PYTHON FOR DATA ANALYSIS PROJECT

YAN PODOLAK

12330 lines

18 variables

No missing values

THE DATASET :
ONLINE SHOPPERS PURCHASING INTENTION DATASET ANALYSIS

WHAT WE NEED TO PREDICT

Variable 'Revenue' to predict

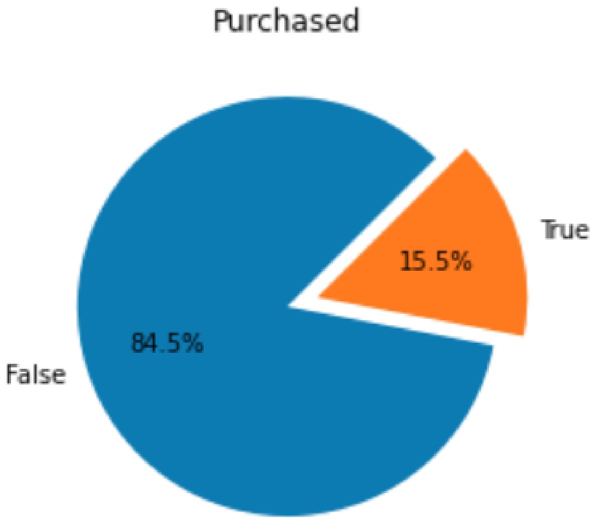
Takes boolean value True/False.

True : The visitor did buy
False : The visitor did not buy

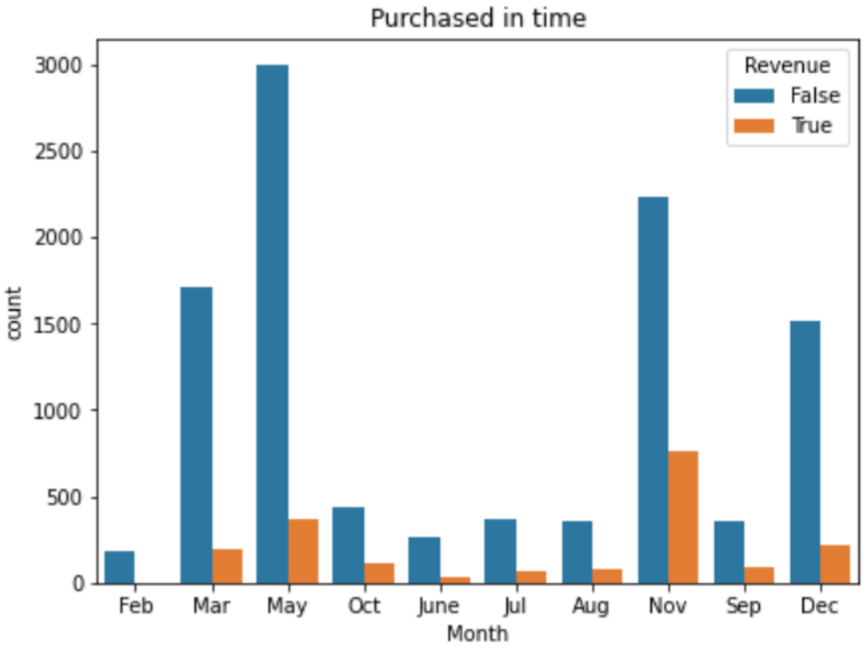
It's a classification problem :
We want to know if the visitor will buy or not on the website

DATA VISUALIZATION

Only 15.5% of visitors spent their money



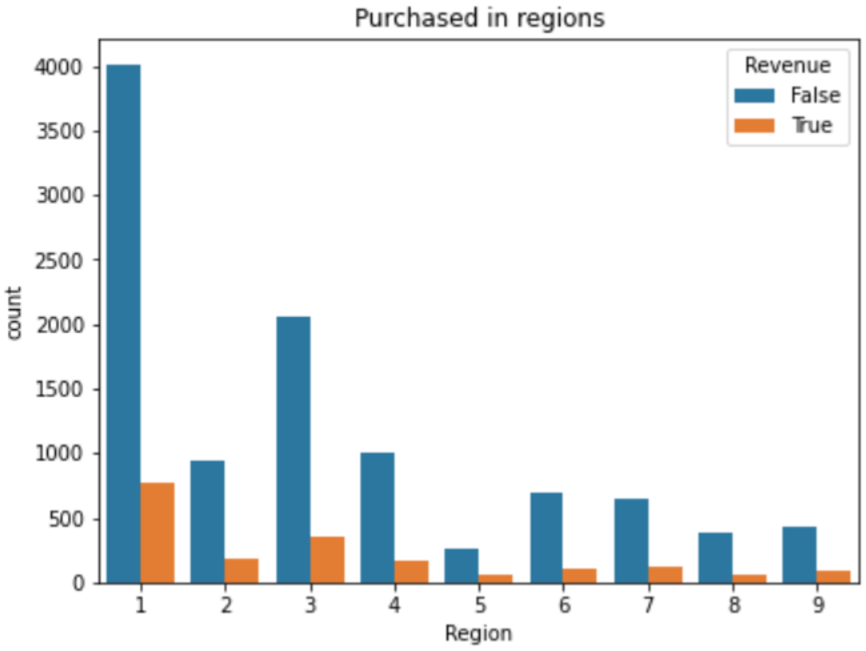
Repartition of the visitors in time



We notice that mainly 4 months have a lot more visitors than the others : March, May, November and December. It may be because of summer and Christmas holidays ? Notice that we don't have data for January and April

