Telecommunications project

Ulises Bonilla

28/5/2021

Introduction

Any business wants to maximize the number of customers. To achieve this goal, it is important not only to try to attract new ones, but also to retain existing ones. Retaining a client will cost the company less than attracting a new one. In addition, a new client may be weakly interested in business services and it will be difficult to work with him, while old clients already have the necessary data on interaction with the service.

The Churn rate (sometimes called attrition rate), in its broadest sense, is a measure of the number of individuals or items moving out of a collective group over a specific period. Accordingly, by predicting the churn, a firm can react in time and try to keep the client who wants to leave. Based on the data about the services that the client uses, the firm can make him a special offer, trying to change his decision to leave the operator. This will make the task of retention easier to implement than the task of attracting new users, about which the firm does not know anything yet.

The data

The `Telecom users dataset` provided by `Radmir Zosimov`is publicly available at https://www.kaggle.com/radmirzosimov/telecom-users-dataset. The data contains information about almost six thousand users, their demographic characteristics, the services they use, the duration of using the operator's services, the method of payment, and the amount of payment. The data can be either downloaded manually or using Kaggle´s API. I proceed with the latter.

The data contains 5986 rows and 21 columns. According to the information provided by the source, the columns are:

1. `customerID` - customer id
2. `gender` - client gender (male / female)
3. `SeniorCitizen` - is the client retired (1, 0)
4. `Partner` - is the client married (Yes, No)
5. `tenure` - how many months a person has been a client of the company
6. `PhoneService` - is the telephone service connected (Yes, No)
7. `MultipleLines` - are multiple phone lines connected (Yes, No, No phone service)
8. `InternetService` - client's Internet service provider (DSL, Fiber optic, No)
9. `OnlineSecurity` - is the online security service connected (Yes, No, No internet service)
10. `OnlineBackup` - is the online backup service activated (Yes, No, No internet service)
11. `DeviceProtection` - does the client have equipment insurance (Yes, No, No internet service)
12. `TechSupport` - is the technical support service connected (Yes, No, No internet service)
13. `StreamingTV` - is the streaming TV service connected (Yes, No, No internet service)
14. `StreamingMovies` - is the streaming cinema service activated (Yes, No, No internet service)
15. `Contract` - type of customer contract (Month-to-month, One year, Two year)
16. `PaperlessBilling` - whether the client uses paperless billing (Yes, No)

17. `PaymentMethod` - payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
18. `MonthlyCharges` - current monthly payment
19. `TotalCharges` - the total amount that the client paid for the services for the entire time
20. `Churn` - whether there was a churn (Yes or No)

Goal

Given these variables, the goal of this project is to predict the variable `Churn`, that is, whether the client remains or not.

Analysis

Data Cleaning and exploration

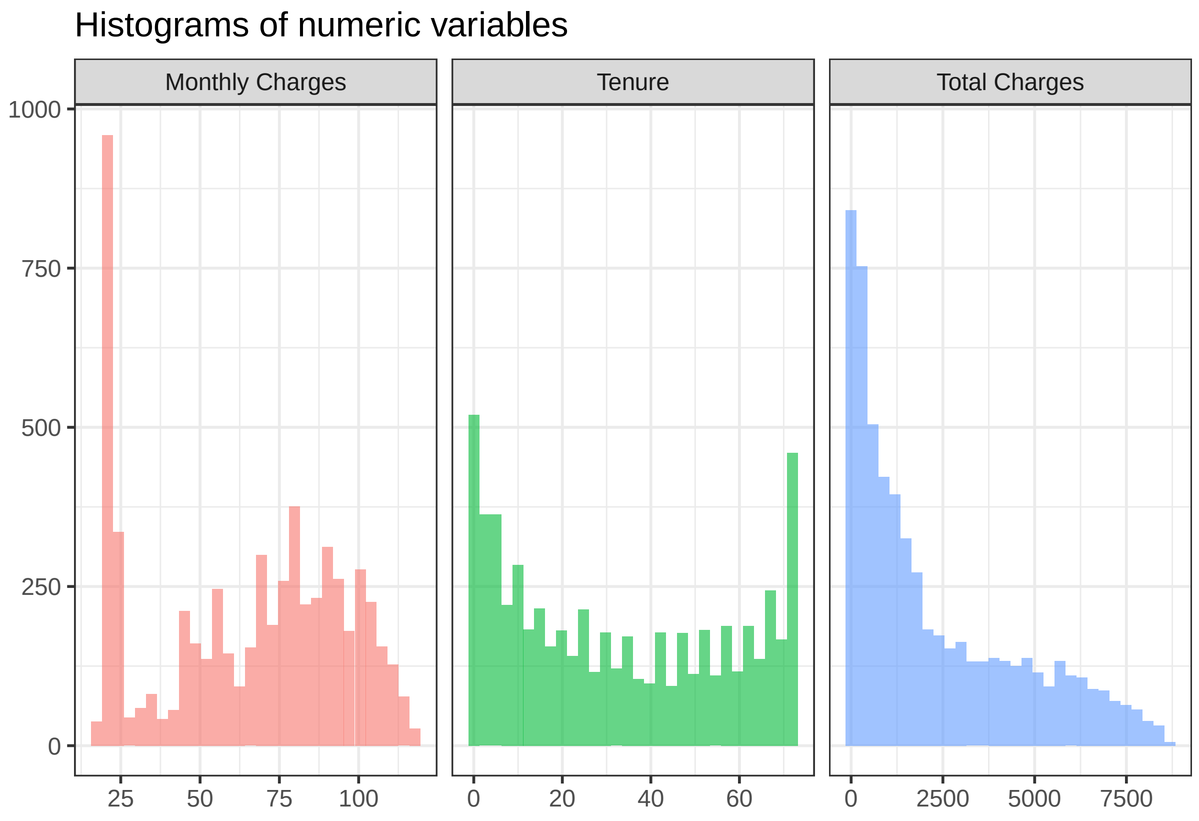
The following first table summarizes the class of each column, as well as how many NAs each contains.

