PYTHON FOR DATA ANALYSIS

Amir MAHMOUDI - Yassine LAHBABI

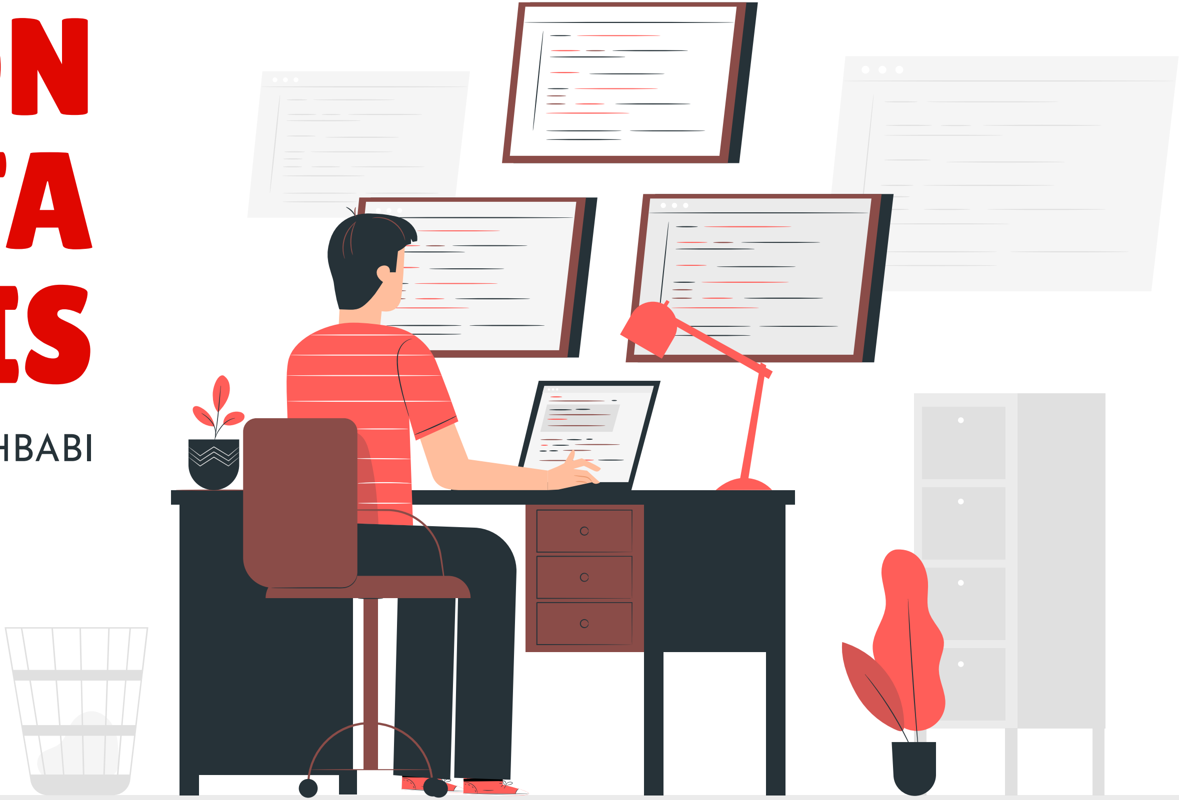


TABLE OF CONTENTS

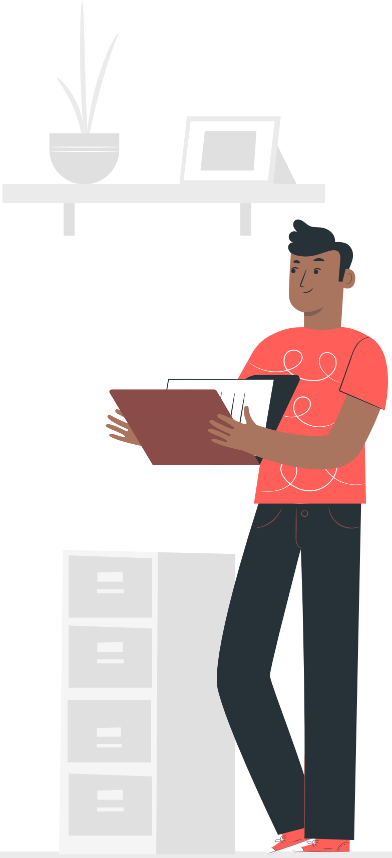
INTRODUCTION

This semester we needed to analyze a data set using python libraries. In our group we are both students from CORE DIA4 at ESILV. We will explain in this presentation our research process. Have a nice reading !