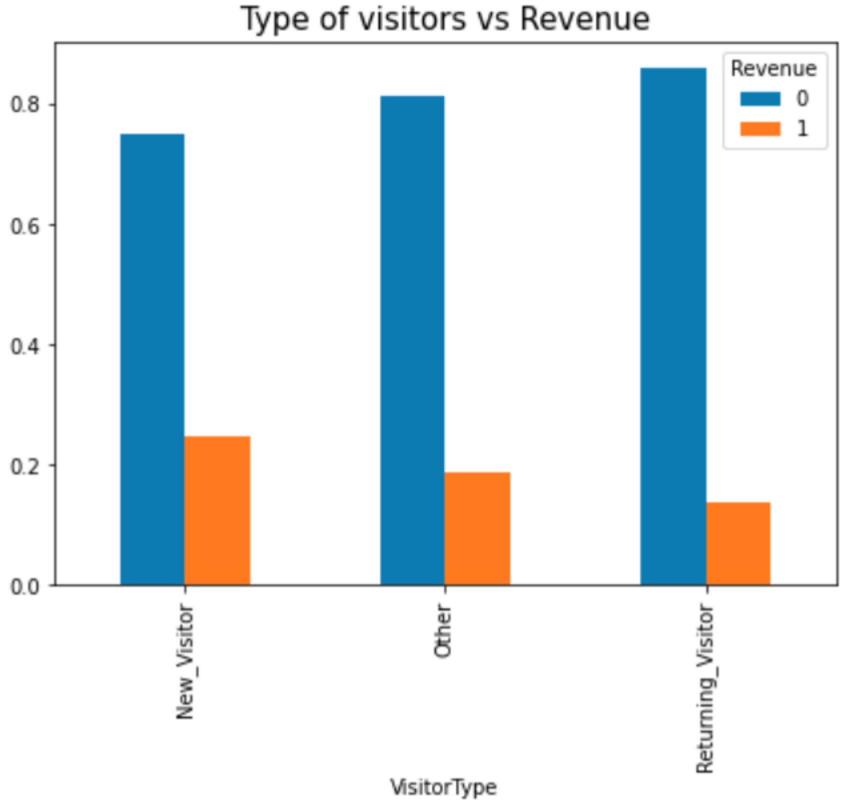
DATA VISUALIZATION

Repartition of the visitors in regions



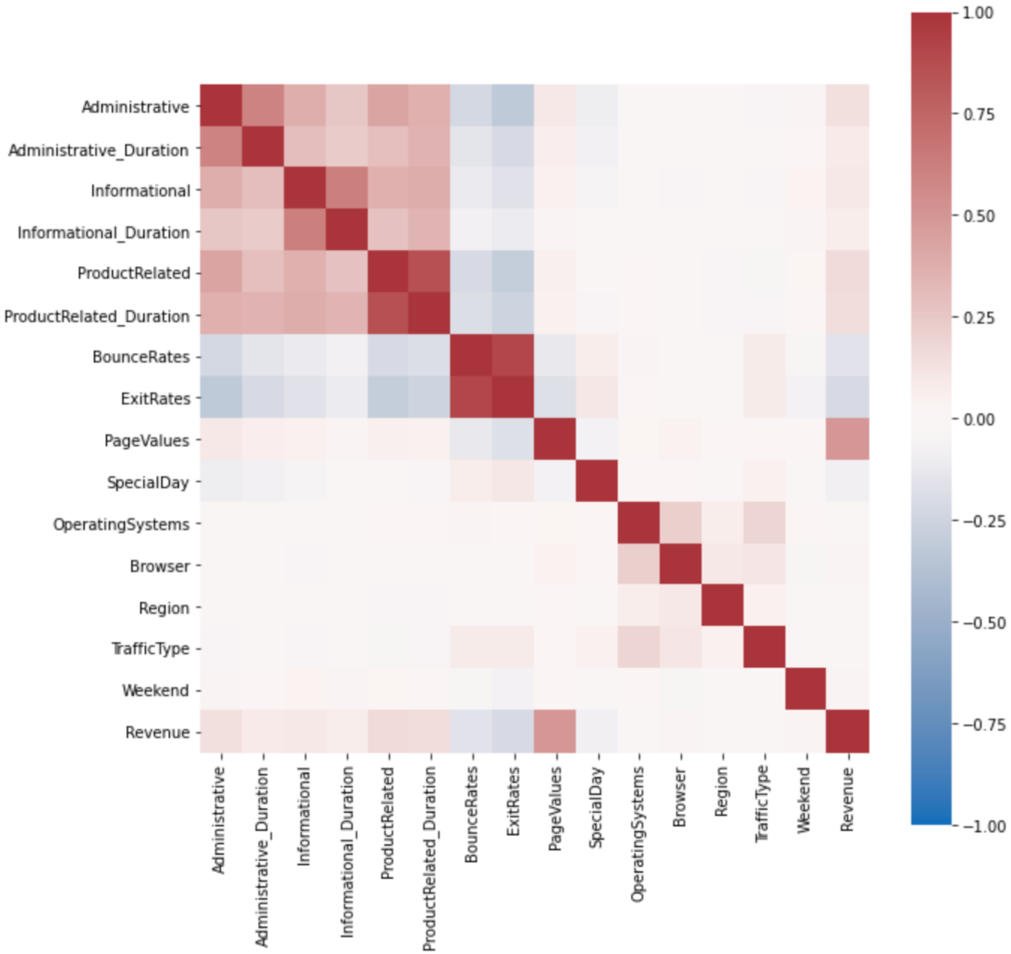
We can see here that most visitors are from region 1

Type of visitors



Returning visitors are not buying more.