Column name	Class	Number of NAs	Number of unique values
customerID	factor	0	5986
gender	factor	0	2
SeniorCitizen	integer	0	2
Partner	factor	0	2
Dependents	factor	0	2
tenure	integer	0	73
PhoneService	factor	0	2
MultipleLines	factor	0	3
InternetService	factor	0	3
OnlineSecurity	factor	0	3
OnlineBackup	factor	0	3
DeviceProtection	factor	0	3
TechSupport	factor	0	3
StreamingTV	factor	0	3
StreamingMovies	factor	0	3
Contract	factor	0	3
PaperlessBilling	factor	0	2
PaymentMethod	factor	0	4
MonthlyCharges	numeric	0	1526
TotalCharges	numeric	10	5611
Churn	factor	0	2

As observed in the table above, there are 4 columns with numeric values and 17 columns with character values. The character columns are categorical variables or factors. However, it is evident that the column `SeniorCitizen` is a categorical value as well because it has only 2 unique values. Moreover, notice that the column `Churn`, which identifies whether there was a churn or not, is categorical. Finally, notice that only the column `TotalCharges` contains NAs. So, i first have to apply a few changes to the data.

I start from the last observation. Recall that the variable `TotalCharges` contains 10 NAs which represents only the 0.17% of the total data. Yet, it may be worth it to review them. In this regard, it is noticed that the clients whose `TotalCharges` are NAs are also those for which their `tenure` is zero. Thus, the actual value should be zero, since they have not paid a single charge yet. This is thus adjusted.

Next, i explore the numerical variables. The following diagrams are histograms of the numeric variables, excluding `SeniorCitizen`.

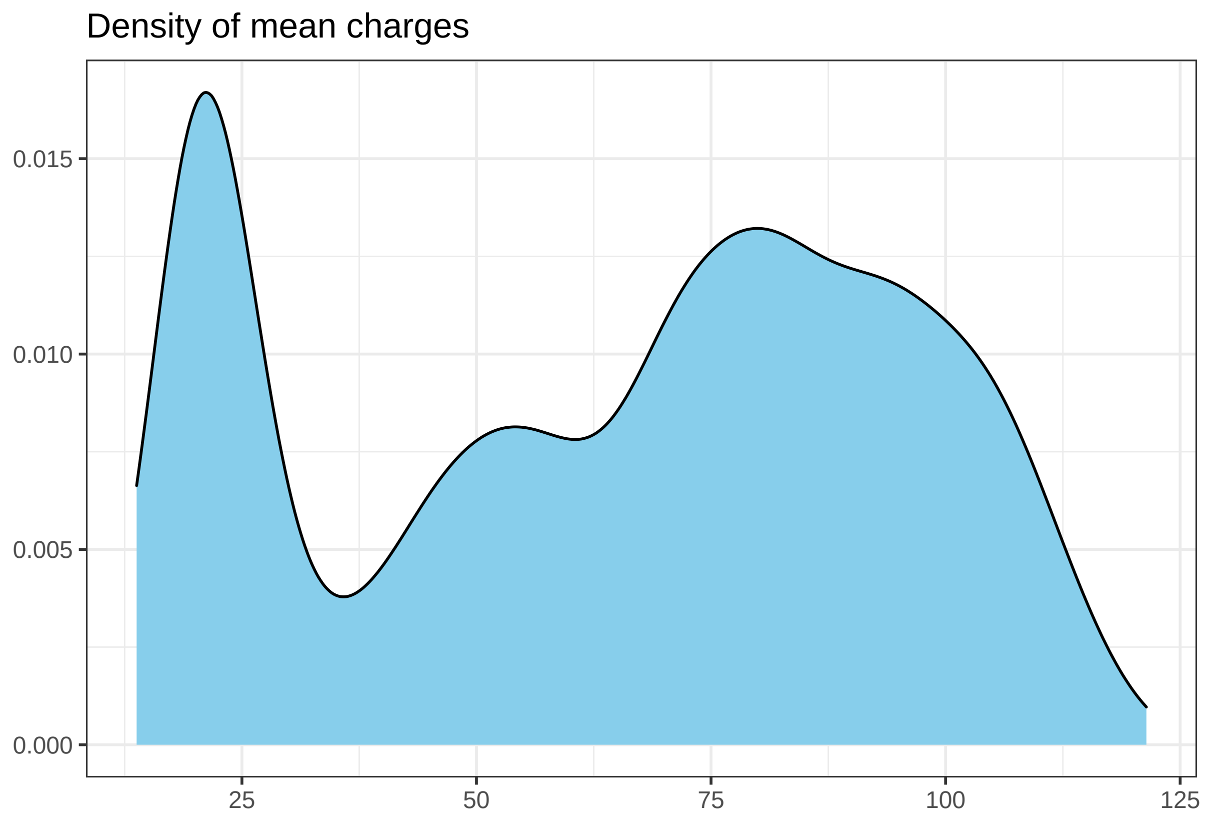


Column name	Mean	Maximum	Minumum	Standard deviation
MonthlyCharges	64.80	118.75	18.25	30.11
tenure	32.47	72.00	0.00	24.52
TotalCharges	2,294.22	8,684.80	0.00	2,274.16

