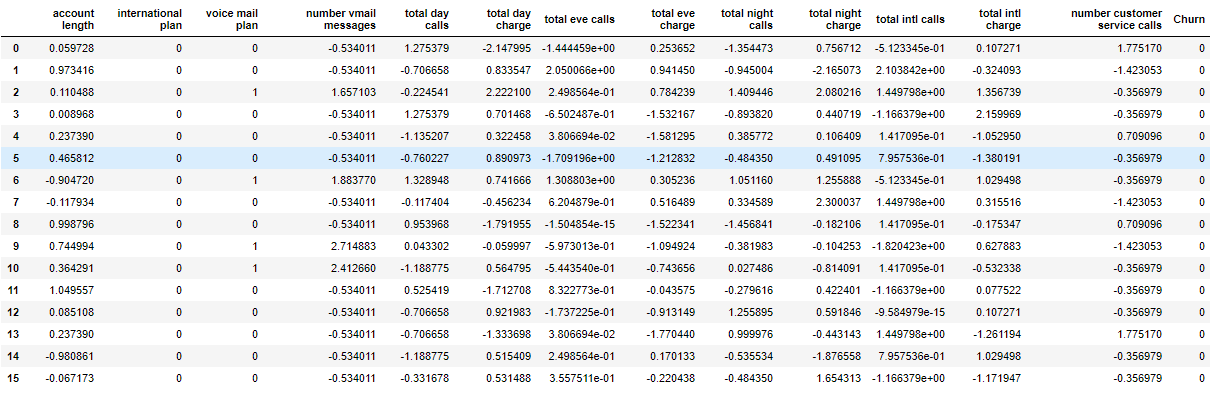
Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step. Since the range of values of raw data varies widely, in some [machine learning](https://en.wikipedia.org/wiki/Machine_learning) algorithms, objective functions will not work properly without [normalization](https://en.wikipedia.org/wiki/Normalization_(statistics)). Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance.

In order to feed the data to our model we need to scale our data so that we can compare and analyze different variables on the same ground. We will use standardization method to scale our numeric data for further analysis.

2.1.4.1 Train dataset after data preprocessing



2.1.4.2 Test dataset after data preprocessing



**2.3 Modeling**

**2.3.1 Model Selection**

In early stage of our analysis process we have come to understand that Our objective is to predict customer behavior, which we can predict in terms of Churn Score i.e. probability of a customer to leave the company or Churn. This problem statement is targeted at enabling churn reduction using analytics concepts. We are provided with a public dataset that has customer usage pattern and if the customer has moved or not. We are expected to develop an algorithm to predict the churn score based on usage pattern

Therefore, we will apply different models such as logistic regression, Gaussian naïve Bayes, and knn on our dataset and choose the model with the best accuracy, but our final model will be either naïve bayes or logistic regression as we want probability as our output as Churn Score.

Here we need to predict the Churn Score which is probability of a customer to leave the company, for that we will finally use the model whose output will be in terms of probability.

After that we will implement our trained model on the test dataset and predict the Churn Score.

**2.3.2 Binary Classification**

Binary or binomial classification is the task of classifying the elements of a given set into two groups (predicting which group each one belongs to) on the basis of a classification rule.

The actual output of many binary classification algorithms is a prediction score. The score indicates the system’s certainty that the given observation belongs to the positive class. To make the decision about whether the observation should be classified as positive or negative, as a consumer of this score, you will interpret the score by picking a classification threshold (cut-off) and compare the score against it. Any observations with scores higher than the threshold are then predicted as the positive class and scores lower than the threshold are predicted as the negative class.

The predictions now fall into four groups based on the actual known answer and the predicted answer: correct positive predictions (true positives), correct negative predictions (true negatives), incorrect positive predictions (false positives) and incorrect negative predictions (false negatives). Which we can see in our Confusion Matrix.

**2.3.2 Logistic regression**

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).

Rather than choosing parameters that minimize the sum of squared errors (like in ordinary regression), estimation in logistic regression chooses parameters that maximize the likelihood of observing the sample values.

**2.3.2 Gaussian Naive Bayes Classifier**

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values.

Naïve Bayes classifier is one of the most effective machine learning algorithms implemented in machine learning projects and distributed MapReduce implementations leveraging Apache Spark. Primarily Naïve Bayes is a linear classifier, which is a supervised machine learning method and works as a probabilistic classifier as well.