PROJECT'S GOAL

The main goal of this project is to analyze the dataset and visualize it so that we are able to see the main features that we need through using multiple libraries(matplotlib, seaborn, bokeh...) and do some modelling to design a machine learning classification system, that is able to predict an online shopper's intention(buy or no buy), based on the values of the given features and using libraries like scikit-learn through using multiple algorithms and comparing between them.

Finally, we will transform the model into a Flask API so that it can be displayed properly through it.



DATASET DESCRIPTION

This dataset consists of different feature vectors belonging to 12 330 sessions.
Specificities of this dataset : It 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.
Of the 12 330 sessions in the dataset, 84,5% were negative class samples that did not end with shopping, the rest were positive samples ending with shopping.

Numerical Features

Feature name	Feature description	Min. val	Max. val	SD
Admin.	#pages visited by the visitor about account management	0	27	3.32
Ad. duration	#seconds spent by the visitor on account management related pages	0	3398	176.70
Info.	#informational pages visited by the visitor	0	24	1.26
Info. durat.	#seconds spent by the visitor on informational pages	0	2549	140.64
Prod.	#pages visited by visitor about product related pages	0	705	44.45
Prod.durat.	#seconds spent by the visitor on product related pages	0	63,973	1912.3
Bounce rate	Average bounce rate value of the pages visited by the visitor	0	0.2	0.04
Exit rate	Average exit rate value of the pages visited by the visitor	0	0.2	0.05
Page value	Average page value of the pages visited by the visitor	0	361	18.55
Special day	Closeness of the site visiting time to a special day	0	1.0	0.19

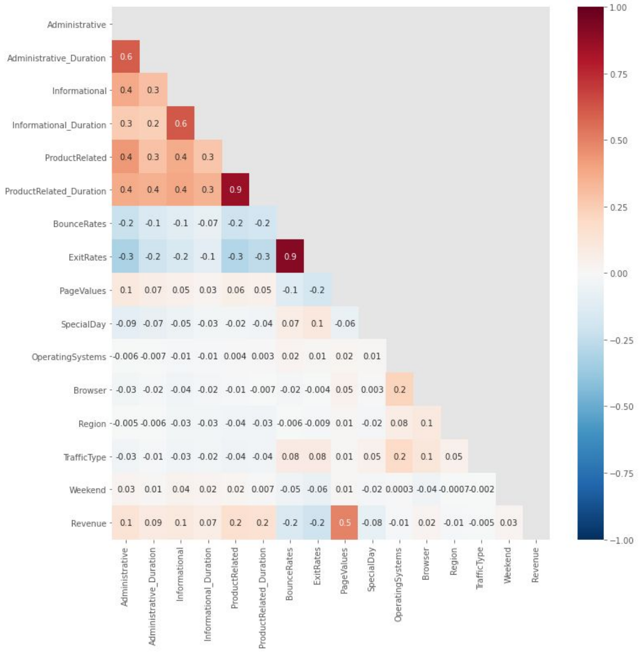
DATASET DESCRIPTION

Categorical Features

Feature name	Feature description	Number of Values
OperatingSystems	Operating system of the visitor	8
Browser	Browser of the visitor	13
Region	Geographic region from which the session has been started by the visitor	9
TrafficType	Traffic source (e.g., banner, SMS, direct)	20
VisitorType	Visitor type as "New Visitor," "Returning Visitor," and "Other"	3
Weekend	Boolean value indicating whether the date of the visit is weekend	2
Month	Month value of the visit date	12
Revenue	Class label: whether the visit has been finalized with a transaction	2

