PROJECT II 2022-2023:

Predict Churn Behavior of Customers from a telecommunication company

Panagiotis G. Vaidomarkakis (p2822203)

Using the provided excel file churn.xls

Tutor: Dimitris Karlis

05/04/2023

Table of Contents

Abstract

In our case, data were collected through a telecommunication company. The whole dataset contains approximately 3333 rows, both with the usage characteristics as well as demographic data for customers. We have 15 numerical variables for call duration, call charge, etc. as far as 6 categorical (factorial) variables about churn status (which is the response for our case), gender, state, etc. Moreover, we have none missing attribute values.

Introduction

In our dataset, we have churn variable which corresponds to if a customer has churned or not churned (values are 0 or 1) and it is the response variable of our model that we will try to present later on.

All the other variables are the following:

- Account Length: the number of days that this account has been active
- VMail Message: presumably the average number of voice mail messages per month
- Day Mins: the total number of calling minutes used during the day
- Eve Mins: the total number of calling minutes used during the evening
- Night Mins: the total number of calling minutes used during the night
- Intl Mins: the total number of calling minutes used on international calls
- CustServ Calls: the number of calls placed to customer service
- Int'l Plan: whether the customer has an international calling plan
- Vmail Plan: whether the customer has a voice mail feature
- Day Calls: the total number of calls placed during the day
- Day Charge: the billed cost of daytime calls
- Eve Calls: the total number of calls placed during the evening
- Eve Charge: the billed cost of evening calls
- Night Calls: the total number of calls placed during the night
- Night Charge: the billed cost of night calls
- Intl Calls: the total number of calls placed internationally
- Intl Charge: the billed cost of international calls
- State: the US state in which the customer resides
- Area Code: the 3-digit area code
- Gender: Male/ Female