**2.3.2 k-Nearest Neighbors**

the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

**Chapter 3**

**Conclusion**

**3.1 Model Evaluation**

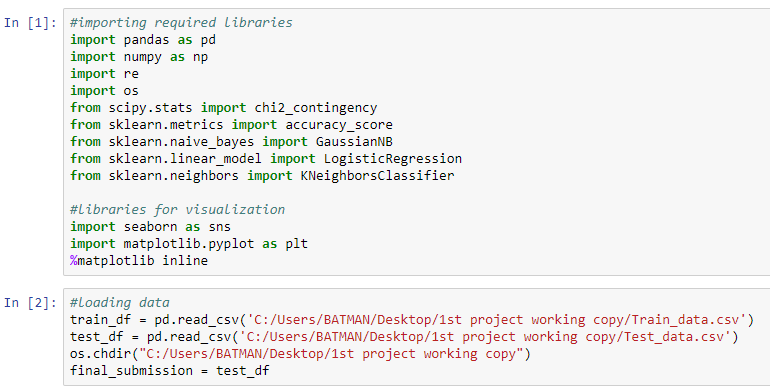
Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

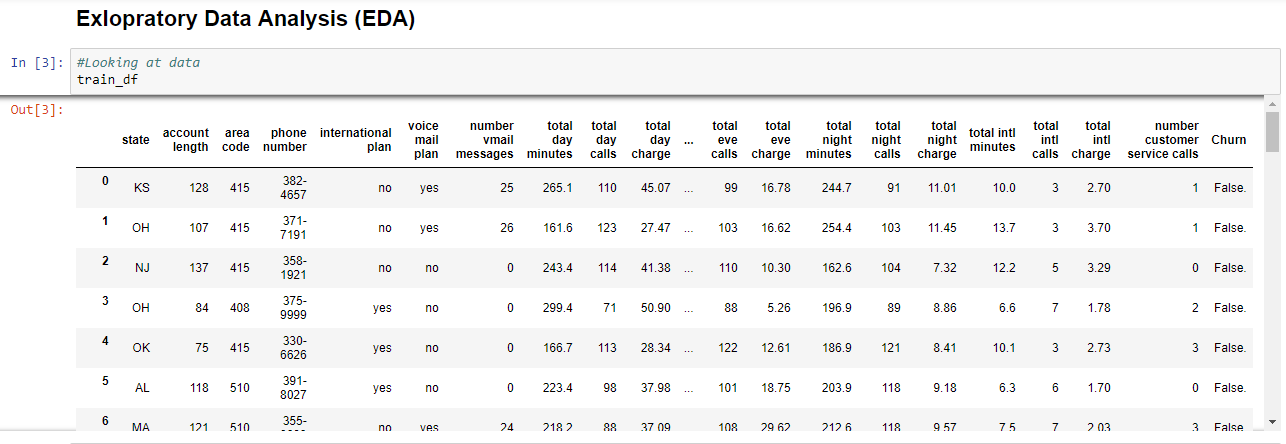
1. Predictive Performance (Accuracy)
2. Required results (probability, rules, etc)

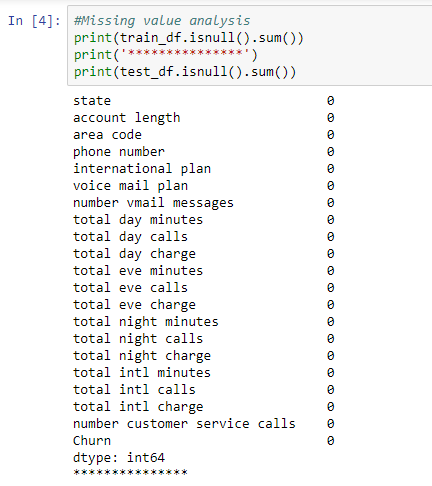
In our case of Churn reduction problem, we have applied three models namely logistic regression, naïve Bayes and knn. and we want our output i.e. Churn Score will be probability of a customer to leave Therefore, we will use logistic regression as it has better accuracy and gives probability as result.

