

# Projet de Data Science (Openclassrooms PJ5)

## Segmenter les comportements de clients





**Comment mieux connaître ses clients  
afin de mieux cibler le discours ?**

**Comment identifier les meilleurs clients ?**



# Approche de modélisation

Identifier des ensembles de client les mieux délimités possibles en fonction de leur comportement, via l'analyse des caractéristiques suivantes :

⇒ Les caractéristiques « RFM » (Recency, Frequency, Monetary)



Récence

Nombre d'achats par mois

Montant d'achat par mois

⇒ La tendance à annuler la commande et à bénéficier de réductions

⇒ L'achat de produits parmi ceux qui rapportent le plus

⇒ Le type de produits achetés (en se fondant sur l'analyse de leur description)

L'approche retenue est **non supervisée** afin de **pouvoir explorer les clusters qui seront obtenus** en analysant leurs spécificités

# Démarche suivie

- 1/ Exploration du dataset, et data cleaning
- 2/ Recherche sur les techniques de segmentation marketing
- 3/ Feature engineering, avec agrégation des commandes au niveau client
- 4/ Mise en œuvre de modèles de réduction dimensionnelle sur différentes combinaisons de features (sans validation croisée, pour commencer), et visualisation des résultats
- 5/ Mise en œuvre d'un pipeline complet avec validation croisée.  
Explication de quelques principes théoriques sur les modèles choisis
- 6/ Analyse des clusters produits sur les meilleurs modèles, et sélection du modèle final
- 7/ Réalisation d'un programme de prédiction (avec en entrée une série de commandes)
- 8/ Perspectives pour aller plus loin



# Exploration du dataset Et data cleaning

## Exemple d'une commande

Invoice No	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom

## Chiffres clé

**540 000** commandes

**8.7 M£** chiffre d'affaires total

**4336** clients

**3700** produits

**2016 £** chiffre d'affaire moyen par client

**92** commandes par client en moyenne

## Nettoyage des données

**5000 doublons supprimés**



**135000 commandes sans  
numéro de client  
supprimées**



**Suppression des frais de livraison et des  
lignes de commande manuelles**



**Normalisation de la description des produits**  
⇒ Suppression des espaces (trim)  
⇒ 1 code produit = 1 description unique



Recherche sur les  
techniques de  
segmentation  
marketing



## L'approche RFM est très utilisée :

**R Récence** ⇒ De quand date le dernier achat ?  
**F Fréquence** ⇒ Quantité de produits achetés ?  
**M Montant** ⇒ Montant dépensé ?

Il est recommandé d'appliquer des ratios, des seuils et des flags  
Par exemple : nombre d'actes par mois, est-ce que le montant acheté dépasse un certain seuil, type de produit ...



Feature engineering  
et agrégation des  
données au niveau  
client

## Rappel des champs d'une commande

Invoice  
No StockCode Description Quantity InvoiceDate UnitPrice CustomerID Country



## Agrégation niveau client et ajout des features

**Recency** Composante R du RFM

**TotalQuantityPerMonth** Composante F du RFM

**TotalPricePerMonth** Composante M du RFM

OU

**RfmScore** Score RFM

(Discretisation de chaque composante  
R,F, M et somme globale)

+

**HasEverCancelled** Le client a déjà annulé au moins une fois

**BoughtTopValueProduct** Le client a déjà acheté un produit dans le top 20 des plus rentables

+

**DescriptionNormalized** Mots clés de la description du produit

**Approche bag of words**

# Feature engineering pour les mots clés

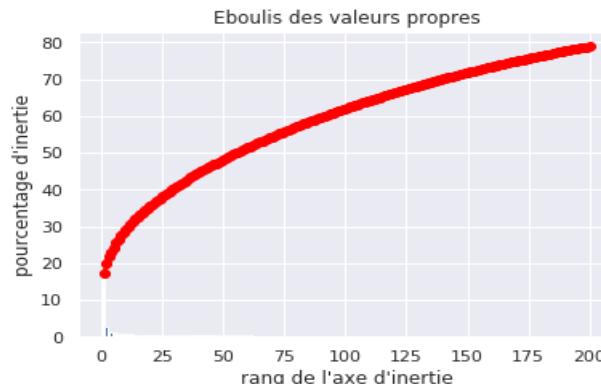
Normalisation des descriptions du dataset : 1 produit = 1 description unique



Approche bag of words : 1 feature par mot clé (valeur 1 ou 0)  $\Rightarrow$  600 features



