Python for Data Analysis

Final Project Report

MAXENCE RAVEAU - GAËLLE RIGAUD

0. Introduction

Before starting the project, we needed to import our dataset. The dataset is called "online_shoppers_intentions" and was donated in August 2018. Thanks to the description of the dataset, we already know that it contains 12,330 rows, described by 18 variables, and that 84.5% (10,422) were negative class samples that did not end with shopping, and the rest (1908) were positive class samples ending with shopping.

To explore this dataset, we decided to install some libraries. Here are the libraries we chose than the use we had of them :

- Pandas : To store our data in a dataframe and apply functions to the whole dataframe.
- Matplotlib.pyplot : To plot our data in different ways (simple plots, bar plots, …)
- Numpy : To manipulate arrays
- Seaborn : To visualize a colorful version of the correlation matix of our dataset

After visualizing the dataset and seeking for a description of it, we succeeded in apprehend it and understand the different variables. Thus, the dataset represents shopping intentions of customers on a web site. According to the source : "The dataset was formed so that each session would belong to a different user in a 1-year period to avoid any tendency to a specific campaign, special day, user profile, or period."

We then clearly saw the 12,330 entries of the dataset and their 18 describing variables. 10 of these variables are numerical and the other 8 variables are categorical. Our target variable is the "Revenue" variable. It's a boolean variable which indicates if there was a sale or not in a particular session. Besides, it's important to note that there is no missing value in the dataset.

After that we have casted a glance to our dataset, we went further and visualized the importance of each variables in the explanation of the target.

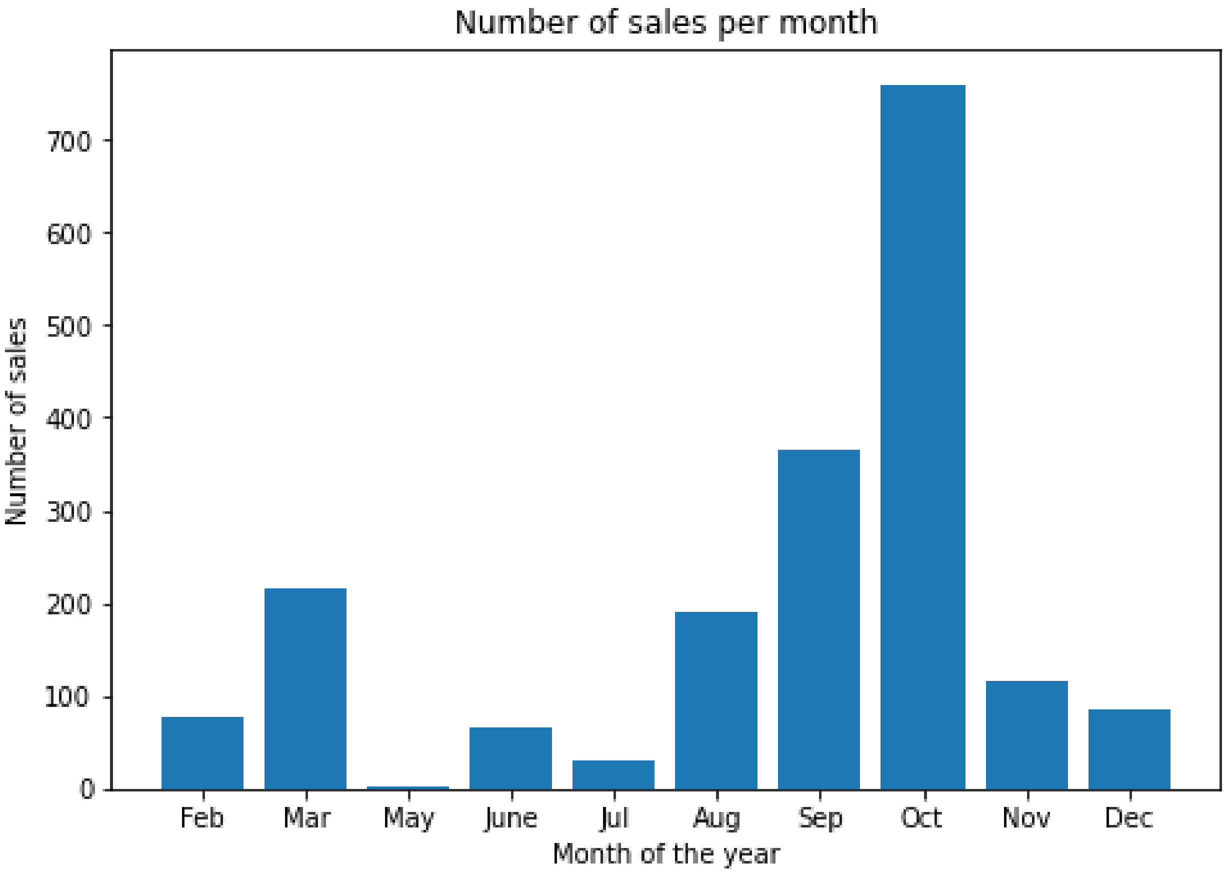
1. Data Visualization

In this part of the project, we apprehended the dataset and to visualized the link between the variables and the target, which is the "Revenue" variable. As a reminder, the "Revenue" column contains boolean values. If it is True, the consumer bought the product designated in the "ProductRelated" column. If it is False, it means that the consumer consulted the page of the product without buying it.