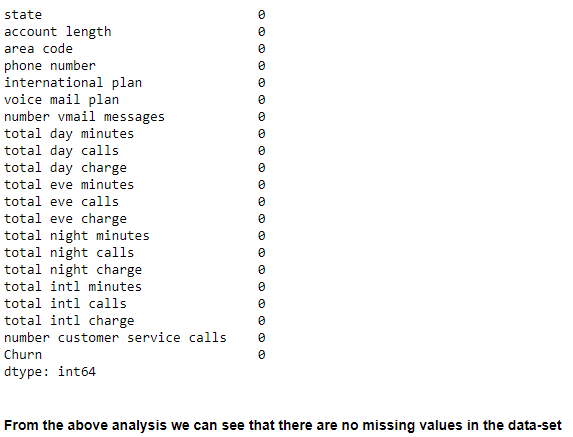
**Appendix – A**

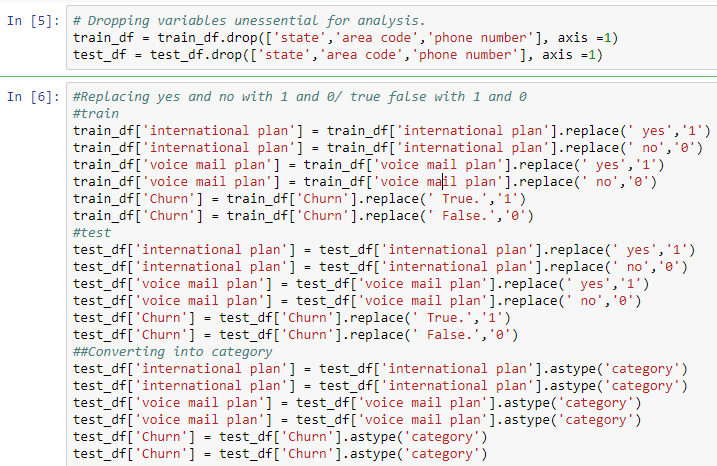
**Python Code**

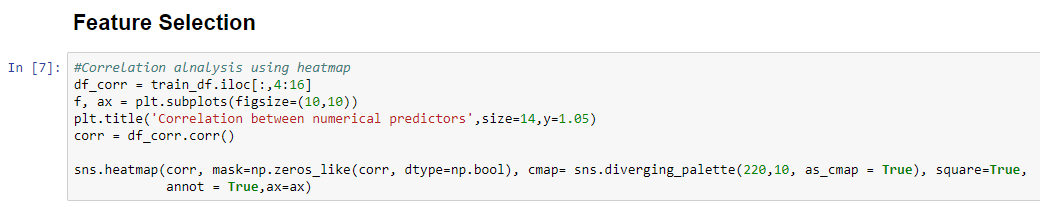


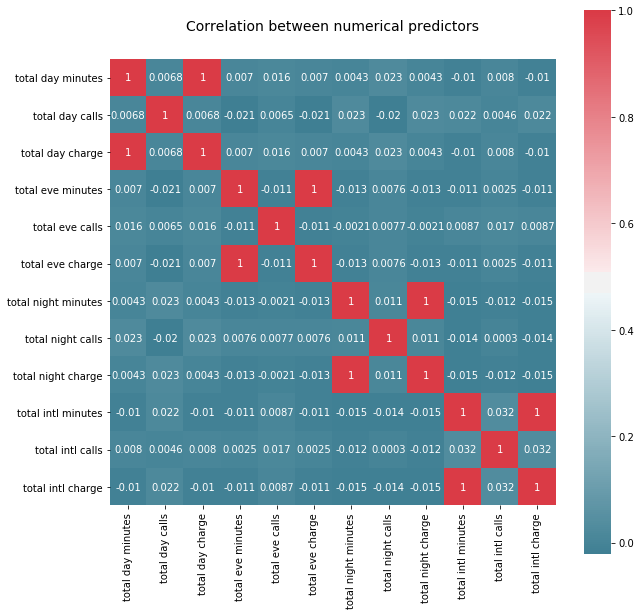


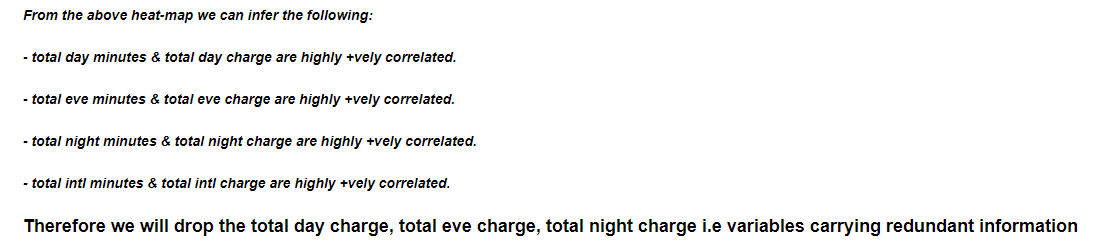


