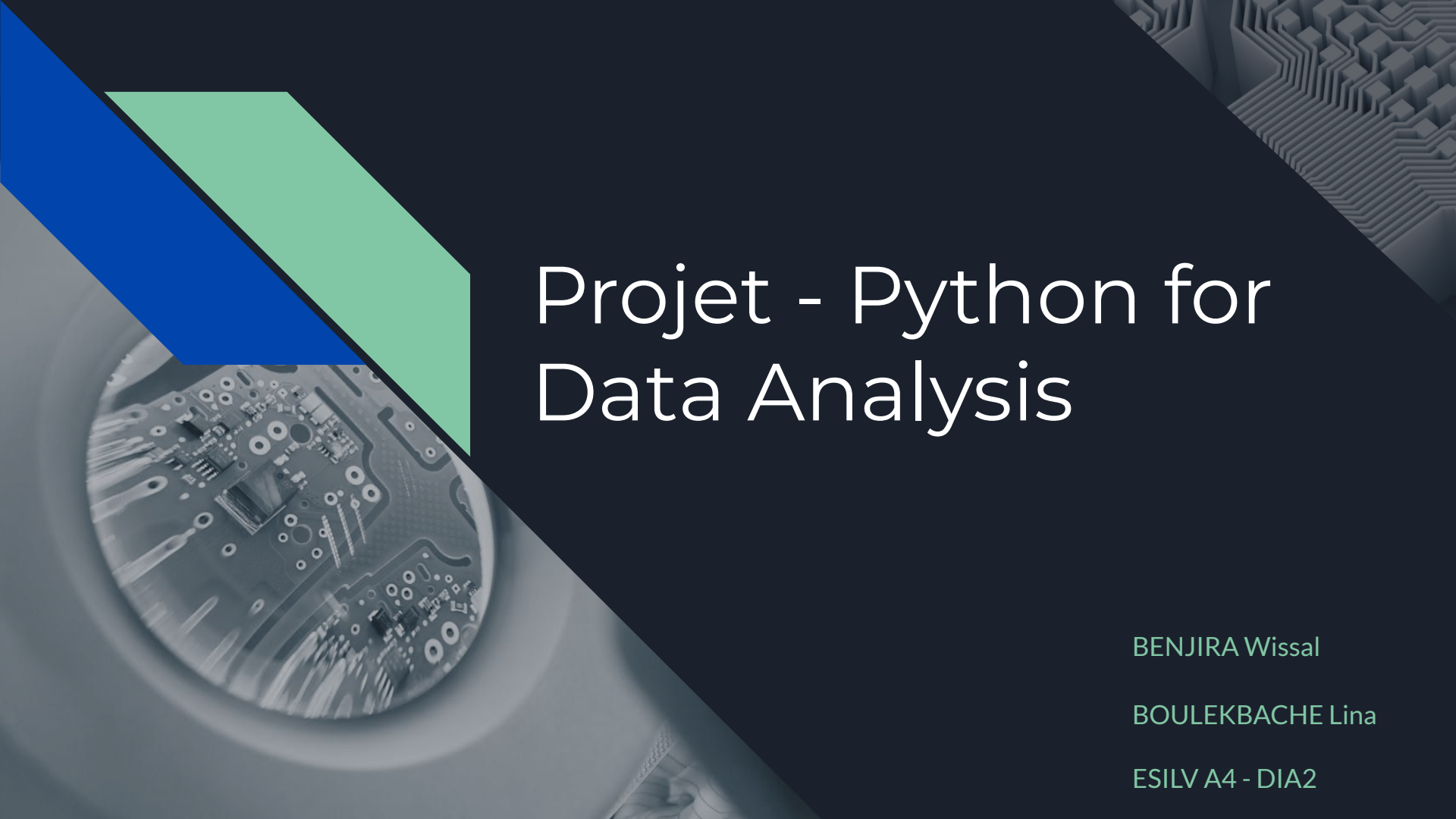


Projet - Python for Data Analysis

BENJIRA Wissal

BOULEKBACHE Lina

ESILV A4 - DIA2

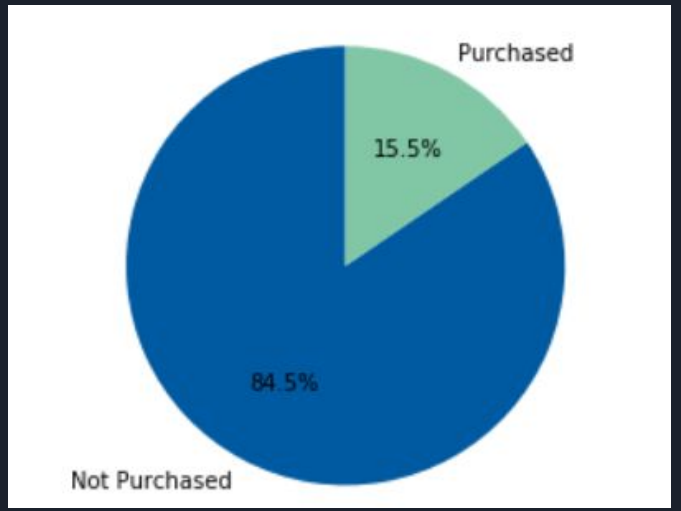
TABLE DES MATIÈRES

1. Introduction du jeu de données

a. Description du dataset

- Cette étude porte sur la base de données suivante : https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset
- Deux scientifiques turques C.Okan Sakar et Yomi Kastro sont à la tête de cette étude.