Explication de la variance via réduction PCA

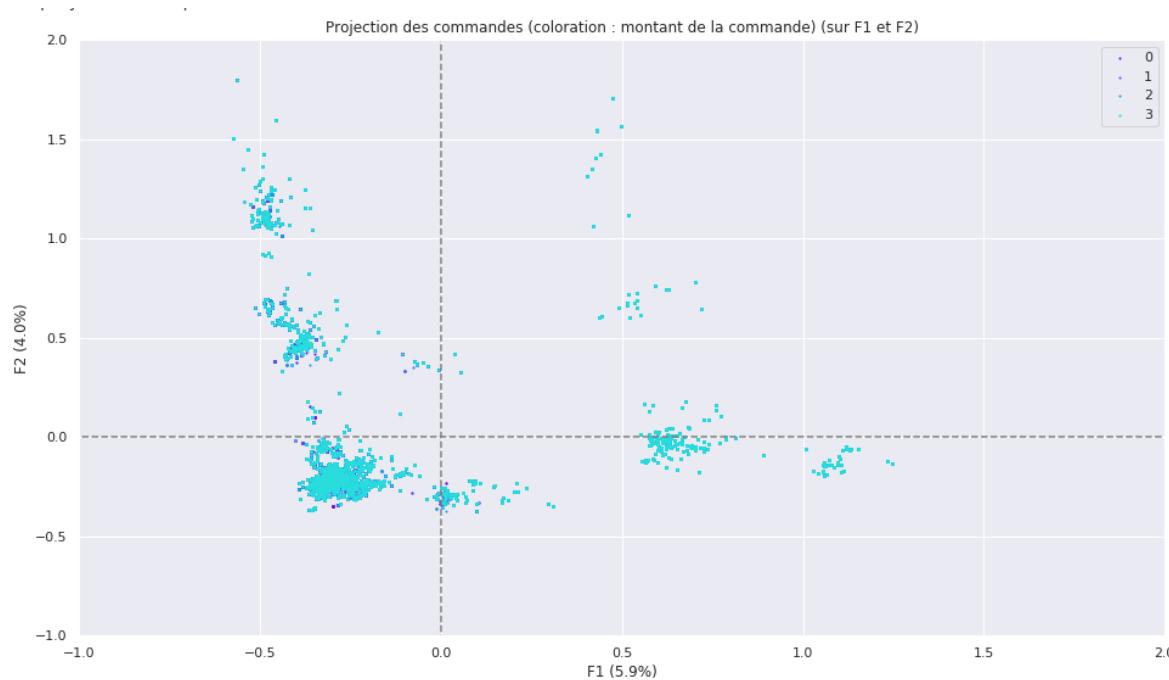


$\Rightarrow$  80 % de la variance expliquée pour 200 features  
 $\Rightarrow$  20 % de la variance expliquée pour 3 features



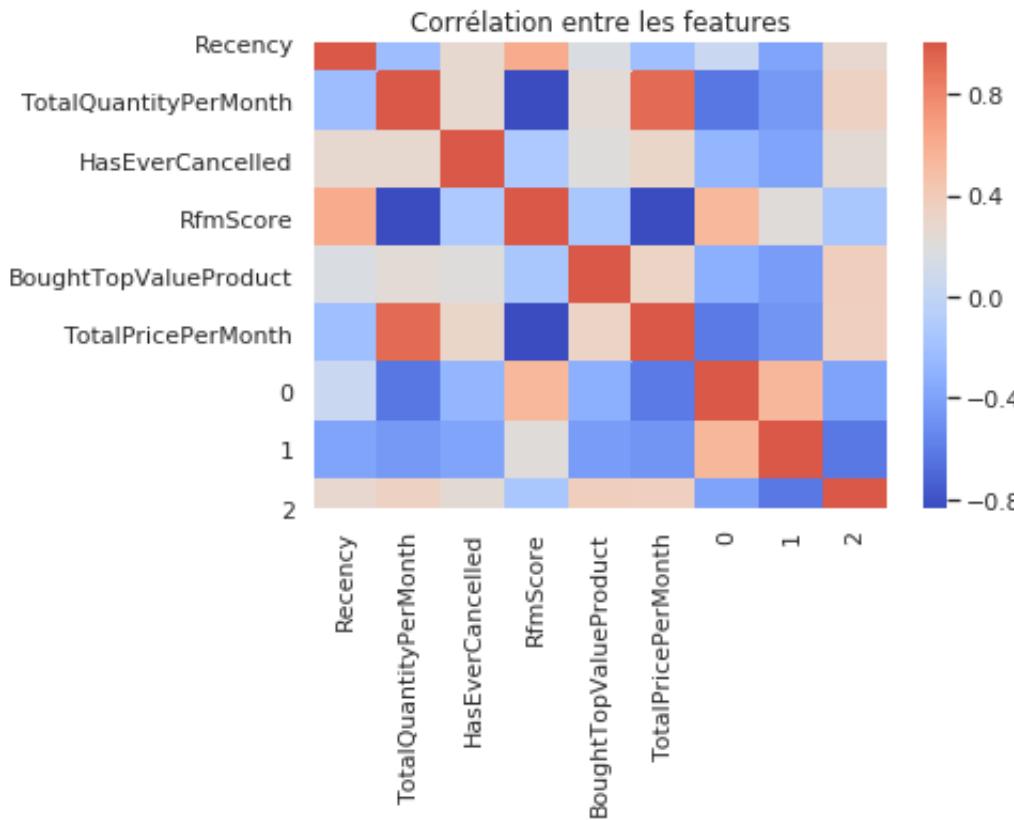
Mise en œuvre de modèles de réduction dimensionnelle sur différentes combinaisons de features (sans validation croisée, pour commencer), et visualisation des résultats

