Decembre 2018

Segmentation Clients

Projet OpenClassrooms

Par Xavier Montamat

Problématique

Nous disposons d’un an d’historique de commandes sur un site de vente en ligne.

Nous devrons tout d’abord segmenter chaque client par groupes pertinents

Puis créer un algorithme prédictif afin de prévoir le segment d’un futur client dès sa première commande

Axes d’approche

La segmentation

- Une pertinente pour le client
- Répartition équilibrée et claire
- Juste (anciens vs nouveaux acheteurs)

La prédiction

- Bien respecter la règle de “première commande”
- Interface compatible avec le format client
- Faire attention au Data leak et overfitting

Plan de réalisation

Nettoyage et exploration

- Format initial
- Nettoyage des données invalides
- Exploration et analyse du dataset
- Feature engineering

Segment et modèle

- Segmentation choisie
- Répartition des segments
- Sélection des données de prédiction
- Test de différents modèles et scores

Modèle final

- Présentation du modèle
- Importance des features
- Résultats
- Erreurs

Format des données

- Peu de colonnes
- Organisation par objet
- Feature engineering necessaire

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.550	17850.000	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.390	17850.000	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.750	17850.000	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.390	17850.000	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.390	17850.000	United Kingdom
5	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.650	17850.000	United Kingdom
6	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.250	17850.000	United Kingdom
7	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.850	17850.000	United Kingdom
8	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.850	17850.000	United Kingdom
9	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.690	13047.000	United Kingdom
10	536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	2010-12-01 08:34:00	2.100	13047.000	United Kingdom
11	536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	2010-12-01 08:34:00	2.100	13047.000	United Kingdom

Nettoyage des données

541K lignes

Données supprimées

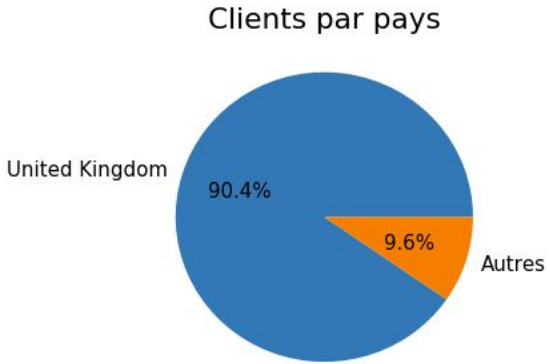
- 135K sans CustomerID
- 5225 duplicats
- 8872 ordres annulés
- 40 prix à 0
- 279 “transactions manuelles”

392K lignes restantes

Les clients

Pays

- 90% United Kingdom
- 10% sur 36 autres pays (dont 8 de plus de 10 clients)