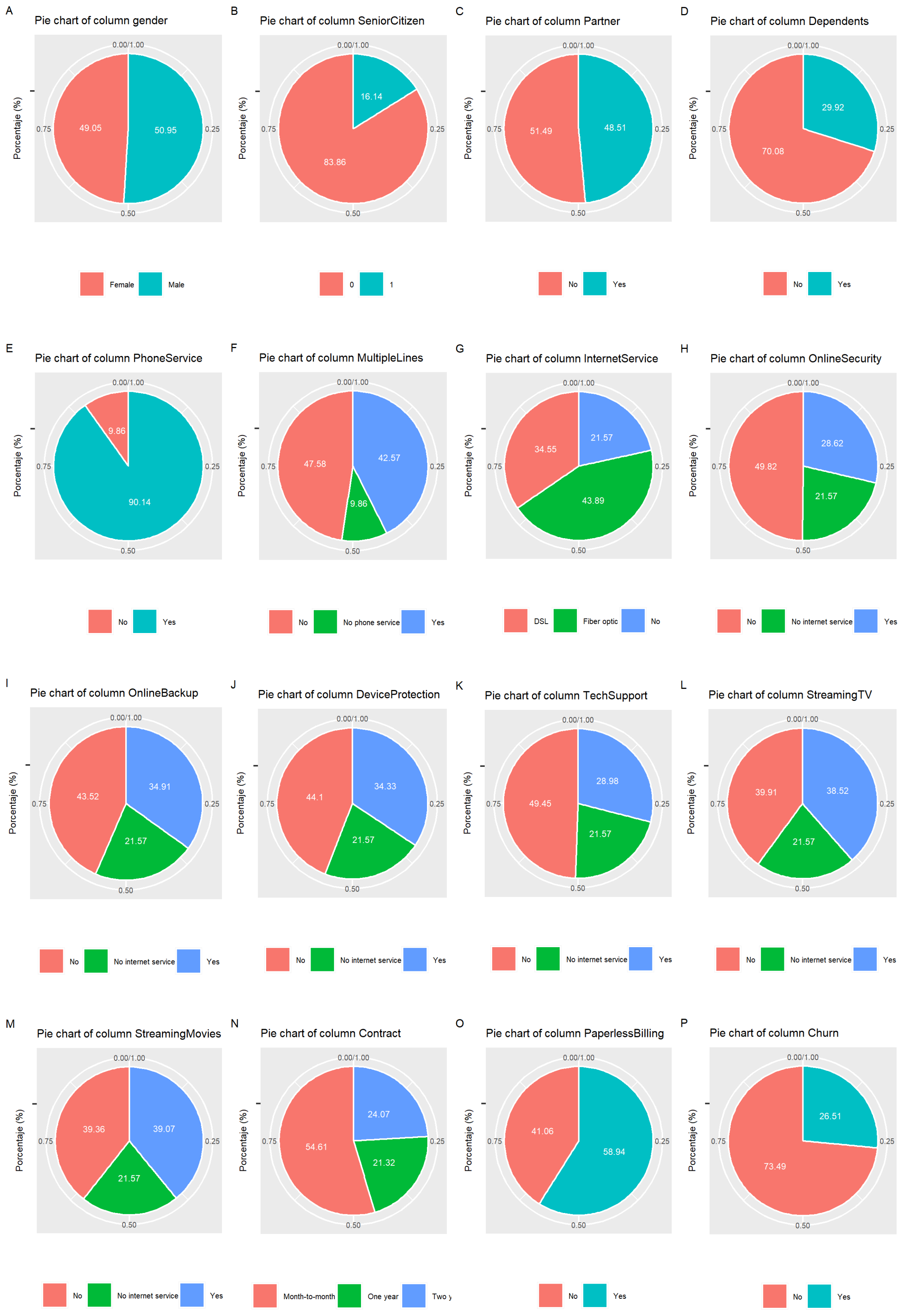
	tenure	MonthlyCharges	TotalCharges
tenure	1.0000000	0.2569833	0.8277555
MonthlyCharges	0.2569833	1.0000000	0.6567617
TotalCharges	0.8277555	0.6567617	1.0000000

The problem with these three variables is that they do not necessarily provide an accurate overview of the data. The `tenure` variable indicates that the mean number of months is 32.47, but its standard deviation is 24.52. This means that the time of contract varies considerably. Because `TotalCharges` highly depends on `tenure`, the former is also misleading. And lastly, `MonthlyCharges` is also missleading since it captures the payment due on the last month of the data. Thus, it may be appropriate to use a better variable, such as `MeanCharges`, which is the average of `TotalCharges` over the `tenure`.

