datazon

SEGMENTEZ LES COMPORTEMENTS DES CLIENTS

Projet 5

Azim Makboulhoussen
18 Avril 2018

Sommaire

- Introduction
- Analyse exploratoire des données
- Classification des clients
- Pistes de modélisations pour classement automatique des clients
- Résultat et implémentation
- Conclusion

Introduction

Objectif du projet

- Mieux comprendre le comportement des clients afin d'augmenter les ventes.
- Segmentation des individus (clients) afin de détecter des catégories de clients.
- Evaluation et amélioration des performances de différents modèles d'apprentissage machine capables de prédire la catégorie d'un client à partir de son historique d'achat.
- Implémentation d'un module capable de classer automatiquement un client dès son premier achat.

Exploration des données

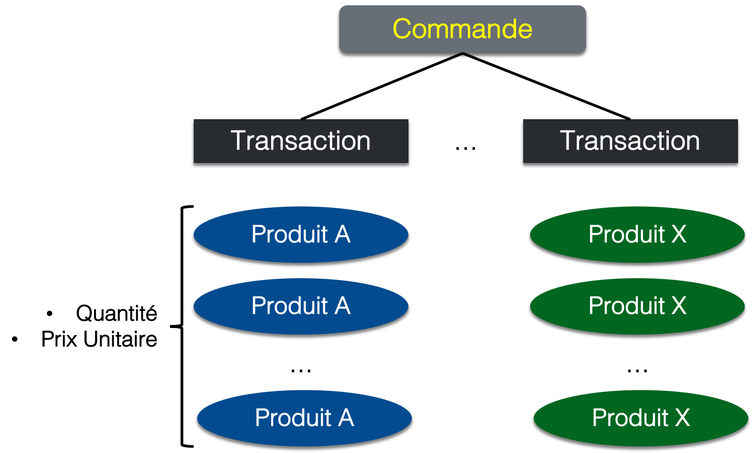
Les données

- Transactions de ventes sur **une année** (Déc. 2010 – Déc. 2011)
- La base contient plus de **500 000** transactions
- Chaque transaction est décrite par **8** variables

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Caractéristiques des données

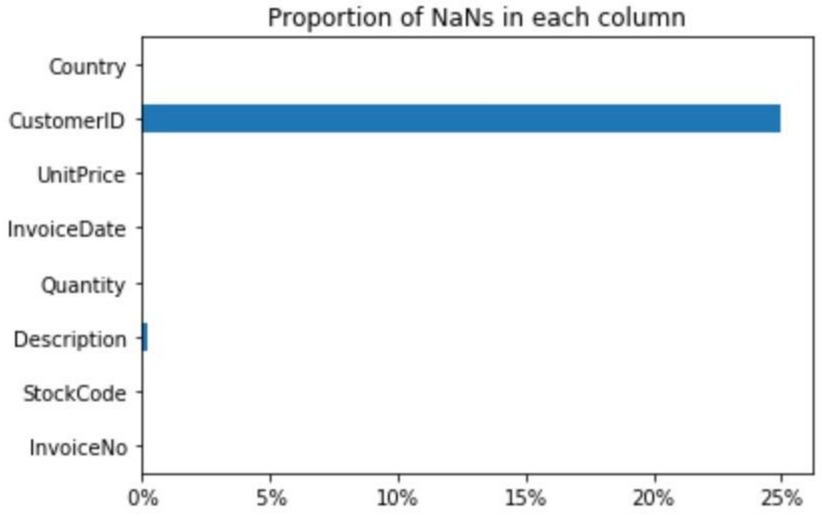
Variable	Type	Description
InvoiceNo	Catégorielle. 6 digits	Numéro de commande unique. commence par c = annulée
StockCode	Catégorielle. 5 digits	Code unique identifiant le produit
Description	Catégorielle. Text	Description du produit
Quantity	Continue. Nombre entier	La quantité du produit commandée dans la transaction
InvoiceDate	Date / Heure (yyyy-mm-jj hh:mi:ss)	Jour et heure de la commande
UnitPrice	Continue. Nombre flottant	Prix par unité en Livre Sterling
CustomerID	Catégorielle. 5 digits	Identifiant unique du client
Country	Catégorielle. Text	Pays de résidence du client



- Compréhension de chacune des variables
- Chaque ligne correspond à une transaction appartenant à une commande.
- Une transaction concerne un produit particulier.
- Les données sont composées de variables :
  - **catégorielles** comme le code du produit ou le pays de l'acheteur
  - **continues** comme la quantité et le prix unitaire