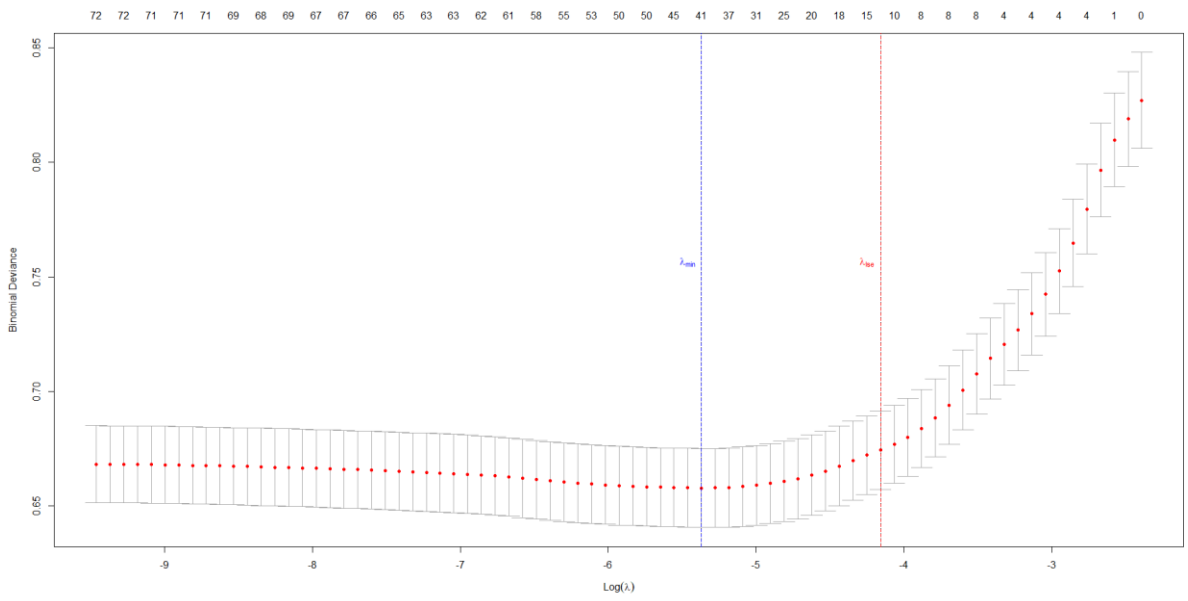
Classification

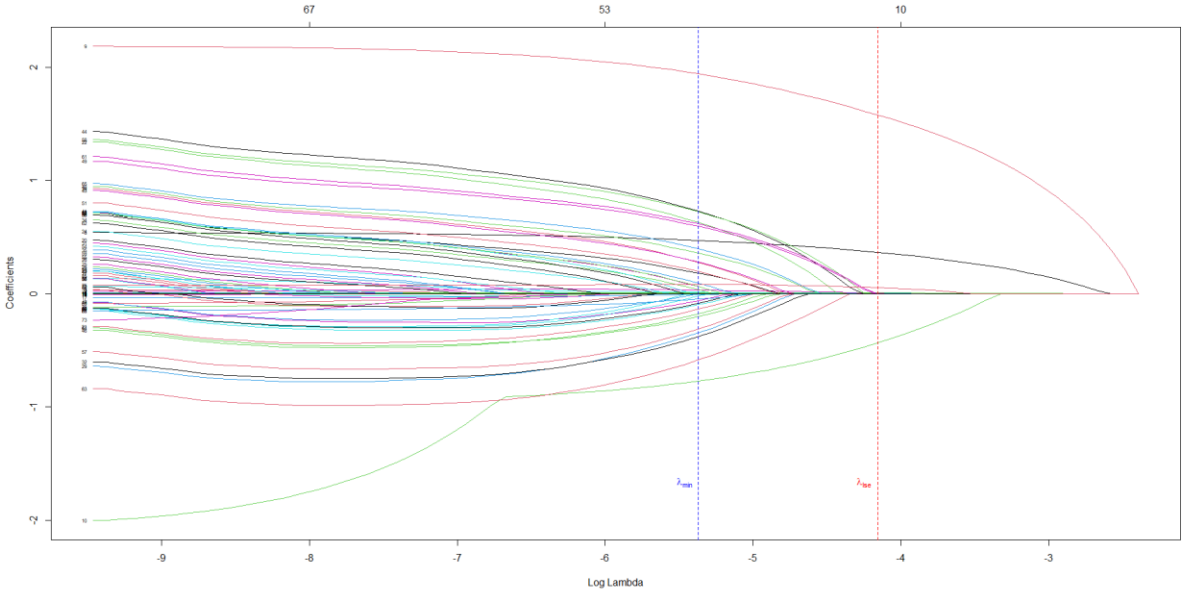
Variable Selection

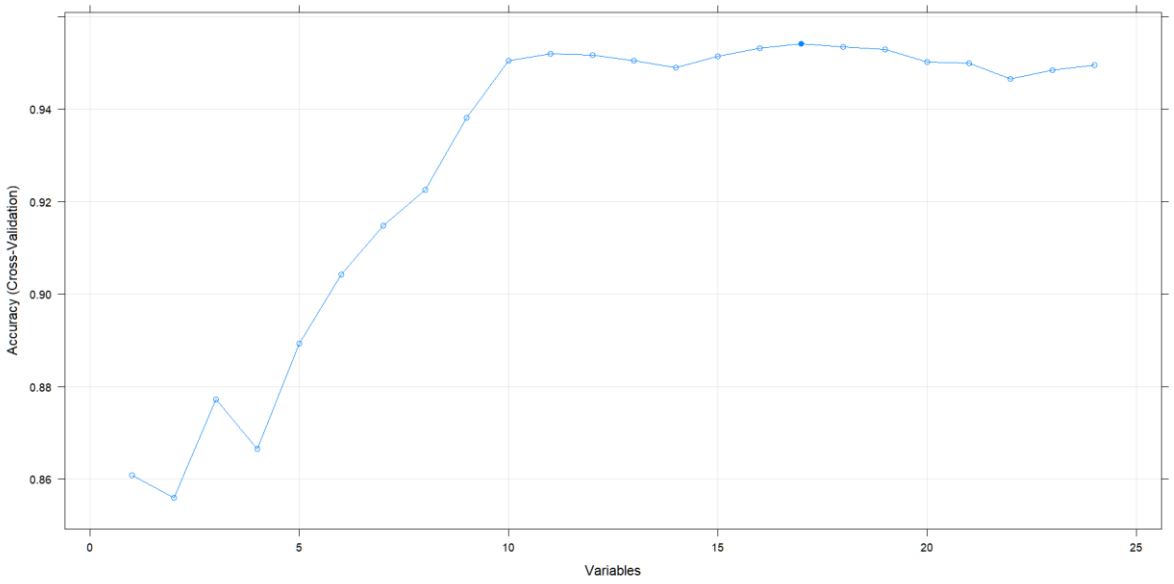
After importing the dataset in RStudio, we need to enrich our dataset with merged information from the columns in order to select which variables are the most significant for a customer to be churner. The features like total day calls and total eve calls measure frequency of usage whereas features such as total day minutes and total eve minutes measure volume of usage.
Another interesting feature to look at would be the average minutes per call. We can measure the average by dividing the total minutes by total calls, for example, the feature average minutes per day call = total day minutes / total day calls and similarly, average minutes per eve call = total eve minutes/ total eve calls. Therefore, we create four new columns which corresponds to average minute per day, per eve, per night and per international call. Some of these columns will have NA values so we make them 0.

In order to make classification more efficient, we need to exclude variables that fills our sample with noise. So, before we begin, we will make a Lasso regression analysis for variable screen and therefore, selection. After performing Lasso, I chose to keep the variables which had minimum log(λ_{min}) because λ_{lse} lose slightly information which is crucial for classification. The output of lasso was 13 variables. Because Lasso typically performs well but doesn't always select the best variables for classification, we will also use a Recursive Feature Elimination (R.F.E) method which uses Random Forest as a method and breaks the set into k-folds and keeps various tree with upper and lower limit of variables. I had set it from 12 variables to 19 variables. After performing the algorithm, the best variables were 17-18 variables. I chose to keep the 18 one in order not to lose more information.

Below are some graphs which help us choose variables from Lasso and variables from R.F.E.:







Preparing and starting classification

Now that we have two datasets to work, we will have the change to compare various methods. Before proceeding to classification, we need create scaled datasets for both sets (Lasso and RFE) because some of the classification algorithms that will follow, need scaled data because the use distances.

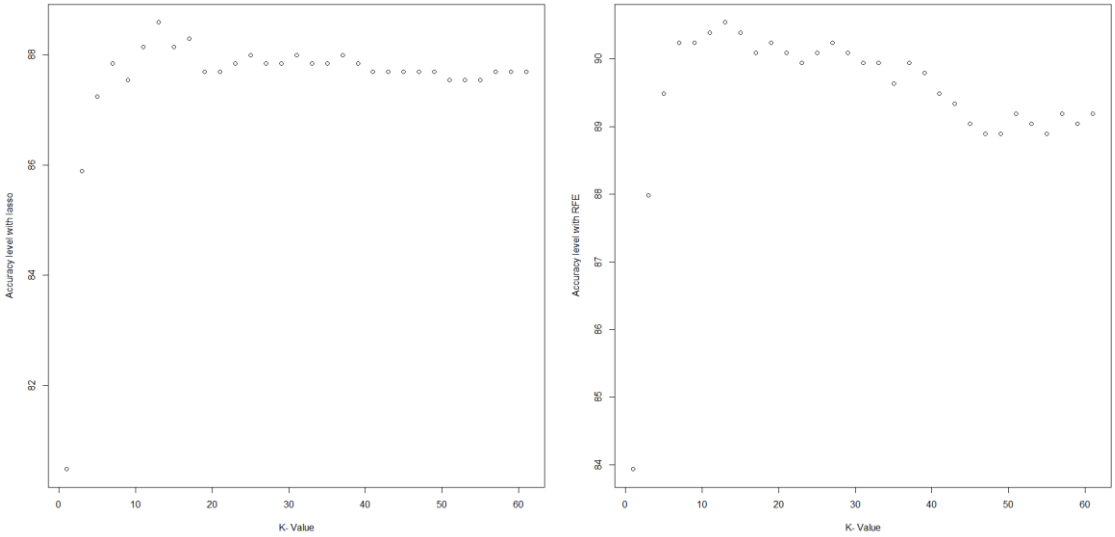
After that, we are ready to separate our sets to training and testing dataset. To do that, we set a seed to our randomizer in order to have the same rows in both Lasso and RFE. After all this procedure, we are ready to begin.