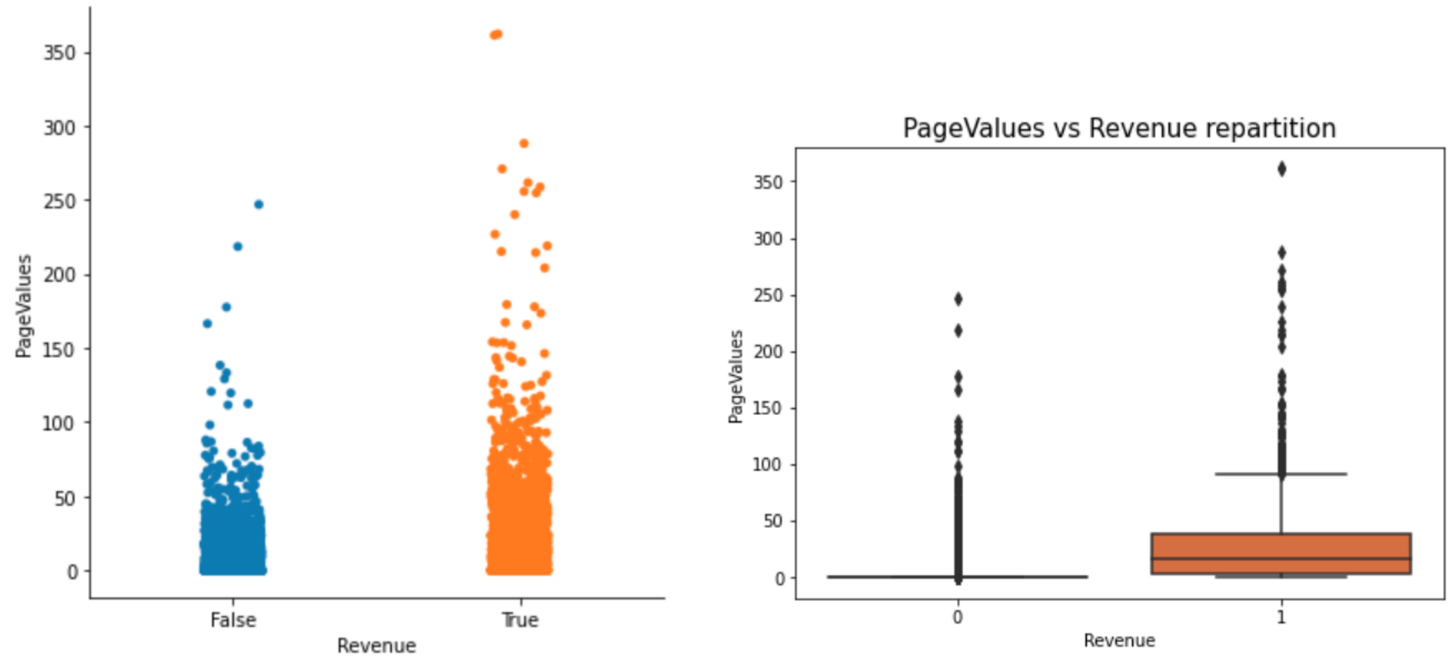
DATA VISUALIZATION



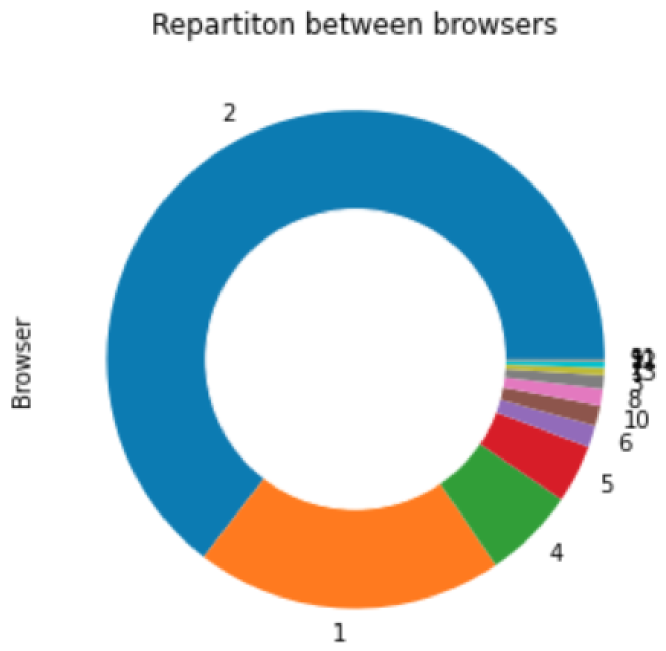
Correlation matrix

According to the matrix we have a correlation between 'Revenue' and 'PageValues' meaning that pages with a high value get more money from the visitor.
We have a strong correlation between BouceRates and ExitRates.

