Segmentation de clients

Thomas Weber

Introduction

- Datazon cherche à comprendre ses différents types de clients afin de pouvoir réaliser des operations marketing plus ciblées et efficaces

- Source des données: 1 an de transactions (déc. 2010 – déc. 2011) : environ 500 000 transactions

- Objectif:
  - Trouver des variables qui permettent de catégoriser les différents types de clients
  - Effectuer un clustering représentatif
  - Entraîner ensuite un modèle de classification qui soit capable de catégoriser de nouveaux clients

Nettoyage

- CustomerID: suppression des valeurs manquantes
- Pays: limitation au Royaume-Uni (+90% des transactions)
- Suppression des doublons (env. 5 000 lignes)
- StockCode: suppression des valeurs non pertinentes
- Quantity: suppression des outliers (> 50000)

```
In [61]: df.nunique()

Out[61]: InvoiceNo      19641
         StockCode       3654
         Description     3853
         Quantity         418
         InvoiceDate    18273
         UnitPrice        435
         CustomerID      3942
         Price           3642
         Cancelled          2
         dtype: int64
```

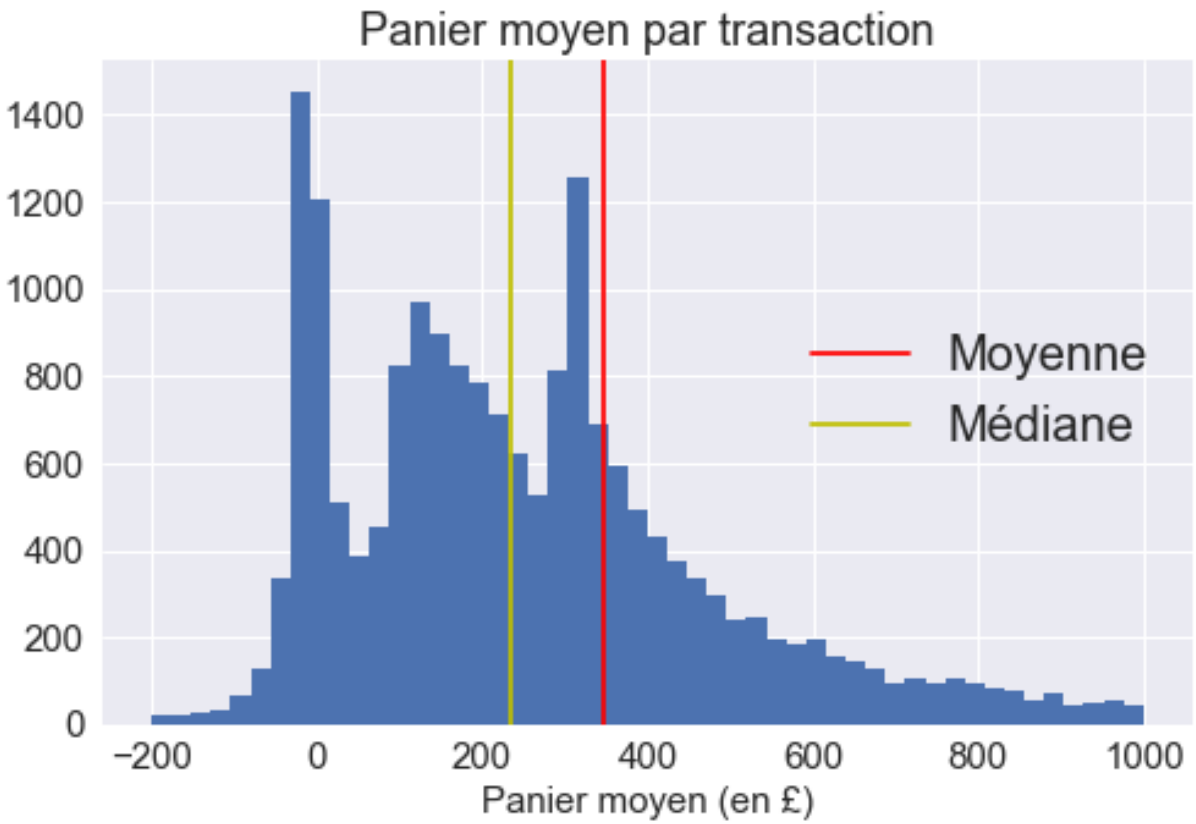
Feature engineering

- Création de 2 variables:
  - Prix total payé par article:

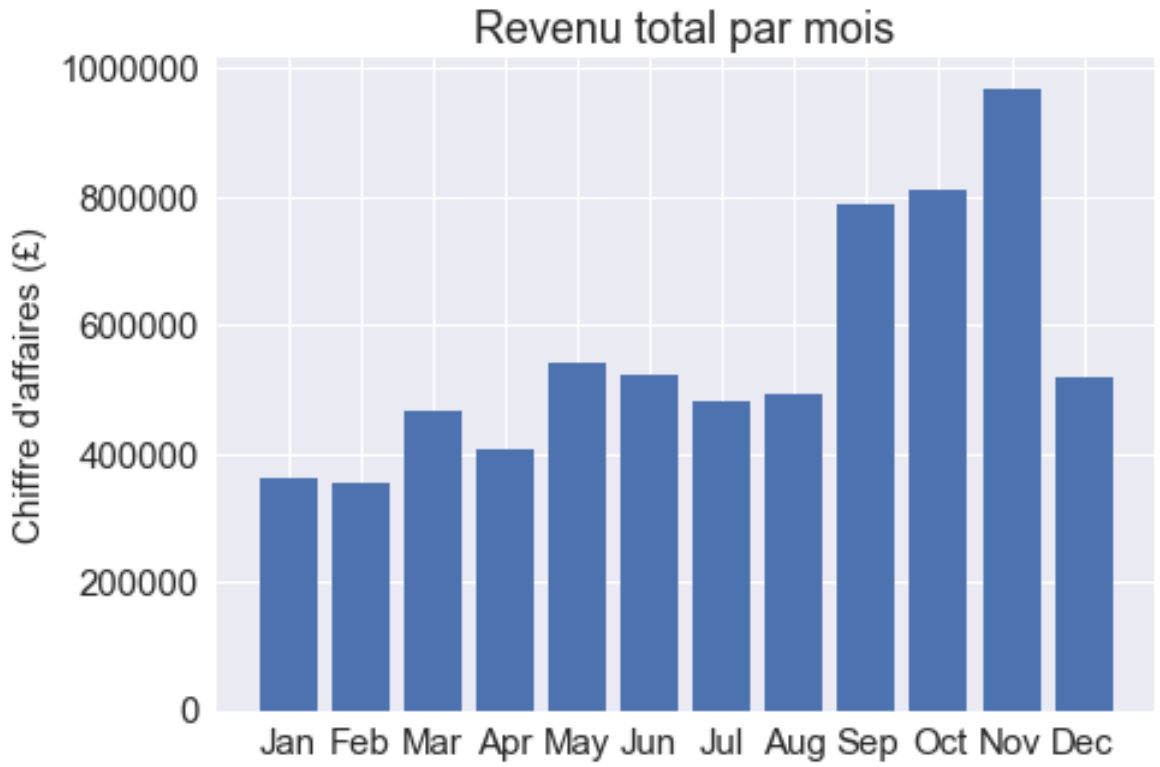
    Price = Quantity * UnitPrice
  - Si la commande est une annulation (valeur booléenne):

    Cancelled = True if InvoiceNo starts with ‘C’

Exploration – Panier moyen



Exploration - Saisonnalité



Sélection des features – Score RFM

- Métrique de base dans la segmentation client:
  - R pour Recency: nombre de jours depuis la dernière transaction
  - F pour Frequency: nombre de transactions
  - M pour Monetary value: montant total des commandes

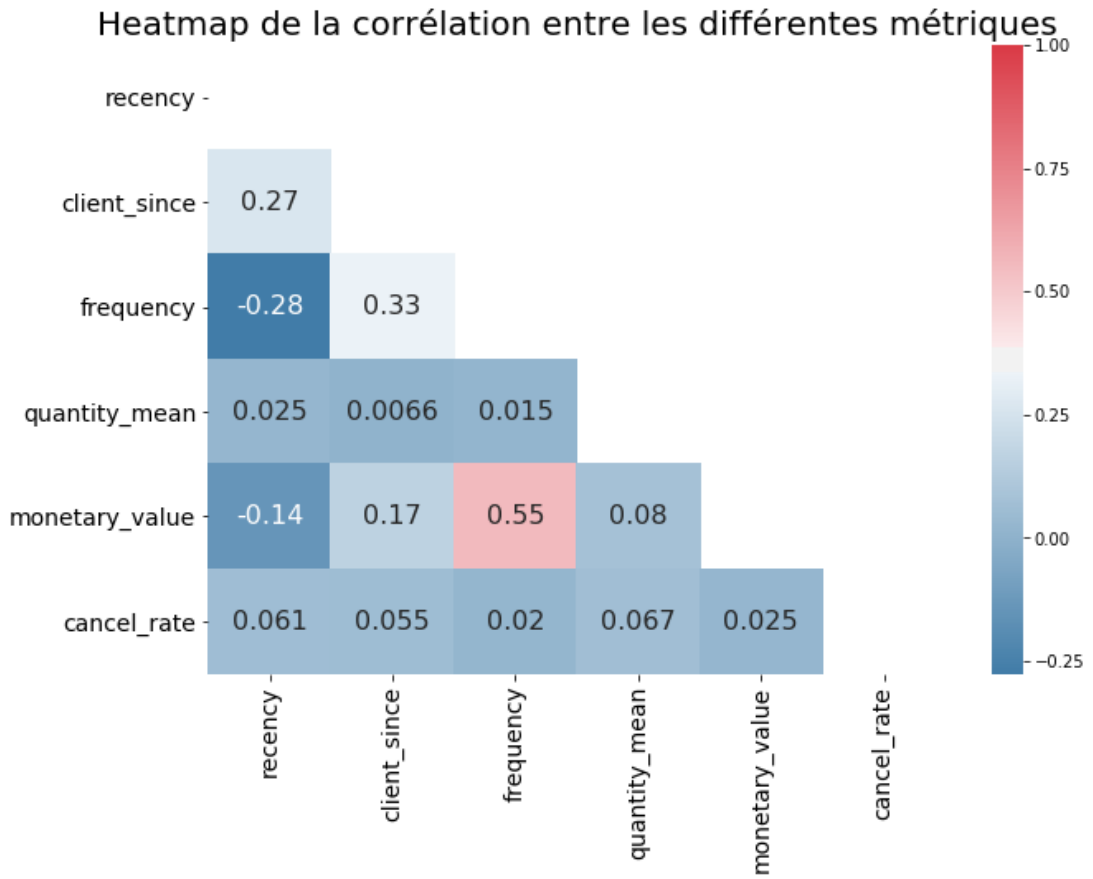
Customer	Recency	Frequency	Monetary	R	F	M
A	53 days	3 tran.	$230	2	2	2
B	120 days	10 tran.	$900	3	3	2
C	10 days	1 tran.	$20	1	1	1

