Soutenance projet N◦ 5

Segmenter les comportements de clients

Gregory Sallen le 01/05/2018

Mentor : Mohammed Sedki

Problématique

datazon

- Analyser les commandes de clients sur une années d'une entreprise de vente en ligne britannique
  - Analyse exploratoire et mise en forme des données
- Segmenter les clients en fonction de leurs comportements dans la durée
- Utiliser les données complètes ou réduites pour proposer un modèle de classification afin de prédire le comportement d'un client dès son premier achat.

Sommaire

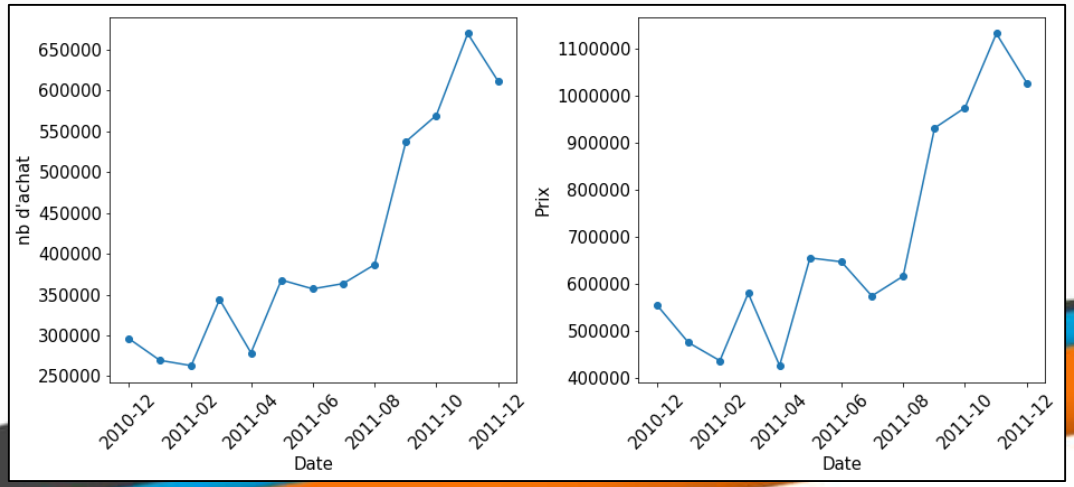
- Analyse exploratoire
- Feature engineering
- Clustering
- Tests de différents modèles de classification
- Sélection, test de performance et optimisation du modèle sélectionné
- Conclusion

Présentation des données

- Données fournies : sous forme d'horizon temporel.
  - Représente un achat
  - 8 variables :
    - InvoiceNo : Numéro de commande
    - StockCode : Code du produit
    - Description : Description du produit
    - Quantity : Nombre de produits achetés (peut être négatif)
    - InvoceDate : Date de la commande
    - UnitPrice : Prix du produit à l'unité
    - CustomerID : Numéro du client
    - Country : Pays du client (90 % Royaume-Uni)

Exploration et nettoyage

- Variable principale : CustomerID
  -> Suppression des valeurs manquantes
- Suppression des quelques valeurs aberrantes
  -> Prix
- Traitement des annulations (Quantité négative)
  -> Certaines annulations n'ont pas la commande correspondante
- Date : Prise en main du format DateTime.
  -> Les données contiennent des commandes entre le 1er décembre 2010 et le 9 décembre 2011



Feature engineering
Construction des données sous forme exploitable

- Chaque ligne de données résume toutes les informations d'un client. (4339 clients)
- Création de 17 variables :
  - **local** : différencie si le client est dans le Royaume-Uni ou pas (binaire)
  - **dep_tot** : dépense totale du client
  - **dep_pos** : dépense positive
  - **dep_neg** : dépense négative (annulation)
  - **dep_com** : dépense moyenne par commande
  - **dep_prod** : dépense moyenne par produit
  - **nb_com** : nombre total de commandes
  - **nb_an** : nombre d'annulations
  - **dep_max** : dépense max
  - **dep_min** : dépense min
  - **nbpr_tot** : nombre total de produits
  - **nbpr_ann** : nombre de produits annulés
  - **nbpr_diff** : nombre de produits différents
  - **nbp_com** : nombre moyen de produits par commande
  - **last_com** : dernière commande (unité = jour)
  - **freq_com** : fréquence des commandes (unité = jour^(-1))
  - **peri_com** : période entre le première et la dernière commande