1.1 Number of sales

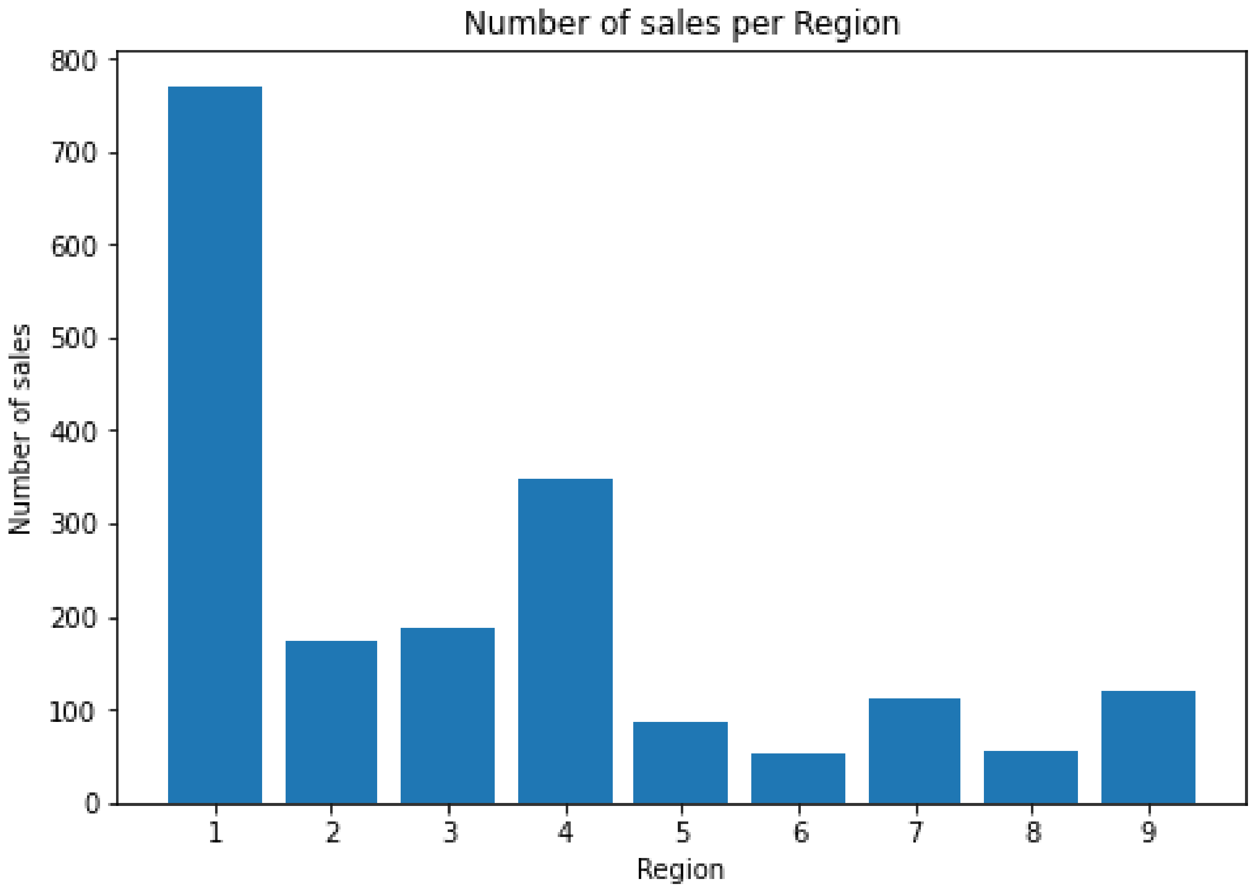
We wanted to know how many sales had been made each month. For this, we needed the list of months that exist in the dataset and the number of "True" values in the column "Revenue", grouped by month. Then, we represent the obtained values in the bar plot below.

Thanks to this plot, we could saw that a lot of sales have been made in October. At this time, we thought that the month variable would be strongly correlated to the Revenue variable.



We also wanted to know how many sales had been made in each region. For this, we used a list of the regions that exist in the dataset and we applied the same logic than previously.

Thanks to this plot, we could saw that a lot of sales have been made in region number 1. This may mean that the region variable is strongly correlated to the Revenue variable.



1.2 Never Purchased Products

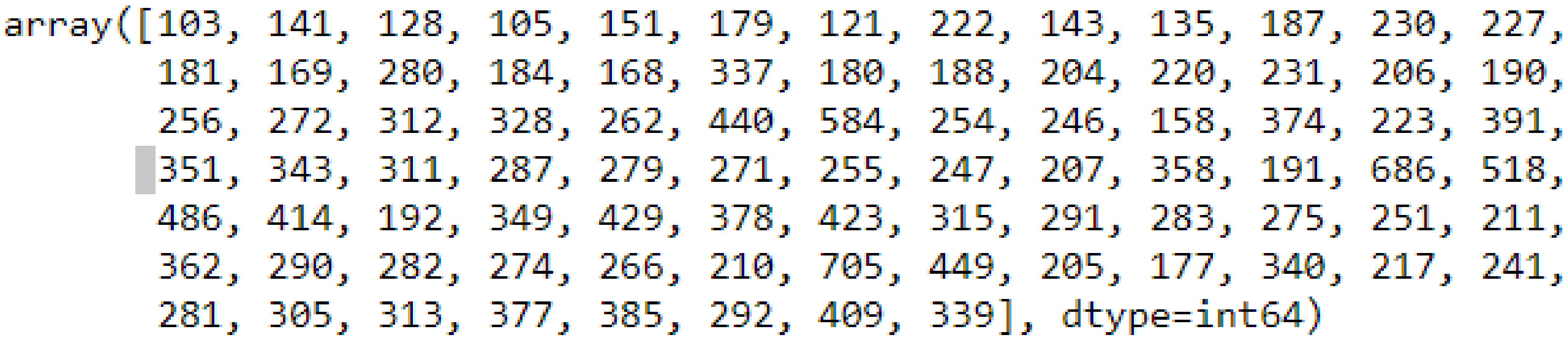
After studying the sales, we wanted to know if there are products in the dataset that are never bought. And if they existed, we also would have like to have the list of them.

To do this, we got the list of all the products there have been consulted but not purchased at least. Indeed, it correspond to the rows with a value "False" in the "Revenue" column. Thus, we have divided the dataset into two sub-datasets : the "allSales" dataset that contains all the rows concluded by a sale and "noSale" dataset that contains all the rows that didn't lead to a sale.

In order to see if there are products that are never bought, we also needed to get the list of the purchased products and the list of all the products that exist in our dataset.

Then, we verified that the list of purchased products is a subset of the list of all products. If it is True, it means that not all products of the dataset are bought, at least once

According to our program, this was True. So, because we knew that they exist, we wanted to get all these products that are never purchased. In order to have this, we created a function that go through the list containing all the products of the dataset. For each product, the function verify if the product is also in the list of the bought products. If not, the function add the product to the list of product that are never purchased. The function returns this list of 86 products.



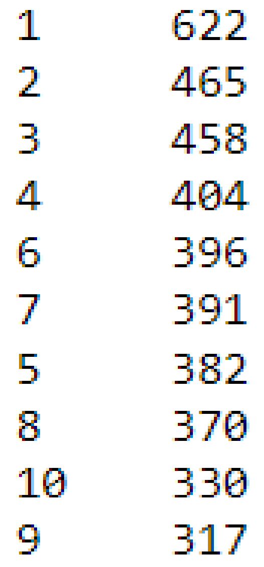
```
array([103, 141, 128, 105, 151, 179, 121, 222, 143, 135, 187, 230, 227,
       181, 169, 280, 184, 168, 337, 180, 188, 204, 220, 231, 206, 190,
       256, 272, 312, 328, 262, 440, 584, 254, 246, 158, 374, 223, 391,
       351, 343, 311, 287, 279, 271, 255, 247, 207, 358, 191, 686, 518,
       486, 414, 192, 349, 429, 378, 423, 315, 291, 283, 275, 251, 211,
       362, 290, 282, 274, 266, 210, 705, 449, 205, 177, 340, 217, 241,
       281, 305, 313, 377, 385, 292, 409, 339], dtype=int64)
```

1.3 Most Consulted and Purchased Products

After finding the products that are never bought, we wanted to see the most consulted and most purchased products.