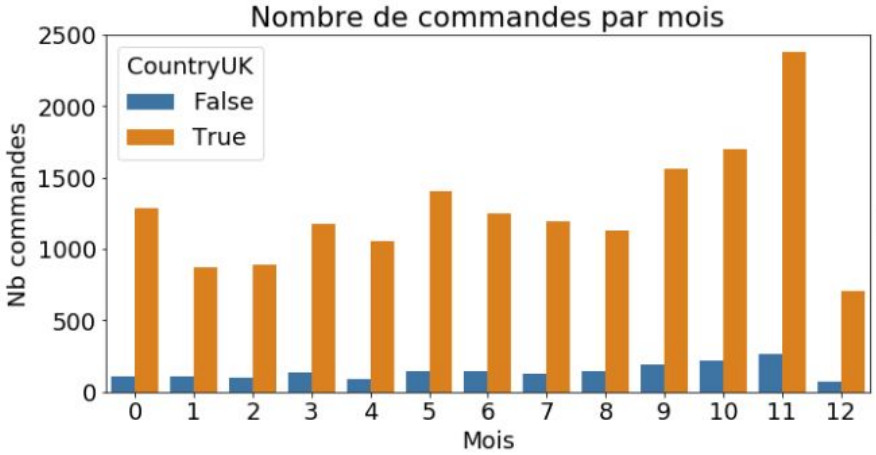
Commandes

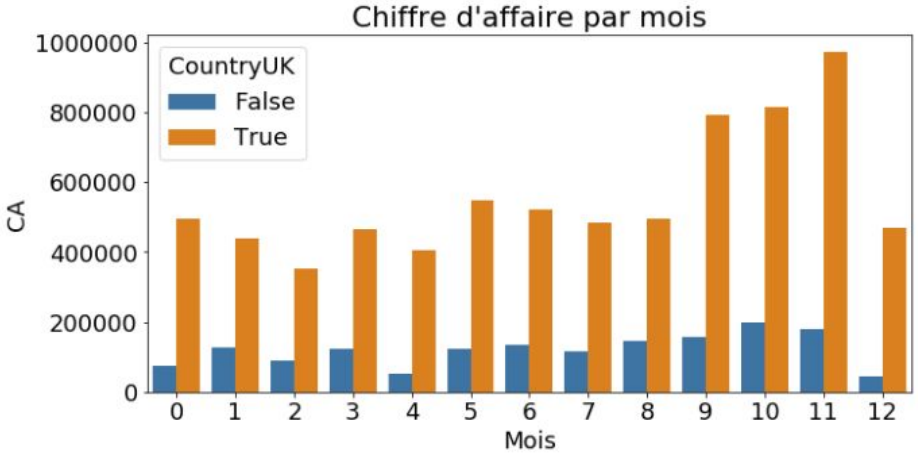
- Mediane de 2 par client (4.2 moy)
- 524 objets commandés pour les clients UK
- 365 pour les autres pays (Médiane)

Dépenses

- Médiane de 645€ de dépenses par client pour le UK
- 1025€ pour les autres pays

Exploration sur l’année





Plutôt équilibré sur l’année

D’avantages de commandes en fin d’année

Segmentation et modèle

Nous devons séparer nos clients en groupes pertinents

Puis utiliser des modèle prédictifs afin de prédire le groupe d'un client en fonction de sa première commande.

Nous analyserons ensuite nos résultats

Métrique choisie

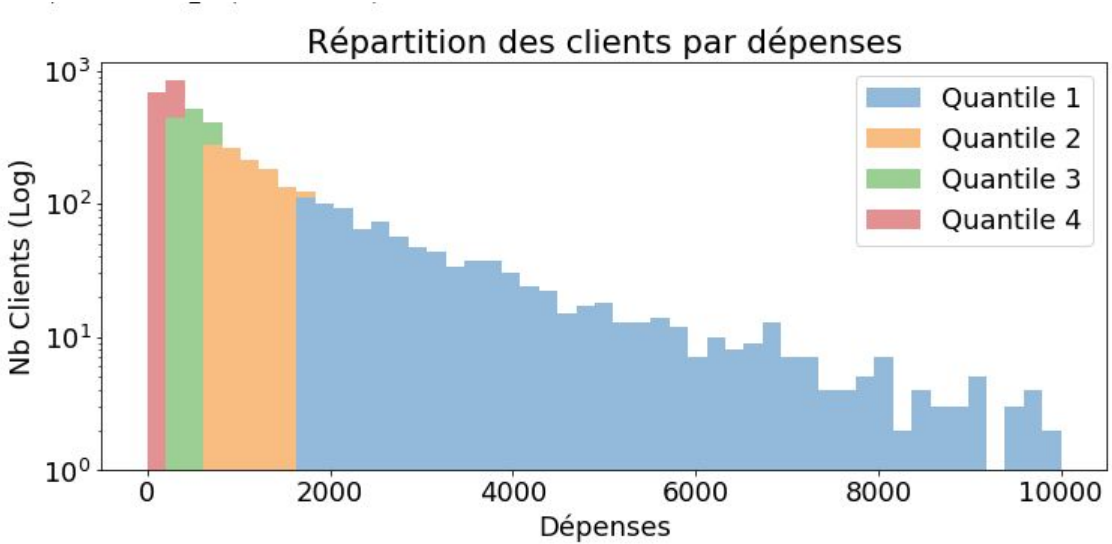
Dépenses du client

- Métrique primordiale
- Simple
- Répartition logarithmique
- Séparation possible en quantiles

Inconvénients

! Biais de l'ancienneté

VS RFM?