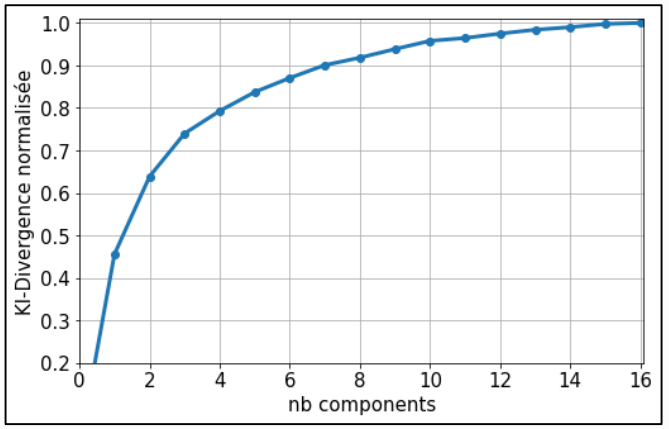
Clustering
segmentation du comportement des clients

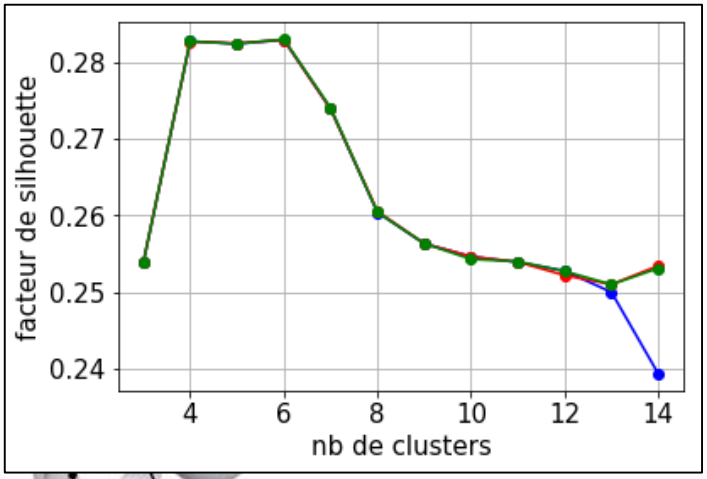
- Test sur différents algorithmes de clustering
- Critère d'évaluation : coefficient de silhouette C_f
- C_f =Cst entre 4 et 8 cluster

méthode	C_f		-1	0	1	2	3	4
K-means	0,28		x	2880	1363	2	11	1
Hiérarchique	0,37		x	3773	9	2	472	1
DBScan	0,35		125	3745	374	6	7	x

Clustering (suite)

- Réduction de dimension : t-SNE
  - Visualisation des clusters : 2 dimensions
  - Influence sur la segmentation
- Limitation à 8 variables

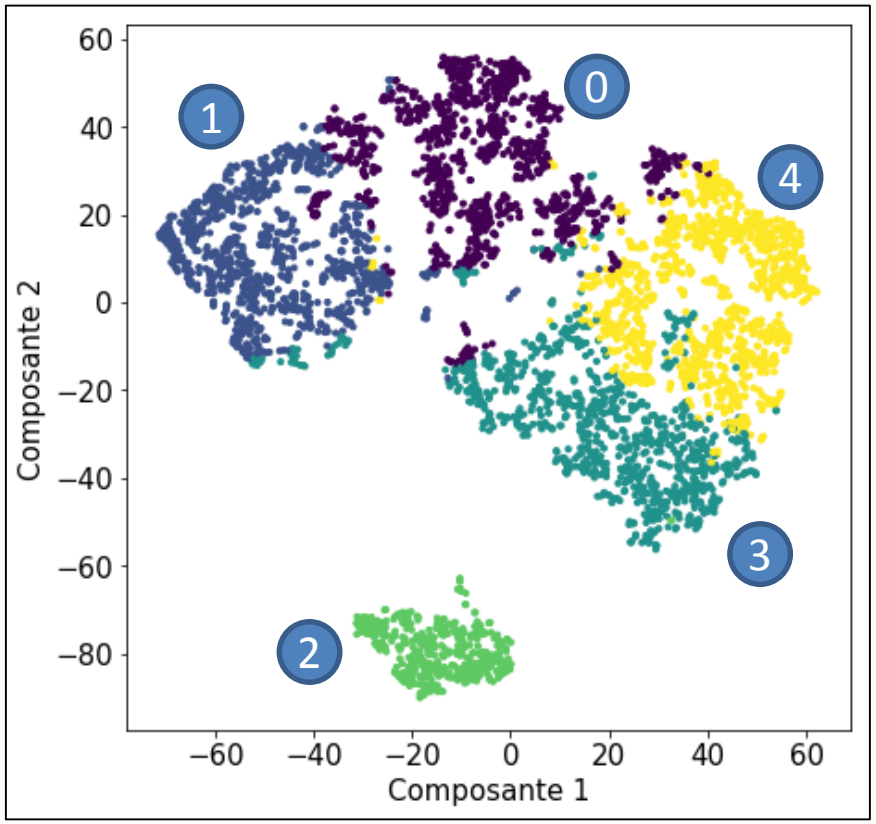




- K-means, 5 clusters

Label	0	1	2	3	4
Pop	1024	933	1013	888	399

Clustering (suite)