For the most consulted products, we use directly the dataset and grouped it by the ProductPages variables. The first ranking is what came out.

For the most purchased products, we used the allSales dataset introduced before and applied a grouping by Revenue. The second ranking is what came out.

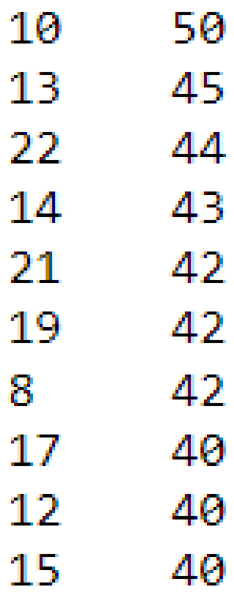


```
1       622
2       465
3       458
4       404
6       396
7       391
5       382
8       370
10      330
9       317
```

Top 10 of most consulted products

Left column : Number of the product

Right column : Number of consultation



```
10      50
13      45
22      44
14      43
21      42
19      42
8       42
17      40
12      40
15      40
```

Top 10 of the most purchased products

Left column : Number of the product

Right column : Number of sales

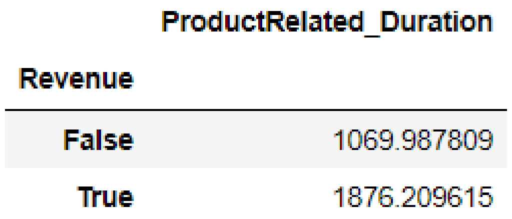
We also looked at the tops 10 of the less consulted and less bought products, but they were irrelevant. In fact, a lot of products are consulted or bought only once in the dataset.

Then, knowing that a lot of products are only consulted once, we wanted to know if these products also belongs to the list of the "never bought" products. It appeared that none of the never purchased products are in the list of the less consulted products.

At the conclusion of this subpart, we could say a lot more about the tops 10 of most consulted and most sold products. In fact, we could see that only two products, the number 8 and the number 10, are in the two lists. This means that most of the most consulted products aren't the most sold. It brings light to the fact that maybe the presentation pages of the theses products are not attractive enough for consumers or that the publicity made on these products brought them visibility but without giving the consumers the desire to buy it.

1.4 Pre-Purchase Behavior

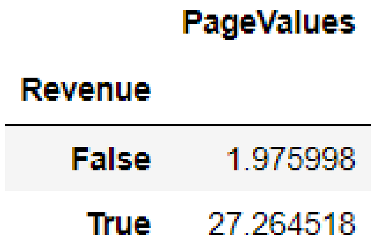
In this part, we wanted to dive further and see the relation between the time spent on a product page and the number of sales of this product.



	ProductRelated_Duration
Revenue	
False	1069.987809
True	1876.209615

Consumers spend on average more time on the page of a product they are about to buy. This shows the correlation between the time spent on a product page and the number of sales of this product.

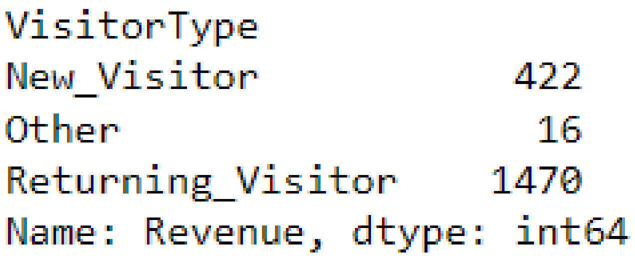
We also wanted to see if the number of pages consulted influence the Revenue variable.



	PageValues
Revenue	
False	1.975998
True	27.264518

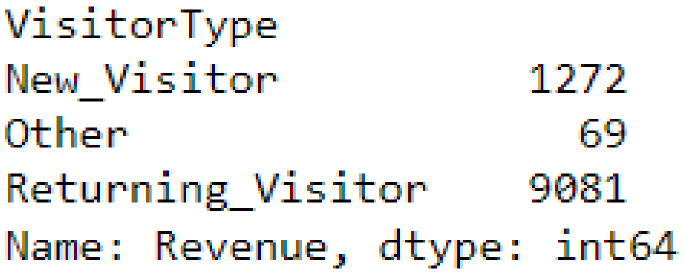
Consumers browse on average more the website when they buy a products. This shows the correlation between the number of page visited by the customer and the probability for a consumer to buy a product from the website.

We also wondered if new customers are more inclined to buy.



```
VisitorType
New_Visitor              422
Other                     16
Returning_Visitor       1470
Name: Revenue, dtype: int64
```

Revenue = False



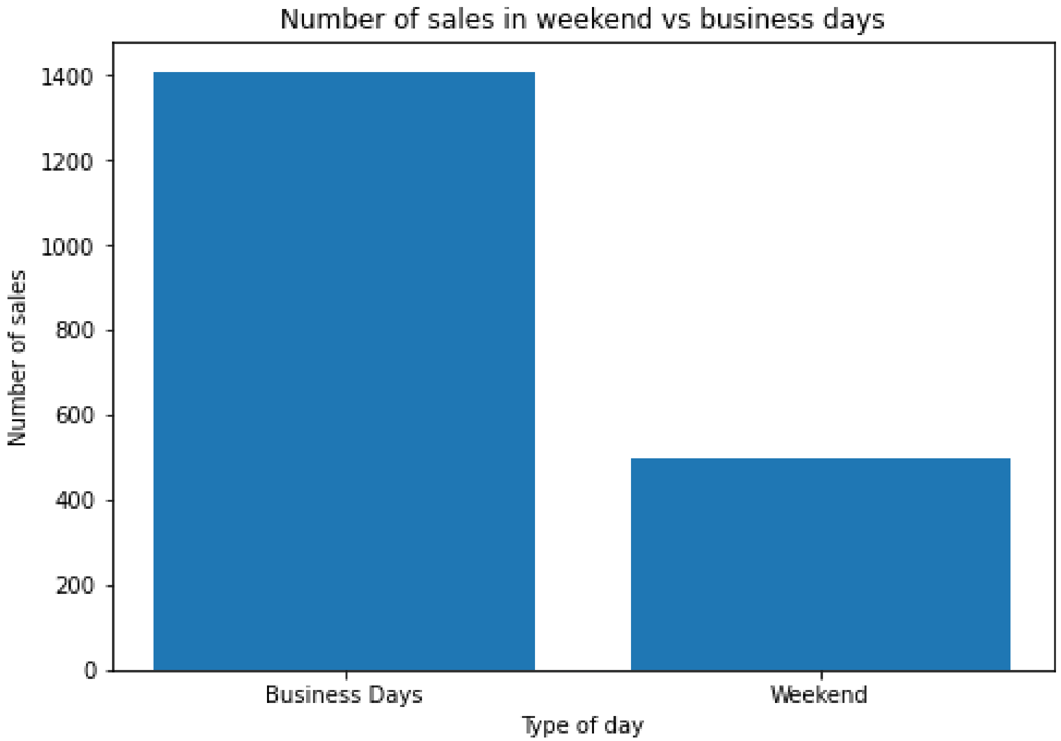
```
VisitorType
New_Visitor             1272
Other                     69
Returning_Visitor       9081
Name: Revenue, dtype: int64
```