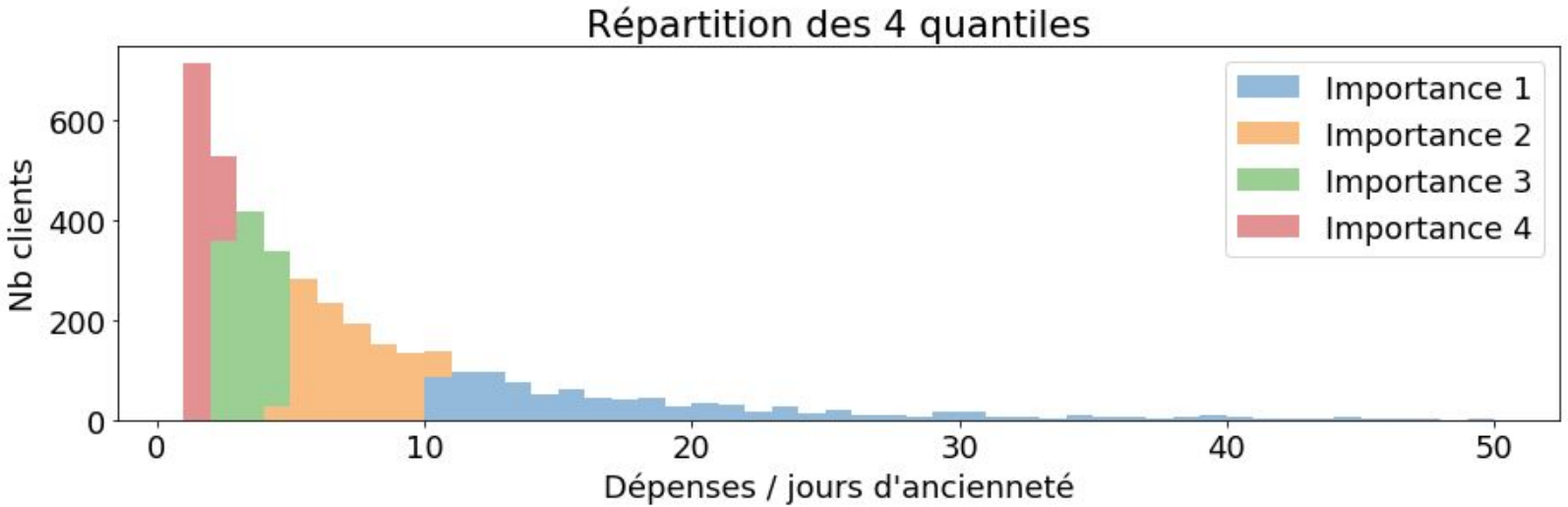


Médiane de 668€

Prise en compte de l'ancienneté

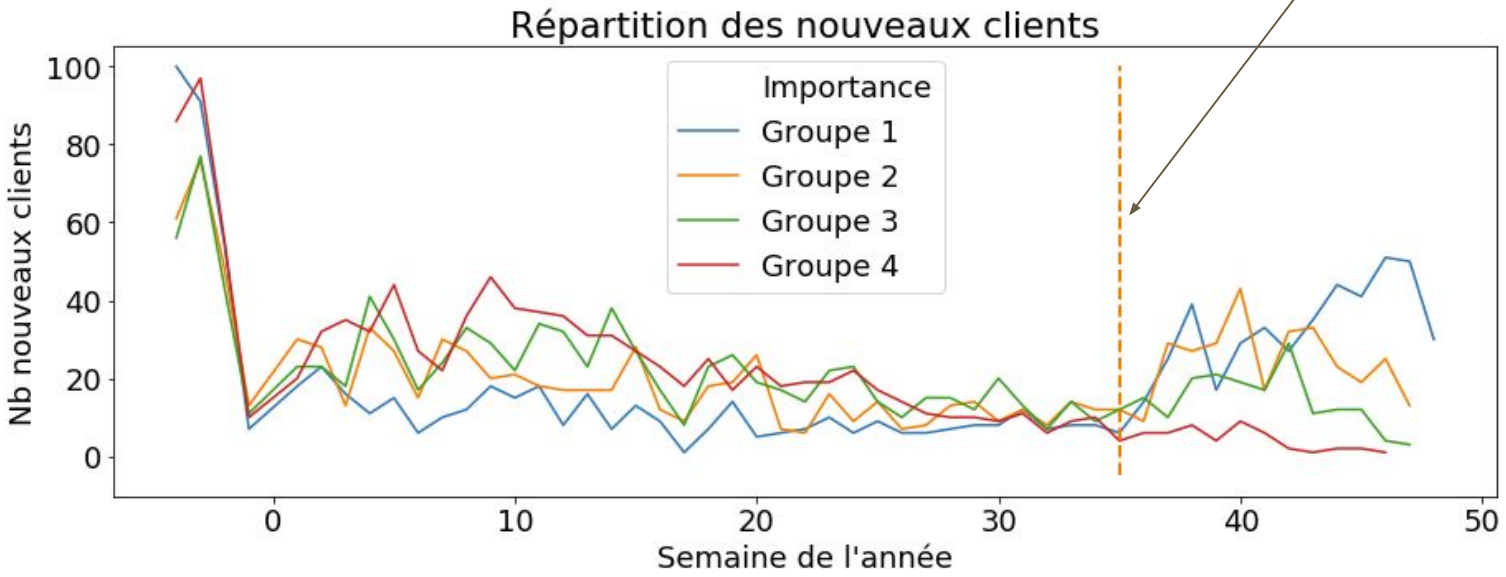
Ancienneté d'un client = Nombre de jours depuis sa première commande