- Le dataset regroupe 12 330 différents clients/acheteurs en ligne sur lesquels sont étudiées 18 variables au cours d'une année.
- La variable cible est "Revenue" (True si le client achète, False sinon)

Répartition du choix des clients



Seuls 1908 sur 12 330 clients achètent .

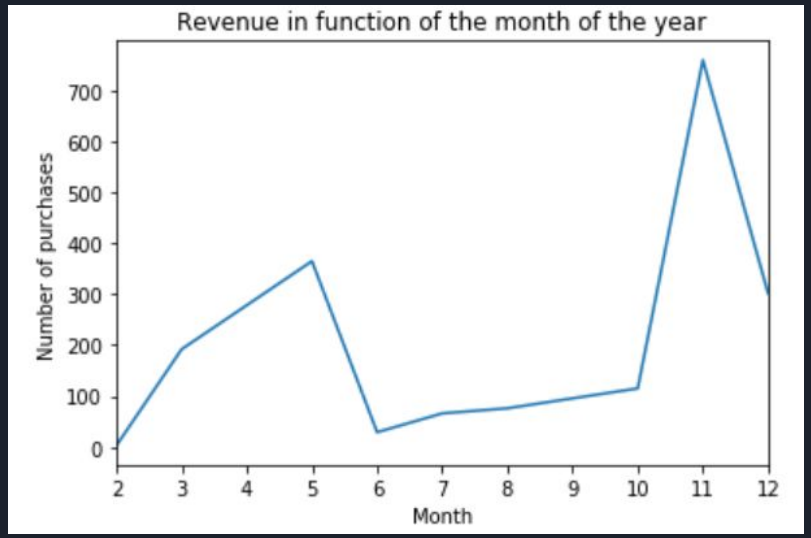
b. Data exploration

Objectif : Analyser ces données pour prédire si le client va consommer sur le site.