We will begin with QDA Algorithm. Our initial thought was to begin with LDA but after performing BoxM Test, we can see that we cannot assume multivariate Distribution which means that we cannot assume eq. variances.

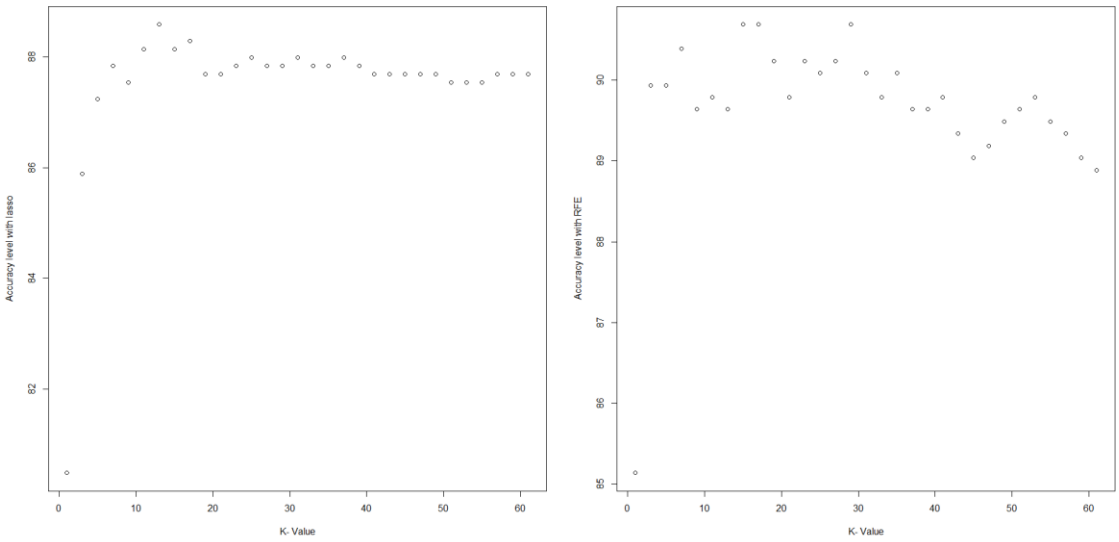
```
        Box's M-test for Homogeneity of Covariance Matrices

data:  churn_data_rfe_num[, -16]
Chi-Sq (approx.) = 1364.4, df = 153, p-value < 2.2e-16
```

So, we will start performing QDA. With Lasso set, we had 82,9% Accuracy as far as with the RFE set which had 82,4%.

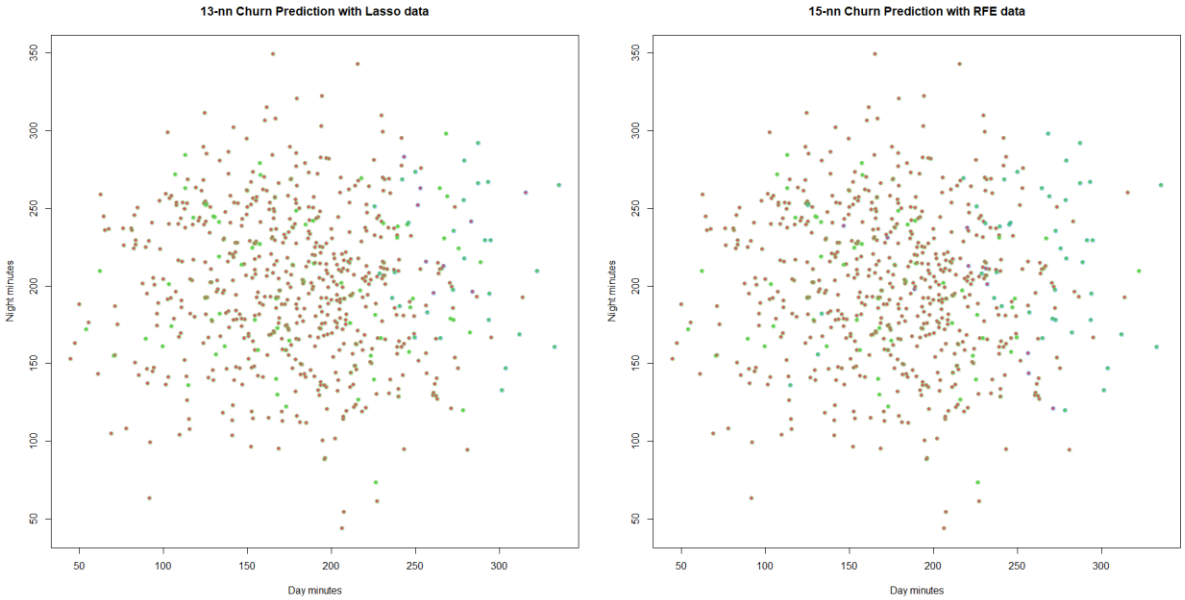
Then, we moved on with the K-nn algorithm. This method needs scaled data in order to have good results. K-nn is a method which 'see' k neighbors near by and put the new input in the class of the most points around it. So, we run a for loop in order to see which k neighbor has the best accuracy. Below is the graph in which k correspond to Accuracy:



The right plot is for the Lasso data and the left one is for RFE data. This plot is without factorial variables in the RFE dataset. Below, is the same graph but with factors in RFE dataset:

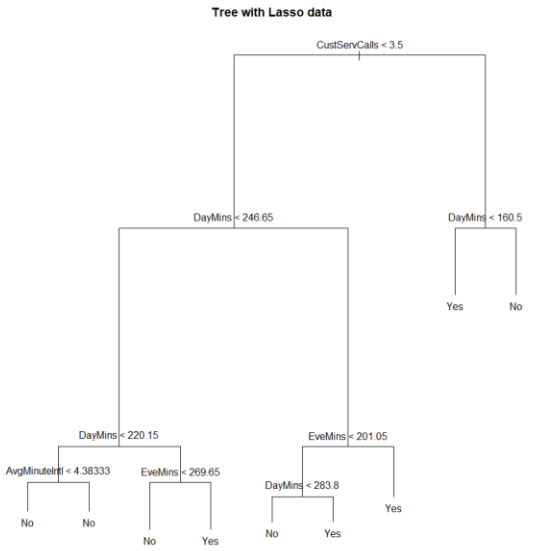


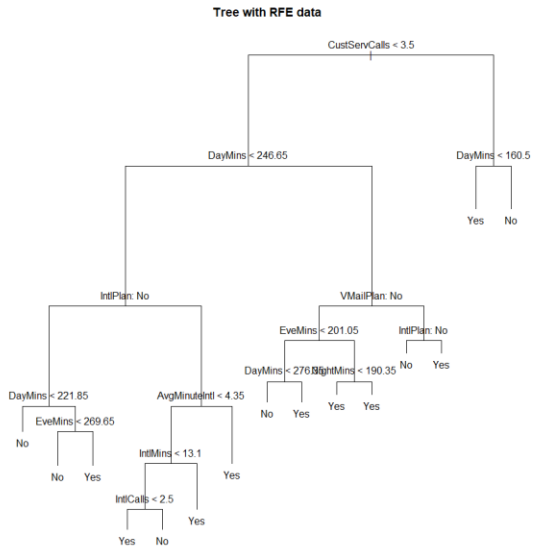
As we can see, the second one is more unstable in comparison to the first one, but we kept this one because we only care about prediction and this has slightly better accuracy.
So, the end results are 13-nn for Lasso scaled data with 88,6% accuracy and 15-nn for RFE scaled data with 90,7% accuracy. Below, you can see the Day to Night minutes classification and prediction with both datasets:



As we can see, the predicted values are hovered by the actual values which leads us that both methods have predictive power. Although, RFE data have slightly more power to predict the actual number (this can be seen in the center left of the plot which has some light blue points).
Moving on, we have tested again with scaled data Multinomial logistic method with 84,5% in RFE in comparison to 85% accuracy in Lasso data.
Later, we tried Naïve Bayes with 84,4% accuracy in Lasso data and 85,7% accuracy in RFE data.
Moreover, we tried the tree method with spectacular results. We had 87,5% accuracy in Lasso data and 91,6% accuracy in RFE data.

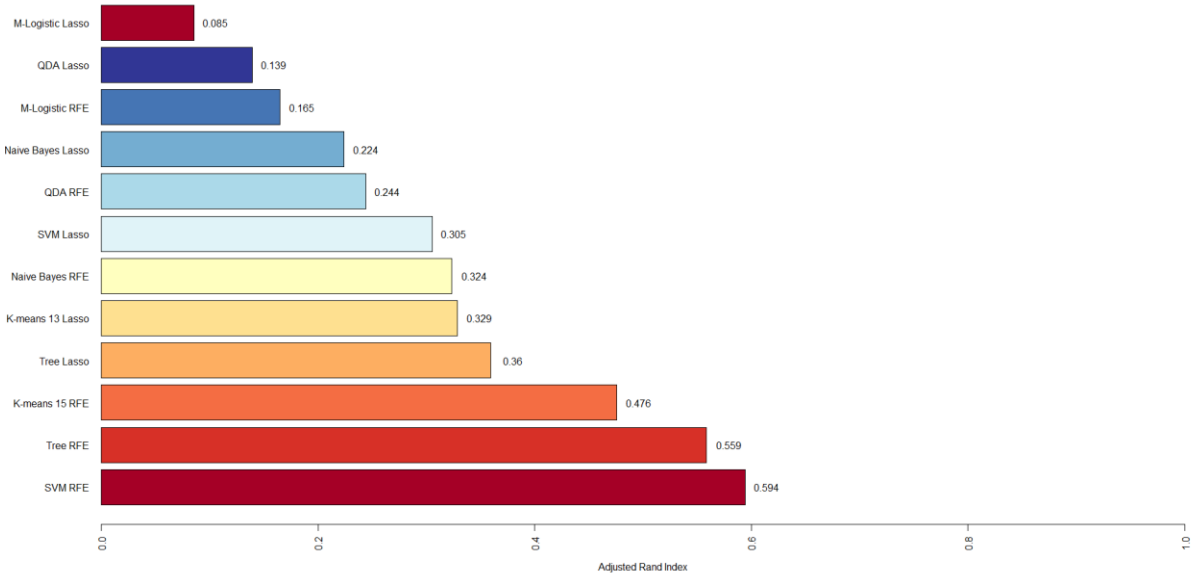
Below, is the final tree with each of our datasets:





Lastly, we tried the support vector machine method with 92,8% accuracy on RFE dataset in comparison to 88,4% in Lasso dataset.

