Group project GAG

Giulia Formiconi, Giulia Macis, Alessandro Ivashkevich

Task 1: Telecom Churn

The objective of this project is predict behavior to retain customers. Analyzing all relevant customer data can help to develop focused customer retention programs. The dataset 'Churn' contains information about the State, Area, Account, Type of plan of the customers, how much they have charged, the schedule they do more calls and others. Here these variables are explored in order to get insights about customer behavior, and Machine Learning Algorithms are used to predict if a customer will continue with its plan or not.

Task 2: Explanatory Data Analysis

Firstly, we load all the libraries that we will need to read and manipulate the dataset, to visualize the plots, to build the model and to get metric scores. Then, we load the dataset.

Before strarting develop our predicting models, we have to get an overview of our dataset. The main objective in the EDA is to explore, visualize the dataset and identify which factors contribute to customer churn. To achieve this objective and perform an efficient EDA we need to pre-process and clean our data. In the pre-processing step we check for duplicates and null values, which are zero. Then we must have a glimpse of our data, and especially of the data structure: get the size of the dataset, the number of variables and its datatype, in order to be able to deal with them.

In this dataframe overview there are 3333 objects of 20 variables. We have to deal mainly with a numerical features, except three categorical ones: State, International.plan,Voice.mail.plan and Churn. Also the target variable is chosen, in order to predict which customer will leave the company the target variable is Churn. It is a binary variable with 2 possible outcomes: Churn=True and NotChurn=False.

The several relationship between the different variables will be analysed to uncover pattern and trends. Basically, we explore with the summary () command the structure of our data from a statistical point of view, in order to get an idea how variables distribution is. Almost all the numerical variables follows a symmetric distributions, meaning that most of our data is balanced and bell-shaped.

It is necessary to convert the target variable 'Churn' into factors, changing respectively False and True to 0 and 1. This improves the clarity and interpretability of the results and to achieve a model compatibility.

Data visualization analysis is done considering the target variable and the most important features, in order to get useful insights.Firstly, we check the churn rate. Checking this is essential for gaining information about customer behavior in our dataset. It provides a foundation for optimizing customer retention strategies.

The pie chart shows that the Churned-False are the 85.51%, whereas the Churn-True are only the 14.49%. This means that the dataset is unbalanced, posing some challenges in the predictive modeling. Since the majority class dominates the training set, the model may learn to make predictions that are biased towards that class and fail to capture the nuances of the minority class.

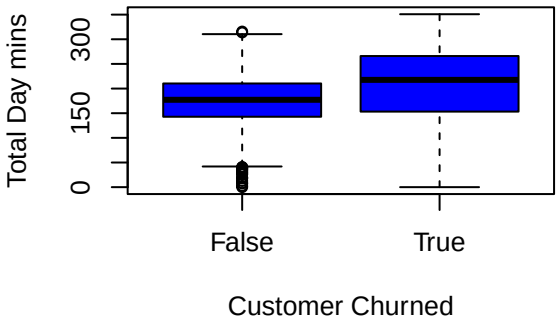
We plot some box plots to understand the various attributes of input variables for churned and not churned customer. It can be noticed that:

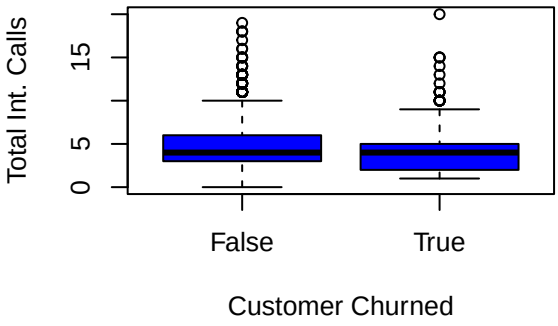
- for Total Mins (Day/Evening/Night/International) churned customers are higher compared to not churned.

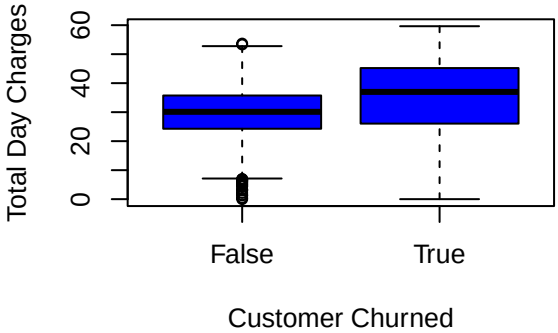
- for Total Calls (Day/Evening/Night/International) churned customers call less but the duration of their calls are relatively high.

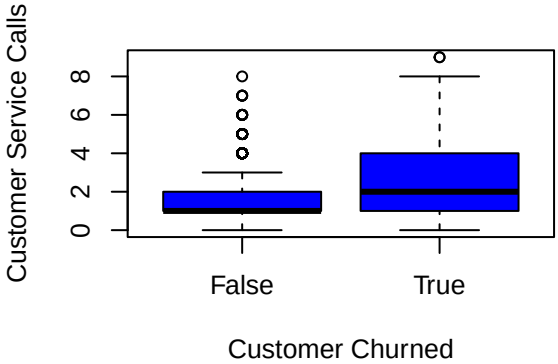
- for Total Charges (Day/Evening/Night/International) not churned customers are relatively higher than Churned Customers.

- for Total Customer Service Calls made it seems that churned customers call Customer Service a lot.