Revenue = True

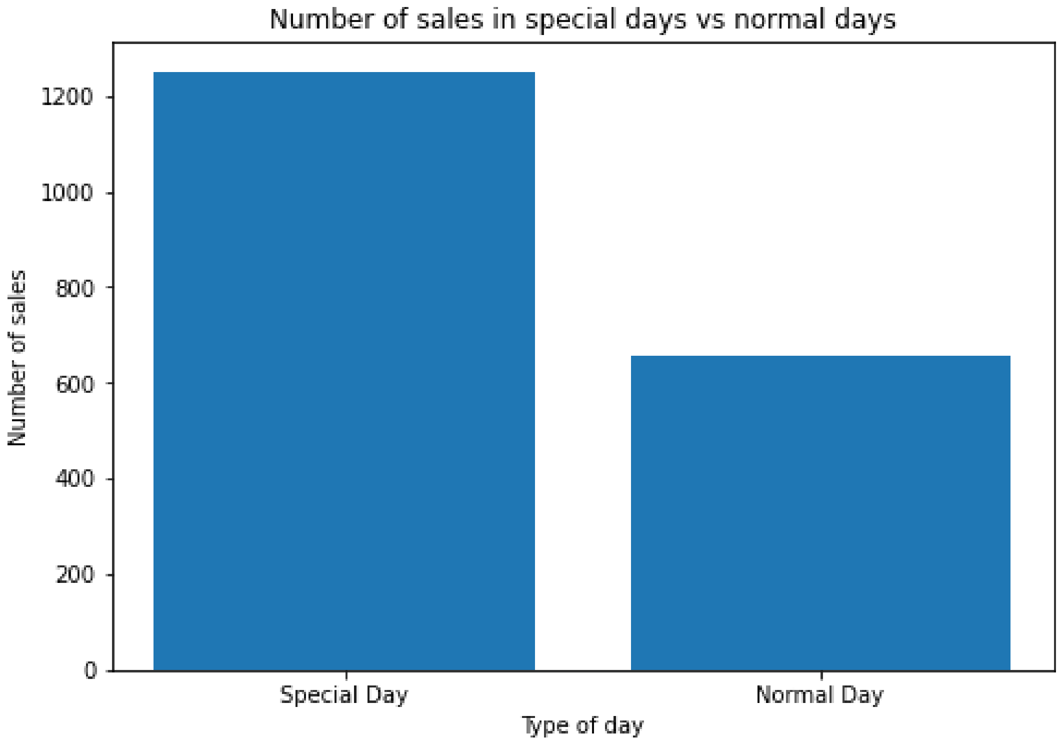
Most of people who buy are returning visitors, but it is also the case for people who don't buy. We'll need the correlation matrix to see the relation between the visitor type and the Revenue variable.

1.5 Seasonal Behavior

In this part of the study, we tried to see if consumers' behaviors change on the weekend or around special days like Valentine's day or Thanksgiving.



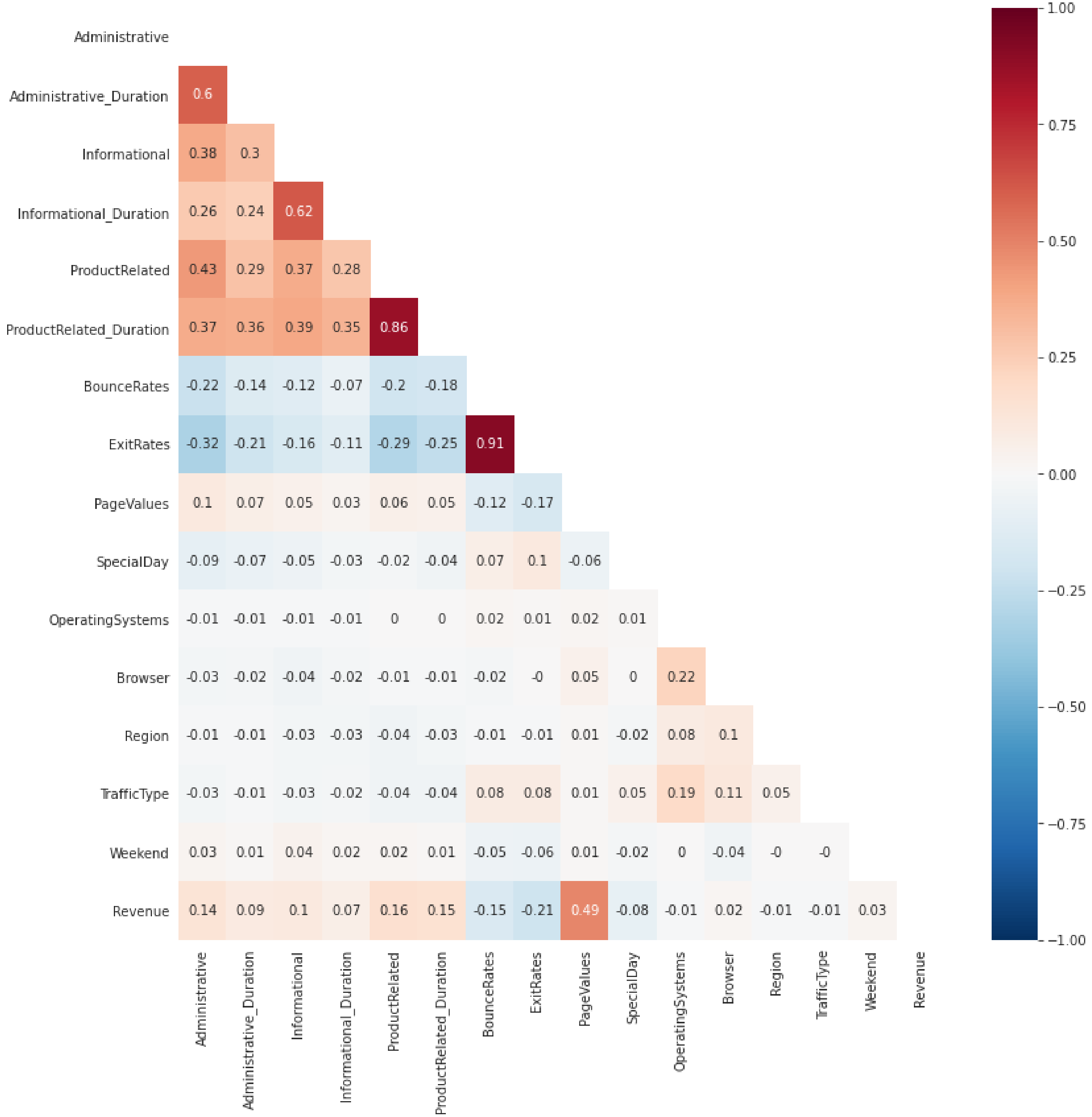
There are way more sales in business days than in the weekend.