After all the above method, we used Adjusted Rand Index (A.R.I.) in order to select the best out of them. Below are the results:

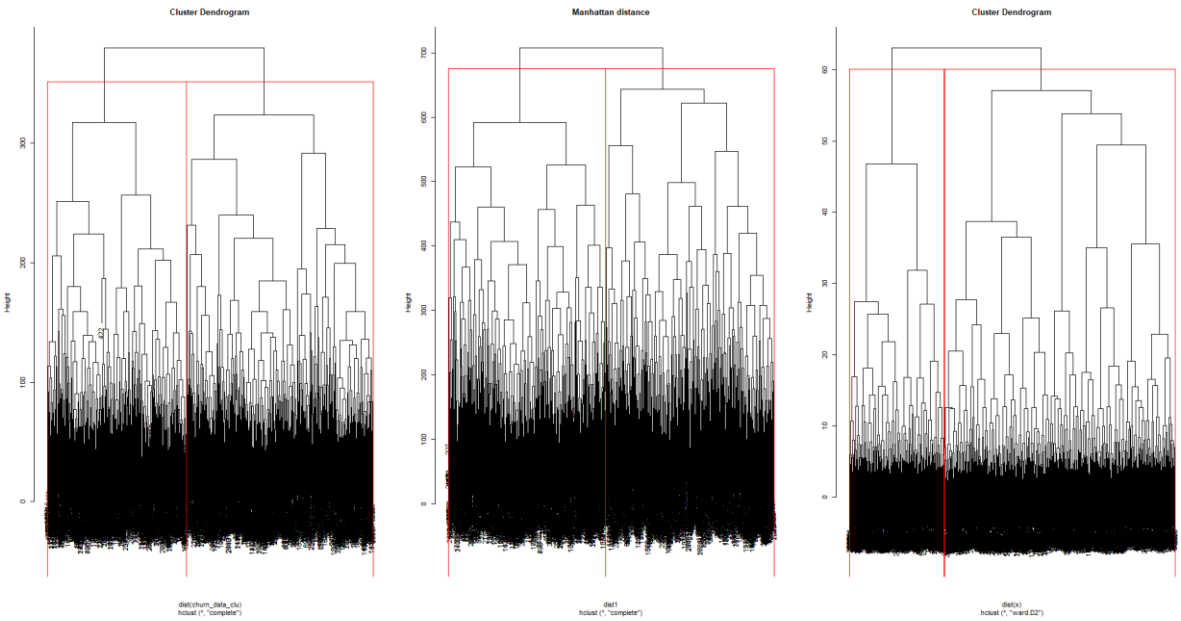


As we can see, RFE dataset tends to train better the models for classification and in the end, the best two methods are Support Vector Machine (S.V.M.) and the Tree method. Multinomial Logistic and QDA where the worst methods for classification.

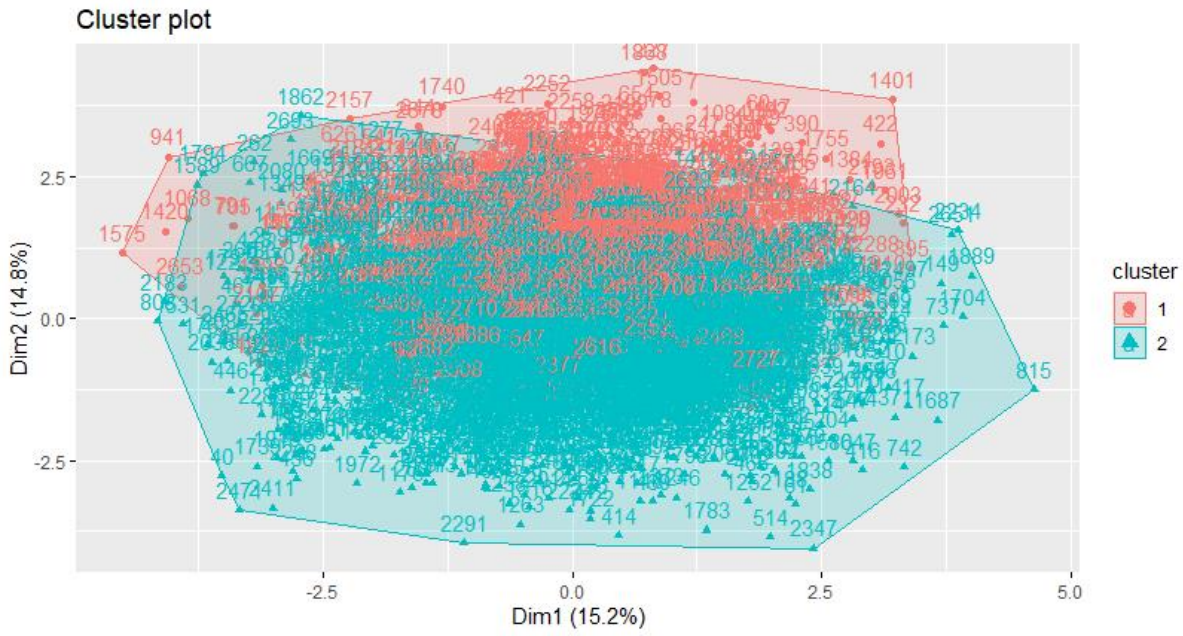
Clustering

For the clustering usage, we kept only the usage metrics and our new added metrics. In order to have nice clusters, it needs to remove outliers in each of our variable in the dataset. From our initial dataset of 3333 observations, we ended up with 2735. After removing outliers, we scaled the data.