Sélection des features

- 3 features supplémentaires:
  - Client_since: nombre de jours depuis la 1^{e} transaction
  - Quantity_mean: quantité moyenne commandée par article
  - Cancel_rate: taux d'annulation des commandes

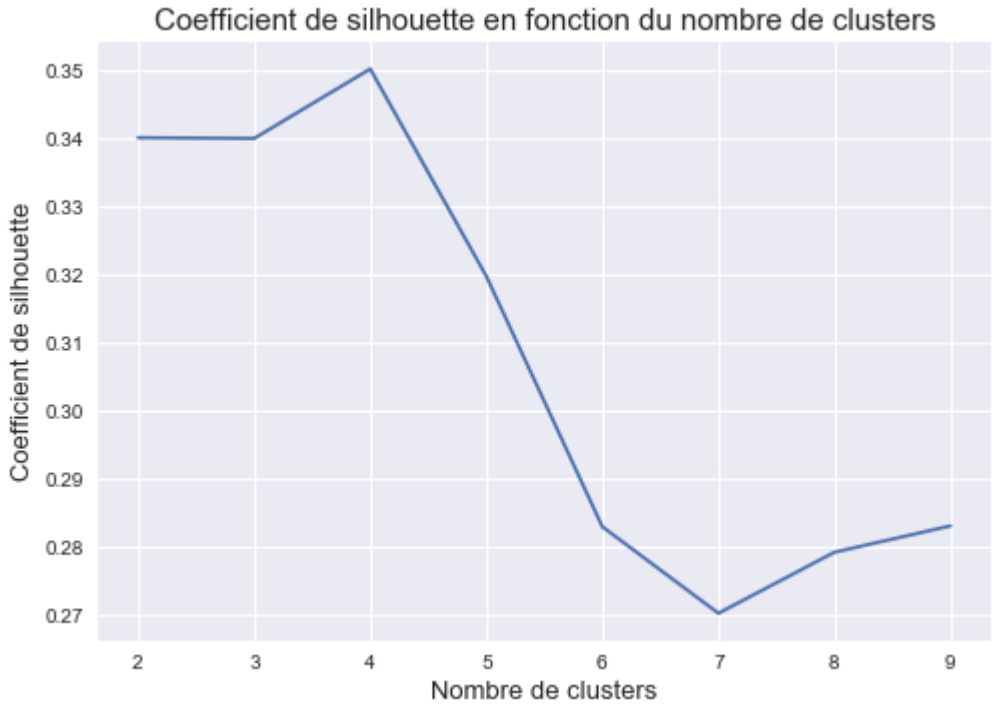
Clustering - Préparation

- Passage d'une liste de transactions à une table clients
- Vérification des corrélations
- Transformation des variables:
  - Passage au log
  - Standardisation
  - ACP (3 composantes)



Clustering - KMeans

- Nombre de clusters testés: 2 à 9
- Calcul du coefficient de silhouette
- Affichage des valeurs médianes de chaque cluster
- Affichage des clusters selon les axes RFM
- Nombre de clusters retenu: 5