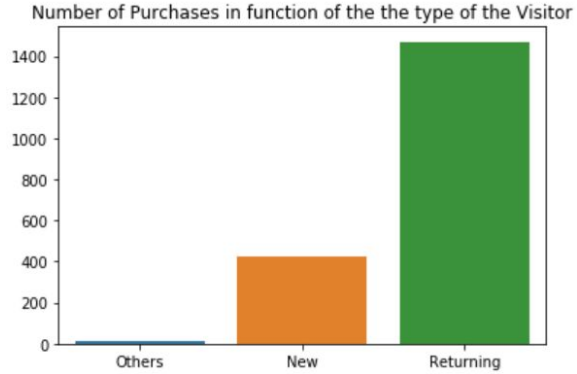
	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues	SpecialDay	Month	OperatingSystems	Browser	Region	TrafficType	VisitorType	Weekend	Revenue
0	0	0.0	0	0.0	1	0.000000	0.200000	0.200000	0.000000	0.0	Feb	1	1	1	1	Returning_Visitor	False	False
1	0	0.0	0	0.0	2	64.000000	0.000000	0.100000	0.000000	0.0	Feb	2	2	1	2	Returning_Visitor	False	False
2	0	0.0	0	0.0	1	0.000000	0.200000	0.200000	0.000000	0.0	Feb	4	1	9	3	Returning_Visitor	False	False
3	0	0.0	0	0.0	2	2.666667	0.050000	0.140000	0.000000	0.0	Feb	3	2	2	4	Returning_Visitor	False	False
4	0	0.0	0	0.0	10	627.500000	0.020000	0.050000	0.000000	0.0	Feb	3	3	1	4	Returning_Visitor	True	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
12325	3	145.0	0	0.0	53	1783.791667	0.007143	0.029031	12.241717	0.0	Dec	4	6	1	1	Returning_Visitor	True	False
12326	0	0.0	0	0.0	5	465.750000	0.000000	0.021333	0.000000	0.0	Nov	3	2	1	8	Returning_Visitor	True	False
12327	0	0.0	0	0.0	6	184.250000	0.083333	0.086667	0.000000	0.0	Nov	3	2	1	13	Returning_Visitor	True	False
12328	4	75.0	0	0.0	15	346.000000	0.000000	0.021053	0.000000	0.0	Nov	2	2	3	11	Returning_Visitor	False	False
12329	0	0.0	0	0.0	3	21.250000	0.000000	0.066667	0.000000	0.0	Nov	3	2	1	2	New_Visitor	True	False

- Administrative, Informal et ProductRelated → Nombre des types de pages visités par le client à chaque session
- Administrative_Duration, Informal_Duration et ProductRelated_Duration → Temps total passé sur chaque catégorie de page
- BounceRate et ExitRate → Pourcentage d'entrées et de sorties sur le site en passant par une page
- Page Value → Métrique calculée par GoogleAnalytics
- SpecialDay → Proximité de la visite du site par rapport à un jour spécial de l'année
- OperationSystem, Browser, Region, TrafficType → Variables étudiées
- VisitorType → New visitor, Returning visitor ou other
- Weekend → Boolean True si la visite a lieu durant le weekend False sinon

c. Data visualisation

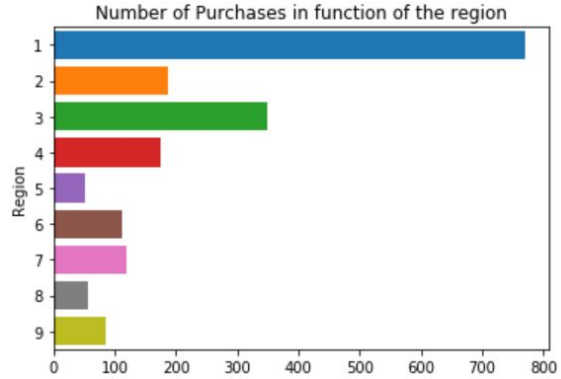


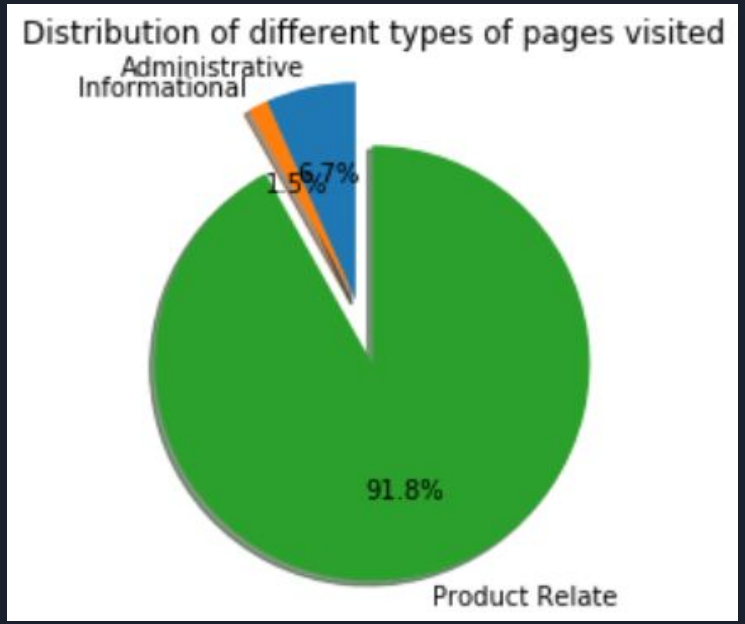
- Le nombre d'achat sur le site est majoritaire en fin d'année tout particulièrement à l'approche de Noël, Black Friday et du nouvel an
- Se présente également un bond entre Mars et Juin