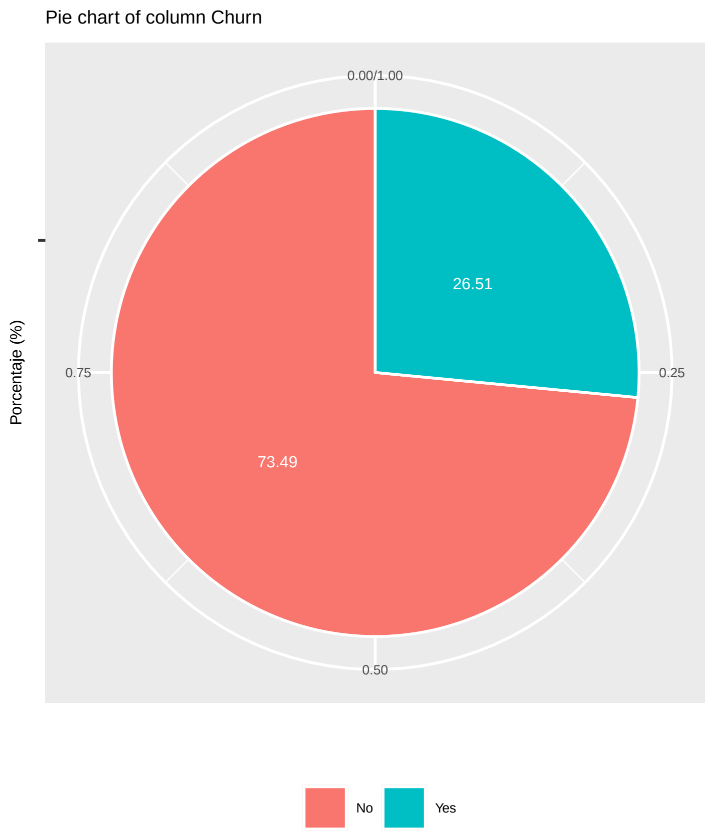
To illustrate the latter, i present the following graph, which contains the density of the mean charges. It is observa that it is similar to the histogram of `MonthlyCharges`. There are two main humps in the distribution. This could suggest a clear differentiation across clients, which could help to improve the predictability power of certain models.



Next, i explore the rest of the variables, which are all categorical, including `Churn`. The following collection of pie charts show the categories present in each variable and their proportions.


A
Pie chart of column gender
49.05
50.95
Female
Male
B
Pie chart of column SeniorCitizen
16.14
83.86
0
1
C
Pie chart of column Partner
51.49
48.51
No
Yes
D
Pie chart of column Dependents
29.92
70.08
No
Yes
E
Pie chart of column PhoneService
9.86
90.14
No
Yes
F
Pie chart of column MultipleLines
47.58
42.57
9.86
No
No phone service
Yes
G
Pie chart of column InternetService
34.55
21.57
43.89
DSL
Fiber optic
No
H
Pie chart of column OnlineSecurity
49.82
28.62
21.57
No
No internet service
Yes
I
Pie chart of column OnlineBackup
43.52
34.91
21.57
No
No internet service
Yes
J
Pie chart of column DeviceProtection
44.1
34.33
21.57
No
No internet service
Yes
K
Pie chart of column TechSupport
49.45
28.98
21.57
No
No internet service
Yes
L
Pie chart of column StreamingTV
39.91
38.52
21.57
No
No internet service
Yes
M
Pie chart of column StreamingMovies
39.36
39.07
21.57
No
No internet service
Yes
N
Pie chart of column Contract
54.61
24.07
21.32
Month-to-month
One year
Two y
O
Pie chart of column PaperlessBilling
41.06
58.94
No
Yes
P
Pie chart of column Churn
26.51
73.49
No
Yes
Porcentaje (%)
0.00/1.00
0.25
0.50
0.75

The most important of the pie charts above is the last one because it is the chart of the independent variable: `Churn`. In fact it is worth it to replicate such chart in a larger scale.



As observed above, 73.49 of `Churn` are `No`. This means that the independent variable is unbalanced. An unbalanced dataset will bias the prediction model towards the more common class.

Models´ implementation

Because the `Churn` column is categorical a classification supervised method would be appropriate to analyze this dataset. There are several ML models that I could use. I have chosen: Logistic regression, Naive bayes, Neural networks, Random forests, and SVM.

The first step for this analysis will be to split the data between training and test set. Afterwards, there are several candidate models of classification that are considered. These include: a. Support Vector Machines, b. Naive Bayes c. Random Forest, d. Neural Networks, and e. Logistic regression. When needed, calibration and selection methods are also implemented. Finally, the evaluation of the models will be done comparing the confusion matrix and the accuracy of each of the models.