DATASET DESCRIPTION

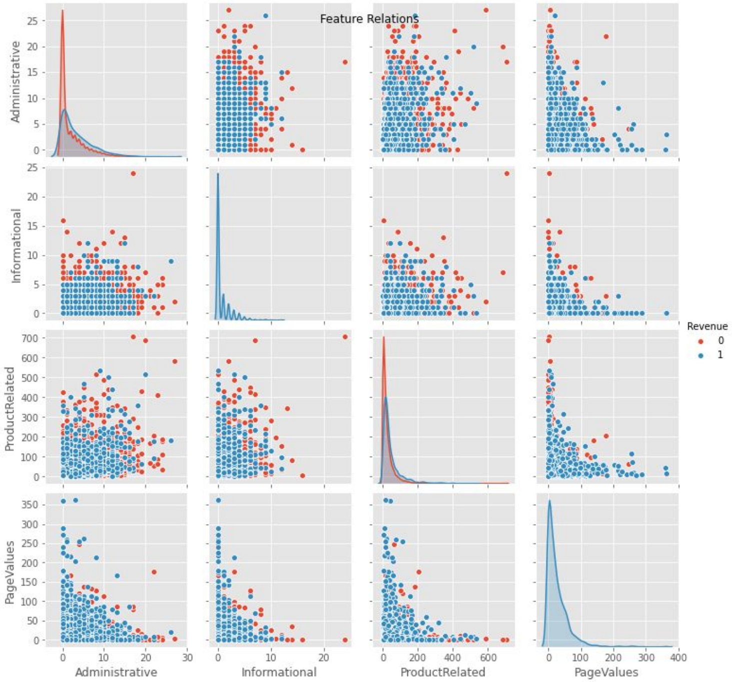
Correlation Analysis



Thanks to this correlation plot we can see that Revenue, our class label isn't very related to our features. For the best, we have a correlation of 0.5 with the Page Values. The rest is really minor we have some features with a correlation of 0.2 such as ProductedRelated. We will also keep them for further analysis.

DATASET DESCRIPTION

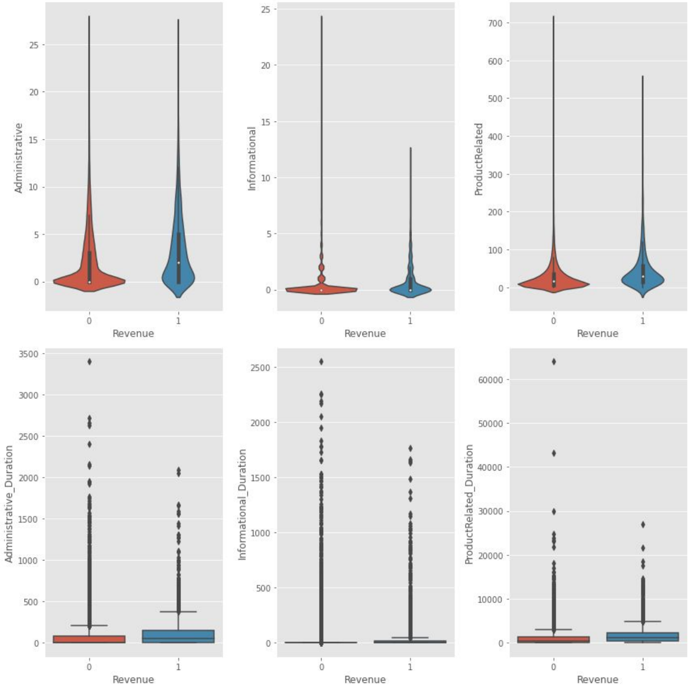
Correlation Analysis



Thanks to this pair plot we can confirm that Revenue, our class label isn't very related to our feature. The repartition of Revenue values (0 or 1) are quite messy we can't discern 2 populations with each pair of feature. The best one is Administrative in y-axis with Product Related in x-axis the red and blue dots are more dispatch.

DATASET DESCRIPTION

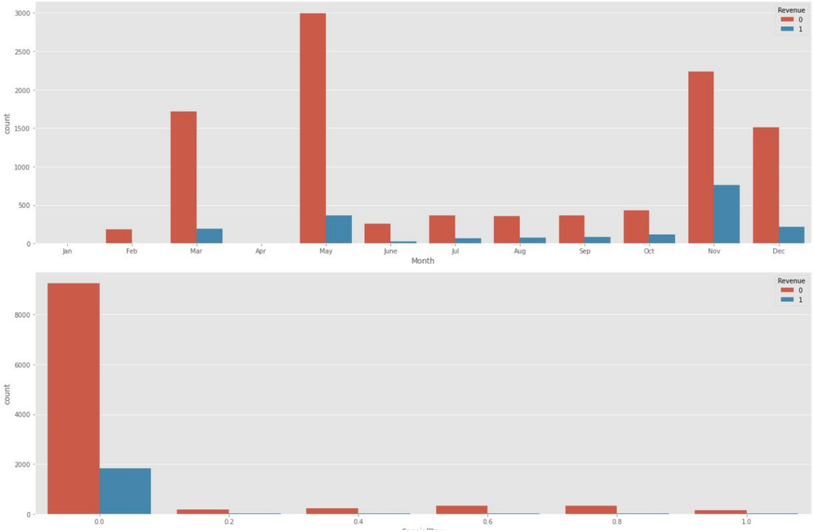
WebPage Analysis



We can see from these boxplots that, the visitors tend to visit less pages, and spend less time, if they are not going to make a purchase. The second analysis that we can make is that the time spent on product related pages is way higher than that for account related of informational pages. The most relevant information for our graphics is that the first 3 features look like they follow a skewed normal distribution.