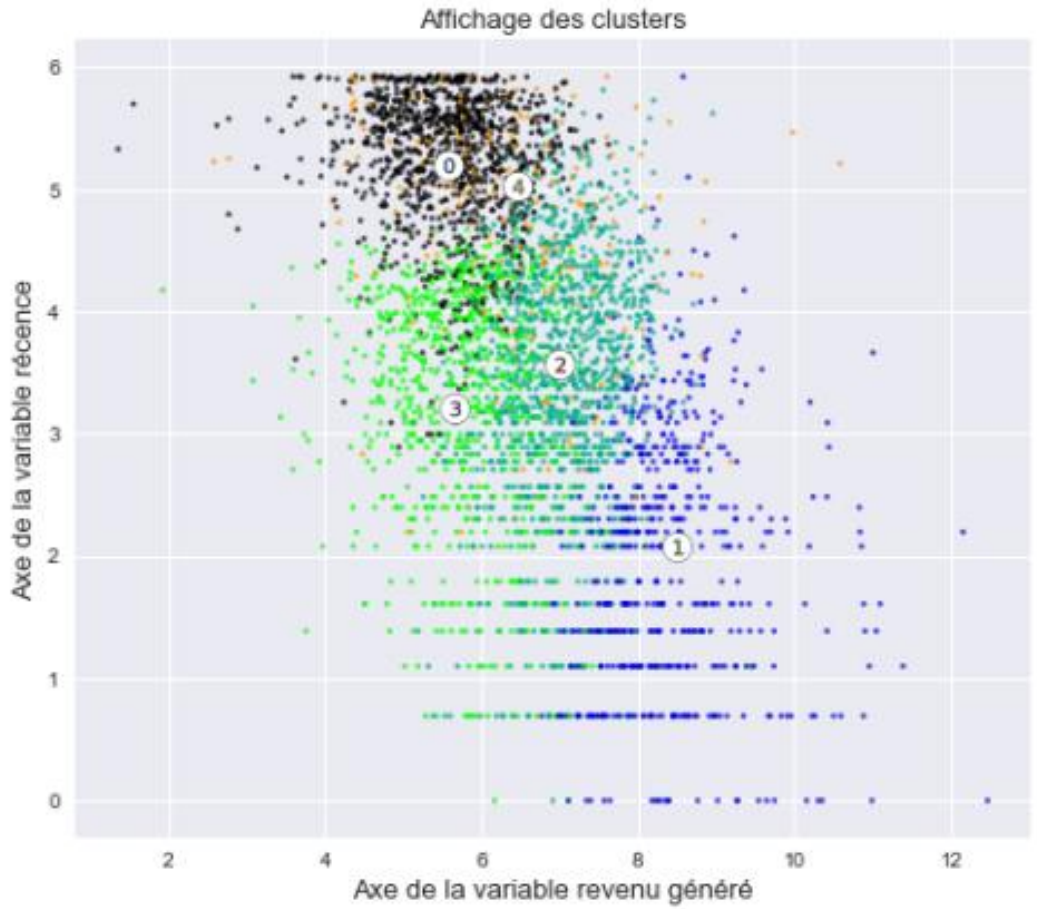
Clustering – Choix entre 4 et 5 clusters

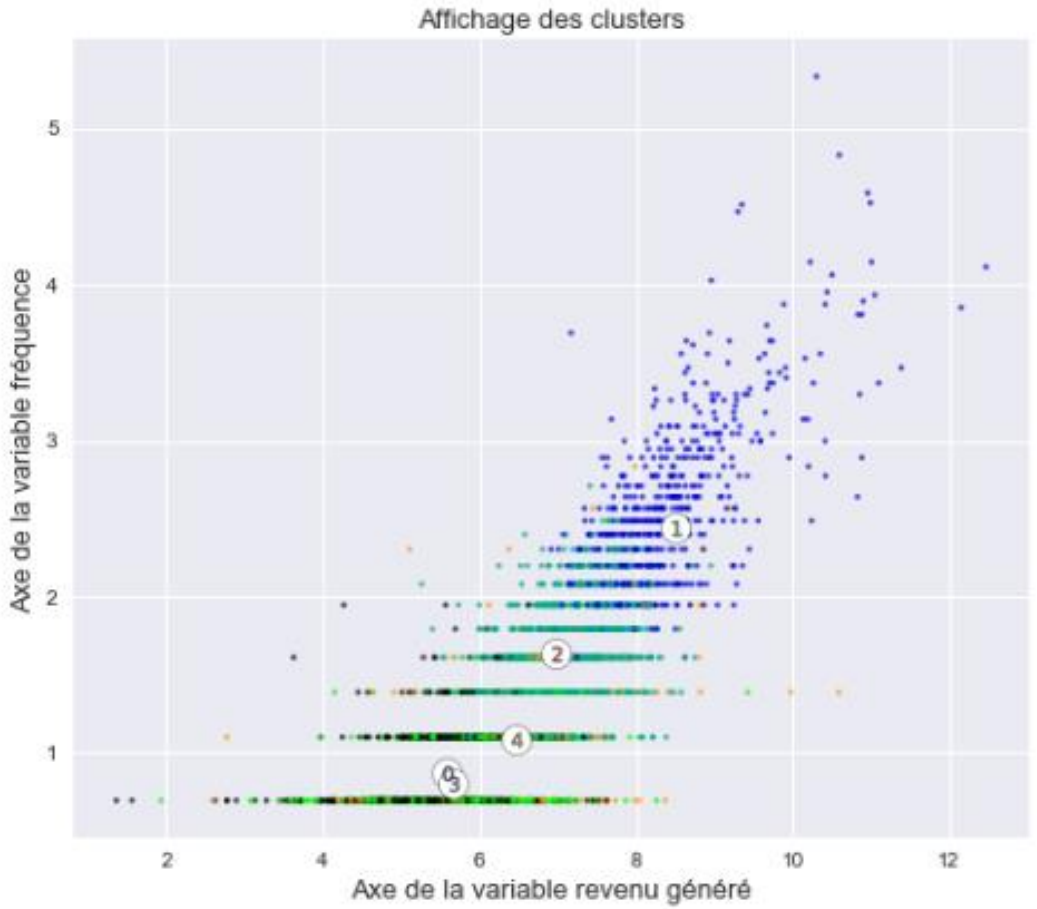
- 5 groupes identifiés:
  - « Meilleurs clients »: ceux qui achètent le plus, le plus souvent (top 15%)
  - « Client réguliers »
  - « Grossistes »: ceux qui achètent de grandes quantités
  - « Clients perdus »: ceux qui ont acheté une fois au moins mais ne sont pas revenus depuis longtemps
  - « Nouveaux clients »: un achat récent

