



# Online Shoppers Purchasing Intention Dataset

Christian AHADJI

Valentin FOSSE

Source : Sakar, C.O., Polat, S.O., Katircioglu, M. et al. *Neural Comput & Applic* (2018)

# Description



Ce dataset porte sur l'intention d'achat des acheteurs en ligne.

Il a été formé sur une période d'un an. Ceci pour éviter toute tendance à une campagne spécifique, un jour spécial, un profil utilisateur ou une période.

Chaque **ligne** appartient à une **session** d'un utilisateur.

Quelques points importants :

- Parmi les **12 330 sessions**, 84.5% (10 422) se terminent sans achat
- Il y a **18 attributs** : 10 numériques et 8 catégoriques
- Aucune valeur n'est manquante

# Problème

C'est un problème de classification.

Chaque session a un attribut nommé *Revenue*, il traduit le fait que l'utilisateur ait procédé à un achat durant cette session.

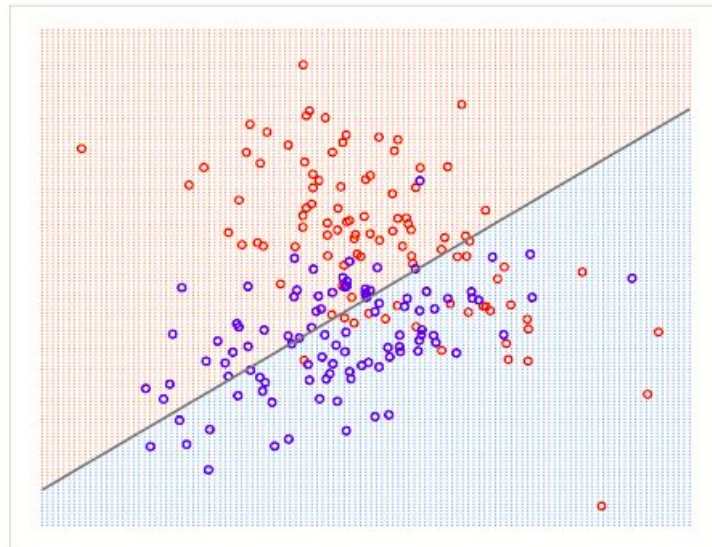
On peut donc, grâce à un modèle, prédire l'intention d'achat de chaque nouveau visiteur sur le site, d'où ce nom de dataset.

Cette classification est dite supervisée car la donnée est bien labellisée. Sa target value est *Revenue*.