DATA VISUALIZATION

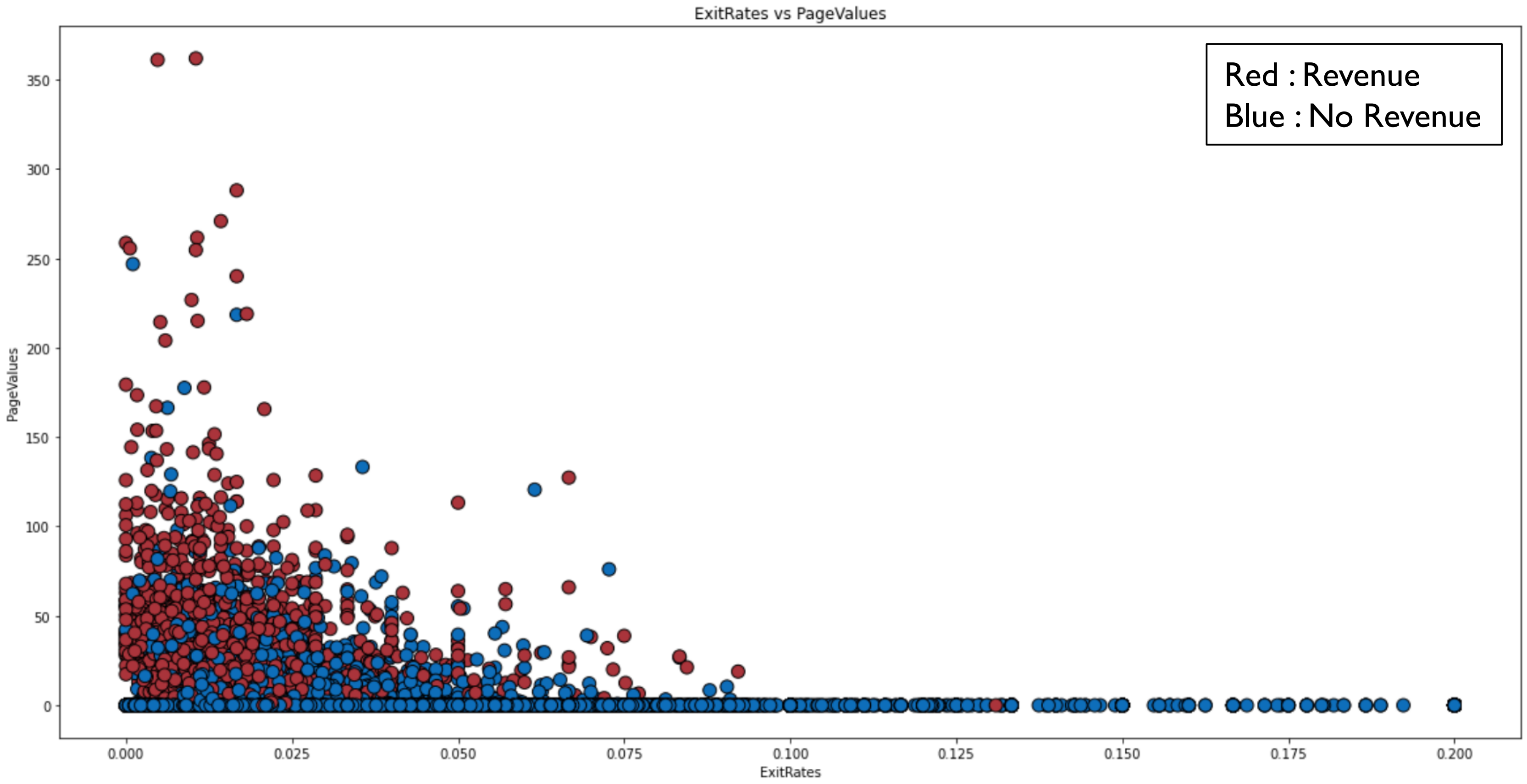


As we saw in the matrix, people tend to buy more on pages with a higher value



We can see that people use mostly the browser 2

DATA VISUALIZATION



We can see again here that the higher the PageValues is, the more people buy.
We see that the higher the ExitRate is, the less visitors buy.

MODELLING

- Pre-processing
- Applying the models
- Exporting the model for the API

PRE-PROCESSING

In order to predict data in a model, we first need to adapt our data :

- Input variables : All variables except ‘Revenue’
- Output variable (to predict) : ‘Revenue’
- We process a one-hot encoding on the input variables
- We modify the output variables to get 0 and 1 instead of False and True.
  - 0 : The visitor will not buy
  - 1 : The visitor will buy
- We split the data :
  - Training data : 70%
  - Testing data : 30%

APPLYING THE MODELS

We have a classification problem with 2 different outputs : 0 and 1. So it is a binary classification problem.

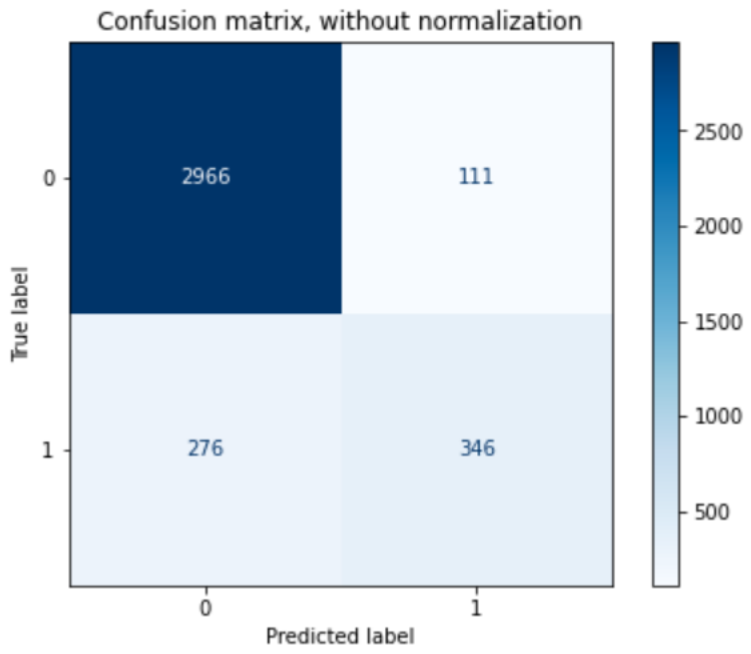
We will apply different models :