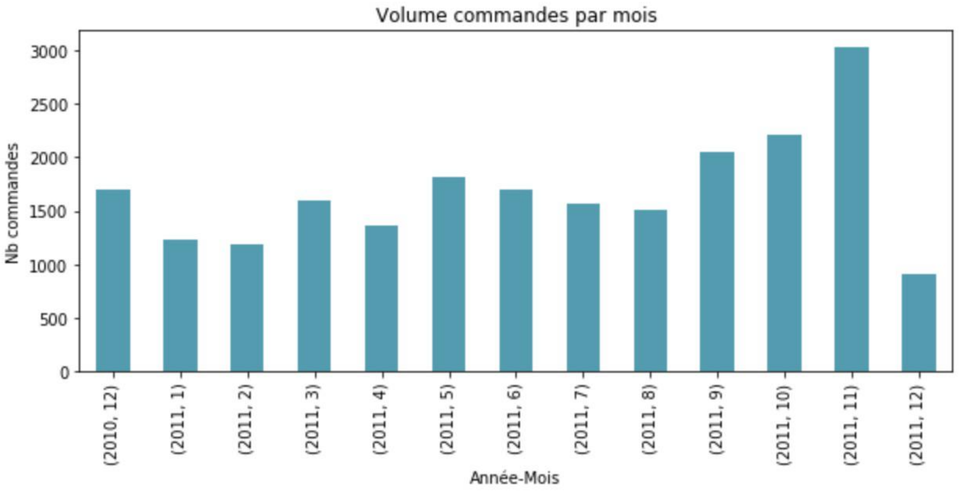
Valeurs manquantes et doublons

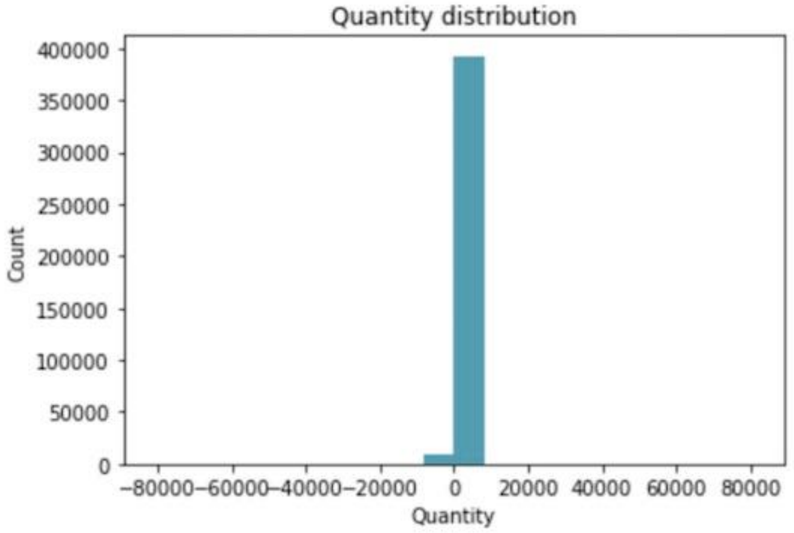
- La base de données est plutôt complète
- La colonne ***CustomerID*** est celle qui contient le plus de valeur vide (25%)
- Quelques valeurs manquantes au niveau ***Description***
- Remplacement des valeurs Description à partir d'autres transactions (même code produit).
- Suppression des lignes sans identifiant client.
- **Doublons** :
  5225 transactions en double => choix de les supprimer



Les commandes

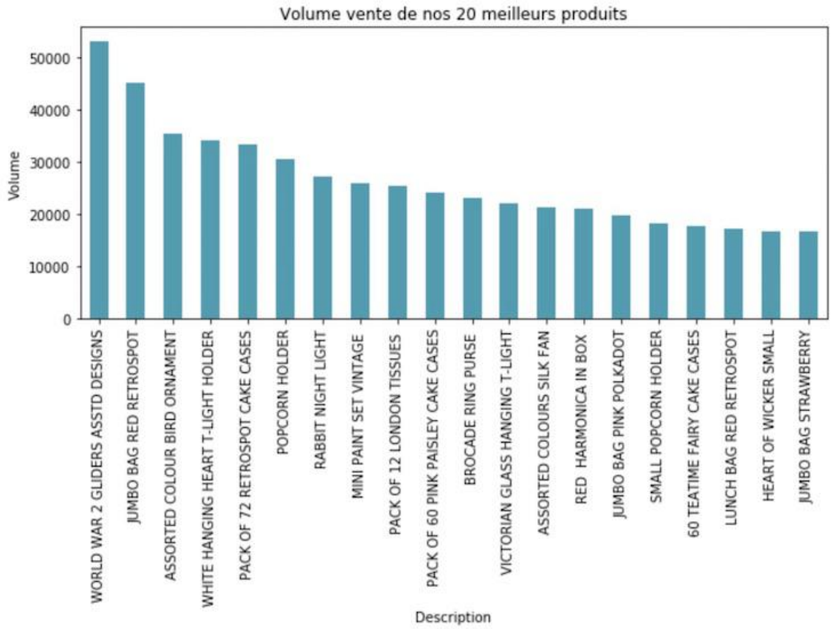
- **Historique :**
  - Du 1er Décembre 2010 au 9 Décembre 2011
  - **4 372 clients** uniques
  - Plus de **22 000** transactions
  - Plus de **3 500** types de produits
  - **8 millions** de £ de CA
  - Augmentation des ventes en fin d'année (à partir septembre)