There are way more sales around special days like mother's day or valentines Day than on normal days.

1.6 Correlation between all the variables

In order to visualize the correlation between all the variables, we built a correlation matrix. For more readability, we presented only a half of the correlation matrix (because of the symmetry of the matrix).



	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues	SpecialDay	OperatingSystems	Browser	Region	TrafficType	Weekend	Revenue
Administrative																
Administrative_Duration	0.6															
Informational	0.38	0.3														
Informational_Duration	0.26	0.24	0.62													
ProductRelated	0.43	0.29	0.37	0.28												
ProductRelated_Duration	0.37	0.36	0.39	0.35	0.86											
BounceRates	-0.22	-0.14	-0.12	-0.07	-0.2	-0.18										
ExitRates	-0.32	-0.21	-0.16	-0.11	-0.29	-0.25	0.91									
PageValues	0.1	0.07	0.05	0.03	0.06	0.05	-0.12	-0.17								
SpecialDay	-0.09	-0.07	-0.05	-0.03	-0.02	-0.04	0.07	0.1	-0.06							
OperatingSystems	-0.01	-0.01	-0.01	-0.01	0	0	0.02	0.01	0.02	0.01						
Browser	-0.03	-0.02	-0.04	-0.02	-0.01	-0.01	-0.02	-0	0.05	0	0.22					
Region	-0.01	-0.01	-0.03	-0.03	-0.04	-0.03	-0.01	-0.01	0.01	-0.02	0.08	0.1				
TrafficType	-0.03	-0.01	-0.03	-0.02	-0.04	-0.04	0.08	0.08	0.01	0.05	0.19	0.11	0.05			
Weekend	0.03	0.01	0.04	0.02	0.02	0.01	-0.05	-0.06	0.01	-0.02	0	-0.04	-0	-0		
Revenue	0.14	0.09	0.1	0.07	0.16	0.15	-0.15	-0.21	0.49	-0.08	-0.01	0.02	-0.01	-0.01	0.03	

Thanks to this correlation matrix, we see that principal correlation links are :

- ExitRate and BounceRate -> 91% of correlation
- ProductRelated_Duration and ProductRelated -> 86% of correlation
- Informational_Duration and Informational -> 62% of correlation
- Administrative_Duration and Administrative -> 60% of correlation
- Revenue and PageValues -> 49% of correlation

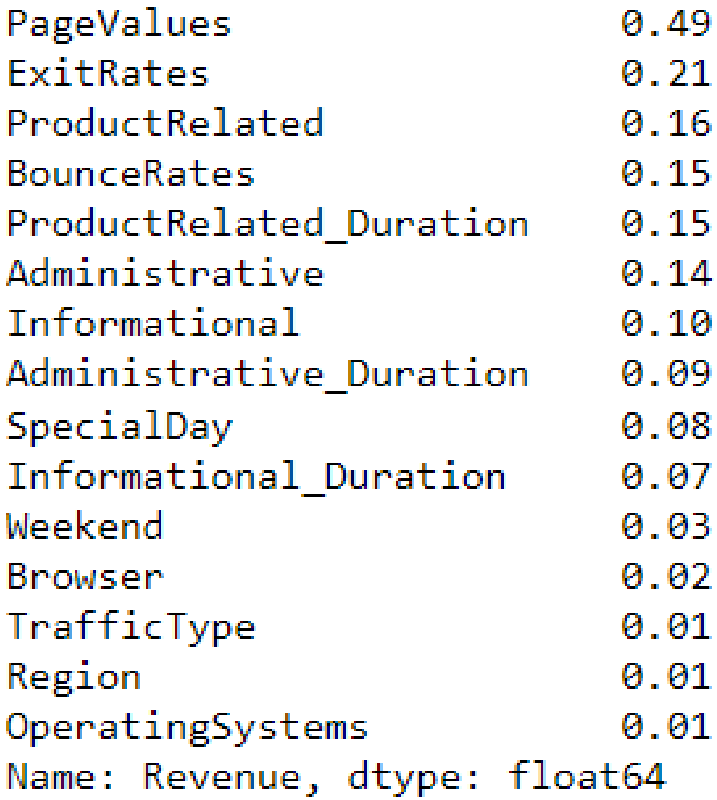
It's important to note that these top correlations are positives.

We also see variables that have almost no correlation:

- OperatingSystem with ProductRelated and ProductRelated_Duration
- Browee with ExitRates and SpecialDay
- Weekend with OperatingSystem, Region and TrafficType

This is a non-exhaustive list. Many other variables have very low correlation rates.

This entire correlation matrix is very interesting but, as we have a target variable, we wanted to focus on it and see which variables explain its variability the most.



```
PageValues                 0.49
ExitRates                  0.21
ProductRelated             0.16
BounceRates                0.15
ProductRelated_Duration    0.15
Administrative             0.14
Informational              0.10
Administrative_Duration    0.09
SpecialDay                 0.08
Informational_Duration     0.07
Weekend                    0.03
Browser                    0.02
TrafficType                0.01
Region                     0.01
OperatingSystems           0.01
Name: Revenue, dtype: float64
```

We chose to sort the absolute values of the correlations so that we see the most important variables, no matter if the correlation is positive or negative.