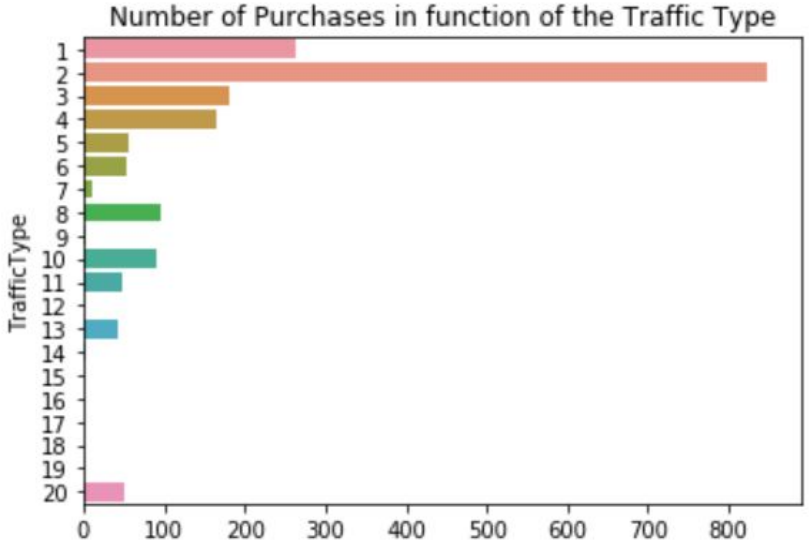
- La grande partie des acheteurs sont des visiteurs qui sont déjà passés par le site auparavant
- Les nouveaux visiteurs représentent un nombre non négligeable d'acheteurs

- Les visiteurs de la région 1 et 3 sont les plus nombreux à acheter sur le site





- La plupart des visiteurs traversent dans la grande majorité des pages liées aux produits vendus sur le site



- Le type de traffic 2 représente un nombre important d'achat, suivi du type 1,3 et 4.

2. Modèles de prédiction

a. Algorithmes manuels

	p_value
TrafficType	5.702434e-01
Region	1.979426e-01
OperatingSystems	1.033943e-01
Browser	7.736888e-03
Weekend	1.140563e-03
Informational_Duration	5.282871e-15
SpecialDay	5.498934e-20
Administrative_Duration	2.146514e-25
Informational	3.174034e-26
VisitorType	5.861360e-28
Month	9.465664e-46
Administrative	3.519760e-54
BounceRates	1.594198e-63
ProductRelated_Duration	6.115338e-65
ProductRelated	3.241187e-70
ExitRates	1.662654e-119

ANalyse of VAriance (ANOVA Test)

Chaque variable est testée par rapport à la target 'Revenue' pour vérifier si la variable de test est significative quant à la prédiction souhaitée.

Au cours de ce test d'hypothèse, on suppose que la variable de test et la réponse sont indépendantes.

La p_value obtenue est comparée au seuil d'erreur choisi α.

Si la p_value < α alors l'hypothèse est rejetée ⇒ la variable est significative.

Dans ce cas, en posant α = 1%, les variables 'TrafficType', 'Region' et 'OperatingSystems' ne sont pas significatives pour prédire 'Revenue'.