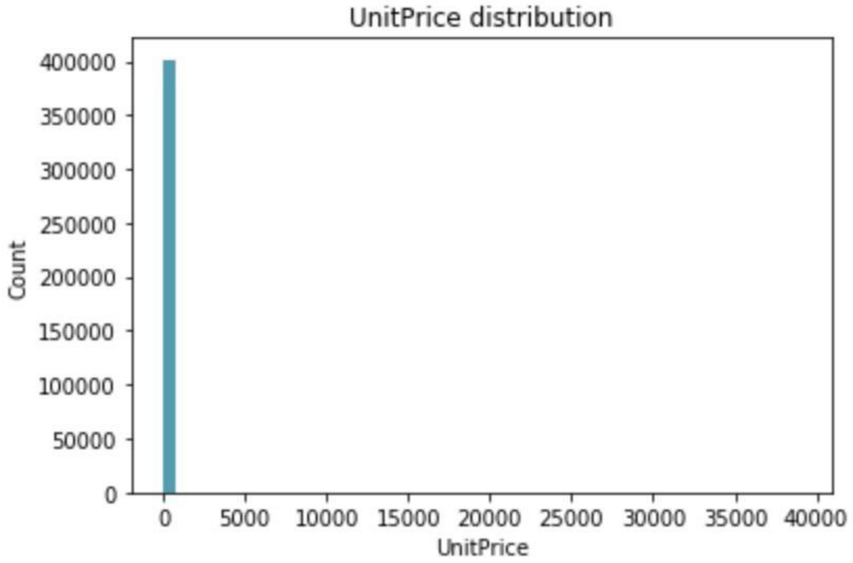




- **Quantité**
  - Quantité < 15 en grande majorité mais existence de valeur extrême allant jusqu'à 80 000
  - Négative = commande annulée
  - Avec code 'D' = Discount
  - Création de colonne spécifique pour indiquer annulation et remise.

Les produits

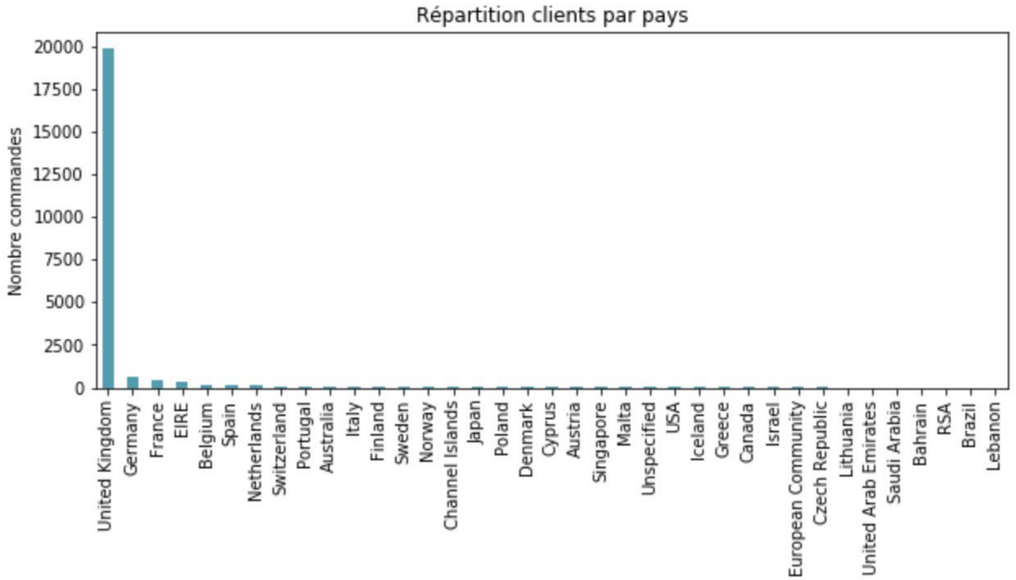




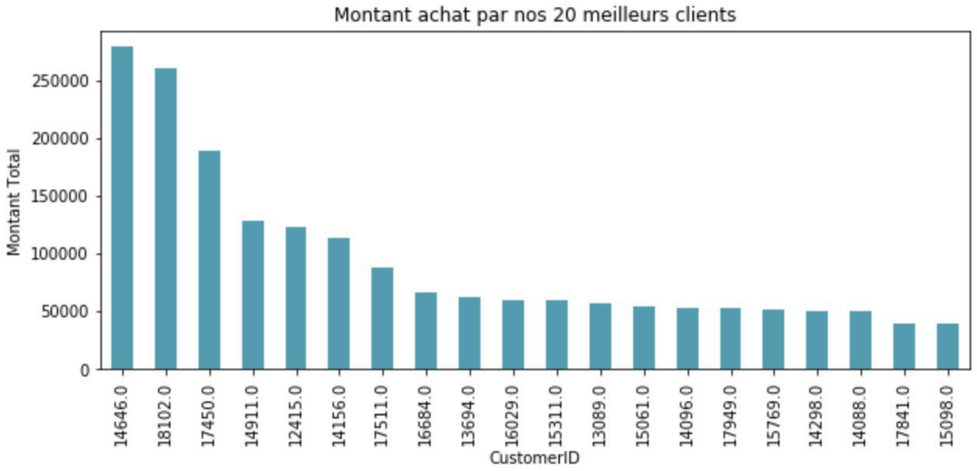
- 20 produits représentent 11% du volume des ventes
- World War 2 gliders et Jumbo Bag Red sont les plus vendus
- Produits avec codes spéciaux (POSTAGE, CARRIAGE, …) => choix de les supprimer pour le projet car correspondent surtout à des frais de ports, ...