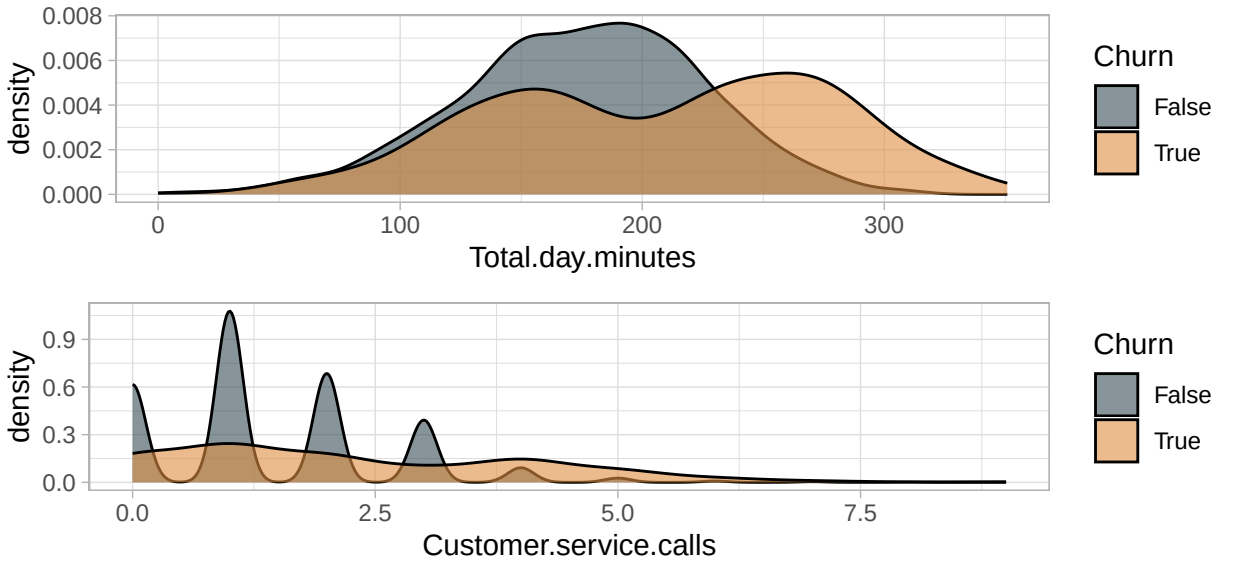




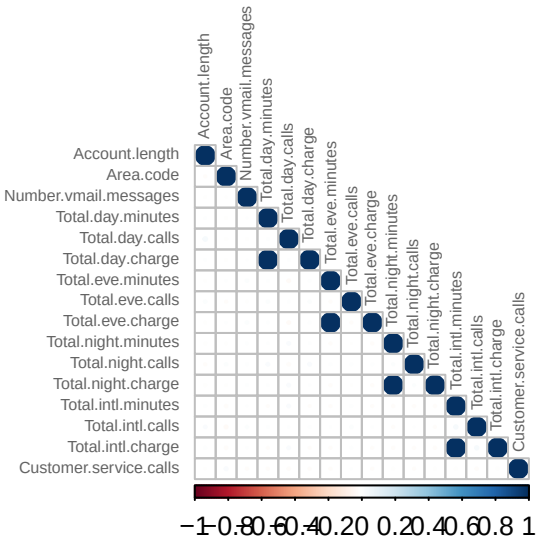




From the following density plots, it can be seen that Total.day.minutes and Customer.service.calls are the variables that influence the most the target variable. Churned customers tend to spend more minutes and for this reason tend frequently to call the customer service.



The correlation map is plotted in order to determine which are the most important features.



It can be noticed that Total.night.charge and total.night.minutes, Total.day.charge and Total.day.minutes, Total.intl.charge and Total.intl.minutes are highly correlated with each other. This is expected because if the person uses more minutes, they will be charged more. Since these variables are highly correlated, it is appropriate to delete one of them (all charges variable) to reduce the redundancy. This because we want to avoid the risk of overfitting, cause by the multicollinearity.

Task 3: Customer Churn Classification

Customer churn is a problem that occurs when customers stop using in our case a call plan service.It can be caused by several factors, including poor customer service, high prices, and competition from other companies. Determining the churn rate involves measuring the percentage of customers who do not renew their plans, negatively impacting a company's revenue and customer loyalty. To deal with customer churn, companies must identify its causes and take action, such as improving service, offering discounts, and increasing marketing efforts.

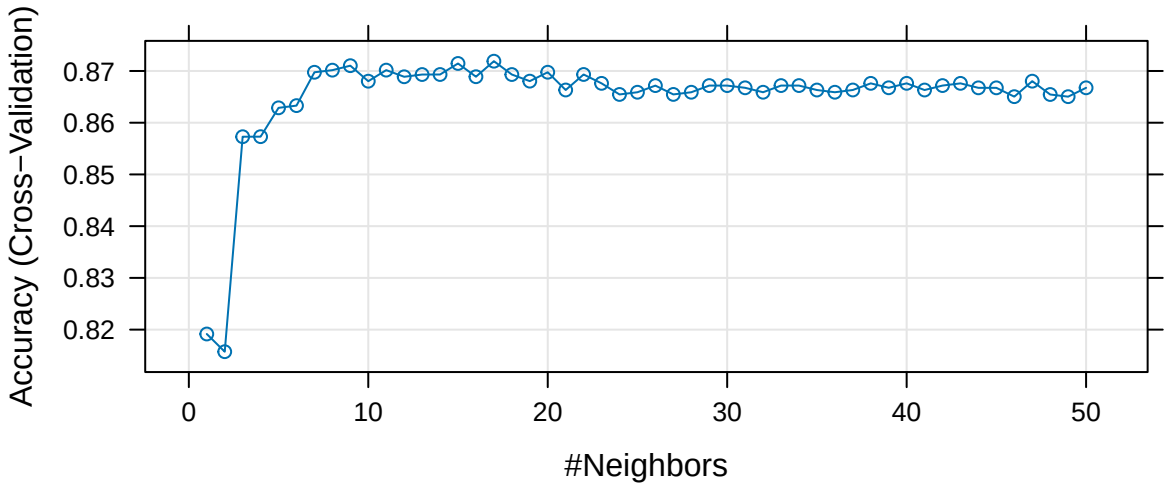
Customer churn prediction involves using machine learning models to identify customers likely to churn and