Thus, we see that almost 50% of Revenue is explained by PagesValues. Besides, this correlation is positive so we can say that when the value of PagesValues increase, the probability of a sale increase too.

The other variables that are the most correlated to Revenue are ExitRates, ProductRelated and BounceRates. At contrary, OperatingSystem, Region and TrafficType are the least correlated variables with Revenue. We could suppress these variables the dataset as the do not explain much the target variable.

2. Supervised Learning

In this part, thanks to machine learning, we tried to produce effective prediction models to know if a certain session will be concluded by a sale or not.

Because our target variable is qualitative, we used classification models like logistic regression, discriminant analysis, regression trees, ...

2.1 Preparation of the dataset

In order to train our models we needed to split the inital dataset into two parts : a training set and a testing set.

But first of all, because our dataset contain qualitative data, we needed to transform this into quantitative data. We represented the months by their number. For boolean values, we considered that True correspond to 0 and False to 1. Finally, here are the correspondences for the VisitorType Variable :

- 0 : Returning_Visitor
- 1 : New_Visitor
- 2 : Other

After that, we splited our dataset into two subsets which will respectively represent the variables and the target of our models.

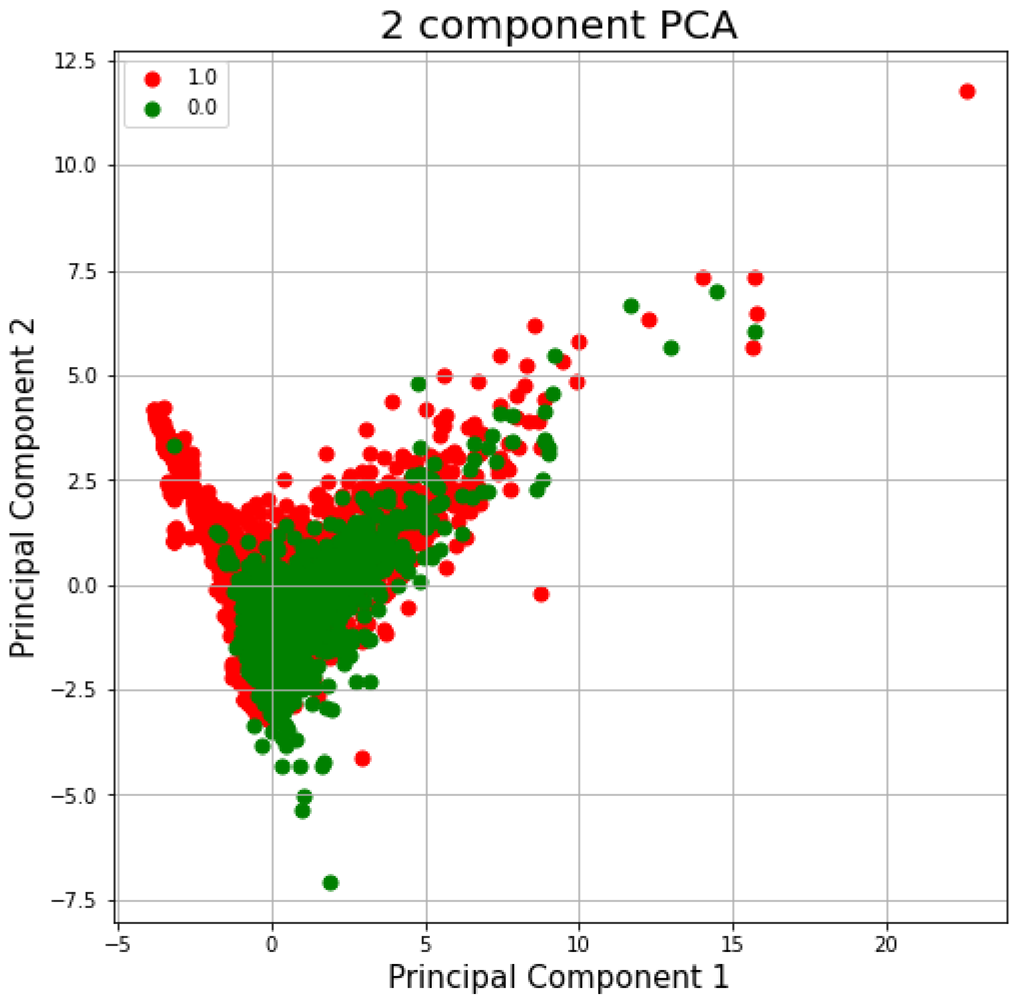
Moreover, we needed to standardize the data to be sure that dimensions related problems won't occur. Because we splited our dataset into a train and a test set randomly and in order to not lose the indexes of our two sets, we save them before the standardization.

2.2 Principal Components Analysis

The principal components analysis isn't a machine learning model. It is a method of dimensional reduction. We made a PCA before creating models because we hoped this would permit us to work with less variables and so reduce computation time.

To begin, we applied the PCA on our training set and we focus on the 2 first principal components. Once we got these two first principal components, we visualized our data in two dimensions.

This visualization didn't seem very relevant. To verify the relevance of these two principal components, we computed their explained variance ratio which are respectively 0.24 an 0.12.