- Prix unitaire essentiellement inférieur à 10 £ mais avec quelques produits allant jusqu'à 38 000 £
- Produit à 0 £ => hypothèse promotion
- Création d'une colonne indiquant '**Promotion**'
- Création d'une colonne '**TotalPrice**'

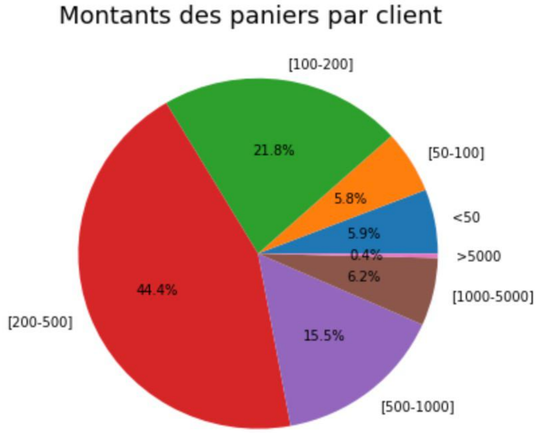
Les clients



- **Pays :**
  - Les clients en très forte majorité Anglais
  - Des commandes à destination de nombreux autres pays (Europe)
  - Ajout d'une colonne pour indiquer si client Anglais ou pas.



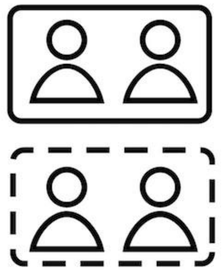
- **Achats :**
  - 20 meilleurs clients (0,5%) représentent 20% du CA
  - Panier moyen client : 380 £
  - La majorité (65%) ont un panier compris entre 100 et 500£



Segmentation clients

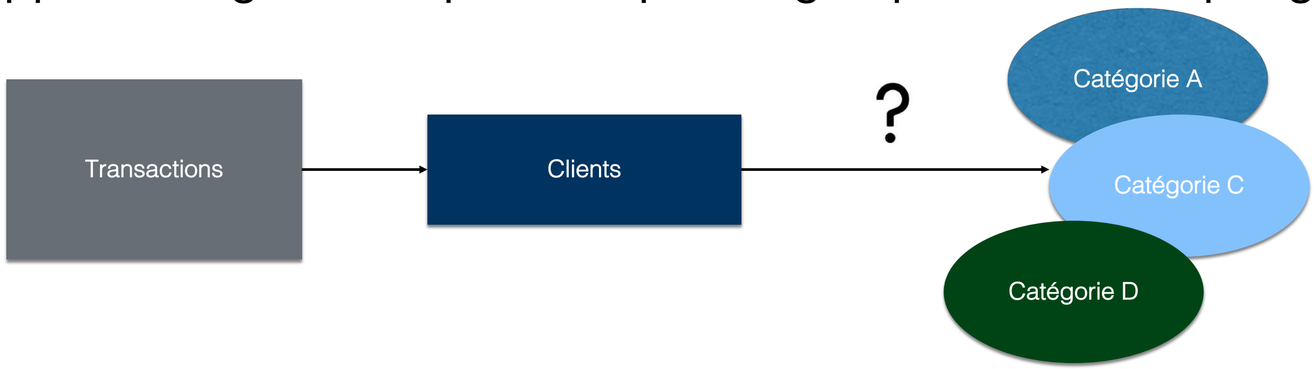
L’objectif

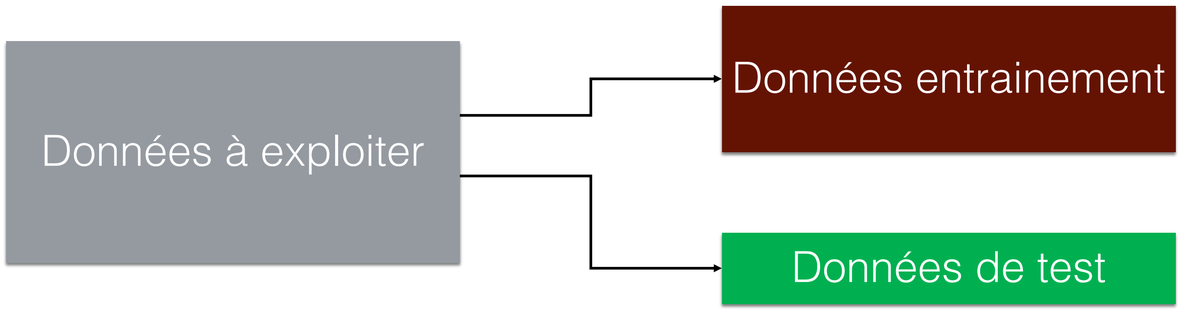
- Identifier des catégories clients à partir des données
- Les catégories doivent regrouper des comportements similaires de clients
- Ces groupes serviront aux marketing pour mieux comprendre les clients et cibler leurs opérations pour augmenter les ventes
- Ils seront utilisés pour l’apprentissage de la classification



Notre démarche

- ❑ Transformation des données en une table par client
- ❑ Recherche des ***features*** qui permettent de détecter des groupes
- ❑ Utiliser une méthode d'apprentissage non supervisée pour regrouper les clients par groupes similaires (clusters).