- Analyse des clusters :

  Représentation de la valeur moyenne et médiane de chaque variable pour tous les clusters

0) Client moyen/petit mais régulier. Dernière commande récente.

1) Petit client, seulement quelques commandes, dernière commande ancienne.

2) Client majoritairement hors Royaume-Uni, client moyen, leur particularité est d'avoir un nombre de commande faible et donc une dépense par commande importante.

3) Petit client, seulement quelques commandes, dernière commande récente.

4) Gros client, dépense total et dépense max importante, achète sur une grosse gamme de produit et régulièrement.

Modélisation

- Méthode naïve
- Régression logistique
- SVM : support vector machine
- Knn Classifier
- Random Forest Classifier
- Gradient Boosting Classifier

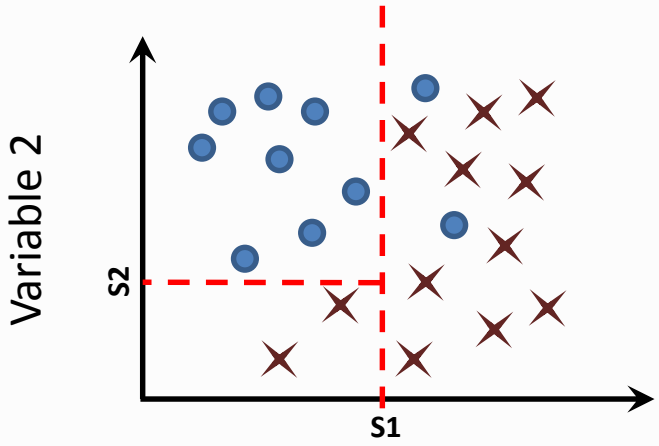
Random Forest Classifier

- Arbre de décision pour la classification
  - Fonctionne également pour la régression

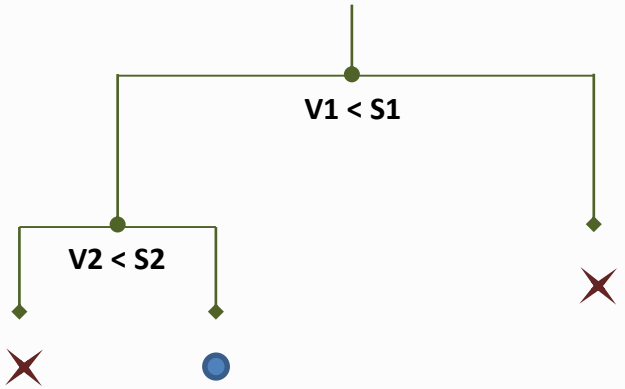
- Bagging : Bootstrap aggregating
  - Méthode de réduction de la variance en combinant plusieurs apprenants