# Projection des commandes sur plan factoriel



Variables projetées :  
Montant total, flag produit top value acheté, et les mots clés des produits

# Agrégation au niveau client et corrélations



⇒ Les features sont globalement corrélées, sans qu'il n'y ait de corrélation trop forte qui justifierait d'exclure des features

⇒ A noter les features 0 et 2 de la description par mots clés qui ont une corrélation avec les autres. Ce qui montre que l'approche par mots clés peut avoir un intérêt

0, 1 et 2 sont les features DescriptionNormalized (bag of words) réduites à 3 dimensions

# Les étapes du modèle

Encodage « bag of words »



Agrégation niveau client et ajout des features vues en slide 12



Sélection des features du modèle



Scaling Min Max (toutes les valeurs entre 0 et 1)



Réduction dimensionnelle

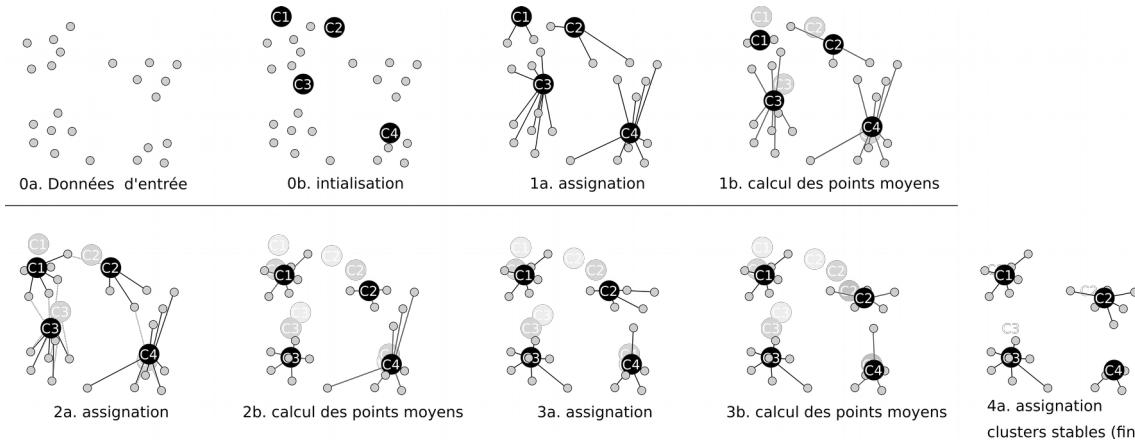


Scaling Min Max (toutes les valeurs entre 0 et 1)



Clustering

# Explication du fonctionnement de l'algorithme KMeans



- 1) Initialisation aléatoire du centre de chaque cluster (le nombre de cluster est fixe et défini à l'avance)
- 2) Répéter les étapes suivantes jusqu'à convergence :
  - 1)Assigner chaque point au cluster le plus proche
  - 2)Recalculer le centre de chaque cluster