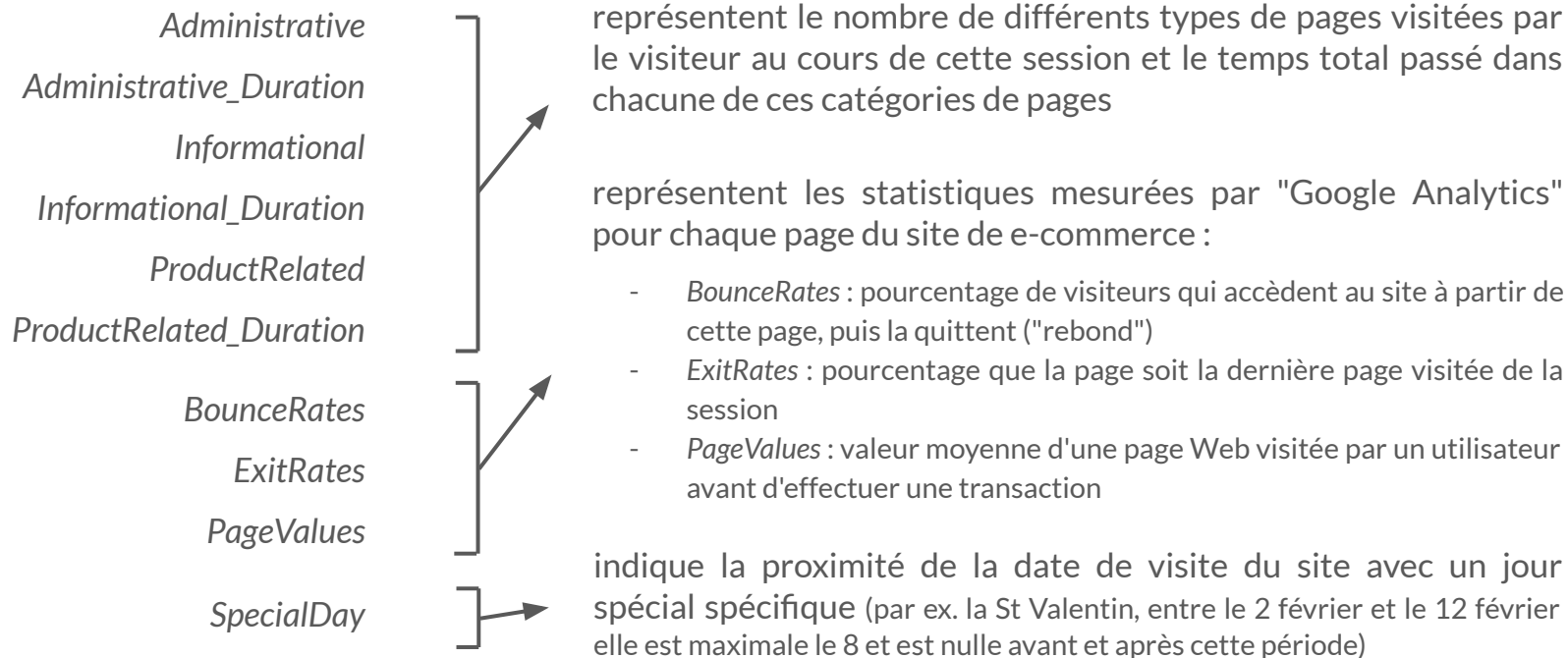
Les autres variables de ce problème sont décrites par la suite.

Supervised Classification

Linear Model



# Variables numériques



# Variables catégoriques



*Month* : le mois durant lequel le visiteur se connecte

*OperatingSystems* : le système d'exploitation

*Browser* : le navigateur internet

*Region* : la région de connexion

*TrafficType* : le type de trafic

*VisitorType* : le type de visiteur, s'il est nouveau ou s'il est déjà venu

*Weekend* : un booléen, vrai si c'est le weekend

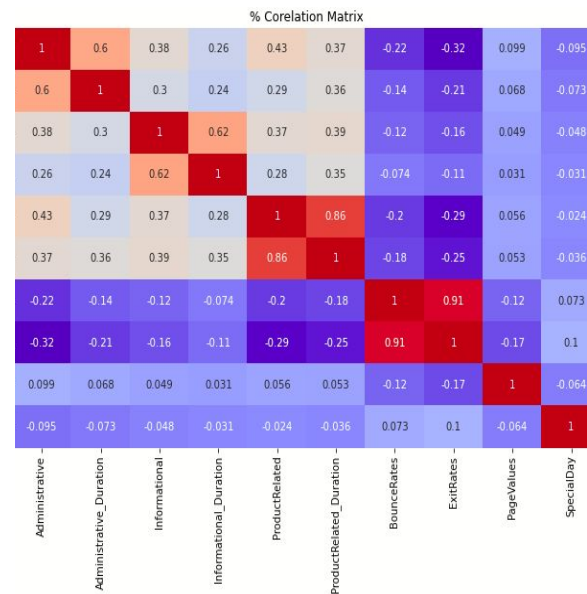
*Revenue* : notre variable de classification

# Variables créées

Deux variables ont été créées car l'étude menée a montré que certains attributs étaient très corrélés :

- *Administrative\_Duration* et *Administrative*  
→ ***ratio\_administrative*** : elle peut être interprétée comme le temps passé par page dans la catégorie *Administrative*.
- *Informational\_Duration* et *Informational*  
→ ***ratio\_informational*** : celle-ci par le temps passé par page dans la catégorie *Informational*.

Cela permet d'avoir un modèle plus simple, sans dédoublement de variables qui influe sur la précision et la stabilité du modèle.



