|  |
| --- |
| Technical Document  Les Clairvoyants |
| **Group members:**   * Chadi Ghosn * Kirran Kayani * Rita Starzyk * YEVGENIY DENKOVYCH |



Contents

[Project Diagram: 3](#_Toc119173347)

[ETL Process: 4](#_Toc119173348)

[Tableau: 5](#_Toc119173349)

[ML Models: 6](#_Toc119173350)

[1. Logistic Regression: 6](#_Toc119173351)

[2. K nearest neighbor: 7](#_Toc119173352)

[3. Random Forest Classifier: 7](#_Toc119173353)

[AI Selenium code: 8](#_Toc119173354)

# A picture containing diagram Description automatically generatedProject Diagram:

The project will start by ingesting a csv dataset into an ETL process for data curation. Once done, a csv of the curated data will be exported to be used with Tableau for data visualization. On the other hand, the curated data will as well be pushed to a PostgreSQL database for machine learning usage. Three different algorithms of machine learning will be applied to try and reach the highest model score available. Finally, the selenium code will contact an online website for AI, where users can input information, as well as the model accuracy achieved, and the website will return insights and percentages of customer retention as well as revenue saving.

# ETL Process:

We start by importing necessary libraries for data curation. Then we read the csv file downloaded from [Kaggle](https://www.kaggle.com/competitions/customer-churn-prediction-2020/data).

The dataset was already split into train, test and sample submission, so we had to append all of the data together to be able to control the split between training and testing data.

Afterwards, we checked that the columns did not have any null values and checked the columns data types to make sure that the int or float columns did not have any strings value that would affect the column data type.

Moreover, we mapped the churn column by replacing yes by 1 and no by 0. Then we dropped irrelevant columns like “State” and “Are code”.

Next, we changed all string values to numerical values to prepare the data for machine learning model usage. And we checked the number of categories that we have by checking the unique values in the columns. We then checked again for null values and data types to make sure data now is ready to be ingested.

Finally, we exported the data as a csv file that will be used with Tableau, and we loaded the data into a PostgreSQL database called “churn\_pred” into a fact table by the name of “fact\_churn”.

This file as well include the selenium code that we will touch on later.

# Tableau:

We created a [Tableau](https://public.tableau.com/app/profile/chadi.ghosn/viz/Project4_16680253761120/Dashboard1?publish=yes) dashboard that helps the business check for any correlation or trend in their data.

The dashboard consists of 7 visuals in total:

* The first 4 visuals are comparing the account lengths (in months) of the account holders, with the total number of minutes used by day, eve, night and international, separately.
* Then we are looking at the number of churns by the date the customers signed up for a contract with the telecom company.
* In addition, we looked at the number of customer service calls by the account length to check for any trend.
* Finally, we compared the churn by day, and the count of sign ups by day, as well as the percentage churn by day for reference.

Based on the visuals created, we were able to conclude the following:

* There is no positive or negative correlation between the total number of minutes and the account lengths.
* Between 2012 and 2014:
  + The company registered the highest number of customers.
  + These customers made the highest numbers of calls until 2020, and also the highest number of calls to customer service.
  + The number of customers who churned was also highest between mentioned years.
* The churn rate remained relatively stable for these customers (avg. 14%), thus:
  + No relationship between the number of months the customer has stayed with the company and the churn rate.

# ML Models:

## Logistic Regression:

We randomly split and scaled the data for training and testing, keeping 25% of the data for testing the model. Both training and testing scores were relatively high and close together after applying the Logistic Regression model, at 83.3% and 81.4% respectively, and a total accuracy of 81%. This was achieved through tuning the parameters of the model: ‘max\_iter’, ‘solver’, ‘penalty’, and ‘l1\_ratio’.

Text

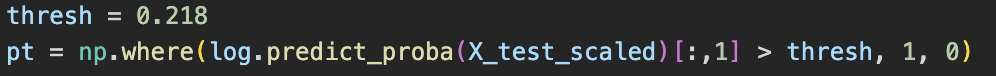
Description automatically generated

However, the proportion of False Negative results was 16.7%. These customers would churn (leave the company) while our model predicted them to stay, and so would be lost customers. In addition, only 2% of the results were True Positives (where our model correctly predicted the customers likely to churn). In comparison, the proportion of False Positive results was low at 2%; these customers are predicted to churn however would remain with the company. As it is more important for our model to predict those who leave, so the company can carry out any necessary retention strategies, we decided to optimise our model to increase its sensitivity.

Chart, treemap chart

Description automatically generated

Chart, treemap chart

Description automatically generatedThrough amending the probability threshold from the default of 0.5 to 0.218, we were able to more than halve the proportion of False Negative results to 8%, while increasing the proportion of True Positives to 11%. Although total accuracy reduced to 77% and the proportion of False Positive results increased, the benefit received from the higher sensitivity would outweigh any cost incurred from the additional False Positive customers.

## K nearest neighbor:

## Random Forest Classifier:

# AI Selenium code:

We started by importing Chrome Driver to access the browser.

Then we’ve set the chrome options to have a “headless” browser, which means the process will be hidden.

Once the hidden browser launches, it will access the URL specified and start applying the set buttons:

* It will accept the cookies based on the html XPath.
* It will ask 5 questions and temporarily store them as variables.
* Then it will find the relevant fields by XPath and fill in the fields with the respective variables.
* Finally it will click the button to process the numbers and bring back the result and print them.