cluster	recency	client_since	frequency	quantity_mean	monetary_value	cancel_rate	cluster_size	size_pct
0	31	50	1	9.666667	335.00	0.000000	854	21.869398
1	16	317	6	10.404580	2128.83	0.009434	1295	33.162612
2	166	263	2	7.928571	365.27	0.000000	1511	38.693982
3	127	260	2	41.545455	601.56	0.100000	245	6.274008

cluster	recency	client_since	frequency	quantity_mean	monetary_value	cancel_rate	cluster_size	size_pct
0	211.0	259.0	1.0	7.592593	286.79	0.000000	1093	27.989757
1	7.0	358.0	11.0	11.563245	3878.31	0.013333	558	14.289373
2	33.0	281.0	4.0	9.333333	1125.07	0.000000	1209	30.960307
3	32.0	47.0	1.0	9.537313	329.34	0.000000	817	20.921895
4	130.5	260.5	2.0	43.927083	601.04	0.109127	228	5.838668

Clustering - Visualisation





Stabilité du clustering

Jeu de données entier

cluster	recency	client_since	frequency	quantity_mean	monetary_value	cancel_rate	cluster_size
0	210.5	259.0	1.0	7.596296	286.265	0.000000	1094
1	33.0	281.0	4.0	9.333333	1125.070	0.000000	1209
2	7.0	358.0	11.0	11.563245	3878.310	0.013333	558
3	32.0	47.0	1.0	9.537313	329.340	0.000000	817
4	130.0	261.0	2.0	43.666667	601.560	0.111111	227

Jeu d’entraînement (75%)

cluster	recency	client_since	frequency	quantity_mean	monetary_value	cancel_rate	cluster_size
0	37	280	4	9.579832	1091.855	0.000000	886
1	133	252	2	42.647059	612.430	0.142857	153
2	8	354	10	11.625000	3559.210	0.013245	455
3	217	259	1	7.601724	266.125	0.000000	826
4	30	47	1	9.400000	329.295	0.000000	608

Stabilité temporelle du clustering

Jeu de données entier

cluster	recency	client_since	frequency	quantity_mean	monetary_value	cancel_rate	cluster_size
0	210.5	259.0	1.0	7.596296	286.265	0.000000	1094
1	33.0	281.0	4.0	9.333333	1125.070	0.000000	1209
2	7.0	358.0	11.0	11.563245	3878.310	0.013333	558
3	32.0	47.0	1.0	9.537313	329.340	0.000000	817
4	130.0	261.0	2.0	43.666667	601.560	0.111111	227

Données jusqu'au 31/8 (env. 9 mois)

cluster	recency	client_since	frequency	quantity_mean	monetary_value	cancel_rate	cluster_size
0	29.0	37.0	1.0	10.422222	303.680	0.000000	342
1	125.0	176.0	1.0	15.000000	381.820	0.222222	145
2	13.0	260.5	9.0	11.979651	3242.470	0.012821	402
3	154.0	173.0	1.0	7.312500	251.210	0.000000	1161
4	49.0	211.0	3.0	9.313853	944.055	0.000000	920