## Implémentation d'une fonction transform pour l'algo. Ward

L'algorithme Ward ne disposant pas de fonction « transform » permettant de prédire le cluster d'appartenance d'un nouveau point, on a implémenté cette fonction :

- Entraînement d'un prédicteur KNN sur la base des points du training set
- Pour prédire le cluster d'appartenance d'un nouveau point :
  - Assignation de ce nouveau point à son plus proche voisin
  - Le cluster d'appartenance sera le cluster affecté à ce plus proche voisin dans le training set

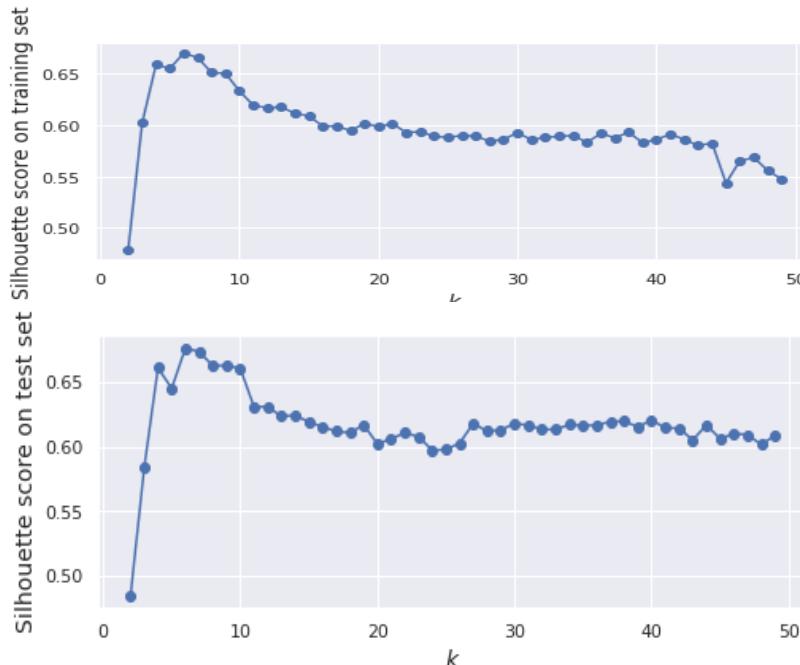
# Visualisation de différents modèles (1/2)

## Caractéristiques

Features : 'TotalPricePerMonth', 'Recency', 'TotalQuantityPerMonth', 'BoughtTopValueProduct', 'HasEverCancelled'

Réduction dimensionnelle : NCA 200

## Score de silhouette



## Réduction à 2 dimensions avec NCA



Ci-dessus on voit un nombre de 4 à 8 clusters se démarquer

## Visualisation de différents modèles (2/2)

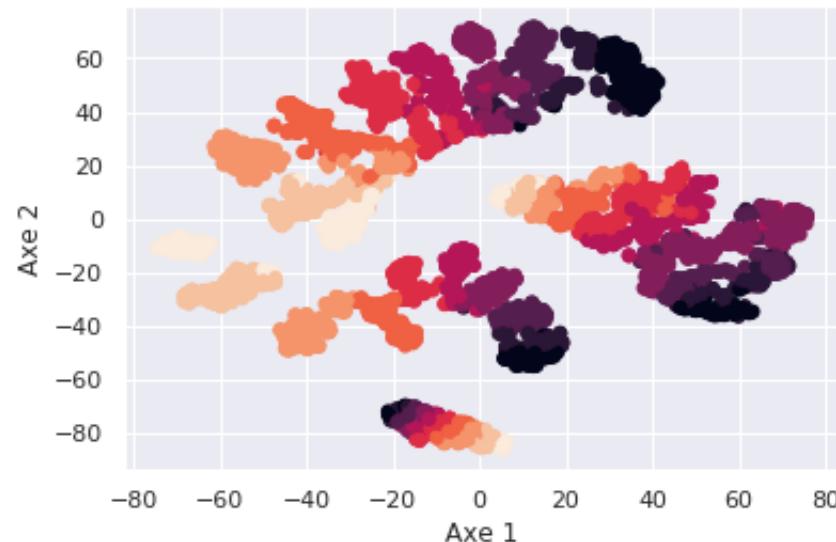
### Caractéristiques

Features : 'DescriptionNormalized', 'BoughtTopValueProduct', 'HasEverCancelled', 'RfmScore'