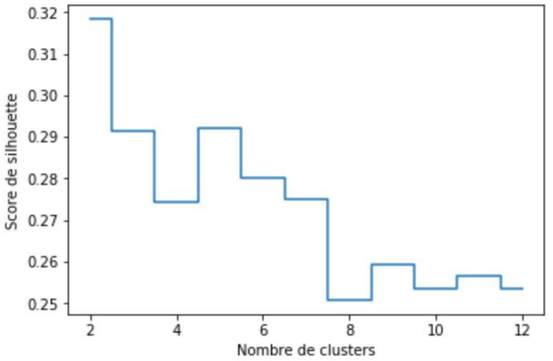




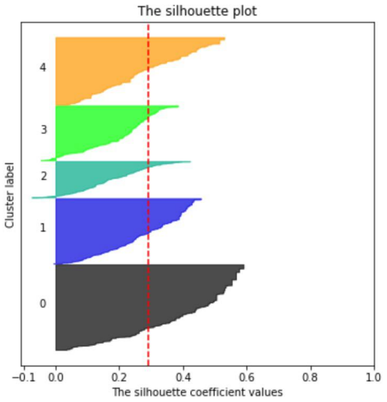
- ❑ Découpage des données en jeu d'entrainement et de test
- ❑ Le clustering est effectué sur données d'entrainement pour éviter le data leakage au niveau des algorithmes de classification.

La recherche de clusters

	CustomerID	NbOrders	TotalSpent	AverageSpent	MinSpent	MaxSpent	TotalQuantity	NbCanceled	NbDiscount	NbPromo	isUK	LastPurchase
0	12346.0	2	0.00	0.000	77183.60	77183.60	0	1	0	0	1	225
1	12347.0	5	2790.86	558.172	382.52	711.79	1590	0	0	0	0	29
2	12348.0	3	1167.24	389.080	187.44	652.80	2116	0	0	0	0	148
3	12350.0	1	294.40	294.400	294.40	294.40	196	0	0	0	0	210
4	12352.0	5	521.18	104.236	104.35	296.50	186	7	0	0	0	162



- Création de nouvelles features pour catégoriser les clients
- Nombreux tests de combinaison de features pour trouver des clusters homogènes et en nombre raisonnable (découpage par mois, score RFM, …)
- Utilisation de l'algorithme K-Means pour trouver les clusters
- Se fait en 3 étapes :
  - Initialisation des k centroïde aléatoirement
  - k clusters sont créés en associant chaque observation au plus proche centroïde
  - On calcule de nouveau le centroïde de chaque cluster et on recommence l'opération
- Utilisation du coefficient de silhouette pour déterminer la meilleure valeur de k

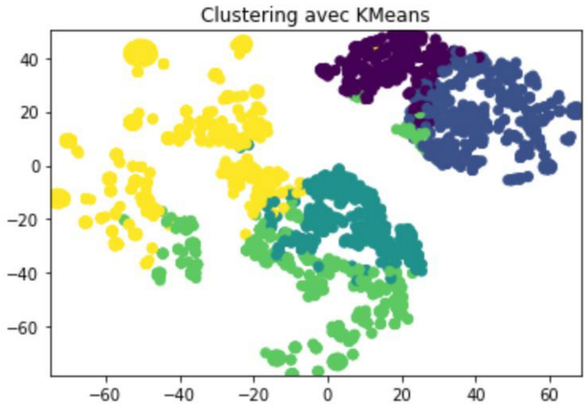


Les scénarios de tests

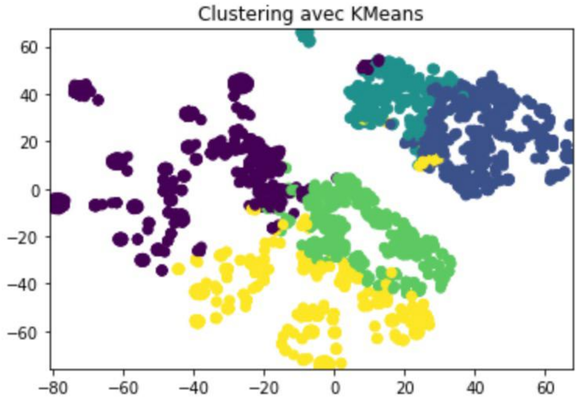
- Découpage de la base clients de manière aléatoire (70% pour clustering et pour l'apprentissage de la classification). Le reste pour vérifier la stabilité du modèle de manière réelle pour de nouveaux clients.
- Découpage temporel des commandes (9 mois de commandes jusqu'à Septembre 2011) et la totalité des commandes afin de mesurer la stabilité temporelle de la classification .