DATASET DESCRIPTION

WebPage Analysis



On March and May, we have a lot of visits (May is the month with the highest number of visits), yet transactions made during those 2 months are not on the same level. We have no visits at all during Jan nor Apr.

Also, Most transactions happen during the end of the year, with Nov as the month with the highest number of confirmed transactions.

We can observe that the closer the visit date to a special day (like black Friday, new year's, … etc) the more likely it will end up in a transaction.

Finaly, most of transactions happen on special days (SpecialDay =0).

Analysis Method

Like we saw in the introduction part, to perform our analysis we need to predict the Revenue features but it has only 2 output True or False (buy or not buy). Hence, using a classification method is the best option because Classification algorithms in machine learning use input training data to predict the likelihood that subsequent data will fall into one of the predetermined categories

Classification Methods

SVM

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new inputs.

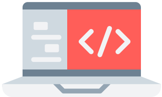
Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable

Random Forest

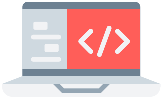
Random forests or random decision forests are an ensemble learning method for classification, it operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes

Classification Methods



Gradient Boost

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.



ADA Boost

AdaBoost, short for "Adaptive Boosting". It focuses on classification problems and aims to convert a set of weak classifiers into a strong one.

Results

SVM

This classifier is known for separating data points using a hyperplane with the largest amount of margin. The SVM used in this project using scikit-learn library finds an optimal hyperplane which helps in classifying new data points.

SVM Initial Performance :

Accuracy	0.8875067604110329
F1-Score	0.5856573705179283
Precision	0.7577319587628866
Recall	0.4772727272727273

Confusion Matrix

1494	47
161	147

Results

SVM

We will tune our SVM Model by modifying his settings :

1. Kernel: Transforms the given dataset into the required form. There are various types of functions such as linear, polynomial, and radial basis function (RBF). Polynomial and RBF are useful for non-linear hyperplane. This transformation can lead to more accurate classifiers.

2. Regularization: C parameter used to maintain regularization. A smaller value of C creates a small-margin hyperplane and a larger value of C creates a larger-margin hyperplane.

3. Gamma: A lower value of Gamma will loosely fit the training dataset, whereas a higher value of gamma will exactly fit the training dataset, which causes over-fitting. A low value of gamma considers only nearby points in calculating the separation line, while a large value of gamma considers all the data points in the calculation of the separation line.

Results

SVM

After fitting 5 folds for each of 80 candidates, totalling 400 fits we obtain :

SVM Tuned Performance :

Accuracy	0.8891292590589508
F1-Score	0.6003898635477583
Precision	0.751219512195122
Recall	0.5

Confusion Matrix

1490	51
154	154

Results

Logistic Regression

Logistic Regression is known to measure the relationship between the categorical dependant variable and the independent variables by estimating probabilities using logistic function.

Logistic Regression Initial Performance :

Accuracy	0.8788534342888048
F1-Score	0.5313807531380752
Precision	0.7470588235294118
Recall	0.41233766233766234

Confusion Matrix

1498	43
181	127

Results

Logistic Regression

After fitting 5 folds for each of 80 candidates, totalling 400 fits we obtain :

Logistic Regression Tuned Performance :

Accuracy	0.879394267171444
F1-Score	0.5285412262156448
Precision	0.7575757575757576
Recall	0.4058441558441584

Confusion Matrix

1501	40
183	125

Results

Random Forest

Random Forest is a meta estimator that uses decision tree classifiers on various sub-sample of the dataset to fit them into theses tree classifiers and uses averaging to improve the predictive accuracy and control the over-fitting.

Random Forest Initial Performance :

Accuracy	0.8988642509464575
F1-Score	0.6581352833638026
Precision	0.7531380753138075
Recall	0.5844155844155844

Confusion Matrix

1482	59
128	180

We can see the random forest classifier gives us higher accuracy and F1 score than other classifiers that we have tested. Let's now try to improve its performance.