Importance client = Dépenses / Ancienneté



Segmentation sur l'année

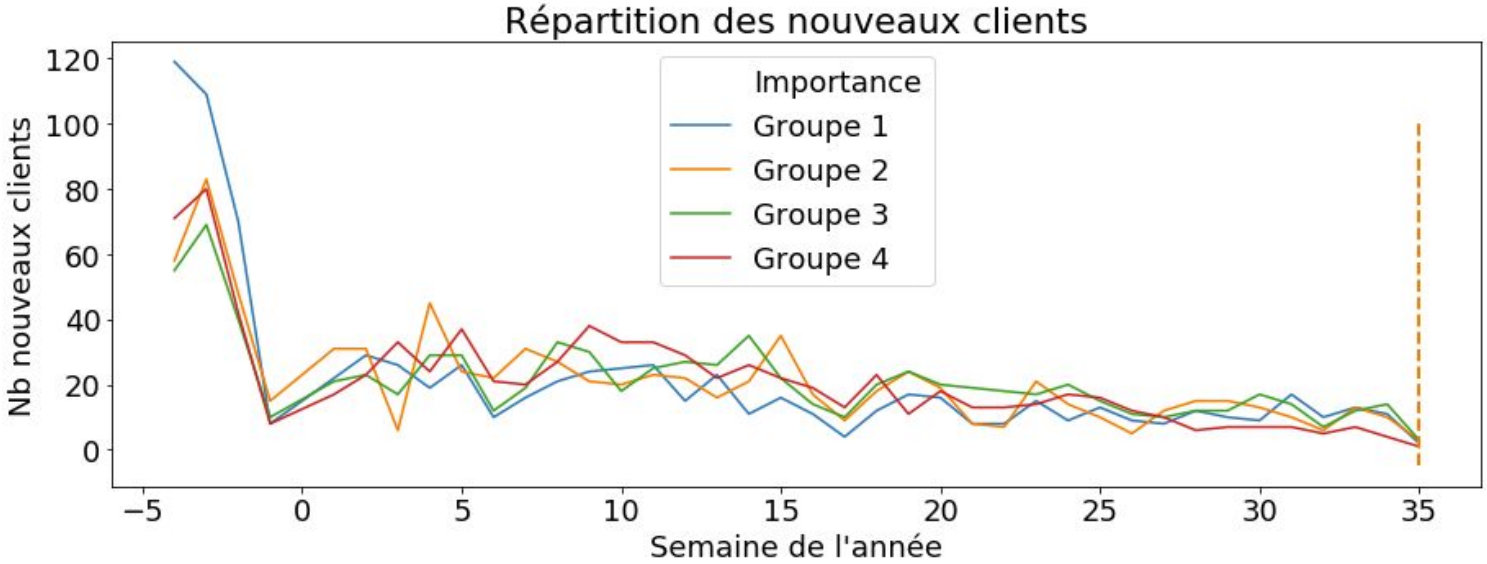
Bien répartis jusqu'à ce point



- Pas assez de recul pour connaître la fidélité
- Nous devrions ignorer les derniers arrivés (outliers)