Before running the proposed models, there are a few pre-processing steps needed. First, I have to deal with the fact that most of the features are categorical variables. The `caret` package includes several functions to pre-process the predictor data. It assumes that all of the data are numeric, with the exception of the target variable `Chrun`, which should be a factor. Second, i have to split the data in a training and a testing set. And finally, third, I have to specify the train control settings that will be used in the models. The structure of the resulting data frame is the following:

```
## 'data.frame':    5986 obs. of  46 variables:
##  $ gender.Female                          : num  0 1 1 0 0 1 1 1 0 0 ...
##  $ gender.Male                           : num  1 0 0 1 1 0 0 0 1 1 ...
##  $ SeniorCitizen.0                       : num  1 1 0 1 1 1 1 1 1 0 ...
##  $ SeniorCitizen.1                       : num  0 0 1 0 0 0 0 0 0 1 ...
##  $ Partner.No                            : num  0 1 0 1 1 0 1 1 1 1 ...
##  $ Partner.Yes                           : num  1 0 1 0 0 1 0 0 0 0 ...
##  $ Dependents.No                         : num  0 1 1 1 1 1 1 1 1 1 ...
##  $ Dependents.Yes                        : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ tenure                                : num  72 44 38 4 2 70 33 1 39 ..
##  $ PhoneService.No                       : num  0 0 0 0 0 1 0 1 1 0 ...
##  $ PhoneService.Yes                      : num  1 1 1 1 1 0 1 0 0 1 ...
##  $ MultipleLines.No                      : num  0 1 0 1 1 0 0 0 0 0 ...
##  $ MultipleLines.No.phone.service        : num  0 0 0 0 0 1 0 1 1 0 ...
##  $ MultipleLines.Yes                     : num  1 0 1 0 0 0 1 0 0 1 ...
##  $ InternetService.DSL                   : num  0 0 0 1 1 1 0 1 1 0 ...
##  $ InternetService.Fiber.optic           : num  0 1 1 0 0 0 1 0 0 1 ...
##  $ InternetService.No                    : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ OnlineSecurity.No                     : num  0 1 1 1 0 0 0 1 1 0 ...
##  $ OnlineSecurity.No.internet.service    : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ OnlineSecurity.Yes                    : num  0 0 0 0 1 1 1 0 0 1 ...
##  $ OnlineBackup.No                       : num  0 0 1 1 1 1 1 1 1 0 ...
##  $ OnlineBackup.No.internet.service      : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ OnlineBackup.Yes                      : num  0 1 0 0 0 0 0 0 0 1 ...
##  $ DeviceProtection.No                   : num  0 0 1 1 0 0 1 1 0 0 ...
##  $ DeviceProtection.No.internet.service  : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ DeviceProtection.Yes                  : num  0 1 0 0 1 1 0 0 1 1 ...
##  $ TechSupport.No                        : num  0 1 1 1 1 0 1 1 0 0 ...
##  $ TechSupport.No.internet.service       : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ TechSupport.Yes                       : num  0 0 0 0 0 1 0 0 1 1 ...
##  $ StreamingTV.No                        : num  0 0 1 1 1 1 1 1 1 0 ...
##  $ StreamingTV.No.internet.service       : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ StreamingTV.Yes                       : num  0 1 0 0 0 0 0 0 0 1 ...
##  $ StreamingMovies.No                    : num  0 1 1 0 1 0 0 1 1 0 ...
##  $ StreamingMovies.No.internet.service   : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ StreamingMovies.Yes                   : num  0 0 0 1 0 1 1 0 0 1 ...
##  $ Contract.Month.to.month               : num  0 1 1 1 1 0 1 1 0 1 ...
##  $ Contract.One.year                     : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ Contract.Two.year                     : num  1 0 0 0 0 1 0 0 0 0 ...
##  $ PaperlessBilling.No                   : num  1 0 0 0 1 0 0 0 1 0 ...
##  $ PaperlessBilling.Yes                  : num  0 1 1 1 0 1 1 1 0 1 ...
##  $ PaymentMethod.Bank.transfer..automatic.: num  0 0 1 0 0 1 0 0 0 0 ...
##  $ PaymentMethod.Credit.card..automatic.  : num  1 1 0 0 0 0 0 0 0 0 ...
##  $ PaymentMethod.Electronic.check        : num  0 0 0 1 1 0 1 0 0 1 ...
##  $ PaymentMethod.Mailed.check            : num  0 0 0 0 0 0 0 1 1 0 ...
##  $ MeanCharges                           : num  24.1 90.3 75.5 59.6 59.8..
##  $ Churn                                 : Factor w/ 2 levels "No","Yes"..
```

The first model i implement is a linear support vector machine (svm). This is a relatively simple model in which no parameter is tuned.The trade of parameter is by default set at `C=1`.[1]

```
## Support Vector Machines with Linear Kernel
##
```

[1]It determines the possible misclassifications. It essentially imposes a penalty to the model for making an error: the higher the value of C, the less likely it is that the SVM algorithm will misclassify a point.

```
## 4790 samples
##   45 predictor
##    2 classes: 'No', 'Yes'
##
## Pre-processing: scaled (45), centered (45)
## Resampling: Cross-Validated (5 fold, repeated 3 times)
## Summary of sample sizes: 3832, 3832, 3832, 3832, 3832, 3832, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8009047  0.4476467
##
## Tuning parameter 'C' was held constant at a value of 1
```

It is noted that, despite the simplicity of this first model, the results obtained from resampling are promising. The resampling estimation of accuracy in this model is 0.8009047.[2] A second metric shown above is the kappa coefficient, which in this case equals 0.4476467.[3]

The second model I use is naive bayes. Naive Bayes is a classification model that assumes that the effect of a particular feature in a class is independent of other variables. This is also a simple model in which its parameters are also kept constant. It can be observed that both accuracy and kappa are lower than the previous model. So, i do not expect this model to be the best performer with the testing data.

```
## Naive Bayes
##
## 4790 samples
##   45 predictor
##    2 classes: 'No', 'Yes'
##
## Pre-processing: scaled (45), centered (45)
## Resampling: Cross-Validated (5 fold, repeated 3 times)
## Summary of sample sizes: 3832, 3832, 3832, 3832, 3832, 3832, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.6922756  0.3790747
##    TRUE      0.7414753  0.4274288
##
## Tuning parameter 'laplace' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were laplace = 0, usekernel = TRUE
##  and adjust = 1.
```

The third model I use is random forest.