predict their behavior within a specific time frame. This is a classification problem and it helps to determine if a customer will churn or not. After cleaning and processing the data, they are split into training and test sets. The training set trains the model by feeding it with churn and non-churn data, enabling the model to learn and adjust its parameters. Predictions are then made on the test set to evaluate accuracy and overfitting.

We have chosen to implement various machine learning algorithms for customer churn prediction: Logistic Regression, LASSO, Decision Tree, XGBoost, Random Forest, BIC, and AIC.

Task 4: Lower-dimensional model with KNN

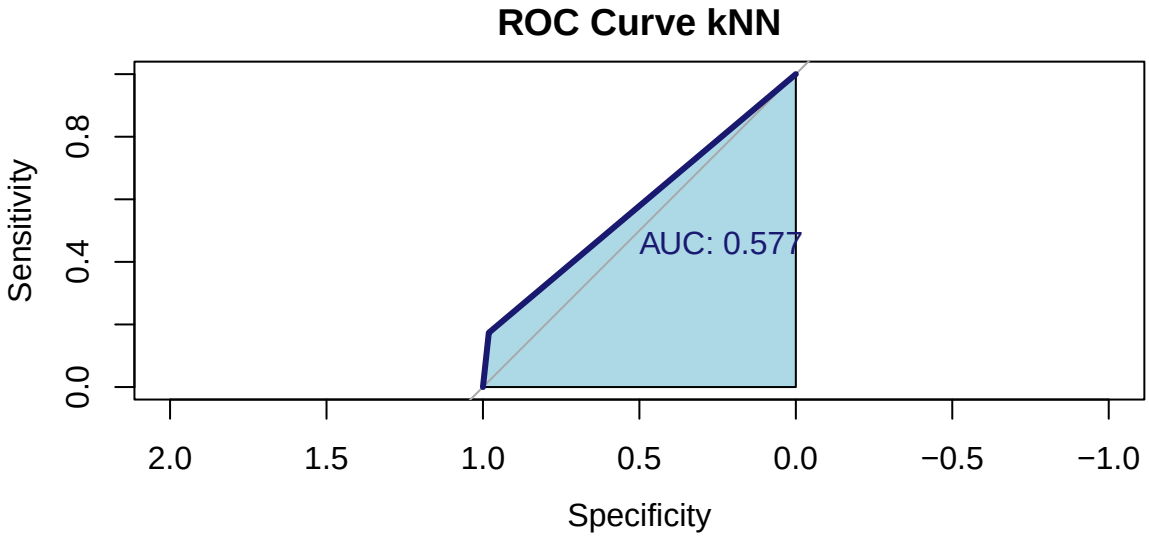
KNN is a straightforward algorithm. When a new customer is presented, the algorithm looks through the database for customers who are most similar to the target customer. We choose as predictor variable the Total.day.minutes and Customer.service.calls, which are the most informative variable. This is understandable, a client calling customer care a lot is probably not satisfied with the service so is more likely to churn. While people who make a lot of calls are probably satisfied with the service and so are unlikely to churn. So these two variables tell us a lot about the probability of a customer churning, as we already highlighted before in the EDA. To find the optimal number of k we use the 10-fold cross validation, which provides a more robust evaluation. This because evaluating the model on a single train-test split may result in an over-optimistic or pessimistic estimate of performance. It can be seen from graph below that the optimal k is 17.



The AUC score is 0.5774 and it is very low, as we expected, because using only two variables we expiriance information loss. Notice also that the unbalance in our target variable influence the results because in KNN the neighbors are weighted equally in the decision-making process. When there is class imbalance, the neighbors from the majority class can dominate the decision, leading to misclassification of minority class instances, such as in this case the Not-Churn prevails the Churn. Moreover, for the same reason this KNN model achieve high accuracy by simply predicting the majority class for most instances. However, this accuracy does not reflect the model's ability to correctly classify the minority class instances.

```
## Setting levels: control = False, case = True
```

```
## Setting direction: controls < cases
```



Task 5: Classification models

Several models are implemented in order to predict customer churn. Firstly, we split the data in 2 subsets: train and test. In this way the algorithm can build the model using the train set and then they can improve this model, making prediction on test set. As linear models, we use 3 different models: Logistic Regression, AIC and BIC stepwise selection.