Nouvelle répartition

Après avoir limité notre dataset de clients



Nos groupes sont équitablement répartis sur l’année

Prédire notre segmentation

Dès la première commande

Données disponible :

- Prix de la commande
  - Prix total
  - Prix moyen par object
  - Prix de l'objet le plus cher
- Pays d'origine
  - Matrice détaillée
  - Local ou non
- Objets détail
  - Quantité totale
  - Quantité par objet
  - Nb objets distinct
  - Matrice détaillée
    - => 3.5K features
- Date
  - Jour/Mois/Semaine
  - Heure

Méthode de test

Sélection des informations

- Première commande

Optimisation des hyper paramètres

- 3 folds Cross Validation

Entraînement du modèle

- Échantillon de 75% stratifié (3.5K samples)

Prediction

- Sur les 25% restant (800 samples)

Performance

- F1 score et rapport de classification

Résultats sur différents modèles

KNeighborsClassifier - F1 0.43 - 15ms

Logistic Regression - F1 0.44 - 155ms

DecisionTreeClassifier - F1 0.40 - 30ms

RandomForestClassifier - F1 0.45 - 700ms

GradientBoostingClassifier - F1 0.43 - 1490ms

Résultats

Des scores très similaires sur les différents modèles

Modèle final

Nous choisissons le modèle des forêts aléatoires comme modèle final

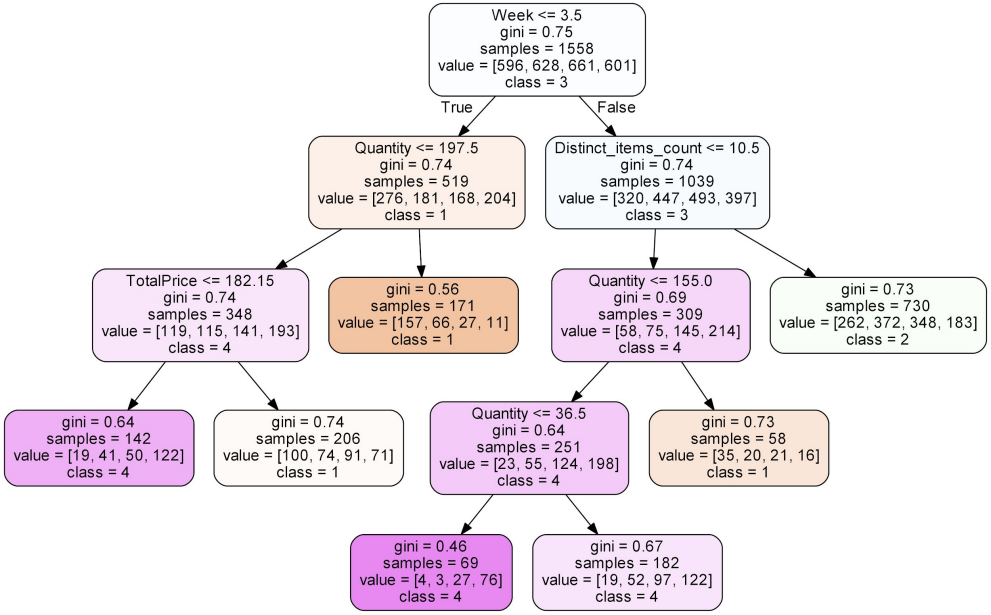
Nous expliquerons le fonctionnement du modèle et de l’importance des features.

Analyse des scores et des erreurs

Forêts aléatoires

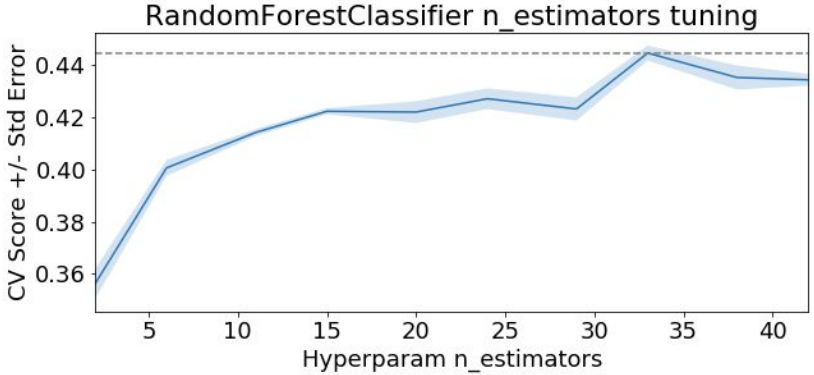
Fonctionnement

- Arbres de décisions binaires
- Plusieurs arbres (estimateurs)