Classification

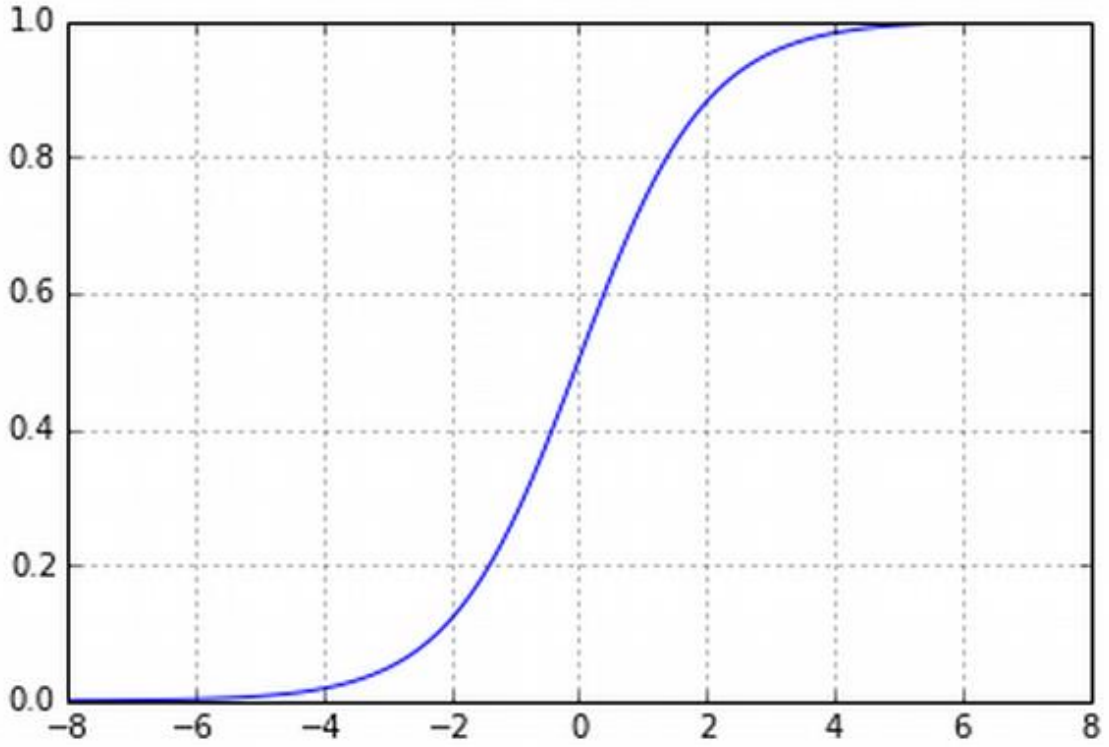
- 4 modèles:
  - Régression logistique
  - SVM
  - K-Nearest Neighbors
  - Random Forest

- 3 stratégies multi-classes:
  - OVO: One versus One
  - OVR: One versus Rest
  - CS: Crammer-Singer (uniquement SVM)

- Choix des hyper-paramètres par validation croisée

Régression logistique

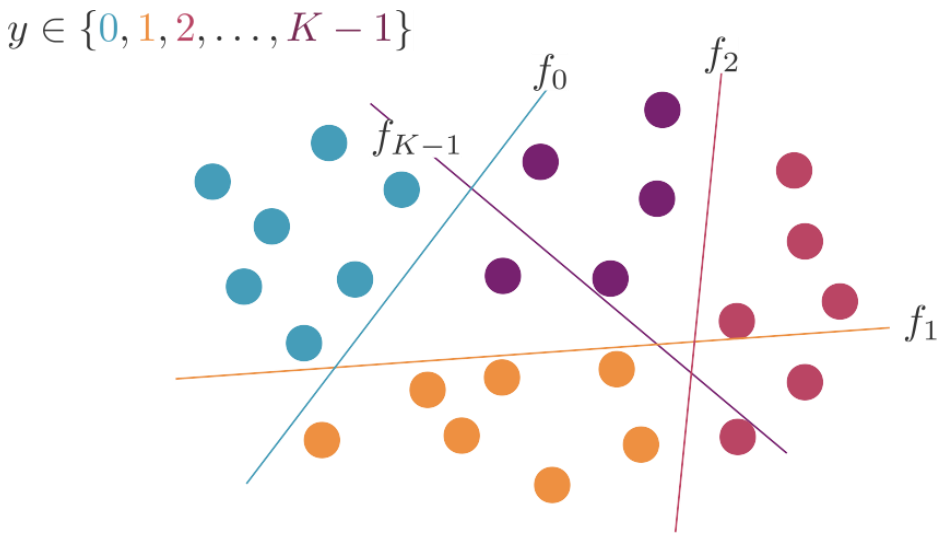
- Modèle de classification binaire
  - $y \in \{0, 1\}$
- Transformation logistique
  - $Logistic(u) = \frac{1}{1+ e^{-u}}$
- Prédiction de $p(Y = 1|x)$
  - $p(Y = 1|x) = \frac{1}{1+e^{-(\beta_0+\sum_{j=1}^{p} \beta_j x_j)}}$
- Calcul des paramètres par la méthode du maximum de vraisemblance