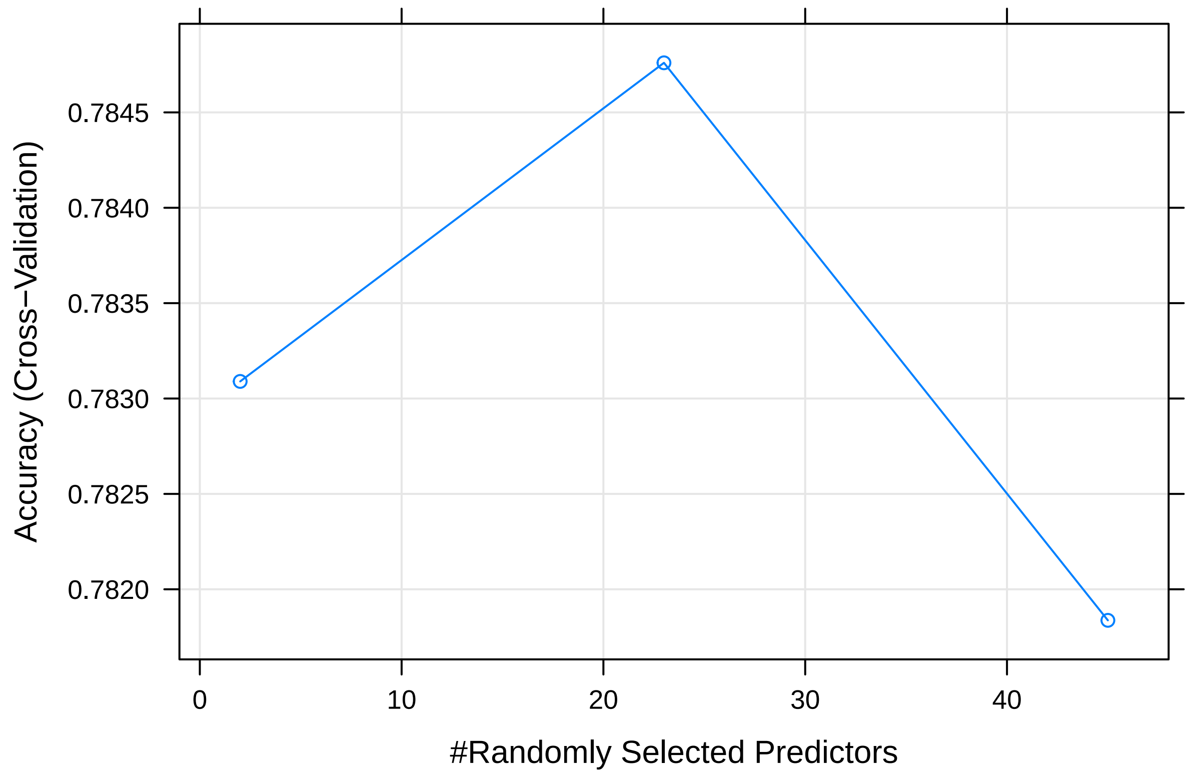
```
## Random Forest
##
## 4790 samples
```

[2] Accuracy is simply a ratio of correctly predicted observation to the total observations.

[3] Cohen's kappa coefficient is a statistic that is used to measure inter-rater reliability (and also intra-rater reliability) for qualitative (categorical) items

```
##   45 predictor
##    2 classes: 'No', 'Yes'
##
## Pre-processing: scaled (45), centered (45)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4311, 4311, 4311, 4311, 4311, 4311, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.7830898  0.3299730
##   23    0.7847599  0.4101143
##   45    0.7818372  0.4012574
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 23.
```

As observed in the results above, the best model is the one that makes use of 23 variables, with an accuracy and a kappa of 0.7847599 and 0.4101143, respectively. The graph below represents this same outcome:

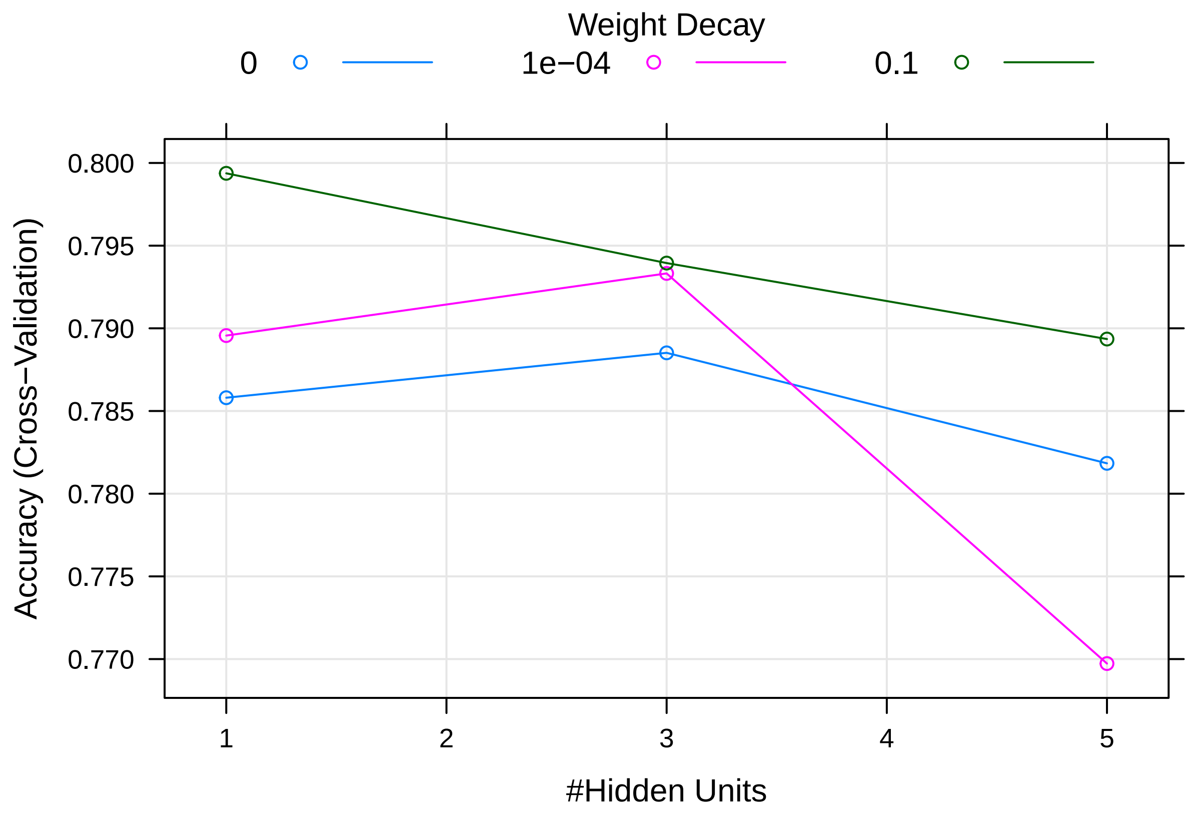


The fourth model in the repertoire is neural networks. The results of this model are presented below:

```
## Neural Network
##
## 4790 samples
##   45 predictor
##    2 classes: 'No', 'Yes'
```

```
##
## Pre-processing: scaled (45), centered (45)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4311, 4311, 4311, 4311, 4311, 4311, ...
## Resampling results across tuning parameters:
##
##   size  decay  Accuracy   Kappa
##   1     0e+00  0.7858038  0.3648875
##   1     1e-04  0.7895616  0.4033196
##   1     1e-01  0.7993737  0.4574584
##   3     0e+00  0.7885177  0.3860171
##   3     1e-04  0.7933194  0.4362927
##   3     1e-01  0.7939457  0.4316366
##   5     0e+00  0.7818372  0.4094277
##   5     1e-04  0.7697286  0.3830899
##   5     1e-01  0.7893528  0.4272860
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were size = 1 and decay = 0.1.
```

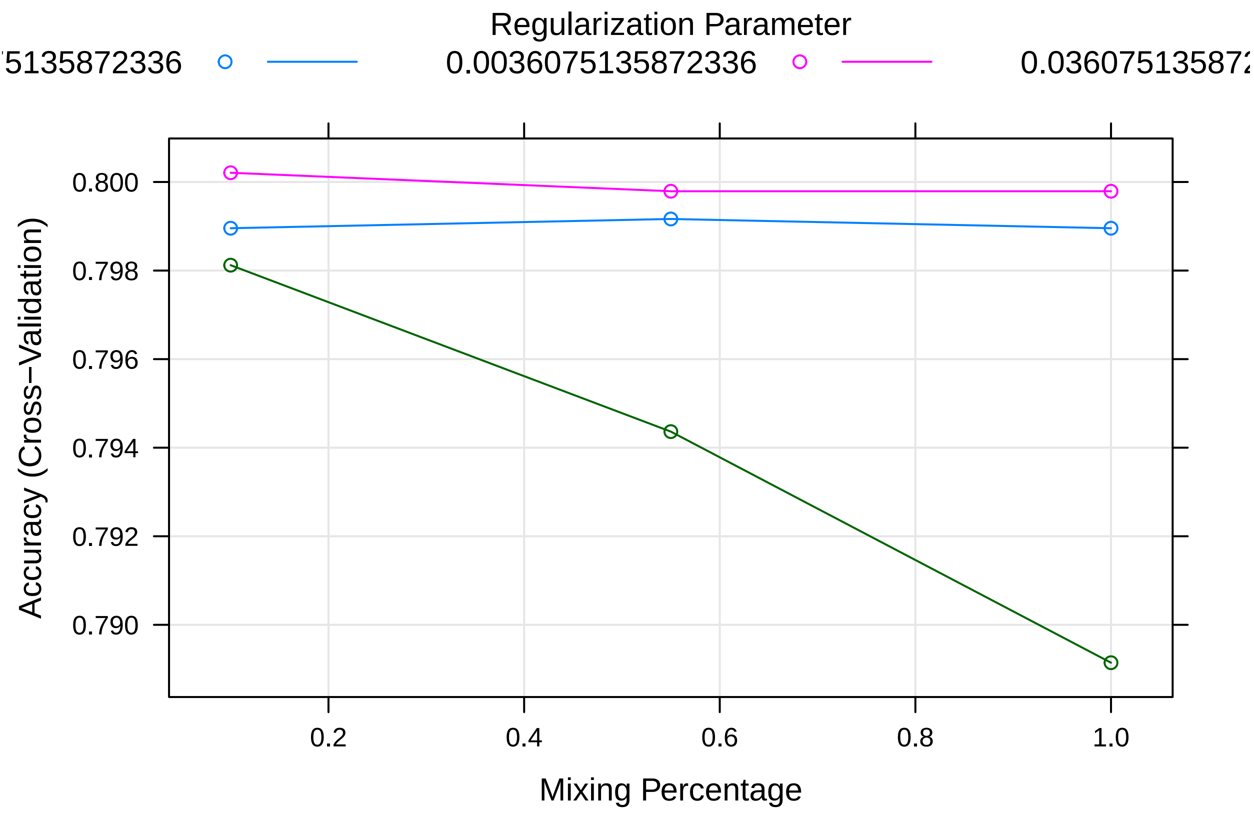
As observed, the best model is the one with size = 1 and decay = 0.1, which has an accuracy of 0.7993737 and a kappa of 0.4574584. Graphically, this is represented below:



Finally, the fourth and last model presented here is the logistic regression. The logistic regression is a classification algorithm that predicts the probability of a categorical dependent variable (`Churn`). For our case, the goal of this model is to predict `P(Y=1)` as a function of the predictors.The fact that the best neural network above resulted in a single hidden unit suggests that a simpler model such as logistic could yield a good performance.

```
## glmnet
##
## 4790 samples
##   45 predictor
##    2 classes: 'No', 'Yes'
##
## Pre-processing: scaled (45), centered (45)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4311, 4311, 4311, 4311, 4311, 4311, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda        Accuracy   Kappa
##   0.10   0.0003607514  0.7989562  0.4532360
##   0.10   0.0036075136  0.8002088  0.4555534
##   0.10   0.0360751359  0.7981211  0.4379952
##   0.55   0.0003607514  0.7991649  0.4536811
##   0.55   0.0036075136  0.7997912  0.4538449
##   0.55   0.0360751359  0.7943633  0.4036059
##   1.00   0.0003607514  0.7989562  0.4533045
##   1.00   0.0036075136  0.7997912  0.4519968
##   1.00   0.0360751359  0.7891441  0.3665653
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were alpha = 0.1 and lambda = 0.003607514.
```

As observed, the best performing model, that of accuracy 0.8002088 and kappa 0.4555534, is one of the best performing among all models considered here.



Results

Even though the resampling methods from the previous section provide an idea of the performance of the models, these still have to be tested against unseen data. In this section i compute several metrics resulting from using the models to predict the `Churn`, in the test data set.

The first table of results provide traditional metrics, including accuracy and kappa, which I have addressed before.

Metric	Logistic Regression	Naive Bayes	Neural Network	Random Forest	Support Vector Machine
Accuracy	0.7918	0.7333	0.7943	0.9540	0.7935
AccuracyLower	0.7677	0.7072	0.7703	0.9406	0.7694
AccuracyNull	0.7349	0.7349	0.7349	0.7349	0.7349
AccuracyPValue	0.0000	0.5670	0.0000	0.0000	0.0000
AccuracyUpper	0.8145	0.7582	0.8169	0.9652	0.8161
Kappa	0.4354	0.4211	0.4452	0.8804	0.4265
McnemarPValue	0.0010	0.0000	0.0027	0.1056	0.0000

As observed, the model with the highest accuracy is Random Forest, with 0.9540, which is even higher than the resampling accuracy estimated before. Likewise, Random Forest also has the highest kappa coefficient with 0.8804.

However, as mentioned before, the data is unbalanced and this fact could affect metrics such as accuracy and kappa. For this reason, other metrics are also considered in the following table:

Metric	Logistic Regression	Naive Bayes	Neural Network	Random Forest	Support Vector Machine
Balanced.Accuracy	0.7061	0.7490	0.7118	0.9344	0.6971
Detection.Prevalence	0.7793	0.5836	0.7751	0.7458	0.7977
Detection.Rate	0.6530	0.5259	0.6522	0.7174	0.6630
F1	0.8625	0.7977	0.8638	0.9689	0.8652
Neg.Pred.Value	0.6288	0.4980	0.6320	0.9309	0.6446
Pos.Pred.Value	0.8380	0.9011	0.8414	0.9619	0.8312
Precision	0.8380	0.9011	0.8414	0.9619	0.8312
Prevalence	0.7349	0.7349	0.7349	0.7349	0.7349
Recall	0.8885	0.7156	0.8874	0.9761	0.9022
Sensitivity	0.8885	0.7156	0.8874	0.9761	0.9022
Specificity	0.5237	0.7823	0.5363	0.8927	0.4921

In particular, the two metrics commonly referred in the precense of unbalance data are the balanced accuracy and the F1 rate. As observed, both metrics coincide in that, once again, the Random forest model has the best performance. This model has the highest balanced accuracy with 0.9344 and the highest f1 rate with 0.9689.

Conclusions

- I used four different ML models, Logistic regression, Naive bayes, Neural networks, Random forests, and SVM, to predict the variable `Churn` from a telecom data base found online. According to my results, the best choice is the Random forest model because it shows a better performance accross different metrics.
- Potentially, this type of exercise can be extended to other markets, besides telecom, and generate great value for the firms.
- The main limitation that i encountered was the size of the data. Now a days, a sample size such as the one that i used may not be considered robust enough. Because of this, future work would likely focus on improving the models by making use of larger data sets.