Arbre de décision

- Principe est de segmenter l'espace des prédicteurs en régions simples



Représentation sous forme d'arbre



- La prédiction sera la classe la plus présente dans chaque segment

Détermination des segments

- Mesure des erreurs de classification : $1 - \hat{p}_{mk}$

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} 1\{y_i = k\}$$

- Où la feuille *m* représente la région R_m avec N_m observations

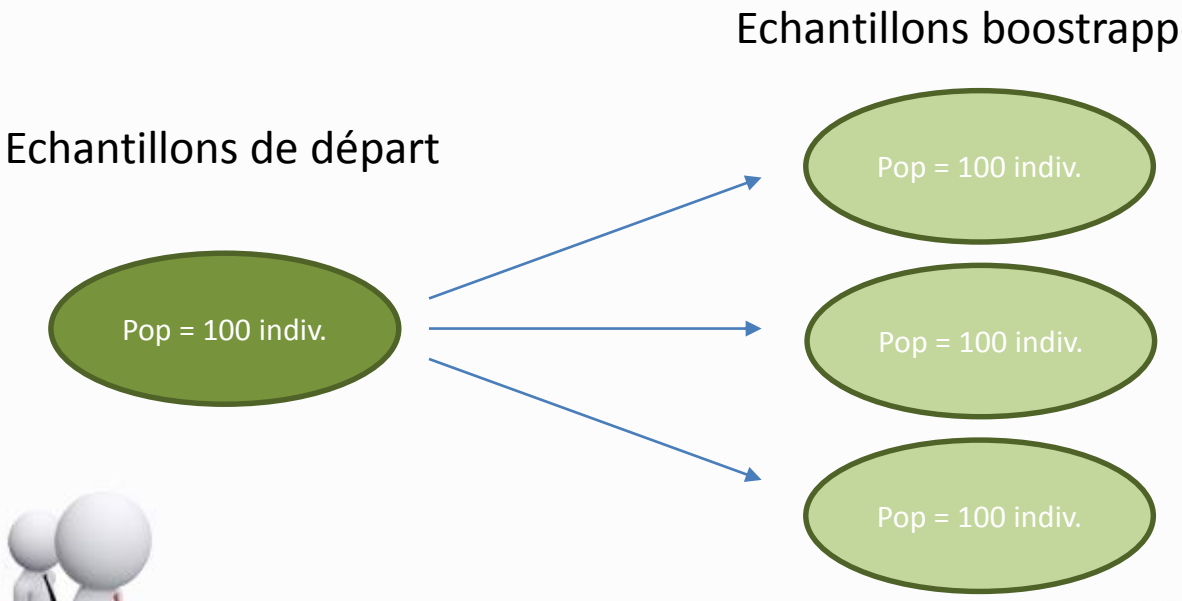
Et *k* la classe.

- Ainsi la valeur du segment sera celle de la classe dominante $C_m = argmax_k(p_{mk})$

- On détermine la position *s* du séparateur en minimisant la somme des erreurs de classification des deux segments.

Le Bagging

- Utiliser plusieurs échantillons pour construire plusieurs arbres de décision ou la prédiction sera un vote à la majorité de chaque arbre.
- Bootstrap : Un échantillon bootstrapper est un échantillon construit à partir des données initiales par tirage avec remise



Random Forest Classifier

- Construction de plusieurs arbres de décision sur des échantillons Bootstrap.
- Prédiction sera un vote à la majorité de chaque arbre.
- Amélioration : Réduction de la variance en décorrélant les différents arbres, utilisation d'un algorithme randomisé.
  - cad pour chaque nœud de chaque arbre l'algorithme n'utilise qu'une partie des prédicteurs tirés aléatoirement. Une nouvelle sélection des prédicteurs est faite à chaque division de nœud.
- Le bagging permet d'éviter les erreurs de sur-apprentissage.

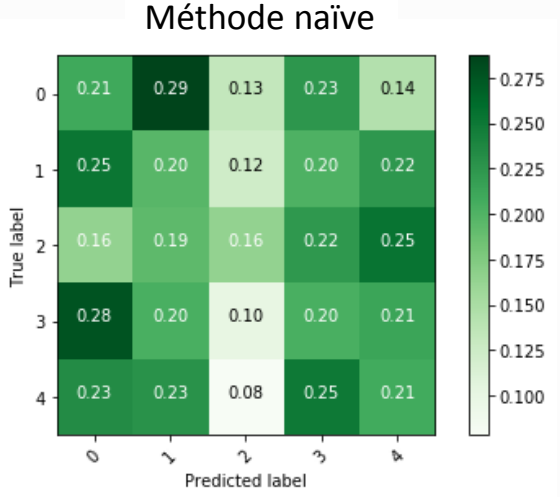
Résultat des classifieurs

- Evaluation des différents classifieurs :
  - Accuracy score :

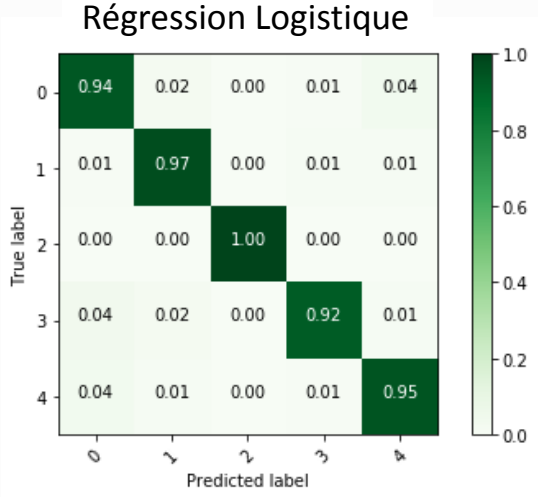
$$\mathrm{acc}(y, \hat{y}) = \frac{1}{N_{samp}} \sum_{i=0}^{N-1} 1\{\hat{y}_i = y_i\}$$