Réduction dimensionnelle : NCA 3

### Réduction à 2 dimensions avec TSNE

Customers 2d representation, RFM colored, training set



⇒ La feature du score RFM conjointement avec les autres permettra peut-être d'obtenir des clusters délimités



Mise en œuvre d'un  
pipeline complet avec  
validation croisée

# Méthode suivie

**Split training set / test set**



**Recherche paramètres avec validation croisée sur le training set**  
⇒ le training set est lui même splitté 5 fois successivement  
(80 % training set, 20 % test set)



**Shortlist des meilleurs modèles via différents critères**  
(principalement le silhouette score)

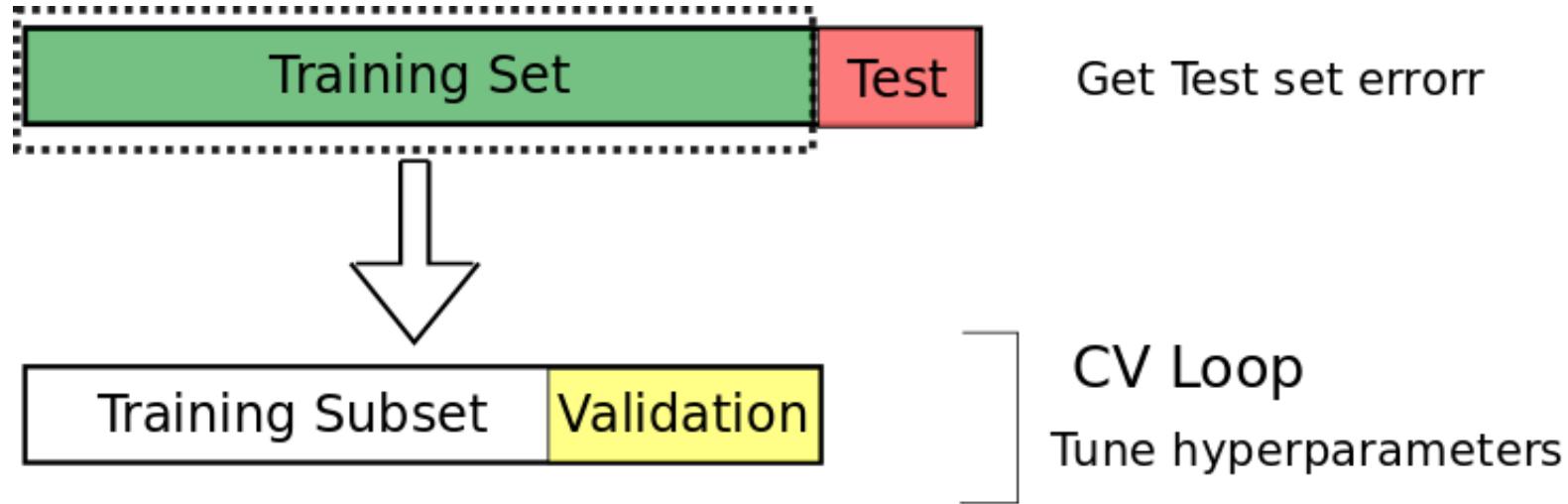


**Interprétation des meilleurs modèles et sélection du modèle final**  
+ un modèle alternatif intéressant



**Vérification des scores sur le test set final**

## Focus sur le split training set et test set : « nested cross validation »



## Test de plusieurs stratifications



Comparaison entre plusieurs stratégies de split training set / test set  
⇒ Peu de différences entre les stratégies comparées

	Silhouette score moyen	Ecart type
<b>Split aléatoire</b>	<b>0,8397</b>	<b>0,004</b>
<b>Split stratifié TotalPrice</b>	<b>0,8402</b>	<b>0,004</b>
<b>Split stratifié InvoiceMonth</b>	<b>0,8401</b>	<b>0,004</b>

# Les étapes de la recherche de paramètres

Encodage « bag of words »



Agrégation niveau client et ajout des features vues en slide 12



**Sélection des features du modèle**



Scaling Min Max



**Réduction dimensionnelle**



Scaling Min Max



**Clustering**

- ⇒ Score RFM globale ou R,F,M séparés
- ⇒ Avec les « bag of words » ou sans

- ⇒ NCA (avec labels bag of words ou score RFM)
  - ⇒ Juste les « bag of words » ou toutes les feats
- ⇒ PCA, LLE, ISOMAP
- ⇒ Nb dimensions : 3, 5, 10, 50, 150, 200, 300