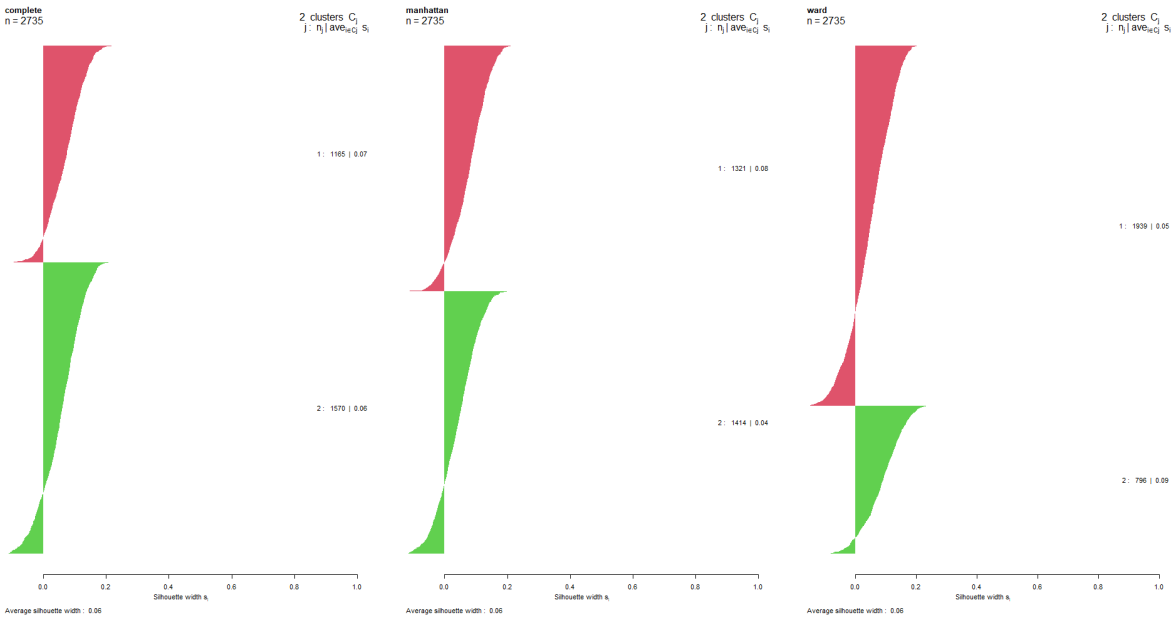
We used hierarchical clustering and after many tests, we ended up that 2 clusters were the best. Below, you can see the silhouette plots:



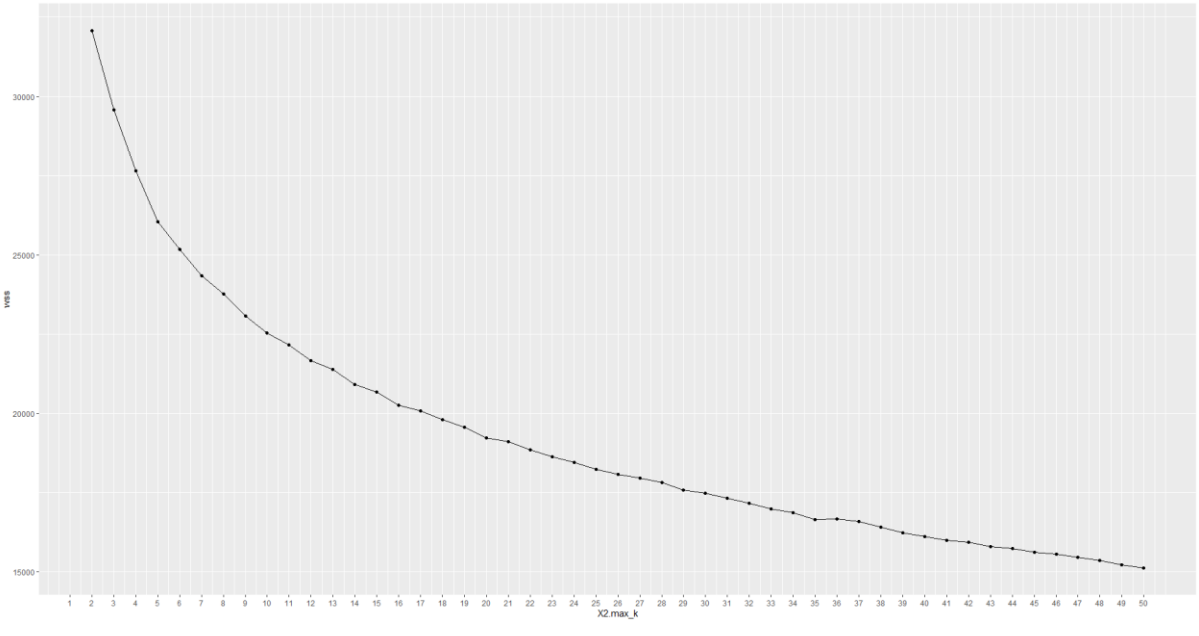
Below you can see a visualization of the clusters from complete method using Principal Component Analysis (P.C.A.):



Below, you can see the silhouette plots:

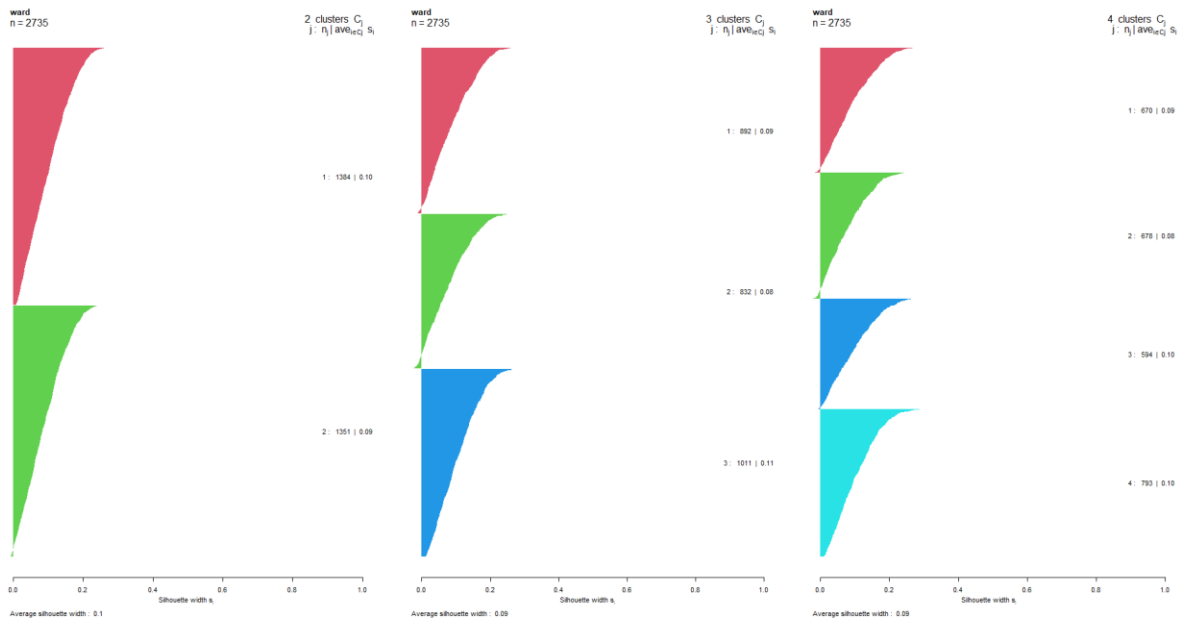


Moving on, we tried K-Means algorithm. We plot Within Sum of Squares (WSS) in order to decide the number of clusters. Below is the plot:

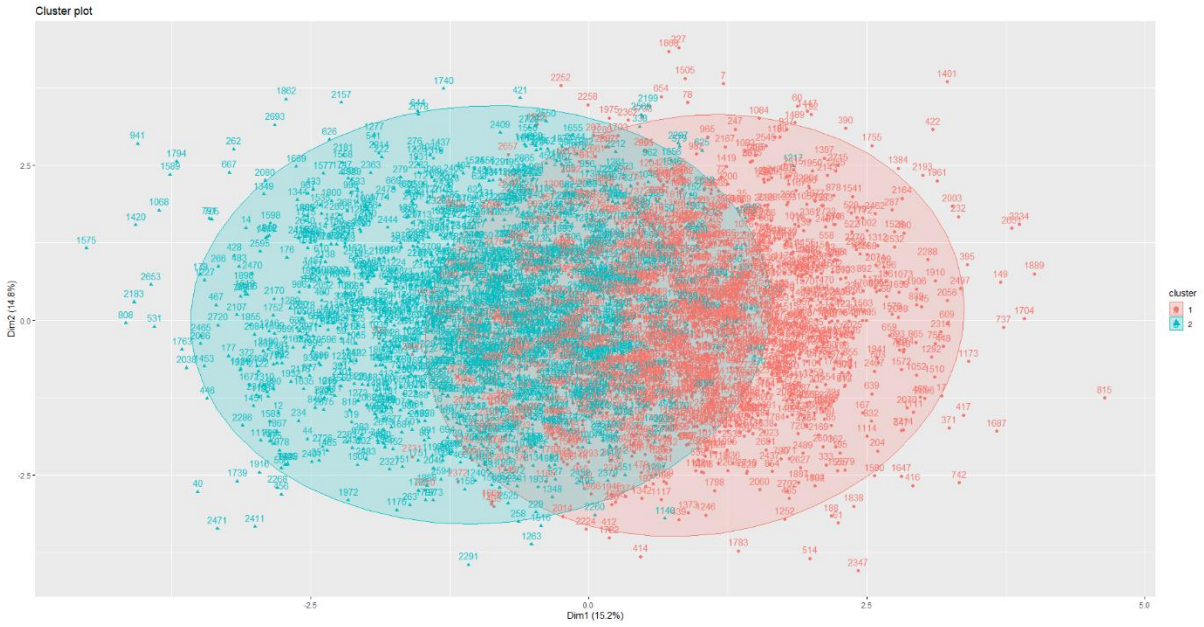


As we can see, 2 again is the best predictor for clustering.

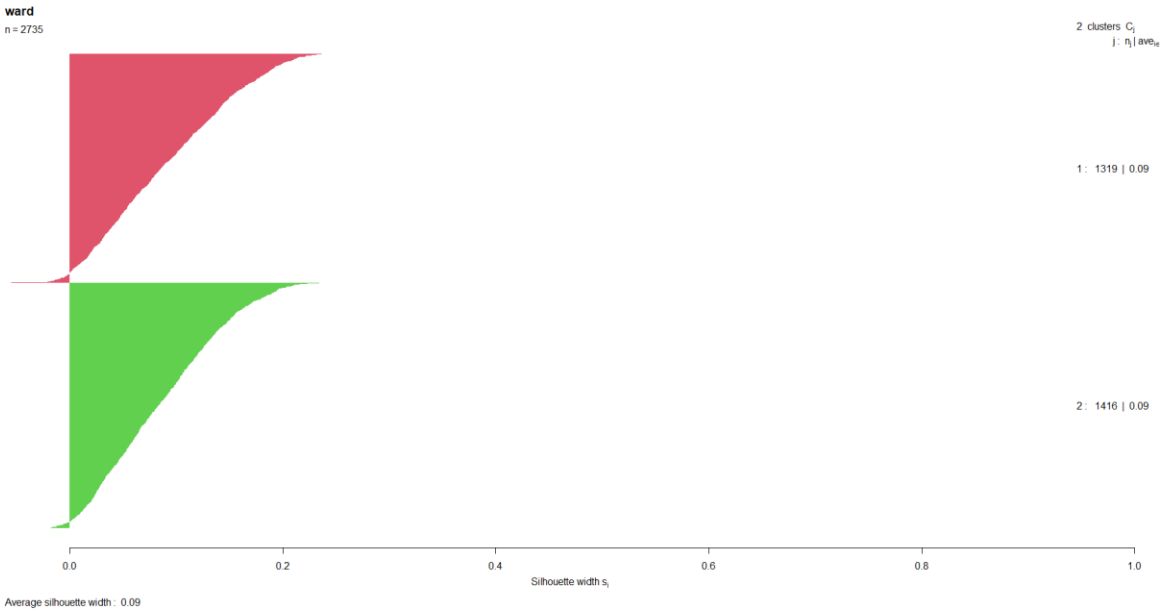
Below, you will find also silhouettes plots for 2 K-means, 3 K-means and 4 K-means.