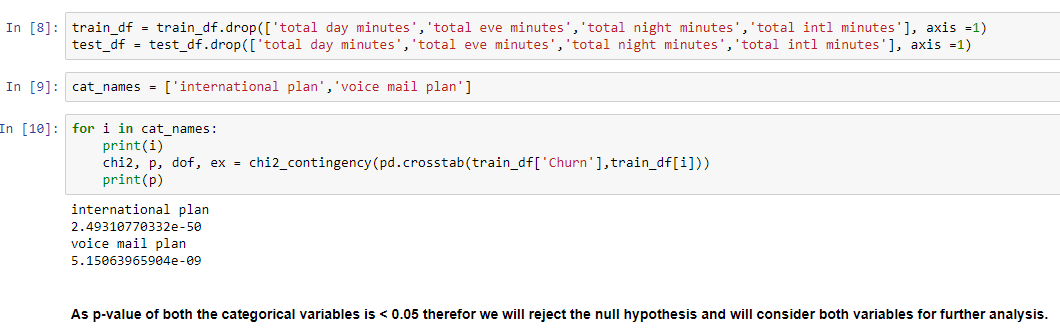


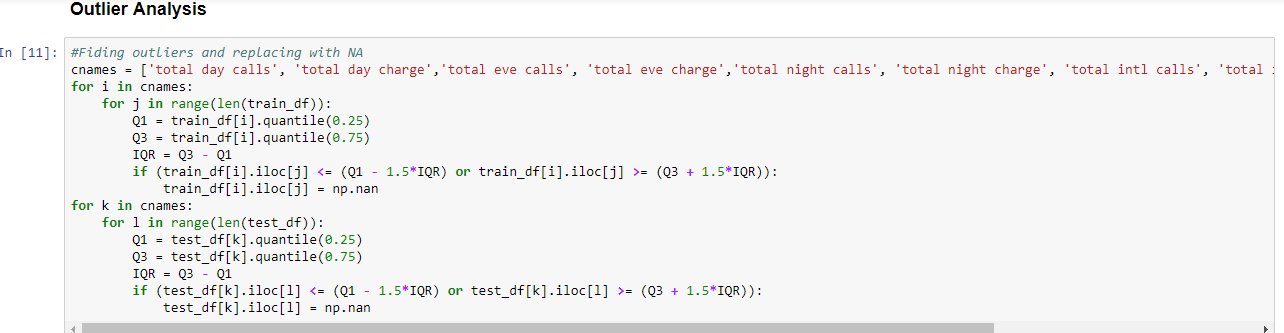


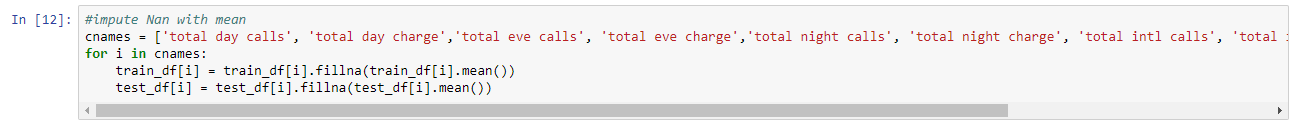


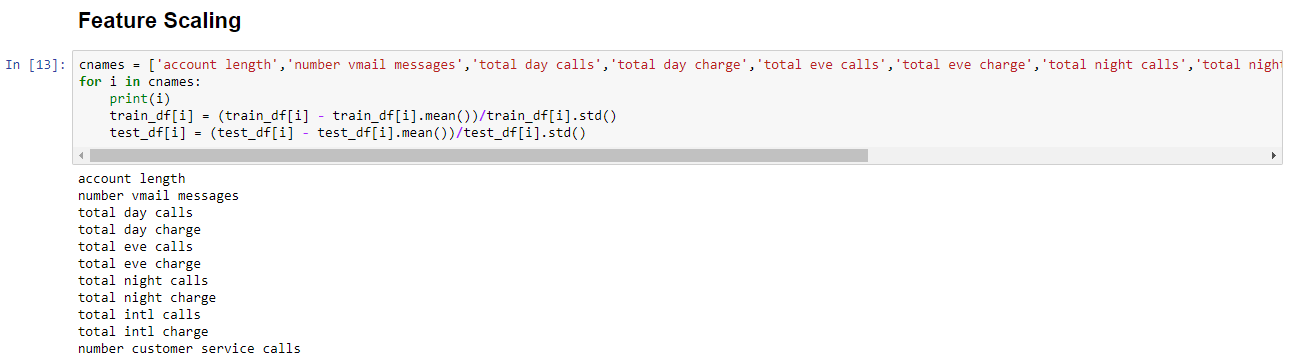
****

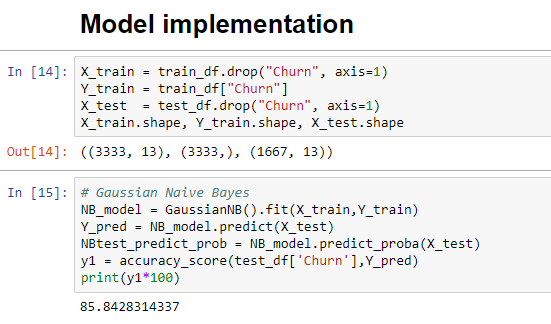




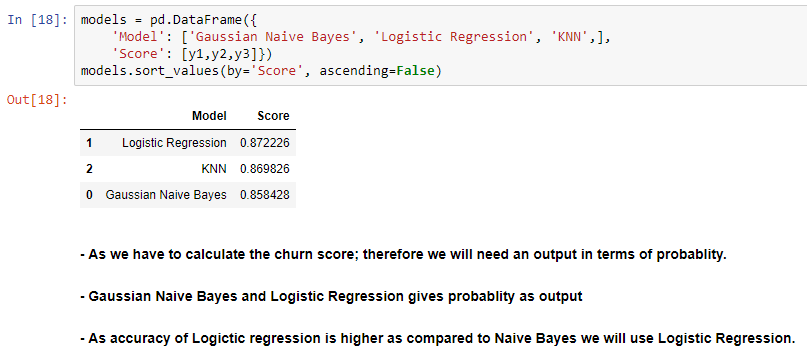