Results

Random Forest

Scikit-Learn documentation tells us the most important settings are the number of trees in the forest (n_estimators) and the number of features considered for splitting at each leaf node (max_features). We will try adjusting the following set of hyperparameters:

1. n_estimators = number of trees in the forest set
2. max_features = max number of features considered for splitting a node
3. max_depth = max number of levels in each decision tree
4. min_samples_split = min number of data points placed in a node before the node is split
5. min_samples_leaf = min number of data points allowed in a leaf node

Due to the large number of parameters and parameter values to be tested, we will use random search this time.

Results

Random Forest

After fitting 5 folds for each of 80 candidates, totalling 400 fits we obtain :

Random Forest Tuned Performance :

Accuracy	0.9015684153596538
F1-Score	0.6604477611940297
Precision	0.7763157894736842
Recall	0.5746753246753247

Confusion Matrix

1490	51
131	177

We can see that the tuned random forest has given us the best performance. Let's try another classification model.

Results

Gradient Boost

Gradient Boosted Regression Trees is a regression classification model used to produce a prediction model in the form of an ensemble of weak prediction models.

Gradient Boost Initial Performance :

Accuracy	0.9053542455381287
F1-Score	0.6891651865008881
Precision	0.7607843137254902
Recall	0.6298701298701299

Confusion Matrix

1480	61
114	194

Results

Gradient Boost

We will try adjusting the following set of hyperparameters:

1. n_estimators = number of trees in the forest
2. loss = loss function to be optimized
3. learning_rate = shrinks the contribution of each classifier
4. subsample = The fraction of samples to be used for fitting the individual base learners
5. max_features = max number of features considered for splitting a node
6. max_depth = max number of levels in each decision tree
7. min_samples_split = min number of data points placed in a node before the node is split
8. min_samples_leaf = min number of data points allowed in a leaf node

Due to the large number of parameters and parameter values to be tested, we will use random search this time.

Results

Gradient Boost

After fitting 5 folds for each of 80 candidates, totalling 400 fits we obtain :

Gradient Boost Tuned Performance :

Accuracy	0.8902109248242294
F1-Score	0.5814432989690722
Precision	0.7966101694915254
Recall	0.4577922077922078

Confusion Matrix

1505	36
167	141

The performance of Gradient Boost with the default parameter values is actually slightly better than that of the tuned version.

Results

ADABoost

It looks like ADABoost has a lower performance than Gradient Boost; We will try to improve it by tuning its parameters.

ADABoost Initial Performance :

Accuracy	0.8848025959978366
F1-Score	0.6148282097649187
Precision	0.6938775510204082
Recall	0.551948051948052

Confusion Matrix

1466	75
138	170

Results

ADABoost

We will try adjusting the following set of hyperparameters:

1. n_estimators = number of trees in the forest
2. learning_rate = shrinks the contribution of each classifier

After fitting 5 folds for each of 80 candidates, totalling 400 fits we obtain :

AdaBoost Tuned Performance :

Accuracy	0.8885884261763115
F1-Score	0.624087591240876
Precision	0.7125
Recall	0.5551948051948052

Confusion Matrix

1472	69
137	171

Results

Model Validation Results

For the Gradient Boosting the parameter that could improve the model's accuracy and performance is the number of estimators so we will increase its value from 100 to 131. Let's see what does it give :

Gradient Boosting Tuned Performance :