- Random Forest
- Logistic Regression
- SVM
- Naïve Bayes
- K-Nearest Neighbors

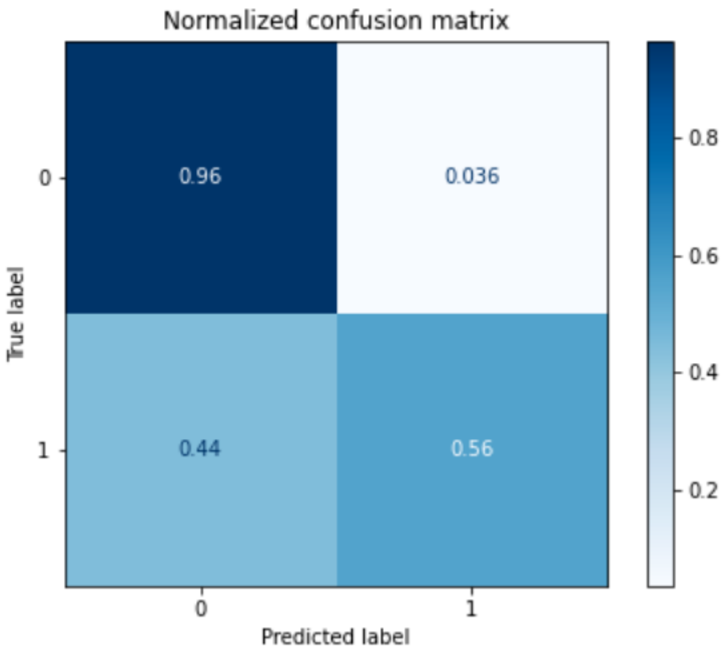
APPLYING THE MODELS

Random Forest

We get an accuracy of 0.8953771289537713



Confusion matrix, without normalization



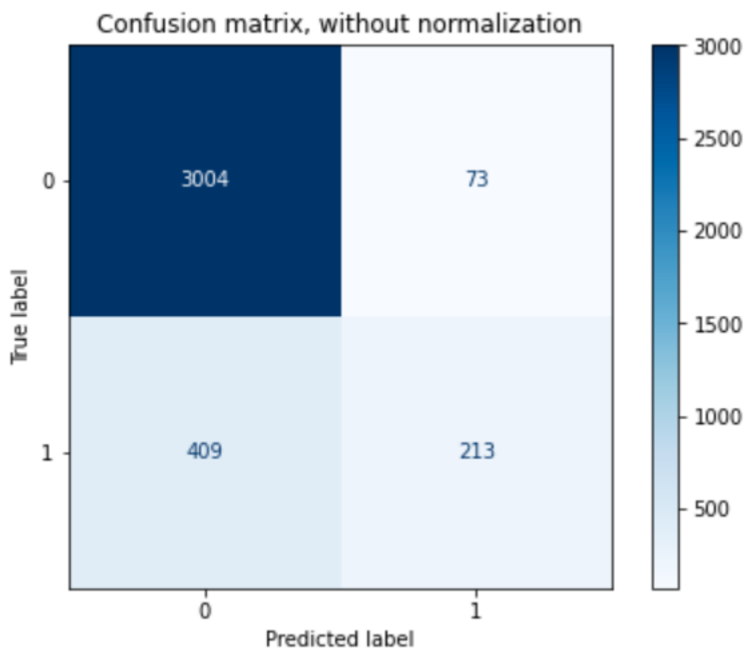
Normalized confusion matrix

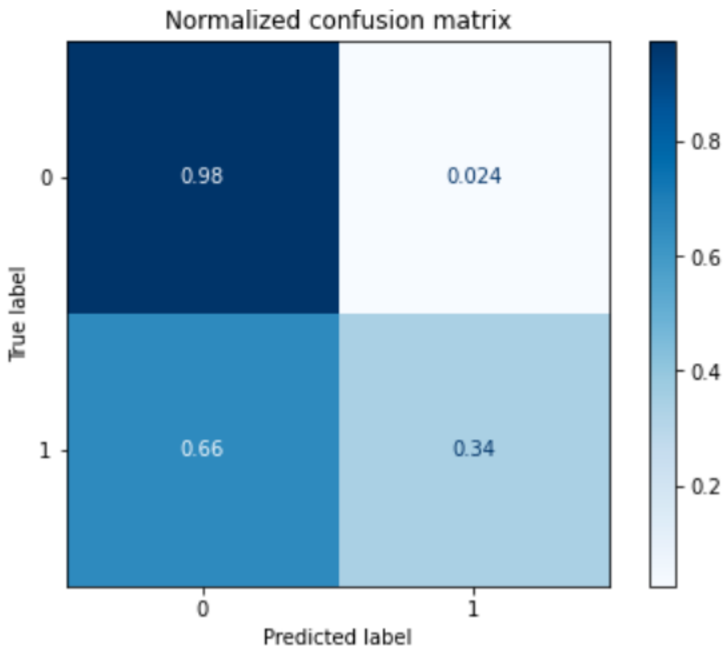
We get a good result on 1 and a very good result on 0

APPLYING THE MODELS

Logistic Regression

We get an accuracy of 0.8696945120302785



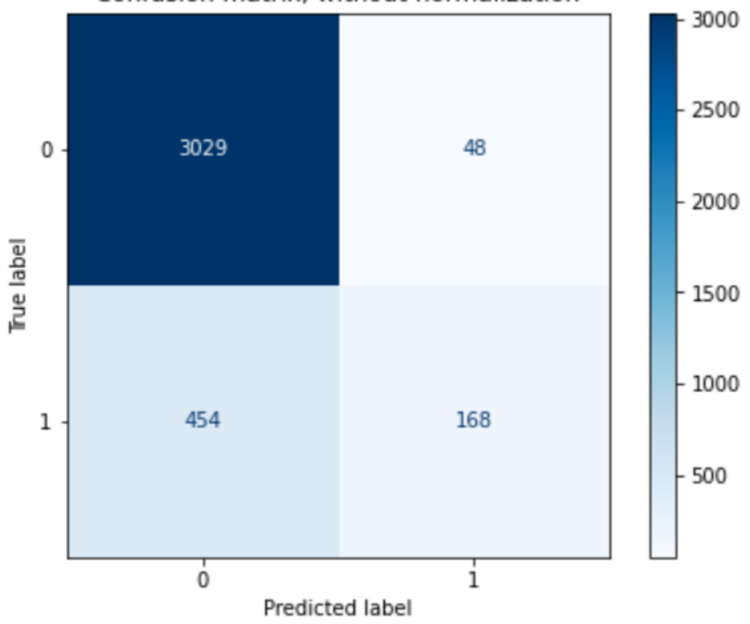


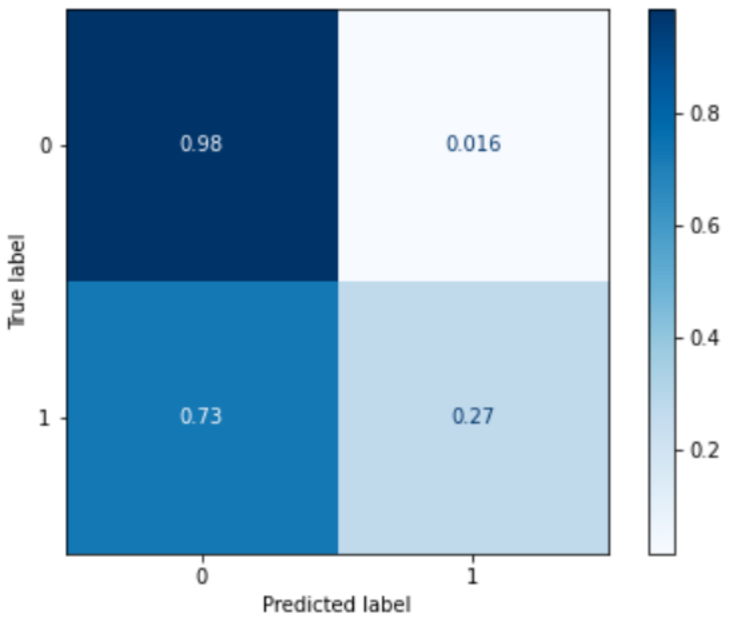
We get a bad result on 1 and a very good result on 0