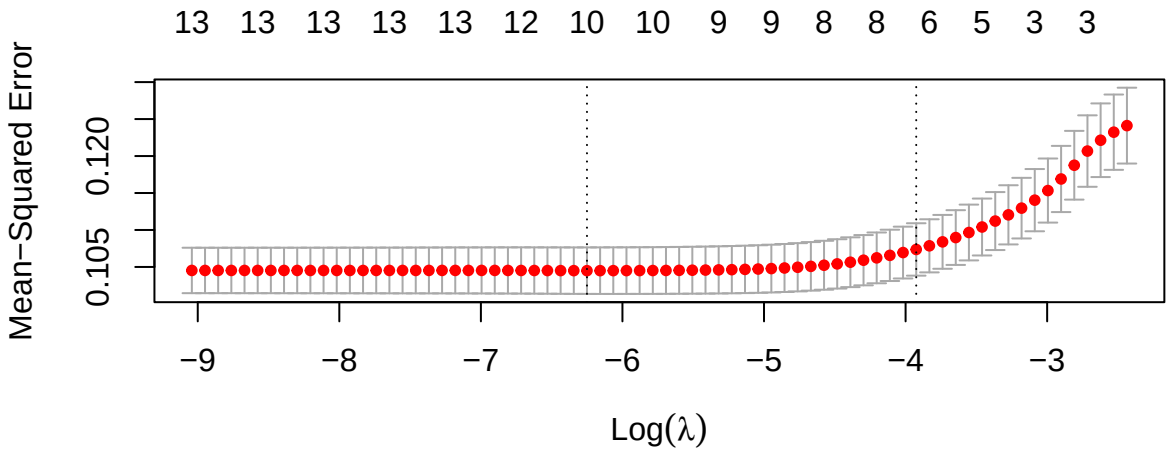
Firstly, we will focus our attention on Logistic Regression model. This latter performs pretty well because it provides a probability interpretation, offering interpretable coefficients. This coefficients provide information about the direction and magnitude of the relationship between predictor variables and the log-odds of churn, helping us understand the factors that influence customer churn. In fact, positive coefficients indicate a positive and strong relationship with the target variable, the contrary for negatives ones. The greater the absolute value of the estimate, the more influential the predictor variable is in predicting the outcome. In our model, the most important are: Total Initial/Eve/Night minutes and Customer.service.calls. This confirm our results of the previous analysis. As we know the threshold used to classify the predicted probabilities into binary outcomes, as default is 0.5, however it might not be suitable for our data. So, we implemented a for loop that plug all the values of the threshold can adopt (from 0.99 to 0.1) and we calculate the sensitivity, specificity and accuracy. The threshold that maximizes the sum of all the metric is the best one, in our case is 0.18.

As previously mentioned, we utilized bic and aic stepwise selection techniques to choose the most significant predictors from a larger pool of potential predictors. These methods exclude certain variables: Aic excludes: state, total day calls, total eve calls, total night calls, account length, and number of voicemail messages. The Bic excludes state, total day calls, total eve calls, total night calls, account length, and number of voicemail messages. It's important to note that these models assume a linear relationship between predictors and the outcome variable. However, in our case, the relationship is nonlinear, which means that these techniques may not identify the optimal subset of variables.

Another important model considering the penzalized approach is LASSO. One of the reasons we decided to choose this algorithm, is because of its interpretability: thanks to the feature selection property, lasso regression can provide a more interpretable model by identifying and emphasizing the most important predictors. Moreover, Lasso regression tends to select one feature from a group of highly correlated features and sets the coefficients of the remaining features in the group to zero. This makes lasso regression less stable when dealing with multicollinear predictors.

```
## [1] 0.01974512
```

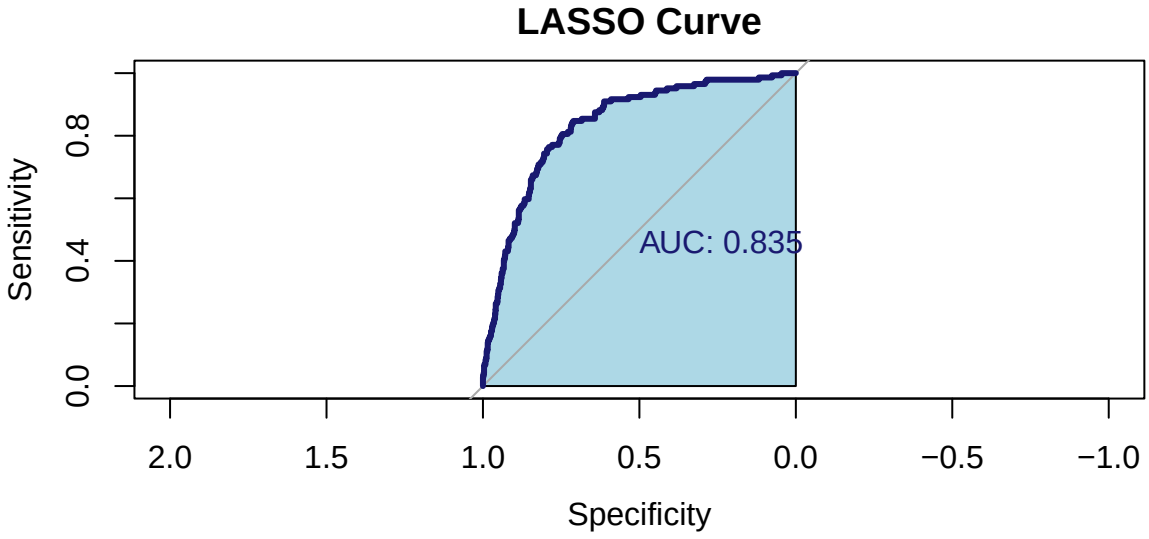
We have found that the best lambda value is 0.0197, then we obtain the the final model produced by the optimal lambda value. The best lambda is that parameter's value with based on the one-standard-error rule, which states that that you should choose the most regularized model (highest lambda value) within one standard error of the minimum cross-validated error. Through the best model, we obtain the predictions for Churn.




```
## Setting levels: control = False, case = True
```

```
## Warning in roc.default(test_y, y_predictions, plot = TRUE, col =
## "midnightblue", : Deprecated use a matrix as predictor. Unexpected results may
## be produced, please pass a numeric vector.
```

```
## Setting direction: controls < cases
```



```
## Area under the curve: 0.835
```