Hyperparamètres

- Nombre d'estimateurs
- Profondeur maximum
- Nombre d'échantillons par feuille



Importance des features

L’algorithme Random Forest estime l’importance de chaque feature pour sa prédiction

Features les plus importantes

- Prix total de la commande
- Quantité
- Nombre d’objets distincts

Retirer les features de moindre importance réduit le bruit

	Importance
TotalPrice	0.520
Quantity	0.249
Distinct_items_count	0.105
Week	0.052
CountryUK	0.027
UnitPrice_avg	0.020
UnitPrice_max	0.019
Time	0.007

Résultats

Rapport de classification

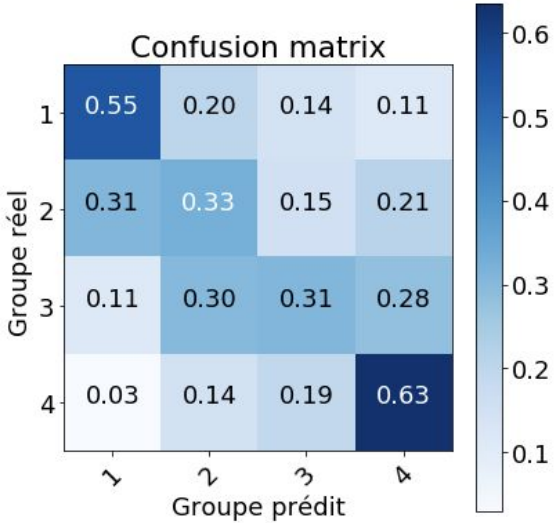
- Le F1 score
- Le minimum (aléatoire)
- Répartition intéressante

693 ms ± 6.21 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Score of : 0.45476477683956573

	precision	recall	f1-score	support
1	0.55	0.55	0.55	207
2	0.34	0.33	0.33	207
3	0.39	0.31	0.35	207
4	0.51	0.63	0.57	208
avg / total	0.45	0.45	0.45	829

Matrice de confusion

- Erreurs cohérentes
- Plutôt vers le centre



Exemples d’erreurs

Client prédit comme importants

- 7 cas
- Grosse première commande
    - Moy 300 Objets
    - Moy 300€
- Pas de deuxième commande
    - Moy 1.1 commande
- Commande ancienne
    - Il y a 345 jours Moy

Clients prédit comme petits

- 35 cas
- Modeste première commande
    - Moy 80 Objets
    - Moy 161€
- Anciens
    - Moy 322 jours
- Mais clients fidèles !
    - Moy 16 Commandes
- Gros chiffres totaux
    - Moy 5800€
    - Moy 3800 Objets

Conclusion

Malgré un taux d’erreurs non négligeable, la prédiction est utile pour notre client

Elle classe de manière efficace les cas les plus extrêmes

Permet de cibler rapidement un acheteur, quitte à ajuster son importance a posteriori

Améliorations possibles

Accumuler plusieurs années de données
Récupérer d'avantage d'informations client
Réajuster la prédiction à chaque commande

Questions

?

Command line

L’outil en ligne de commande permet de prédire les segments de clients

A partir de leur historique de commande dans le format d’origine

Il prend un fichier d’entrée en paramètre --file

Et envoie les résultats dans un .csv en sortie

```
(base) C:\Users\Maly-Fenix\OC_Project5>python 3_App_Segmenting_clients.py --file "Online Retail.xlsx"
This python app will categorize customers into predefined segments.
Importing Online Retail.xlsx
541909 rows and 8 columns found in the file.
10624 items with negative quantity (cancels). Dropping
4339 customers found
970 customers in segment 1
1163 customers in segment 2
1025 customers in segment 3
1181 customers in segment 4
Results exported into Online Retail_export.csv
```