  - Matrice de confusion :

Méthode naïve



Régression Logistique



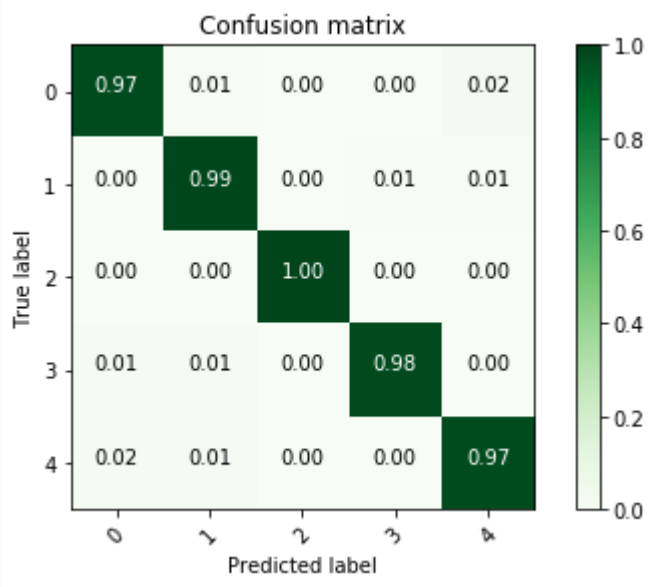
Résultat des classifieurs

Méthode	Algorithme	Accuracy
naïve	DummyClassifier()	0,2
Régression logistique	LogisticRegression()	0,96
support vector machine	LinearSVC()	0,95
Knn	KNeighborsClassifier()	0,96
Random Forest	RandomForestClassifier()	0,97
Gradient Boosting	GradientBoostingClassifier()	0,98
Vote des classifieurs précèdents	VotingClassifier()	0,98

- Pour chaque méthode les hyperparamètres sont optimisés avec la fonction gridsearch en cross validation.

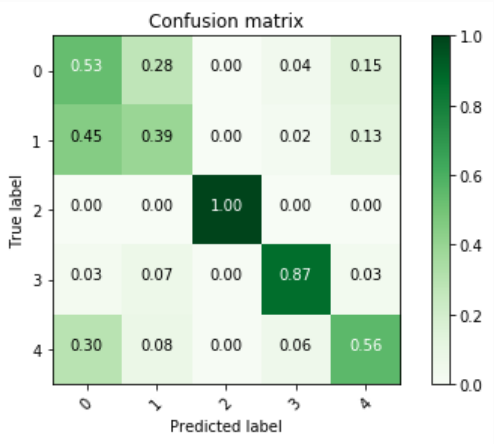
Modèle retenu

- Aucun modèle ne se démarque des autres, l'ensemble des modèles ont de très bons résultats (>95%) sur les données complètes.
- Modèle retenu : voting Classifier

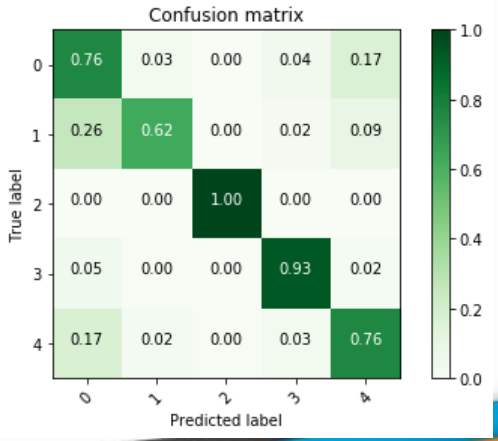


Test de performance horizon temporel réduit

- Sélection du premier achat de chaque client :
  - Accuracy = 0,63
  - Confusion entre les labels 0, 1 et 4
  - Label 2 et 3 marqués par la localité (hors RU) et une faible quantité de commande
  - Label 4, achat important et régulier
  - Label 0, achat moyen et régulier



- Sélection des deux premiers achats de chaque client :
  - Accuracy = 0,80



Conclusion

- Analyse exploratoire et mise en forme des données.
- Segmentation des clients sur les données complètes. Analyse des 5 segments.
- Evaluation et sélection de différents modèles de classification.
- Entrainement ciblé pour classé un client le plus rapidement possible.

Rappel

- Coefficient de silhouette : $\frac{b - a}{\max(a, b)}$

  - a : distance moyenne intra-cluster
  - b : distance moyenne des clusters voisins