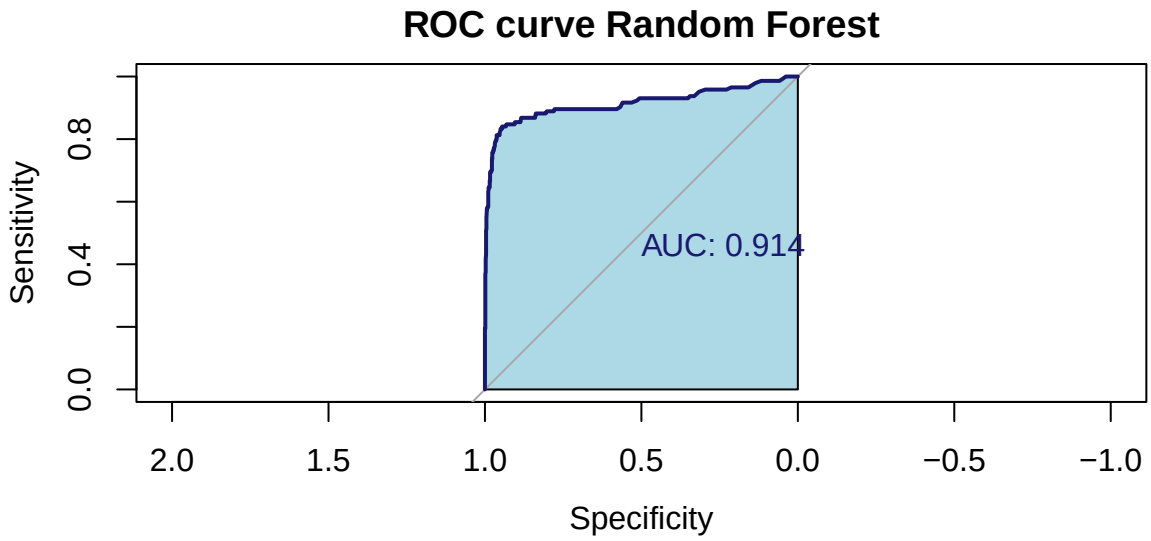
Therefore, an AUC of 0.835 indicates that the LASSO model has achieved a reasonably high level of discrimination power in distinguishing between the positive and negative classes.

To predict customer churn with non-linear models we focus on tree-based model. Firstly, we perform a Decision Tree model, which thanks to its transparency they are to understand and interpret. However, in our case Random Forest is more suitable. It is superior to a single Decision Tree for customer churn prediction in classification tasks because it reduces overfitting, handles outliers and noisy data better, provides feature importance estimation (which decision tree does not provide), is more suitable for high-dimensional data, and benefits from ensemble learning. Random Forest is an advanced non-parametric prediction model. It does not make assumptions about the data distribution and can handle skewed data and categorical data. It is generally considered a very accurate predictive model and it can handle a large number of independent predictor variables. Firstly, we build the model considering as the predictors all the variables. We perform the model on the train set. The relative AUC score is 0.915, which a very high result in real life cases.

```
## Setting levels: control = False, case = True
```

```
## Setting direction: controls < cases
```