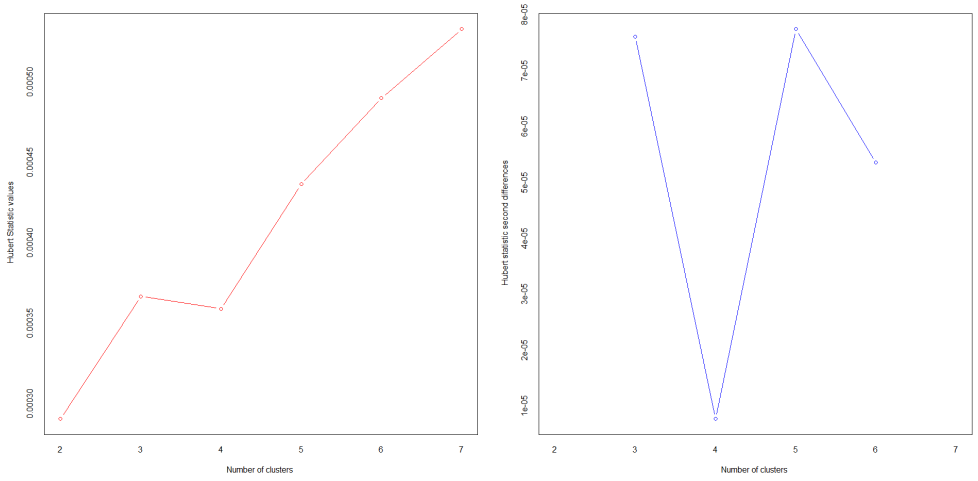
Below is a visualization from 2 K-Means with predicted clusters using again P.C.A.:

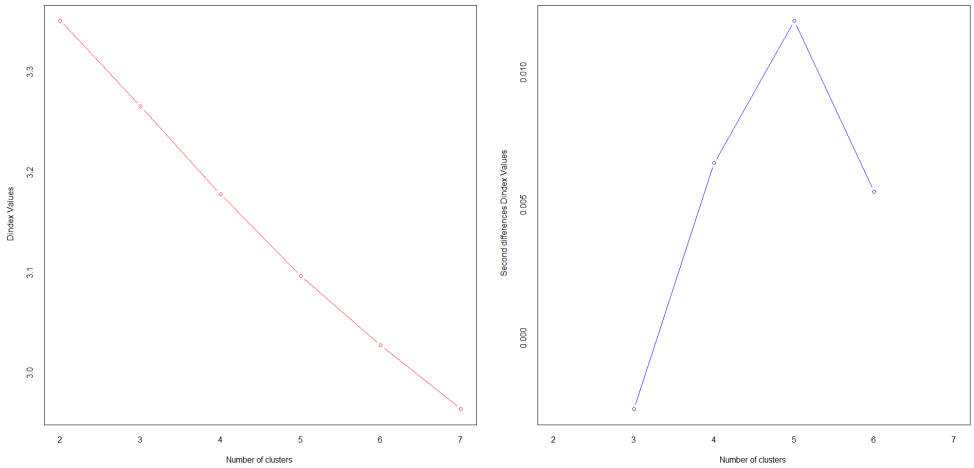


In the end, we've used Mclust and below you can see the results from silhouette plot:



Lastly, we've used Nbclust method in a much smaller random sample of 547 observation (because it is time consuming) in order to have a hind of how many clusters are good for our data. The output of the algorithm was 2. Below, you can see the output plots:





```
*** : The Hubert index is a graphical method of determining the number of cluster
s.
                In the plot of Hubert index, we seek a significant knee that corre
sponds to a
                significant increase of the value of the measure i.e the significa
nt peak in Hubert
                index second differences plot.

*** : The D index is a graphical method of determining the number of clusters.
                In the plot of D index, we seek a significant knee (the significan
t peak in Dindex
                second differences plot) that corresponds to a significant increas
e of the value of
                the measure.

*******************************************************************
* Among all indices:
* 9 proposed 2 as the best number of clusters
* 1 proposed 3 as the best number of clusters
* 4 proposed 4 as the best number of clusters
* 2 proposed 5 as the best number of clusters
* 1 proposed 6 as the best number of clusters
* 6 proposed 7 as the best number of clusters

                   ***** Conclusion *****

* According to the majority rule, the best number of clusters is  2
```

Comparing this with the previous silhouette plots, the best clustering was provided from 2 K-means. Below, you will the mean value of each variable in each new cluster in order to understand which cluster has which customer:

```
  cluster VMailMessage  DayMins  EveMins NightMins IntlMins  DayCalls
1       1     7.380058 163.4661 189.6358  225.0512 10.01893 105.21098
2       2     8.820873 193.0996 208.3293  173.5226 10.43560  97.20577
   EveCalls NightCalls IntlCalls AvgMinuteDay AvgMinuteEve AvgMinuteNight
1 103.99639   94.17052  4.744942     1.598701     1.873886       2.443089
2  97.69356  107.45892  4.165803     2.045971     2.208385       1.640834
  AvgMinuteIntl
1      2.526612
2      3.031031
```

As we can see above, the major gap between the 2 clusters is DayMins, EveMins and NightMins which lead us to this conclusion. The first cluster contains all customers who make many Night Calls with a lot of talking time during night while the second cluster contains all customers who make many Day Calls and Eve Calls with a lot of talking time during day and evening.