Les résultats

- Pour que le découpage soit exploitable par le Marketing nous avons ciblé un nombre de clusters supérieur à 3 et plus petit que 7.
- Nous avons choisi k=5 qui donne un bon score de coefficient de silhouette
- Les groupes sont homogènes et équilibrés
- Nous obtenons des résultats similaires avec les 2 scénarios (t-SNE pour réduire les dimensions pour affichage)



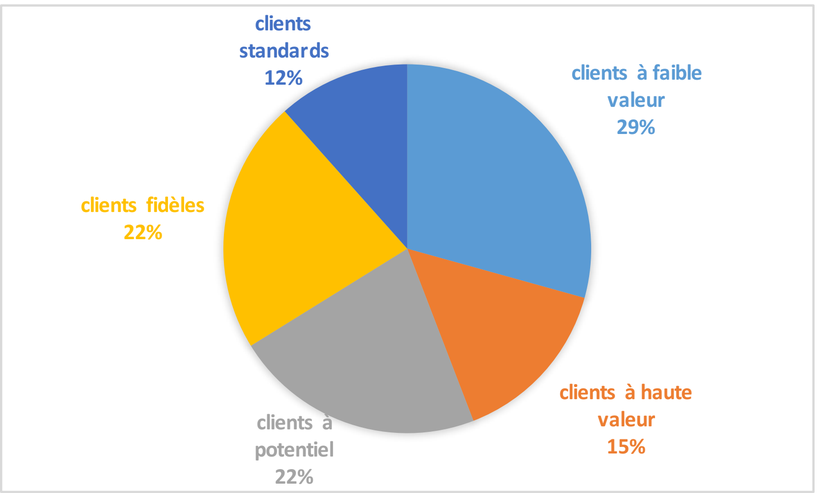
Découpage aléatoire



Découpage temporel

Interprétation des clusters

Cluster	NbOrders	TotalSpent	AverageSpent	MinSpent	MaxSpent	TotalQuantity	NbCanceled	NbDiscount	NbPromo	isUK	LastPurchase	size
0.0	3.660057	446.063711	114.443959	437.337628	1058.760180	282.220963	3.342776	0.005666	0.008499	0.903683	118.898017	353
2.0	5.426667	2512.452289	391.203969	177.592022	710.726978	1425.148889	0.000000	0.000000	0.002222	0.902222	34.053333	450
3.0	1.568862	862.307545	556.375666	519.807186	669.996243	504.220060	0.103293	0.000000	0.001497	0.872754	116.976048	668
1.0	13.096582	5434.607400	362.581830	192.127935	1065.432199	3357.264487	7.286776	0.084695	0.029718	0.895988	27.369985	673
4.0	1.591216	236.303457	155.755487	137.643311	175.729200	155.766892	0.004505	0.000000	0.000000	0.942568	135.308559	888

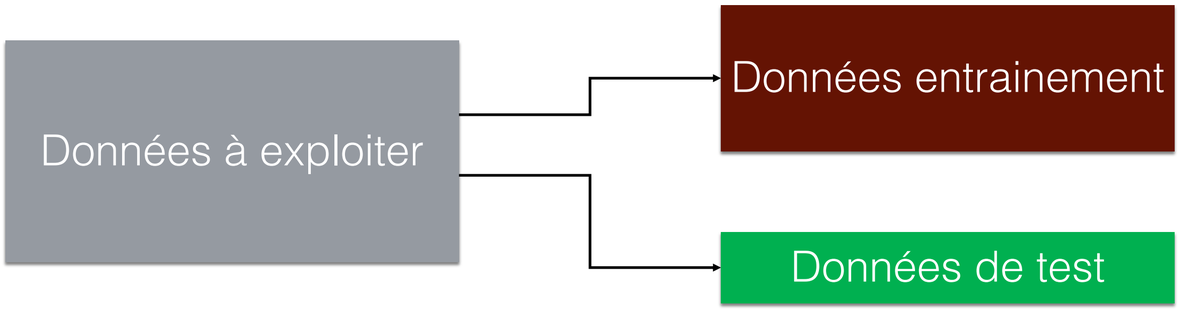


- 1 : **clients fidèles** : ont acheté régulièrement et plus que la moyenne.
- 2 : **clients à haute val**eur : ont un panier moyen élevé et ils reviennent faire des achats.
- 3 : **clients à potentiel** : clients rares mais qui ont un panier moyen très élevé.
- 0 : **clients standards** : clients qui ont acheté plusieurs fois mais pas de commandes très élevées.
- 4 : **clients à faible valeur** : ont de petits paniers d'achats et ne sont pas revenus pour la plupart