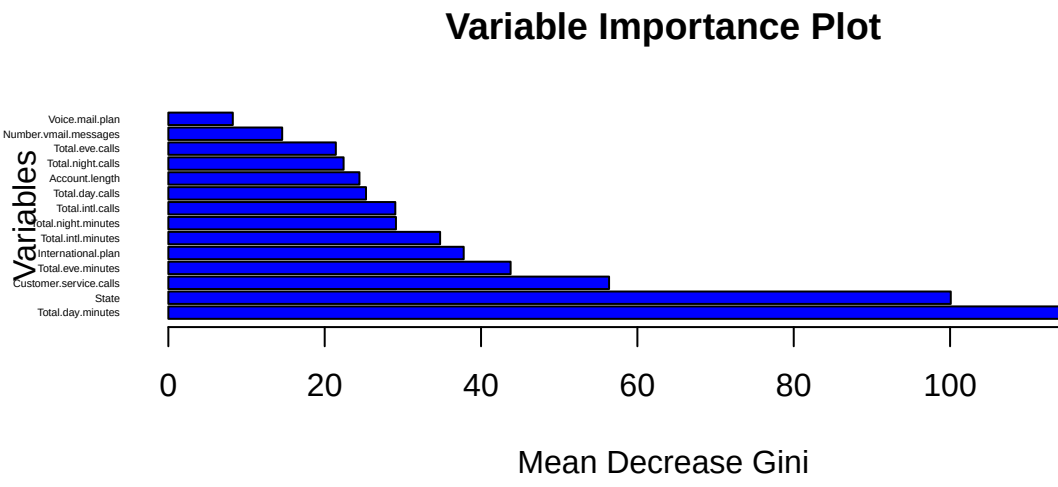
```
## Area under the curve: 0.9141
```



It is significant to notice that in this model the variable importance enables capturing synergistic effects, detecting feature interactions, identifying redundancy, incorporating contextual relevance, and ensuring model stability and robustness. By evaluating variables collectively, the model can better understand the complex relationships between features and make more accurate predictions. So, we decide to build a second Random Forest model, specifying the selected variables: Total.day/eve/intl/night.minutes,State, Customer.service.calls, International.plan, Total.intl/day/night.calls and Account.length. We calculate the Mean Decrease Gini with a sorted way in order to find these variables. The Gini index is important in customer churn prediction as it guides the random forest construction process, determines feature importance, and serves as an evaluation metric for model performance. It facilitates the identification of informative features, enables the creation of accurate and robust predictive models, and provides a measure of the model's ability to discriminate between churners and non-churners.

```
##                               False       True MeanDecreaseAccuracy
```

```
## Total.day.minutes      54.94617242 70.2916012          72.1974366
## State                  -0.43381665  4.3862505           1.8658572
## Customer.service.calls 39.02650288 55.1753512          53.7527163
## Total.eve.minutes      13.56986288 26.1347474          23.8221826
## International.plan     35.04171106 45.1276870          45.1771478
## Total.intl.minutes      9.95108748 21.4912352          18.0184830
## Total.night.minutes     2.38103997  6.6497836           5.3691723
## Total.intl.calls       14.14040406 21.8044851          21.3381212
## Total.day.calls         1.21847395 -0.3879550           0.9440193
## Account.length         -0.88886292  0.7246851          -0.4646901
## Total.night.calls      -0.72636926 -0.9052755          -1.0354920
## Total.eve.calls        -0.03474802 -0.4070208          -0.2837999
## Number.vmail.messages   9.47317741 16.2481473          14.6912335
## Voice.mail.plan         8.89274675 16.0613958          14.8287210
##                        MeanDecreaseGini
## Total.day.minutes            127.908987
## State                        100.086938
## Customer.service.calls        56.375390
## Total.eve.minutes             43.781644
## International.plan            37.780930
## Total.intl.minutes            34.764192
## Total.night.minutes           29.114924
## Total.intl.calls              29.030974
## Total.day.calls               25.283187
## Account.length                24.445531
## Total.night.calls             22.417345
## Total.eve.calls               21.425480
## Number.vmail.messages         14.561151
## Voice.mail.plan                8.243415
```

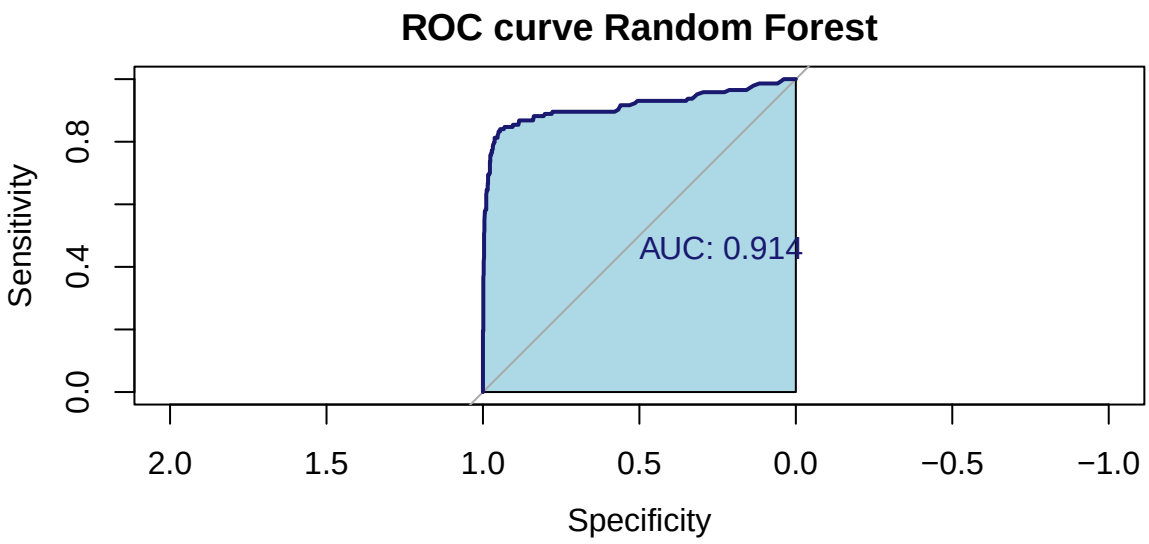


We plot the importance variable.

```
## Setting levels: control = False, case = True
```

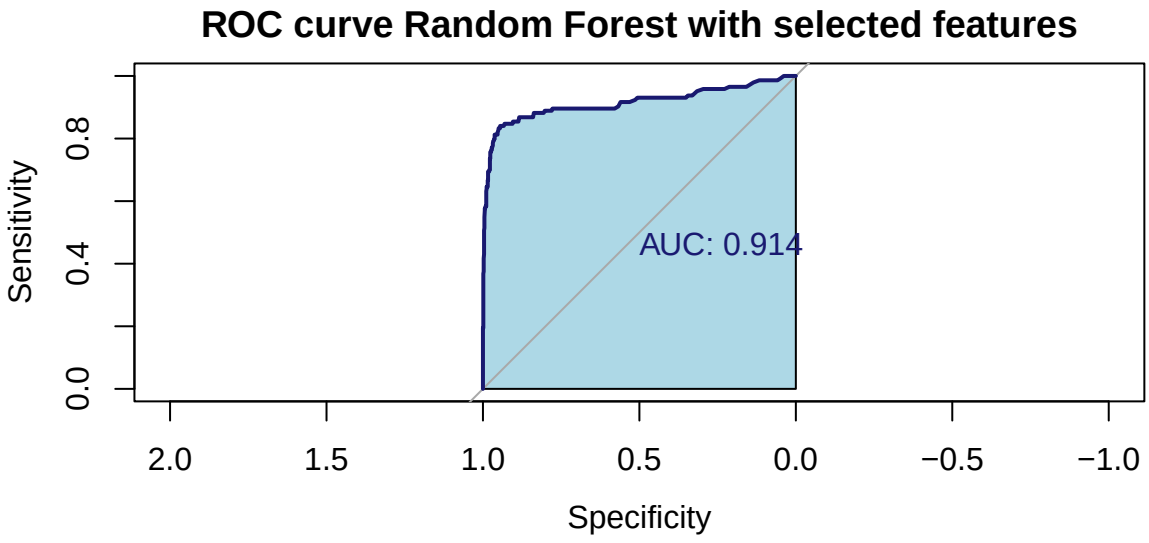
```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9141
```




```
## Setting levels: control = False, case = True
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9039
```



However, we notice the AUC scores of the two models are the same, meaning that the variables we have chosen before are already highly descriptive.