APPLYING THE MODELS

SVM

We get an accuracy of 0.8642876453095432


Confusion matrix, without normalization

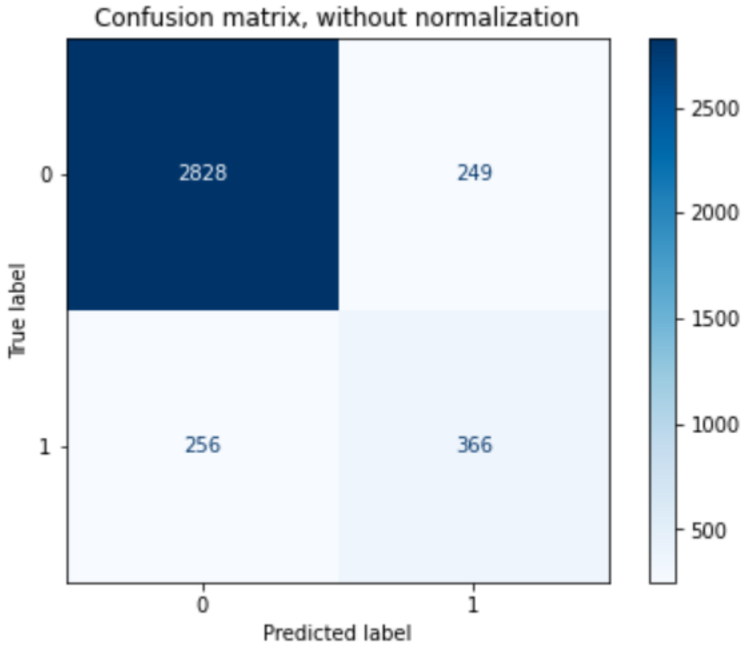

Normalized confusion matrix

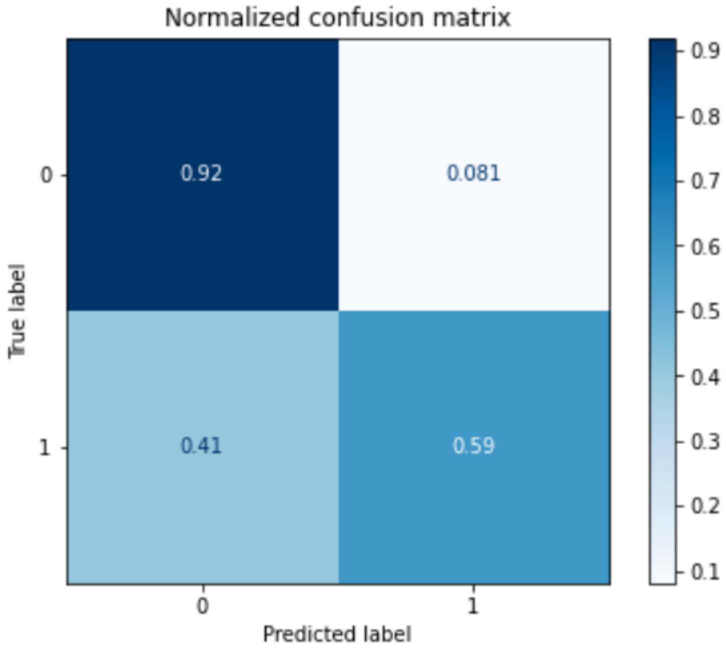
We get a bad result on 1
and a very good result on 0