Apprentissage de la classification

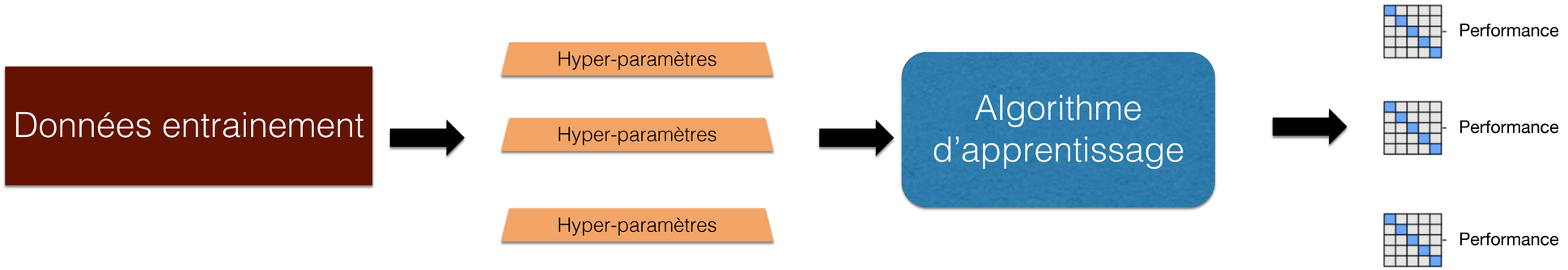
L’objectif

- Trouver un modèle capable de prédire la catégorie d’un client
- Le clustering défini précédemment servira comme variable cible pour l’apprentissage
- Nous allons utiliser les jeux d’entrainement des 2 scénarios pour « tuner » nos algorithmes de classification
- Les jeux de tests (non utilisés pour le clustering) seront utilisés pour comparer les modèles et vérifier la qualité des prédictions.

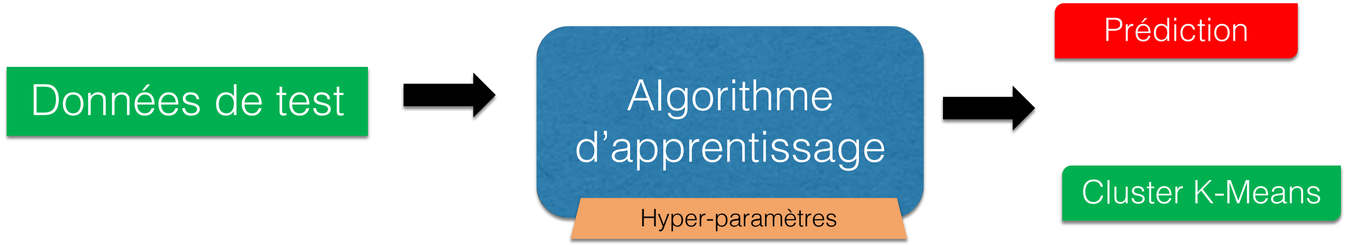


1. Séparation en jeu d’entrainement et jeu de test

Notre démarche d'évaluation de modèle



2. Evaluation de différentes valeurs d'hyper-paramètres par une rechercher sur grille et une validation croisée pour trouver la meilleure performance



3. On évalue notre algorithme sur les meilleures valeurs des hyper-paramètres avec les données de test.

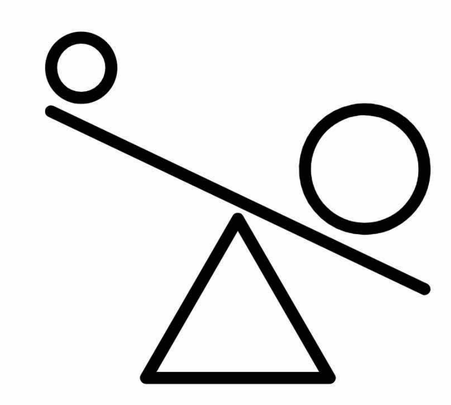
Les algorithmes testés – hyper-paramètres

SVM	• Compromis entre largeurs des marges et erreurs (C)
Régression Logistique	• Le paramètre de régularisation (alpha) • L'inverse de la force de régularisation
Arbre de décision	• Fonctions de mesure de la qualité du découpage • Nombre maximal de features à considérer pour le best split (max_features) • Profondeur
Plus proches voisins (K-NN)	• Nombre de voisins • Algorithme pour calcul des voisins
Forêt Aléatoire	• Nombre d'estimateurs • Profondeur
Gradient Boosting	• Nombre estimateurs

Evaluation des algorithmes

- Validation croisée et recherche sur grille :
  - Donne les meilleurs hyper-paramètres

- Comparaison entre les algorithmes :
  - Accuracy Score
  - Matrice de confusion



Résultats et implémentation

Résultats – Accuracy Score

	Découpage aléatoire	Découpage temporel
SVM	97,25%	97,11%
Logistic Regression	98,13%	97,41%
K-NN	97,80%	97,61%
Decision Tree	87,58%	87,86%
Random Forest	98,46%	98,61%
Gradient Boosting	99,12	99,00%
XGBoost	95,38%	94,83%