Toutefois, nous décidons de ne pas les retirer du dataset pour avoir une prédiction totale.

```
#We split the dataset
X=data[['Administrative','Administrative_Duration','Informational','Informational_Duration','ProductRelated',
        'ProductRelated_Duration','BounceRates','ExitRates','PageValues','SpecialDay',
       'Month','OperatingSystems','Browser','Region','TrafficType','VisitorType','Weekend']]
y=np.ravel(data[['Revenue']])
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
```

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(random_state=0)
clf.fit(X, y)
clf_pred = clf.predict(X_test)
clf_score = clf.score(X_test,y_test)
clf_score
```

- Le dataset est séparé en 2 tableaux
  - X : tableau contenant les prédicteurs
  - y : tableau de la réponse
- Chaque tableau est divisé en 2 sous tableaux de training et de testing

- Chaque modèle est testé à la manière de RandomForestClassifier ci-dessous :
  - .fit() : établit le modèle sur le dataset complet
  - .predict() : effectue la prédiction sur le tableau de test
  - .score() : calcule à l'aide d'une métrique, la performance du modèle

b. Grid Search

- **Grid Search est utilisé pour tuner et pour déterminer les hyper paramètres qui maximisent la performance du modèle (permet d'éviter l'overfitting)**

```
=========================================================
 Results from Grid Search
=========================================================

 The best estimator across ALL searched params:
 GradientBoostingClassifier(criterion='friedman_mse', init=None,
                            learning_rate=0.01, loss='deviance', max_depth=4,
                            max_features=None, max_leaf_nodes=None,
                            min_impurity_decrease=0.0, min_impurity_split=None,
                            min_samples_leaf=1, min_samples_split=2,
                            min_weight_fraction_leaf=0.0, n_estimators=500,
                            n_iter_no_change=None, presort='auto',
                            random_state=None, subsample=0.9, tol=0.0001,
                            validation_fraction=0.1, verbose=0,
                            warm_start=False)

 The best score across ALL searched params:
 0.9012652752243971

 The best parameters across ALL searched params:
 {'learning_rate': 0.01, 'max_depth': 4, 'n_estimators': 500, 'subsample': 0.9}

 =========================================================
```

c. Version automatisée : xgboost

- Cette fonction parcourt tous les sous-modules de sklearn et sélectionne toutes celles qui présentent une fonction ‘fit’.
- xgboost teste tous les modèles de Machine Learning qui existent et calcule leur score à la suite de la prédiction.
- Certaines erreurs apparaissent telles que :

```
'TruncatedSVD' object has no attribute 'score'
```

du fait que certaines fonctions ont un ‘ fit’ mais ne sont pas des modèles. Nous ne traiterons pas ces dernières.

- Il en est de même pour certains Warnings comme le suivant :