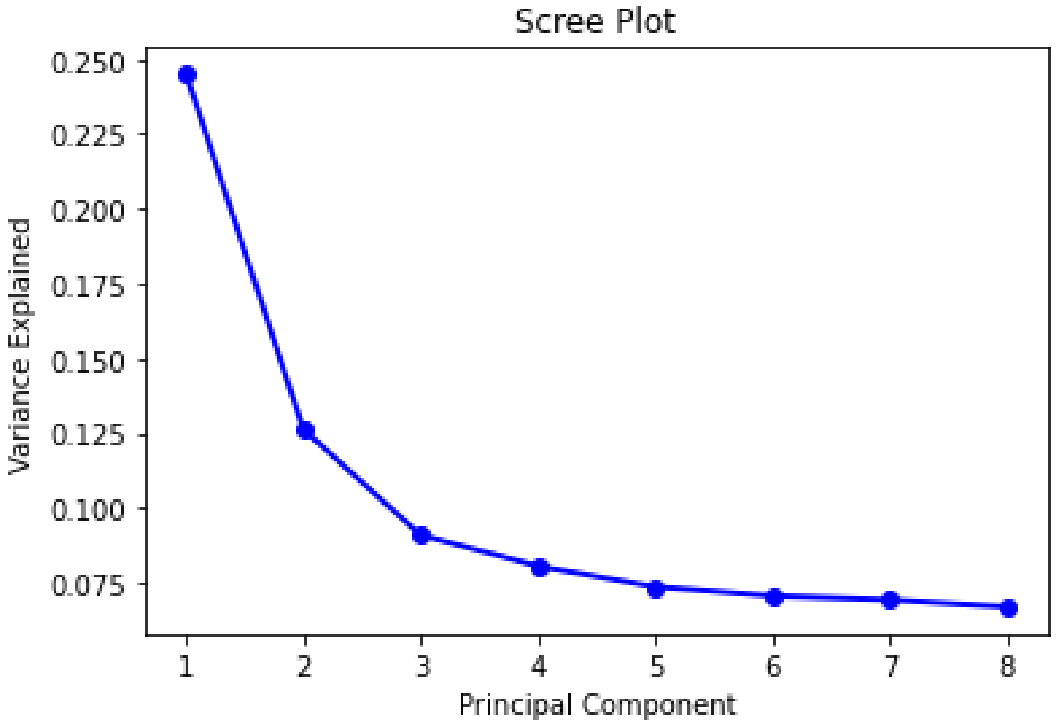


With these explained variance ratios, we saw that our two first principal components only explain 36% of the variance. This percentage is quite low. But our aim was not to specially to visualize our data in a 2D plot, so we could add other principal components to increase the cumulative explained variance ratio.

Instead of repeating this process again and again, we created a function that gives us the number of principal components we need in order to obtain a cumulative explained variance ratio greater than a given percentage.

Thus, in order to represent at least 75% of the variance of the dataset we need 7 principal components. This represents the half of the number of variables we have in our data set.

We can conclude that the principal components analysis was not very concluant on our data. We decide to not use this representation of our data in the following models.



2.3 Find the best parameters of a model

In order to find the better model, we tested different machine learning model on our dataset. And even in each of these models, we knew that different parameters can be chosen. So, to find the best parameters for each models, we used the GridSearchCV function from the sklearn library. We automized this process by creating a function that takes in entry the model we want to test and the possible parameters of this model. This function return the best score obtained with this model and the parameters used in the model to obtain this score.

After creating this function, we tested it on a model. We chose the TweedieRegressor model and we tested with some parameters. The function returned a score and a list of parameters. But the first obtained score was low. We tested this model with the all the parameters we could and the best score was still quite low. This model wasn't adapted to our dataset. But this was just a test and we applied other models on our dataset.

2.4 Test different models

In this part, we tested different classification models. We saved the best score of each model into a table and exploited it in the last section.

2.4.1 Logistic Regression

First, we tried the logistic regression model. For this, we imported the LogisticRegression function from sklearn library and we applied the function we created earlier.

With only one random parameter, the model already seem very good with a score of 88.22% of good predictions.

With the best parameters, the score increased by 0.3%. So we can conclude that the best score we can have on this model is 88.52% of good predictions.

2.4.2 Linear Discriminant Analysis

Secondly, we tried the linear discriminant analysis model. For this, we imported the LinearDiscriminantAnalysis function from sklearn library and we applied the function we created earlier.

With only one random parameter, the model already seem very good with a score of 87.87% of good predictions.

Even with the best parameters, the best score didn't change. So we could conclude that the best score we can have on this model is 87.87% of good predictions.

2.4.3 Quadratic Discriminant Analysis

Then, we tried the quadratic discriminant analysis model. For this, we imported the QuadraticDiscriminantAnalysis function from sklearn library and we applied the function we created earlier.

With only one random parameter, the model already seem very good with a score of 83.15% of good predictions.

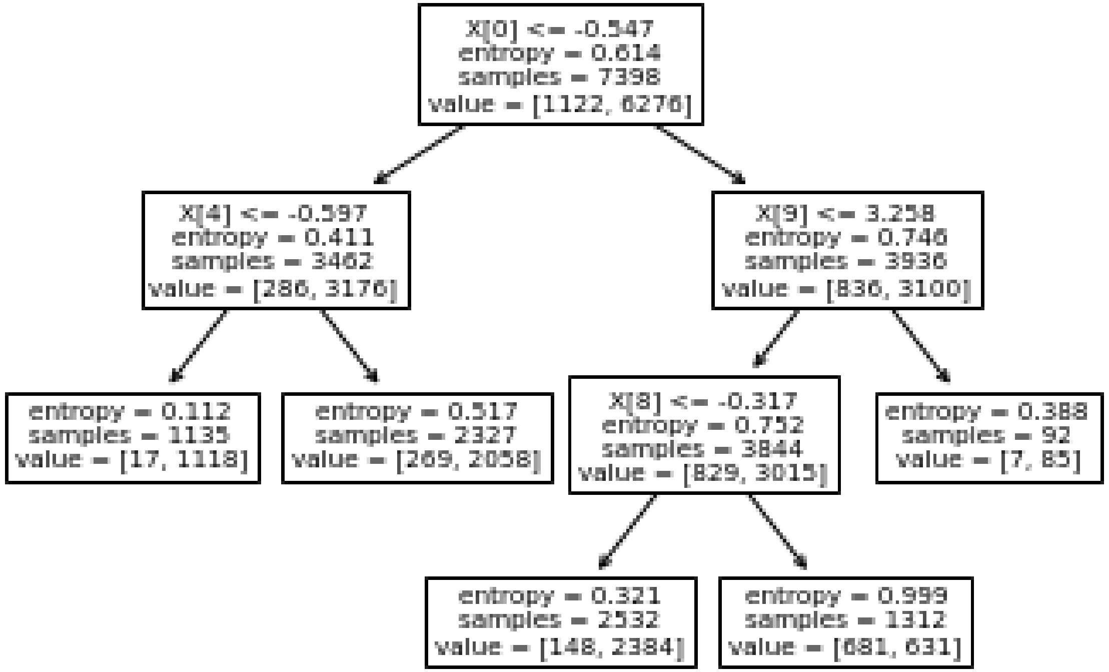
With the best parameters, the score increased by 2%. So we could conclude that the best score we can have on this model is 85.45% of good predictions.

2.4.4 Decision Tree Classifier

Later, we tried the decision tree classifier model. For this, we imported the DecisionTreeClassifier function from sklearn library and we applied the function we created earlier.

With only one random parameter, the model already seem very good with a score of 85.96% of good predictions.

With the best parameters, the score increased by 1.6%. So we can conclude that the best score we can have on this model is 87.61% of good predictions.