To evaluate the performance of Random Forest we also use the Out-of-Bag error, an important metric. It provides an estimate of prediction accuracy, helps in avoiding overfitting, supports model selection and hyperparameter tuning, and assists in assessing variable importance. It is an important metric for evaluating the performance and reliability of the Random Forest model in churn prediction tasks. In the first built model our OOB rate is 6%. Whereas, in the second model with the selected feature this rate increases and it becomes the 8%.

```
## [1] 0.06653815
```

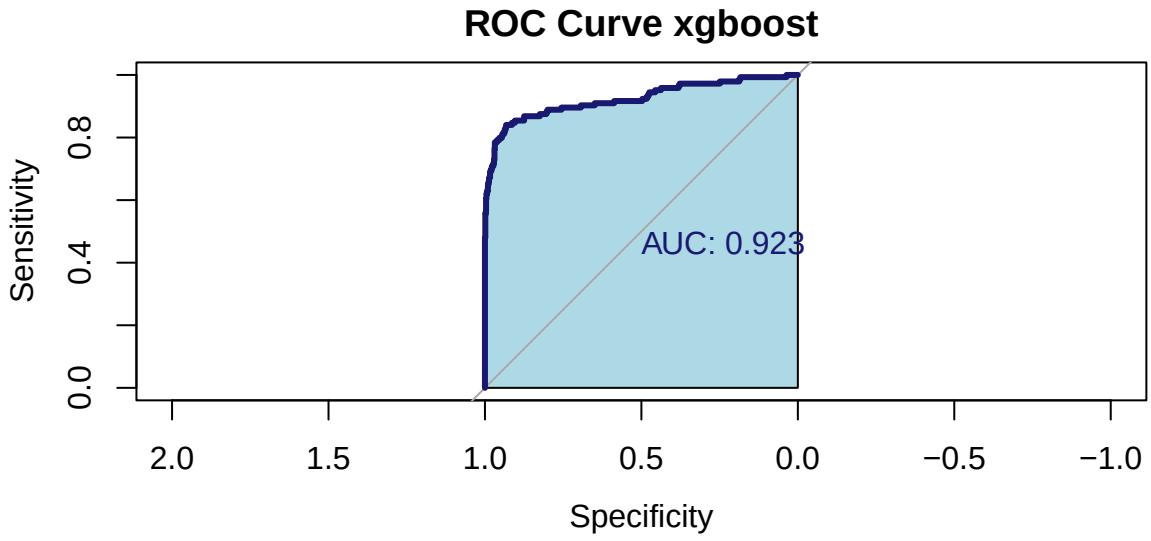
```
## [1] 0.07899142
```

Finally, another model is implemented: XGBoost, which stands for Extreme Gradient Boosting. It is a scalable, distributed gradient-boosted decision tree (DGBDT). It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems. It is a supervised machine learning method which is based on decision trees and improves on other methods such as random forest and gradient boost. Due to its nature, has a low interpretability but, on the other hand, it can give really great results To implement this model our binary variable we need to predict, must be numeric, not factor. However we didn't use as.integer(as.logical(train$Churn)) because it yields Nas values. In order to not change permanently our train and test dataset we will copy them, and use these copies for this model. Basically, we choose this model because it is good option for unbalanced dataset, such as in our case.

As we expected it yields the highest AUC score, which is 0.923.

```
## Setting levels: control = False, case = True
```

```
## Setting direction: controls < cases
```



Task 6: Conclusions

In conclusion, based on our analysis, linear models are not suitable for our data as there is no evidence of a linear relationship between the features and the target variable. Therefore, non-linear models are more appropriate for our churn prediction task. Among the models evaluated, XGBoost demonstrates the best performance in terms of AUC. It effectively handles complex patterns and provides accurate predictions.

However, it is worth noting that Random Forest outperformed all other models in terms of interpretability and variable selection. By examining the correlation matrix and distribution of the data, we observed that the target variable is highly dependent on a few key variables such as total day minutes, state, customer service calls, total eve minutes, and international plan.

Based on these findings, we recommend that telecommunications companies pay attention to customers who spend significant amounts of time on calls, frequently contact customer service, and have an international plan. These customers are more likely to churn, and therefore, companies should target them in their marketing campaigns. Additionally, improving customer service and offering competitive tariffs for international plans may help in retaining these customers and reducing churn rates.

Overall, the combination of the XGBoost model's predictive accuracy and the insights gained from Random Forest's interpretability and variable selection provides valuable guidance for telecom companies to effectively address customer churn and implement targeted strategies for customer retention.