Classification multi-classes

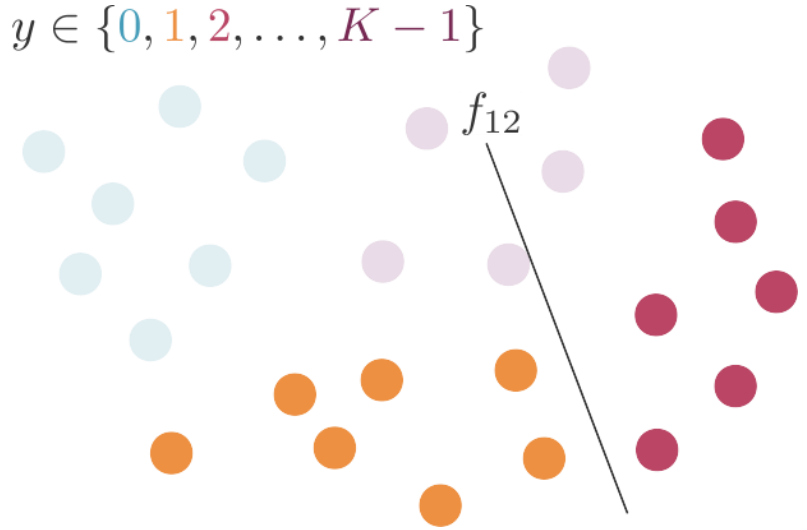
Stratégie One-versus-Rest

- 1 classifieur par classe pour séparer les points de cette classe de tous les autres points
- Pour chaque point on a alors sa probabilité d'appartenir à chacune des k classes

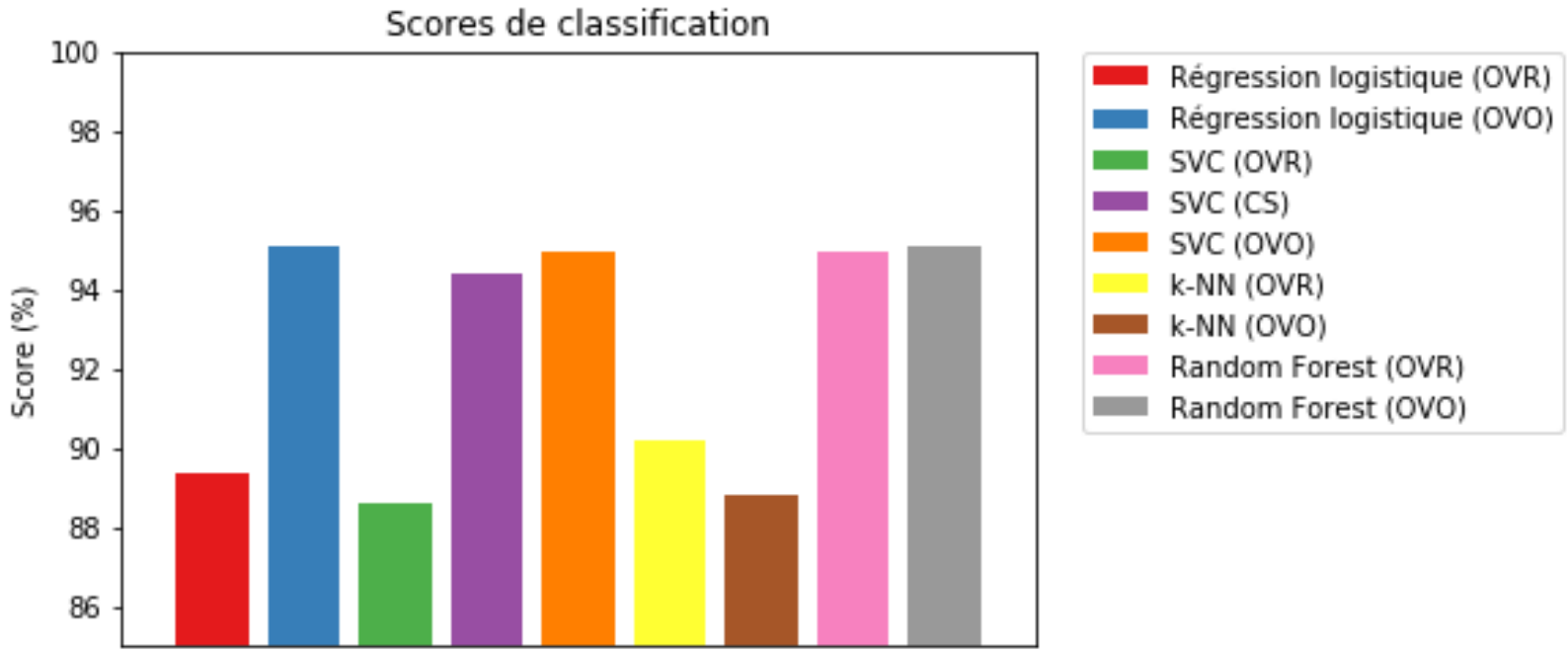


Stratégie One-versus-One

- 1 classifieur par couple de classes (k(k-1) classifieurs en tout)
- Pour chaque point, la classe prédite en majorité est retenue