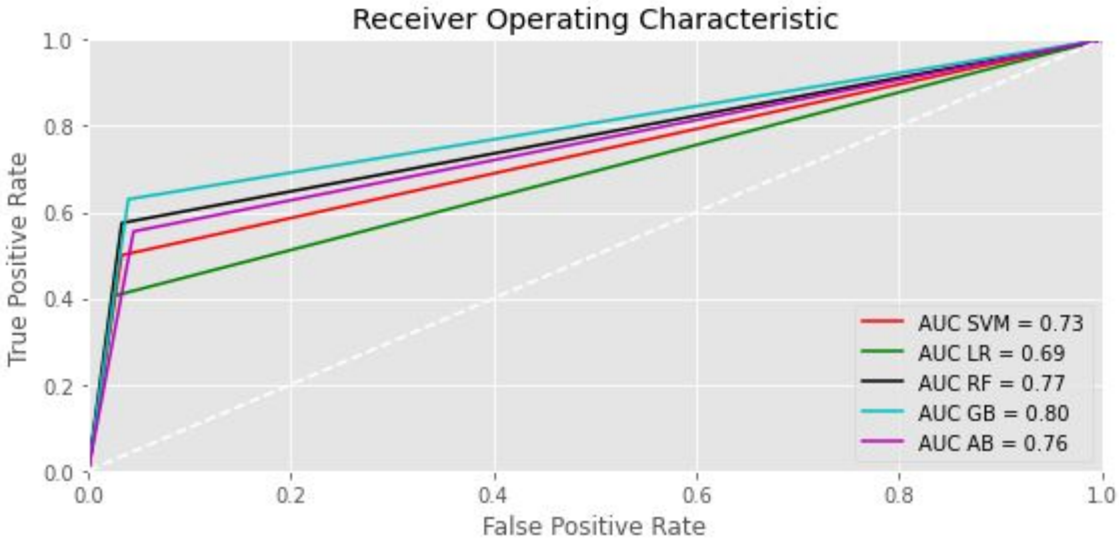
Accuracy	0.9064359113034073
F1-Score	0.6927175843694494
Precision	0.7647058823529411
Recall	0.6331168831168831

Confusion Matrix

1481	60
113	195

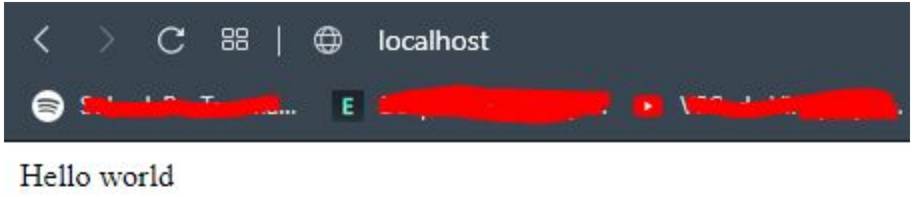
Results

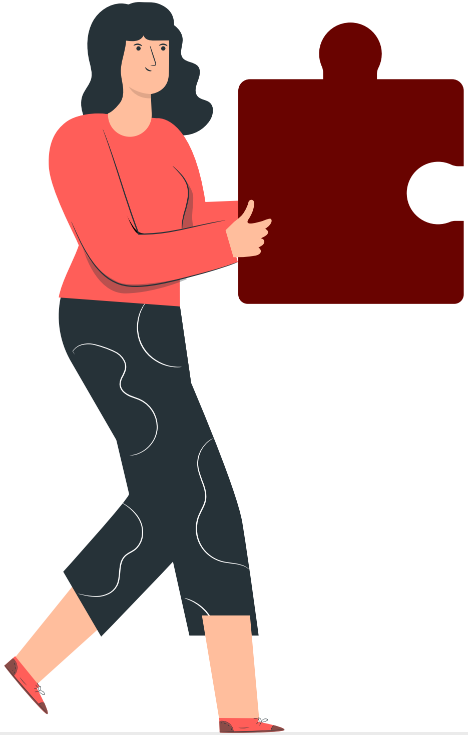
ROC Curves



API

For the API part of the project, we used the Flask API so that we can pre-process the data that someone sends.
With pickle we saved our model in the .SAV format so that we can load it and then we apply it to the data that goes through our API.



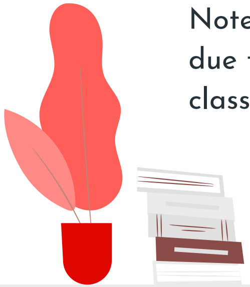


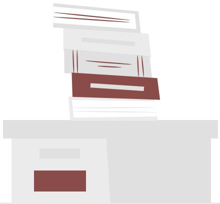
CONCLUSION

In this project, we used Online Shoppers Intention dataset to build models that can classify website visitor, and predict which of them is likely going to make a purchase on the website. 5 different learning classifiers (Logistic Regression, Random Forest, Gradient Boosting, and Adaboosting) were tested and optimized, and we have achieved the best classification performance using Gradient Boost classifier, followed by random Forest, and then Adaboost.
The best classification performance:

Accuracy: 91%
F1 Score: 0.66

Note: There is a clear difference of classification performance between the 2 classes, that is mainly due to the unbalanced nature of our dataset, where around 85% of our data points belong to 1 class, and less than 15% belong to the other.





THANK YOU

Amir MAHMOUDI
Yassine LAHBABI