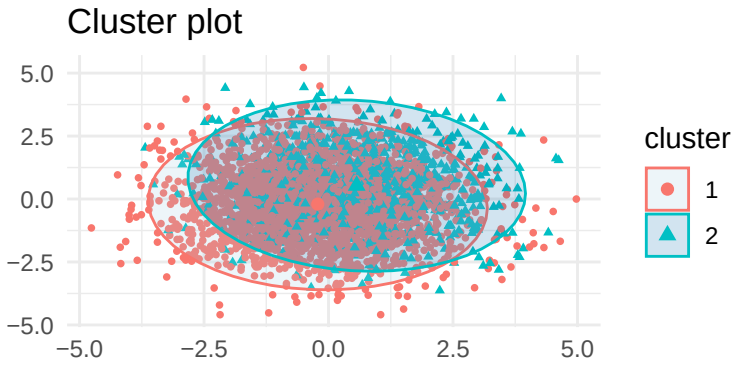
Task 7: Clustering

In this part we attempt to cluster customers,which could help in dividing customers into distinct groups based on their behavior. This segmentation allows businesses to understand different customer profiles and tailor their marketing strategies, products, and services accordingly. However, we didn't work with all variables of the dataset but we decided to omit the ones that we already ommitted for the classification task, plus the Churn variable. This because Clustering is an unsupervised learning technique that focuses on identifying natural groupings or patterns within the data without any reference to a specific target variable. Although we will see how this variable is distributed in the various clusters.

We also considered correct to scale the dataframe to work with more homogeneous data.

We started with the clustering task by using a simple K-Means (Euclidean distance based). Firstly, we found the optimal number of clusters with cross validation using the Silhouette-score and WSS method, and them performed the actual K-Means algorithm, storing the clusters label in our initial dataset TChurn.

This is the clusters that K-Means found:



At this point we implemented the Hierarchical methods,(finding new optimal number of clusters) to see if they perform better. In particular, we firstly used the distance based Hierarchical clustering (using Euclidean distance), using three different linkage methods: Ward, Complete and Single. Then we also used the correlation based Hierarchical clustering, using the same methods as before. Notice that we didn't actually run the dendrograms, since they were quite difficult to interpret because we are dealing with a lot of data. Finally, as we did before, we stored the clusters' labels in the TChurn dataframe. Here is an example of how the algorithm is implemented:

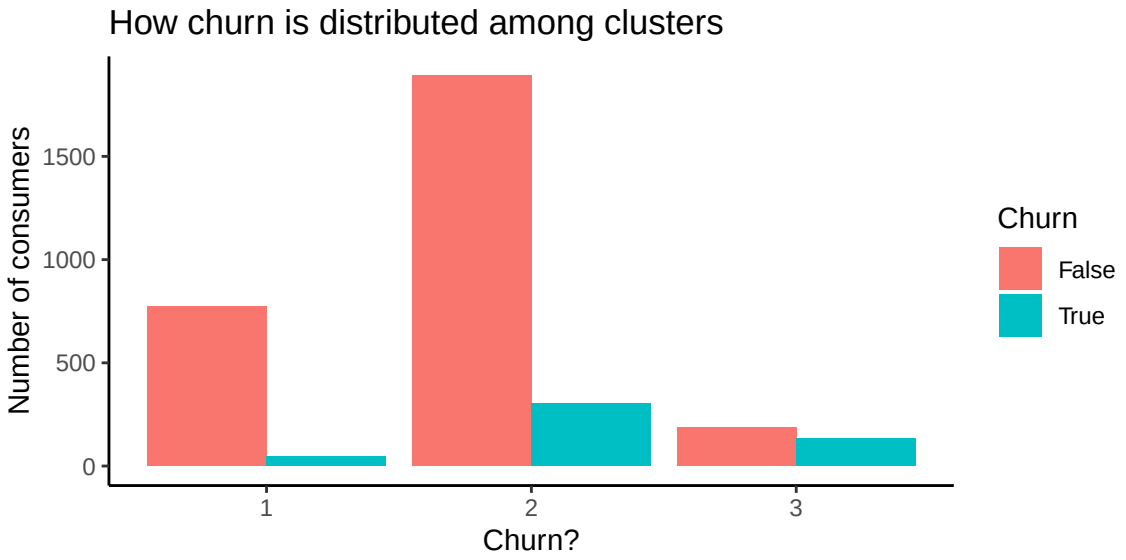
```
hc_ward <- factoextra::hcut(x = dist_eucl,
                            k = 3,
                            hc_method = "ward.D2")
#h3=factoextra::fviz_dend(x = hc_ward)
TChurn$hc_ward= as.integer(cutree(hc_ward, k=3))
```

Now that we have computed different clusters with different techniques we analyzed the one that best suits our data. We decided to evaluate the various clustering methods using the Average Silhouette-score, which is a widely used metric for evaluating clustering results due to its simplicity, interpretability, unsupervised nature, and ability to measure both cohesion and separation of clusters. After computing the Average Silhouette-score for all clusters methods (omitting the 'single' methods because we realized they did not

perform well from the clusters visualization), we concluded that the algorithm that performed better is the distance based Hierarchical clustering using the Ward method.

Let's see the clusters size:

Finally we also analyzed how Churn variable is distributed among clusters,to see any similarities. To achieve this, we performed graphs and tables to see the relation for all the segmentations that worked well. For example, here is how churn is distributed among the clusters found with our best model:



As the Adjusted Rand Index confirm (ARI measures the similarity between two clustering solutions) the Hierarchical clustering solution does not effectively capture the underlying clustering structure of the data in relation to the churn labels, but may rely more on other variables.

```
adjustedRandIndex(TChurn$hc_ward, TChurn$Churn)
```

```
## [1] 0.04161516
```