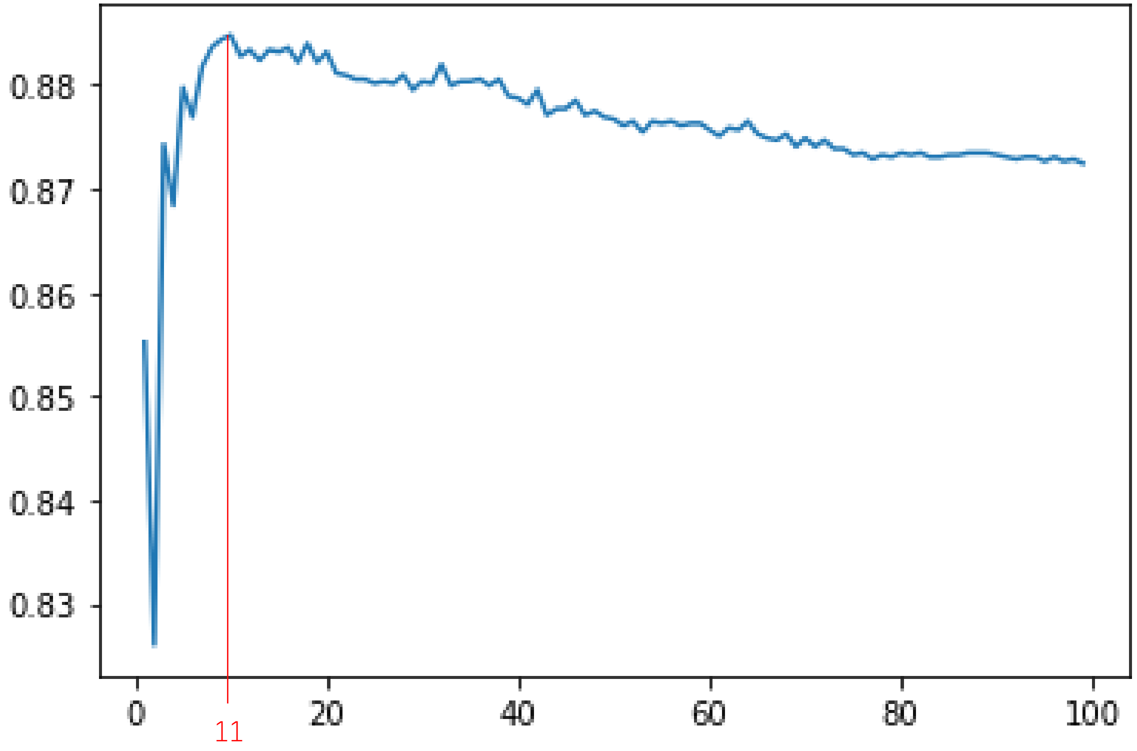
2.4.5 K-Neighbors Classifier

Finally, we tried the K-Neighbors classifier model. For this, we imported the KNeighborsClassifier function from sklearn library and we applied the function we created earlier.

With only one random parameter, the model already seem very good with a score of 88.24% of good predictions. We wanted to improve this score by testing this model with other parameters. But before that, we got to find the best number of neighbors. To do so, we created a function that test all the number of neighbors from 1 to 100. The function returns the best number of neighbors according to the "good predictions" score. So, here we knew that we would have 11 neighbors in our model.

With the best parameters, the score inscreased by 0.1%. So we can conclude that the best score we can have on this model is 88.34% of good predictions.

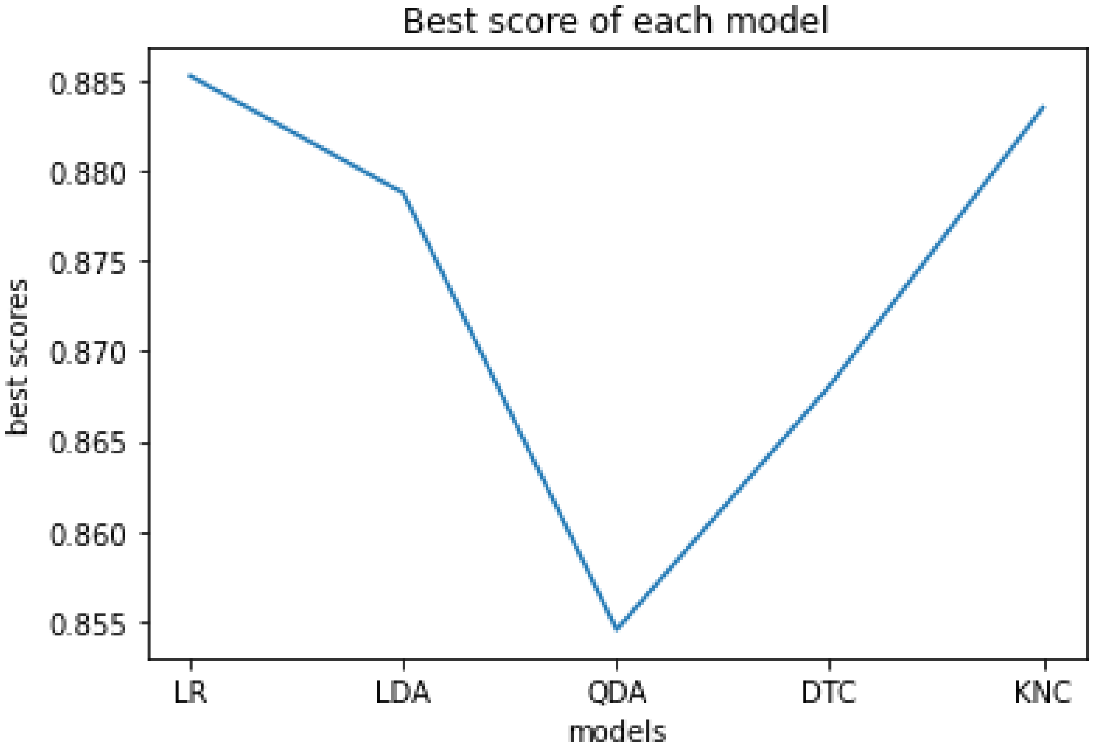
Score of the K-Neighbors Classifier according to the number of neighbors



2.5 Conclusion

To conclude, we tested 5 machine learning models. Here is a plot to recap the score of each model.

We saw that all the scores are between 85% and 89%. These scores are all quite good, which means that all the models could be used to predict data from our dataset. But the best model still is the logistic regression with almost 89% of good predictions, as we can see on the plot.


Best score of each model
best scores
LR
LDA
QDA
DTC
KNC
models