- ⇒ Kmeans
- ⇒ Ward
- ⇒ Nb clusters : 3, 4, 5, 6, 7, 8, 9, 10, 11, 20, 30, 40, 50

# Résultats de la recherche de paramètres

Clustering	n Clusters	Dim Reduction	Dim Reduction Features	Dim Reduction N dim	features_selector__features_toselect	mean_test_score
WARD	50				['BoughtTopValueProduct', 'HasEverCancelled', 'RfmScore']	1,00
KMEANS	50				['BoughtTopValueProduct', 'HasEverCancelled', 'RfmScore']	1,00
WARD	4				['BoughtTopValueProduct', 'HasEverCancelled', 'RfmScore']	0,70
KMEANS	4				['BoughtTopValueProduct', 'HasEverCancelled', 'RfmScore']	0,70
WARD	20				['BoughtTopValueProduct', 'HasEverCancelled', 'RfmScore']	0,69
KMEANS	20				['BoughtTopValueProduct', 'HasEverCancelled', 'RfmScore']	0,69
WARD	4				['BoughtTopValueProduct', 'HasEverCancelled', 'TotalPricePerMonth', 'TotalQuantityPerMonth', 'Recency']	0,66
KMEANS	4				['BoughtTopValueProduct', 'HasEverCancelled', 'TotalPricePerMonth', 'TotalQuantityPerMonth', 'Recency']	0,66
KMEANS	8				['BoughtTopValueProduct', 'HasEverCancelled', 'TotalPricePerMonth', 'TotalQuantityPerMonth', 'Recency']	0,66
WARD	4	NCA	['DescriptionNormalized']	3	['DescriptionNormalized', 'BoughtTopValueProduct', 'HasEverCancelled', 'RfmScore']	0,64
KMEANS	4	NCA	['DescriptionNormalized']	3	['DescriptionNormalized', 'BoughtTopValueProduct', 'HasEverCancelled', 'RfmScore']	0,64
WARD	4	ISOMAP	['DescriptionNormalized']	3	['DescriptionNormalized', 'BoughtTopValueProduct', 'HasEverCancelled', 'TotalPricePerMonth', 'TotalQuantityPerMonth', 'Recency']	0,64
KMEANS	4	ISOMAP	['DescriptionNormalized']	3	['DescriptionNormalized', 'BoughtTopValueProduct', 'HasEverCancelled', 'TotalPricePerMonth', 'TotalQuantityPerMonth', 'Recency']	0,64
WARD	8				['BoughtTopValueProduct', 'HasEverCancelled', 'TotalPricePerMonth', 'TotalQuantityPerMonth', 'Recency']	0,64
KMEANS	6	ISOMAP	['DescriptionNormalized']	3	['DescriptionNormalized', 'BoughtTopValueProduct', 'HasEverCancelled', 'TotalPricePerMonth', 'TotalQuantityPerMonth', 'Recency']	0,63
KMEANS	6	ISOMAP	['DescriptionNormalized']	3	['DescriptionNormalized', 'TotalPricePerMonth', 'TotalQuantityPerMonth', 'Recency']	0,46
KMEANS	4	NCA	['DescriptionNormalized']	3	['DescriptionNormalized', 'RfmScore']	0,39

- ⇒ Les modèles en jaune et vert ont été shortlistés pour une analyse plus précise sur la base des critères suivants :
- Un silhouette score pas trop faible (mais pas trop élevé pour éviter l'overfit)
  - Eviter d'avoir trop de clusters (pour ne pas nuire à l'interprétabilité, et éviter l'overfit)
  - Essayer aussi bien des modèles qui utilisent le RfmScore, que les features R,F,M éclatées
  - Essayer les modèles avec les features de mots clés (bag of words), en incluant ou non les features « BoughtTopValueProduct » et « TotalPricePerMonth »