Choix du modèle final

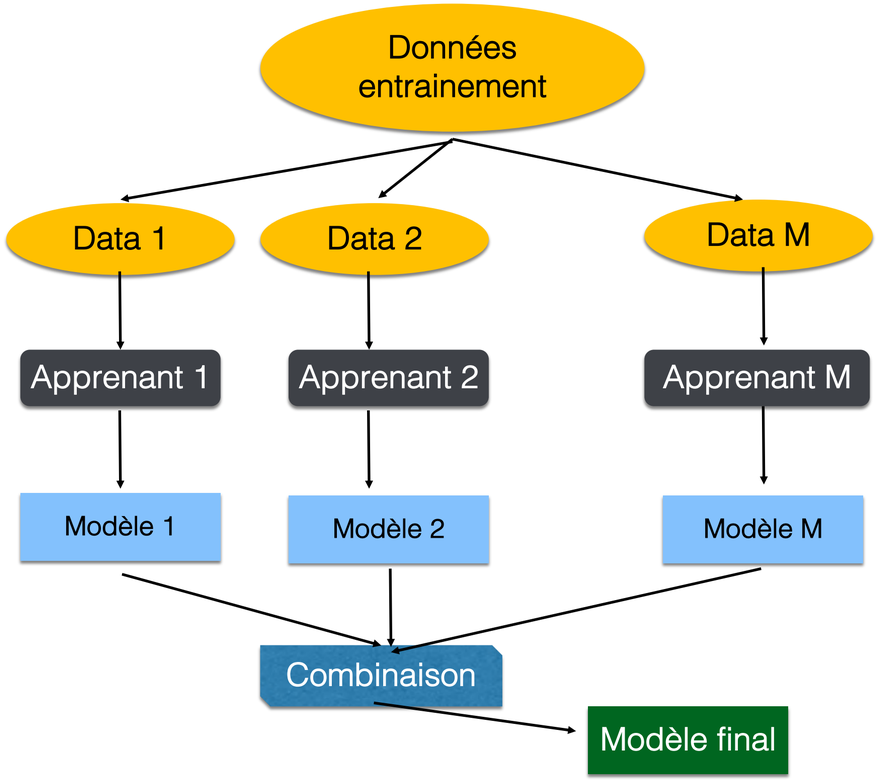
- Validation des résultats par le test avec la base clients non utilisée pour le clustering.
- Choix de l'algorithme qui donne le meilleur score : Gradient Boosting
- Implémentation au niveau d'un module python qui prend en entrée un fichier Excel de transactions clients et génère en sortie un fichier Excel qui prédit la catégorie des clients.

Les sources sont disponibles ici : https://github.com/makboulhoussen/projet5

Présentation de l’algorithme Gradient Boosting

Les méthodes ensemblistes

- Combinaison des résultats de plusieurs modèles pour faire la prédiction finale
- Concept : meilleure prédiction à partir d'une combinaison intelligente des résultats de plusieurs modèles plutôt que d'un seul
- **Bagging (parallèle) :**
  - Sous échantillonnage des données et on fait générer à l'algorithme un modèle pour chaque sous échantillon
  - On fait une ensuite une moyenne ou un vote des différentes prédictions
- **Boosting (séquentiel) :**
  - On va donner plus d'importance aux valeurs difficiles à prédire correctement
  - Les modèles suivants vont apprendre des erreurs des modèles précédents.



Gradient Boosting = Boosting + Descente de gradient

- Algorithme de type ensembliste, majoritairement employé avec les arbres de décision
- **Boosting** : méthodes fonctionnant sur ce principe d'assemblage en série d'apprenant faibles.
- **Descente de gradient** : algorithme qui à chaque itération va chercher les valeurs des coefficient qui vont minimiser la fonction de perte.
- Le « **Gradient Boosting** » est un type spécial de boosting qui va chercher à réduire l'erreur de manière séquentielle.
- A chaque itération, le modèle essaye de corriger les erreurs du modèle précédent.

Gradient Boosting : algorithme

1. Application d'un modèle simple sur les données
2. Calcul de la perte résiduelle.
3. Modélisation du résidu avec un nouveau modèle M2
4. Répéter les étapes 2 à 4 jusqu'à ce que le nombre d'itération atteint.
5. Le modèle finale sera l'association de tous ces modèles qui seront pondérés.

- **Résidu** = gradient négatif. La modélisation des résidus correspond donc à minimiser la fonction de coût global.

- **Pour classification :**
  - le principe est globalement le même mais avec une fonction de coût adaptée au classement.
  - A chaque itération, un poids (inversement proportionnel au taux d'erreur) est affecté aux observations

Gradient Boosting

Quelques paramètres pour la classification :

- Nombre d'arbres : nombre d'itérations
- La profondeur des arbres
- Learning_rate : le coefficient de rétrécissement.
- La fonctions de perte à utiliser

- **Avantage** :
  - Généralement robustes par rapport aux outliers
  - Peuvent apprendre de modèles non linéaires
  - Performance

- **Inconvénients** :
  - Sur-apprentissage
  - Paramètres nombreux
  - Occupation mémoire de tous les arbres

Conclusion

Conclusion

- L’analyse exploratoire a permis une meilleure compréhension de nos données et de les préparer pour la segmentation.
- La recherche des clusters a demandé de nombreux tests et combinaisons de features pour trouver des catégories clients ayant un sens
- Nous avons amélioré les performances de notre modèle de classification par des entrainements et la recherche des paramètres optimums. Nous avons séparé les données avant le clustering pour éviter le dataleakage.
- Le modèle qui a donné le meilleur résultat : Gradient Boosting a été implémenté au niveau du module final.
- Axes d’amélioration :
  - Travailler avec l’équipe marketing pour la segmentation
  - Segmentation sans utiliser le clustering (score RFM)
  - Avoir des informations sur catégories de produits pour mieux segmenter les préférences des clients

Merci à mon mentor Amine Abdaoui pour sa disponibilité, ses explications et ses précieux conseils