```
C:\Users\33664\Anaconda3\lib\site-packages\sklearn\decomposition\sparse_pca.py:170: DeprecationWarning: normalize_components=Fa
lse is a backward-compatible setting that implements a non-standard definition of sparse PCA. This compatibility mode will be r
emoved in 0.22.
  DeprecationWarning)
```

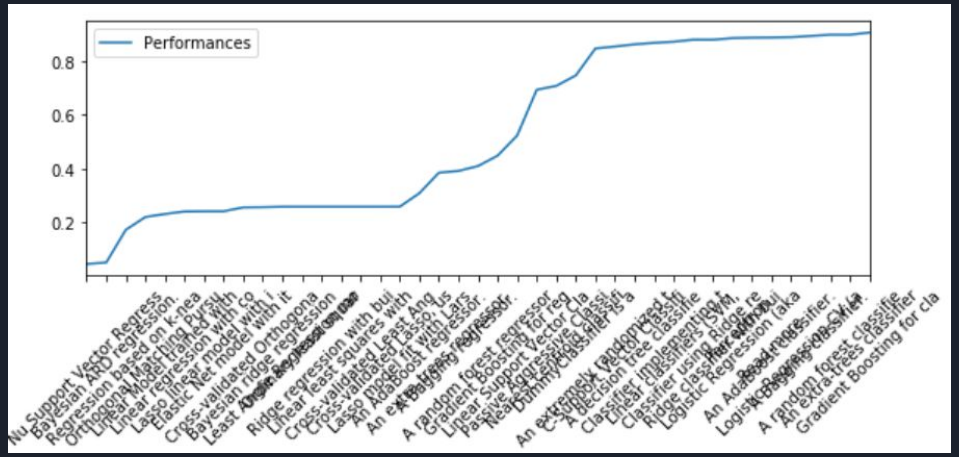
3. Sélection du modèle

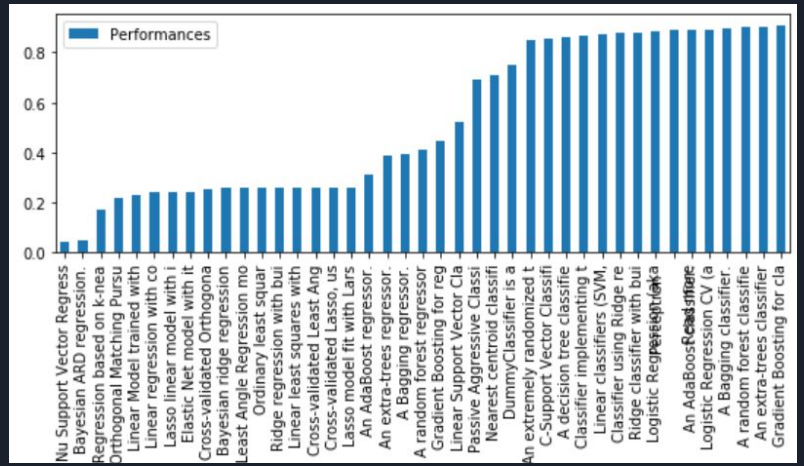
a. Calcul des métriques

- Dans ce dataframe sont regroupés tous les modèles manuels testés un à un ainsi que le score de leur prédiction
- Le score est d'autant plus élevé que la prédiction est correcte et précise
- Les meilleurs modèles sont des modèles de classification
- D'après cette étude, le Gradient Boosting Classifier est le plus performant des 7 modèles testés

	Scores
GradientBoostingClassifier	0.906909
RandomForestClassifier	0.825375
BaggingRegressor	0.400063
LinearRegression	0.258889
KNeighborsRegressor	0.172872
DecisionTreeRegressor	0.075381
SVR	0.002868

b. Graphe de comparaison





- Le choix du modèle de Gradient Boosting Classifier se confirme par le test automatisé de xgboost

4. Réalisation de l'API Flask

a. Création de l'API

Utilisation de la librairie Joblib pour créer un fichier *model_saved.joblib* contenant le modèle retenu (*Gradient Boosting Classifier)*

Le fichier *app.py* permet de lancer le projet reliant l'interface, le modèle et les templates.

request.py a servi à la validation du modèle et de l'app.py avant la création de l'interface.

index.html contient le formulaire à remplir avec les données dont on souhaite prédire la réponse.

result.html correspond à la page qui affiche la prédiction.

- *model_saved.joblib*
- *app.py*
- *request.py*
- *index.html*
- *result.html*

b. Test en local

Lancement des fichiers *app.py* et *request.py*:

Les données d'un visiteur (une ligne du dataset) sont entrées dans le modèle.

Le fichier *request.py* retourne la prédiction.

```
C:\Users\33664\Documents\S7\python_for_data_analysis\flask>python app.py
C:\Users\33664\AppData\Local\Programs\Python\Python38\lib\site-packages\sklearn\base.py:329: UserWarning
: Trying to unpickle estimator GaussianNB from version 0.21.3 when using version 0.23.2. This might lead
 to breaking code or invalid results. Use at your own risk.
  warnings.warn(
 * 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
C:\Users\33664\AppData\Local\Programs\Python\Python38\lib\site-packages\sklearn\base.py:329: UserWarning
: Trying to unpickle estimator GaussianNB from version 0.21.3 when using version 0.23.2. This might lead
 to breaking code or invalid results. Use at your own risk.
  warnings.warn(
 * Debugger is active!
 * Debugger PIN: 392-932-484
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [10/Jan/2021 00:05:12] "[37mPOST /local/ HTTP/1.1[0m" 200 -
```

app.py

```
C:\Users\33664\Documents\S7\python_for_data_analysis\flask>python request.py
<Response [200]> "The visitor is not going to purchase."
```

request.py

c . Démonstration

Lancement du fichier *app.py* après création des templates.

SUBMIT

The visitor is going to purchase.