APPLYING THE MODELS

Naïve Bayes

We get an accuracy of 0.8634766153014328


Confusion matrix, without normalization

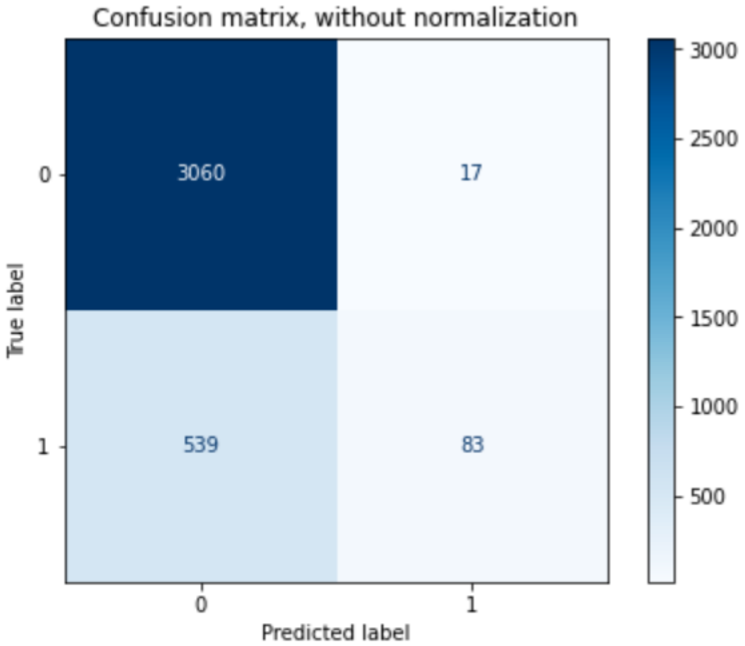

Normalized confusion matrix

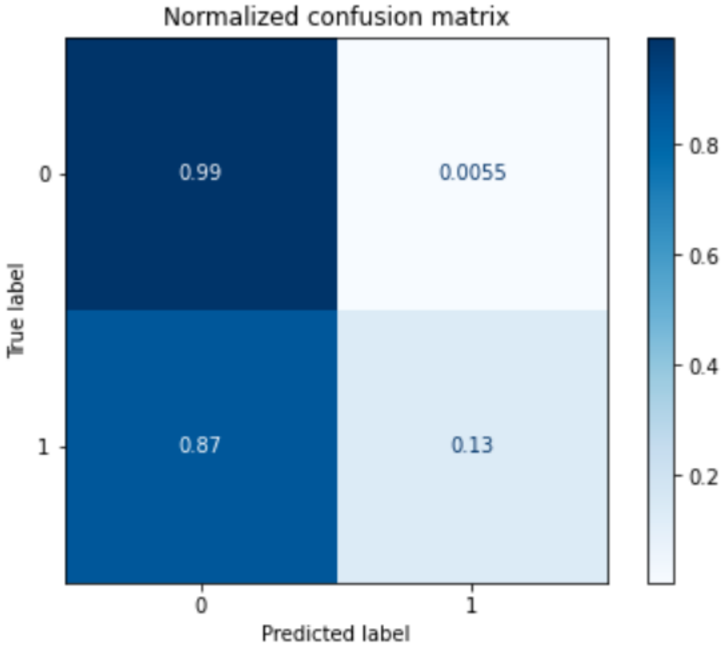
We get a quite good result on 1 and a very good result on 0

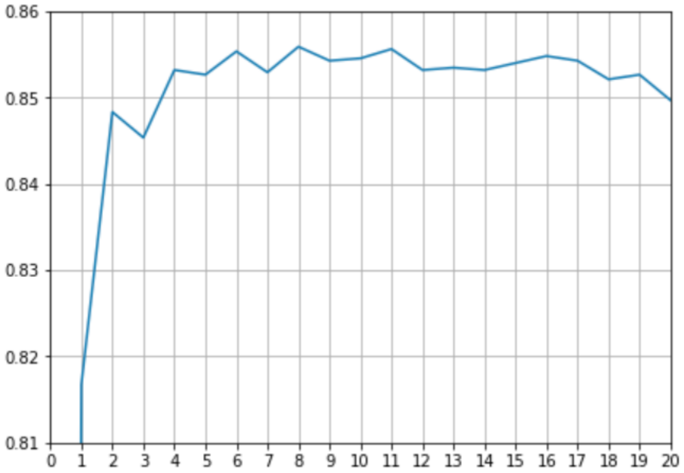
APPLYING THE MODELS

K-Nearest Neighbors

The accuracy depends on the number of neighbors k


Confusion matrix, without normalization


Normalized confusion matrix

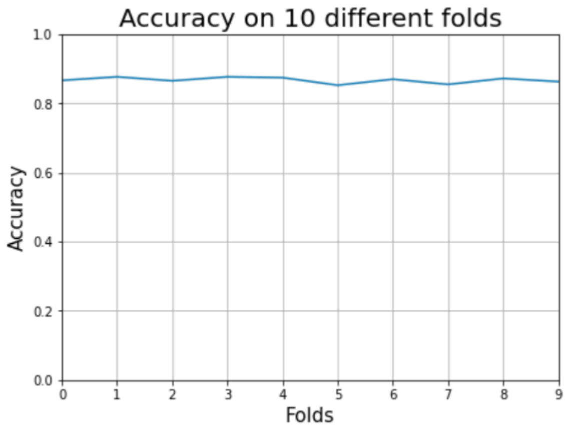


Accuracy depending on k

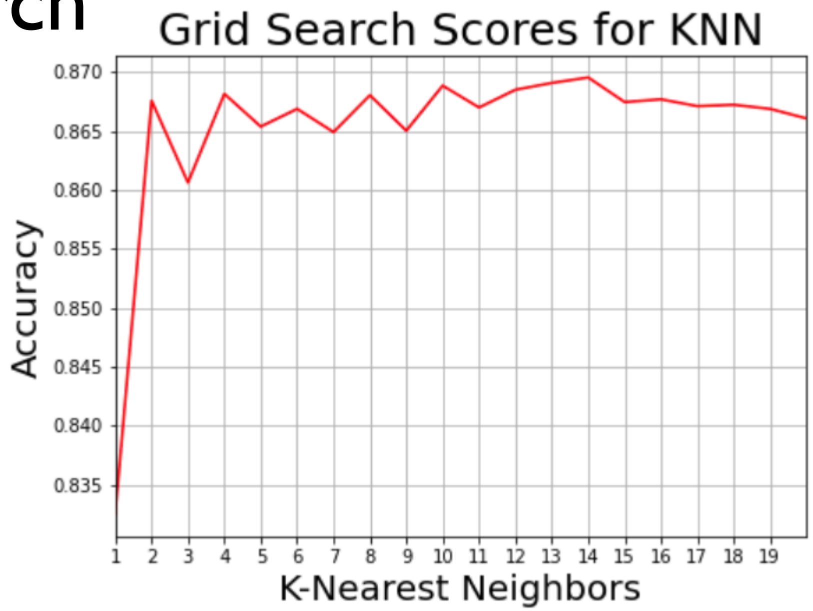
We get a really bad result on 1 and a very good result on 0

APPLYING THE MODELS

K-Nearest Neighbors using a Grid Search

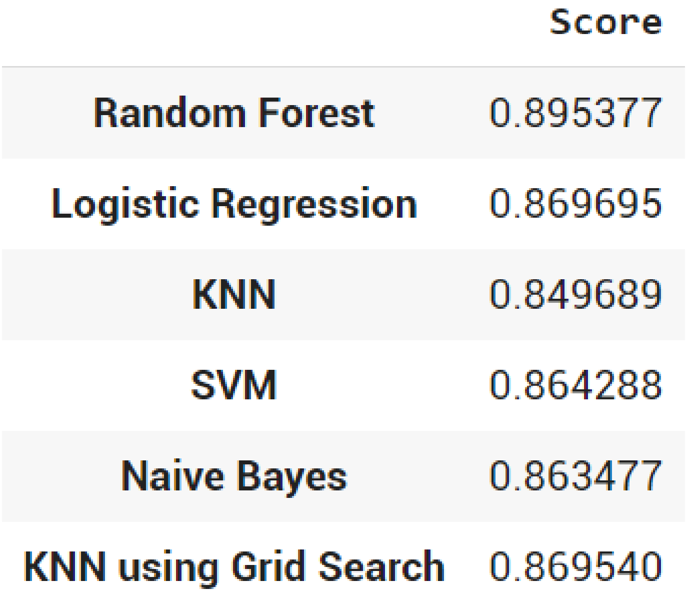


With a cross-validation with 10 folds, we get a mean of 0.8675704637140036



With the Grid Search, we get the best results for k=14 with 0.8695402021372474.