# Choix du modèle (1)



Après un pre-processing des données (fusion de variables corrélées, création de dummies variables, partitionnement du dataset en train et test set, normalisation des données), elles sont maintenant exploitables.

Neuf algorithmes ont été appliqués pour trouver le meilleur modèle, avec une cross validation et/ou une grille de réglage des hyperparamètres suivant les cas. Voici leur score final :

- *KNN* : 0.8691
- *Régression logistique* : 0.8758
- *Arbre de décision de classification* : 0.8933
- *Forêt aléatoire* : 0.9015
- *Support Vector Machine* : 0.8764

# Choix du modèle (2)



- *Bagging*: 0.8691
- *AdaBoost Classifier* : 0.8758
- *ExtraTree Classifier* : 0.8933
- *Forêt aléatoire* : 0.9015
- ***Stochastic Gradient Boosting*** : 0.9061

Nous retenons donc le modèle du **Gradient Stochastique**, qui obtient le meilleur score.

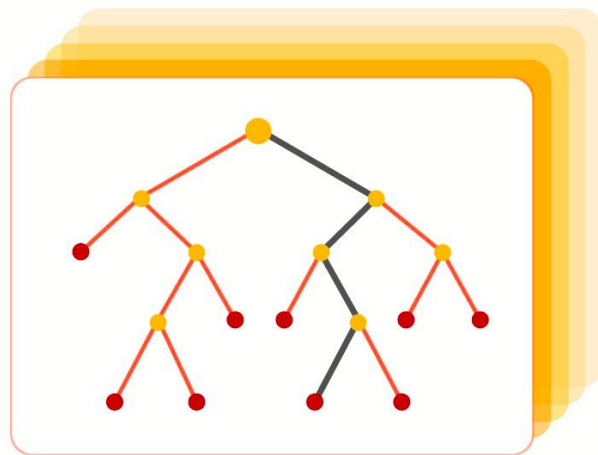


# Résultat

On a appliqué au modèle de forêt aléatoire une grille de recherche pour optimiser les hyperparamètres du modèle afin d'obtenir le meilleur score.

Celui-ci est obtenu pour ces paramètres :

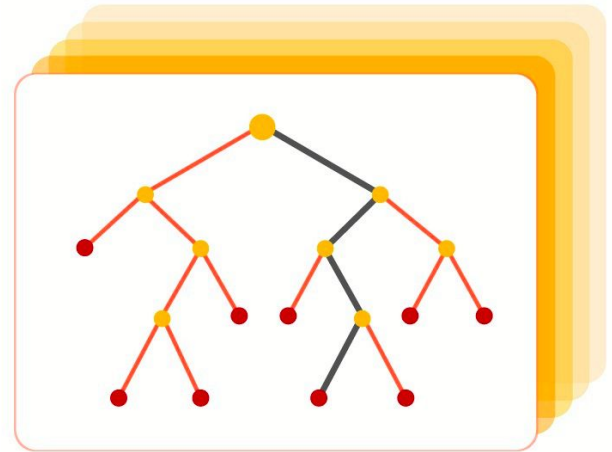
- Nombre d'arbres séquentiels à modéliser (**n\_estimators**) : 88
- Nombre minimal d'échantillons (ou d'observations) qui sont requis dans un nœud pour être pris en compte pour la division (**min\_sample\_split**) : 750
- Échantillons minimum requis dans un nœud terminal ou une feuille (**min\_samples\_leaf**) : 398



# Résultat

- Learning rate: 0.1
- Le nombre de fonctionnalités à prendre en compte lors de la recherche de la meilleure répartition (**max\_features**): 50
- La profondeur maximale de l'arbre (**max\_depth**): 8
- La proportion d'observations à sélectionner pour chaque arbre (**subsample**): 0.7

Le score retenu est **0.9061**.



# Outils utilisés

- Visualisation :  
**Bokeh** (graphiques interactifs), **Seaborn**, **Matplotlib**
- Traitement des données :  
**Pandas**, **Numpy**
- Etude statistiques :  
**Scipy.stats**, **Statsmodels**
- Modélisation :  
**Scikitlearn**