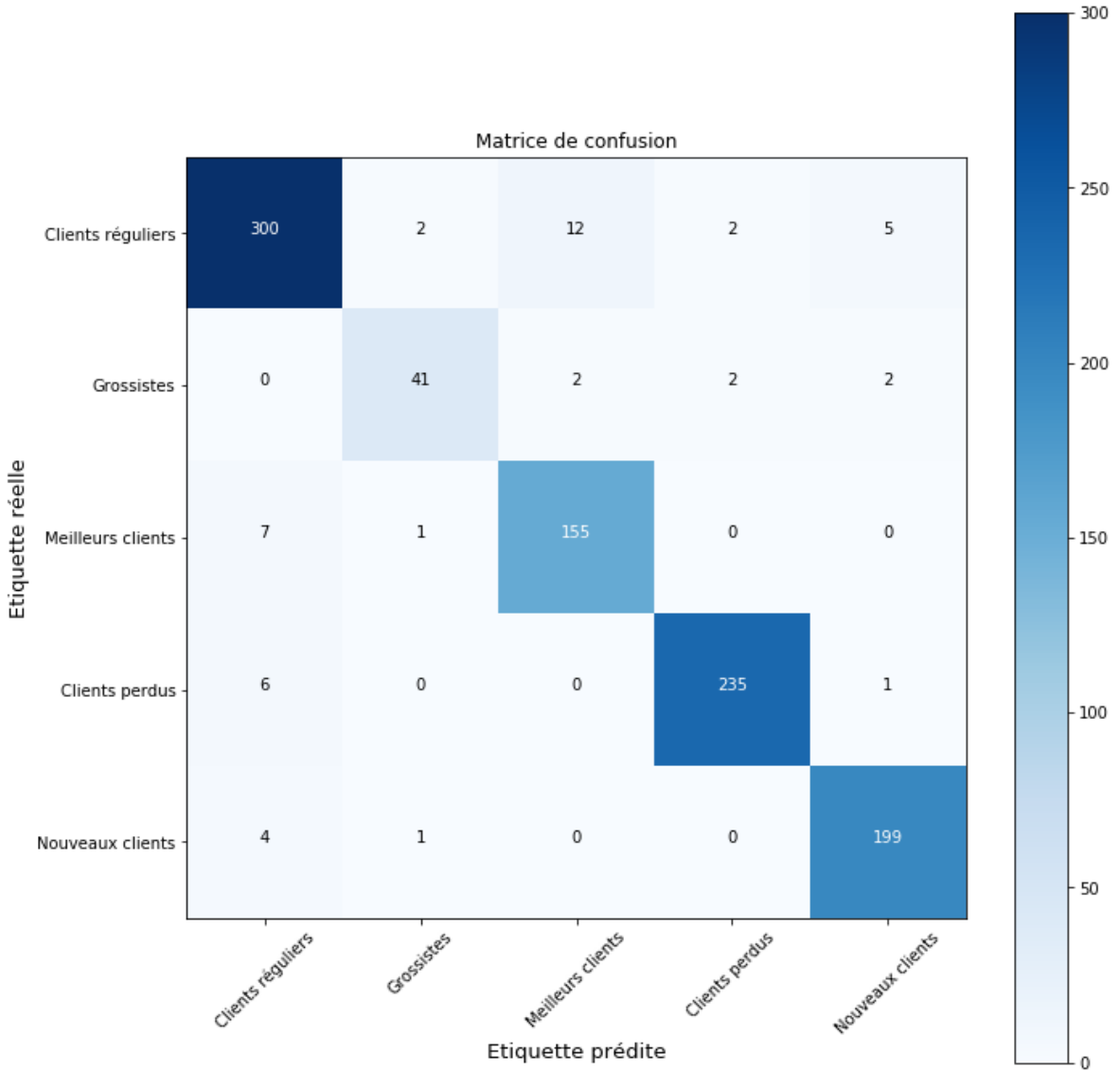


Résultats


Scores de classification
Score (%)
100
98
96
94
92
90
88
86
Régression logistique (OVR)
Régression logistique (OVO)
SVC (OVR)
SVC (CS)
SVC (OVO)
k-NN (OVR)
k-NN (OVO)
Random Forest (OVR)
Random Forest (OVO)

Matrice de confusion

- Peu d’erreurs
- Client réguliers: le plus d’erreurs
  - Normal car c’est la classe « intermédiaire »
- Pas de confusion entre nouveaux clients et clients perdus



Nouveaux clients

- Paramètres de la régression logistique et de la standardisation sont exportés via pickle
- En entrée, le programme prend un fichier csv avec la liste des transactions (même format que le jeu de données)
- En sortie, il donne la liste des clients (id) avec le groupe auquel ils appartiennent

Nouveaux clients - Test

- Client 1: 4 transactions (tous les 3 mois à peu près)
- Client 2: 9 transactions (tous les 1-2 mois)
- Client 3: 1 transaction lors du dernier mois
- Client 4: 1 transaction il y a plus de 6 mois
- Client 5: 1 transaction, avec de très grosses quantités

```
CustomerID
1    Clients réguliers
2    Meilleurs clients
3     Nouveaux clients
4       Clients perdus
5            Grossistes
Name: cluster_name, dtype: object
```