EXPORTING THE MODEL FOR THE API

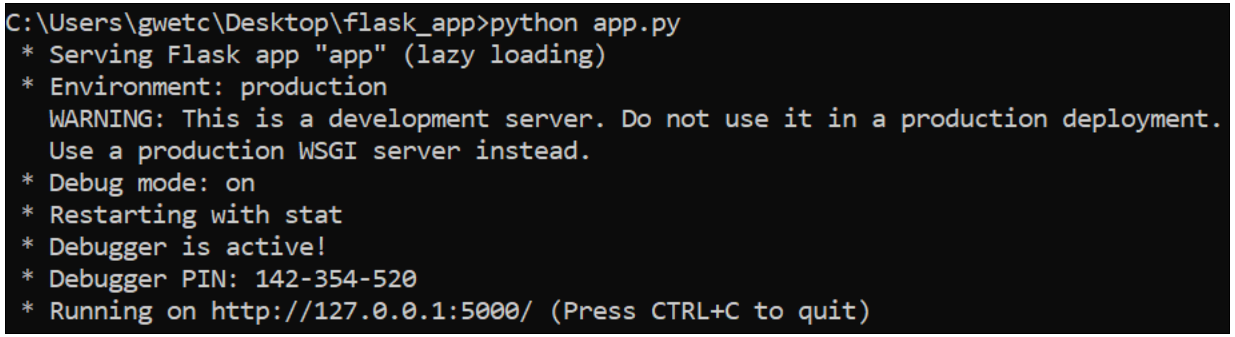


	Score
Random Forest	0.895377
Logistic Regression	0.869695
KNN	0.849689
SVM	0.864288
Naive Bayes	0.863477
KNN using Grid Search	0.869540

As we got the best score with the Random Forest, it is this model that we will use for the API.
It's also the model with the best accuracy on the I which is harder for the models

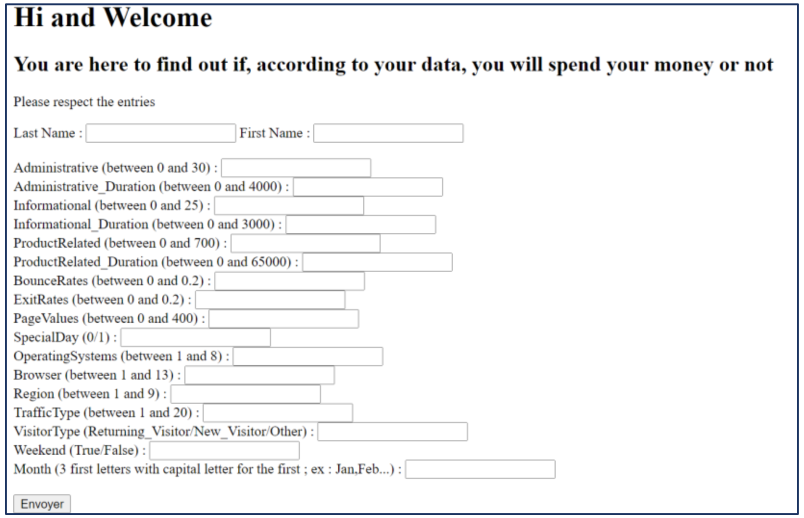
API

API



```
C:\Users\gwetc\Desktop\flask_app>python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 142-354-520
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

If you connect to http://127.0.0.1.5000/ :



Hi and Welcome

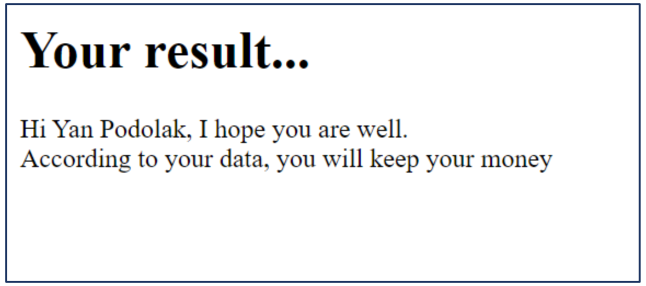
You are here to find out if, according to your data, you will spend your money or not

Please respect the entries

Last Name : First Name :

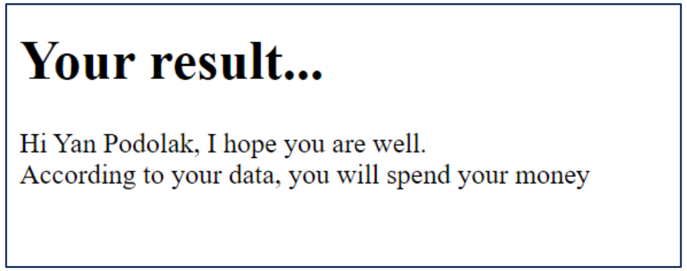
Administrative (between 0 and 30) :
Administrative_Duration (between 0 and 4000) :
Informational (between 0 and 25) :
Informational_Duration (between 0 and 3000) :
ProductRelated (between 0 and 700) :
ProductRelated_Duration (between 0 and 65000) :
BounceRates (between 0 and 0.2) :
ExitRates (between 0 and 0.2) :
PageValues (between 0 and 400) :
SpecialDay (0/1) :
OperatingSystems (between 1 and 8) :
Browser (between 1 and 13) :
Region (between 1 and 9) :
TrafficType (between 1 and 20) :
VisitorType (Returning_Visitor/New_Visitor/Other) :
Weekend (True/False) :
Month (3 first letters with capital letter for the first ; ex : Jan,Feb...) :

According to the problem, you've got 2 possible outputs



Your result...

Hi Yan Podolak, I hope you are well.
According to your data, you will keep your money



Your result...

Hi Yan Podolak, I hope you are well.
According to your data, you will spend your money