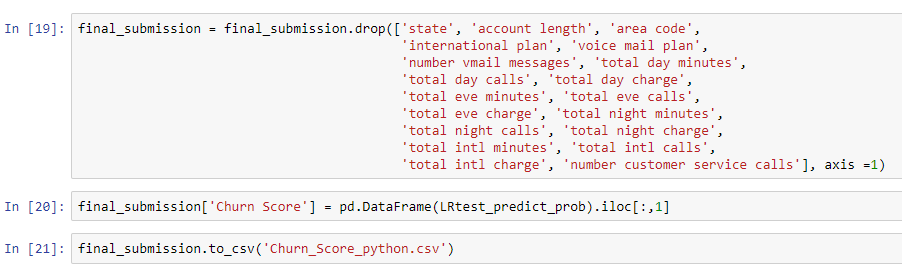






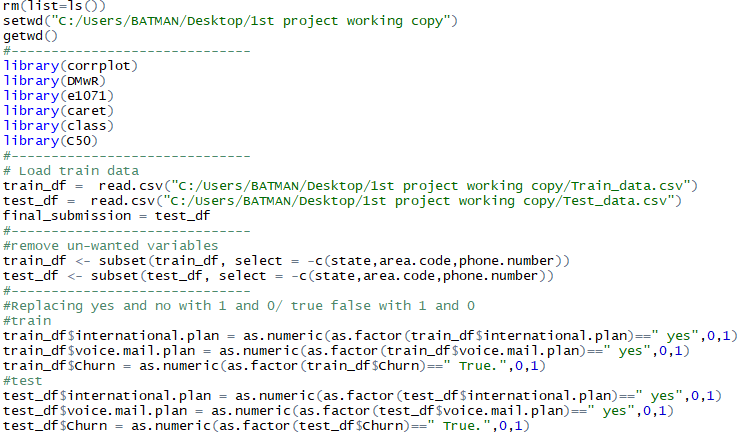


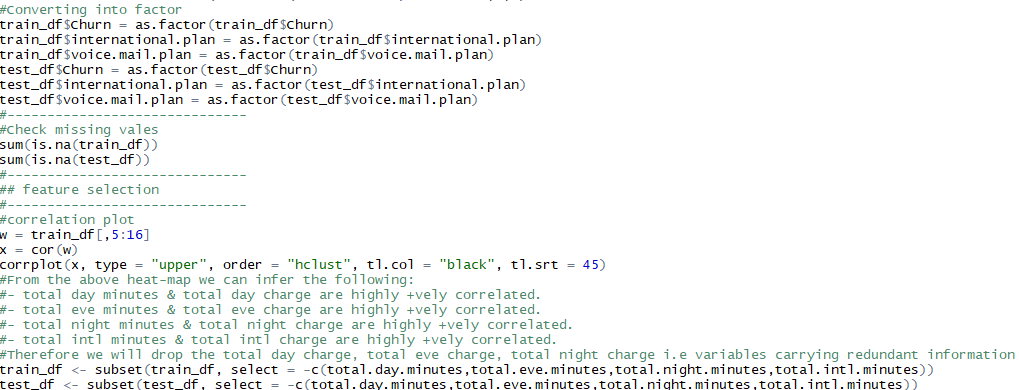


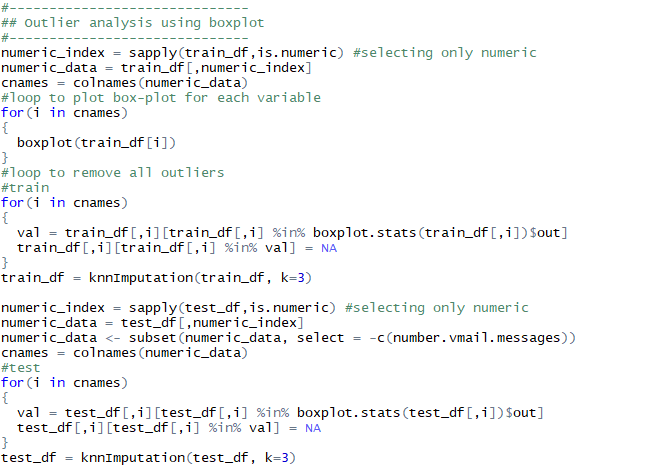


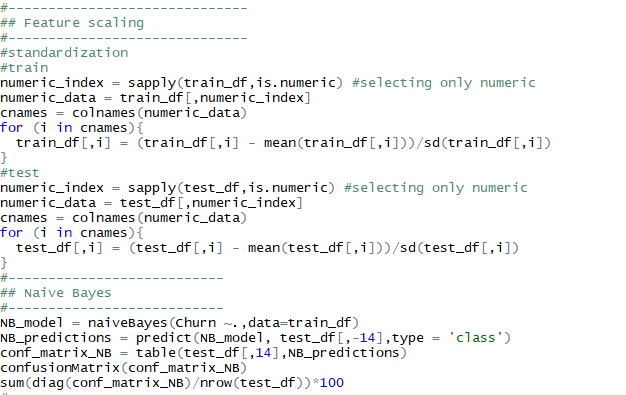
**Appendix – B**

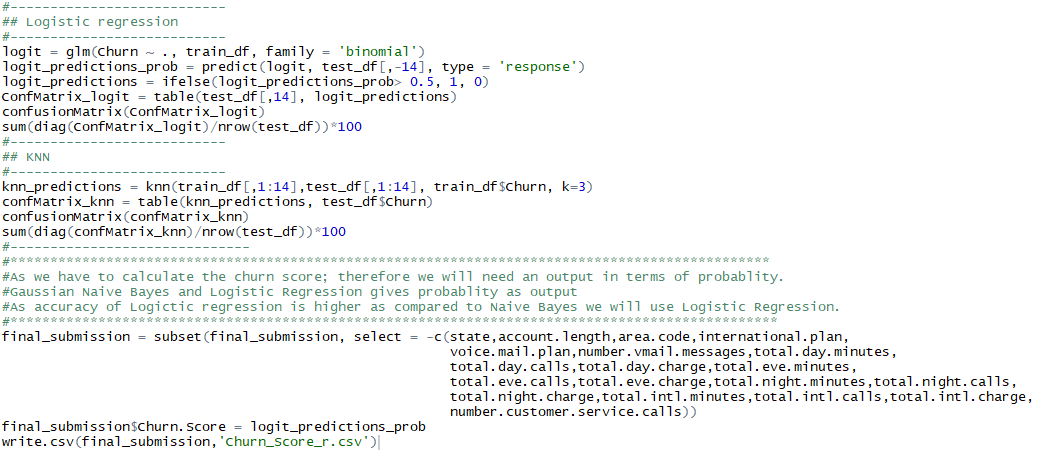
**(R Code)**











**References**

[**https://www.analyticsvidhya.com/**](https://www.analyticsvidhya.com/)

[**https://machinelearningmastery.com**](https://machinelearningmastery.com)

[**https://docs.python.org/3/**](https://docs.python.org/3/)