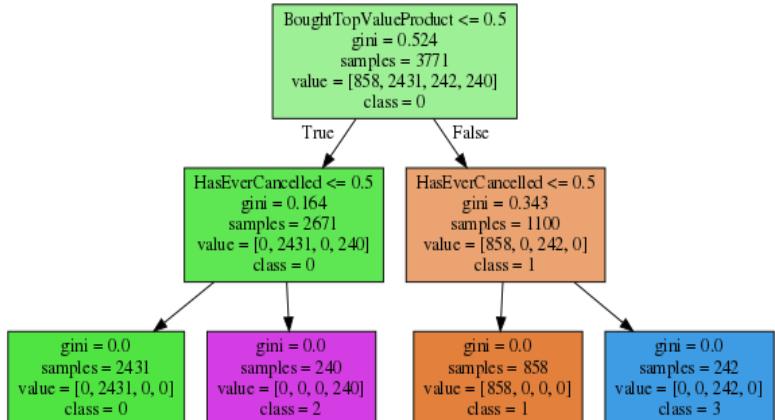
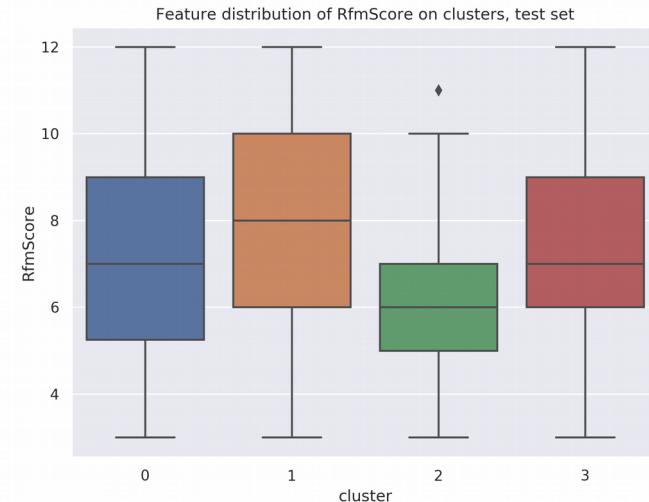
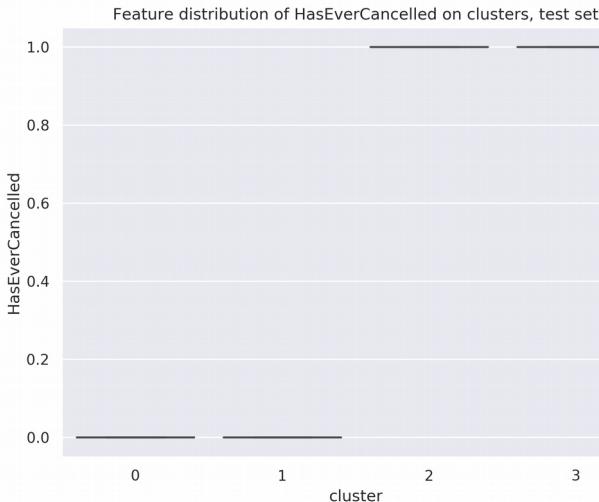
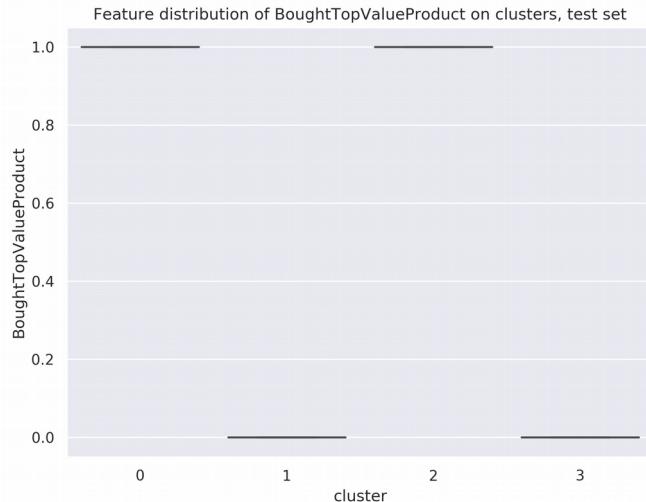
Analyse des clusters  
produits sur les  
meilleurs modèles, et  
sélection du modèle  
final



On a inspecté chaque modèle en regardant :

- ⇒ Le score sur le test set final
- ⇒ La répartition des features dans les différents clusters via un boxplot pour chaque feature
- ⇒ L'entraînement d'un arbre de décision sur le test set (pour prendre en compte la généralisation du modèle) : regarder les 3 premiers étages de l'arbre de décision pour interpréter le modèle
- ⇒ Les notes de feature importance de cet arbre : la diversité des features importances et leur équilibre  
NB : la feature importance est calculée en regardant pour chaque nœud la diminution du score d'impureté (gini) obtenu, pondéré par la probabilité d'atteindre ce nœud (proba = n samples node \* n\_samples total)

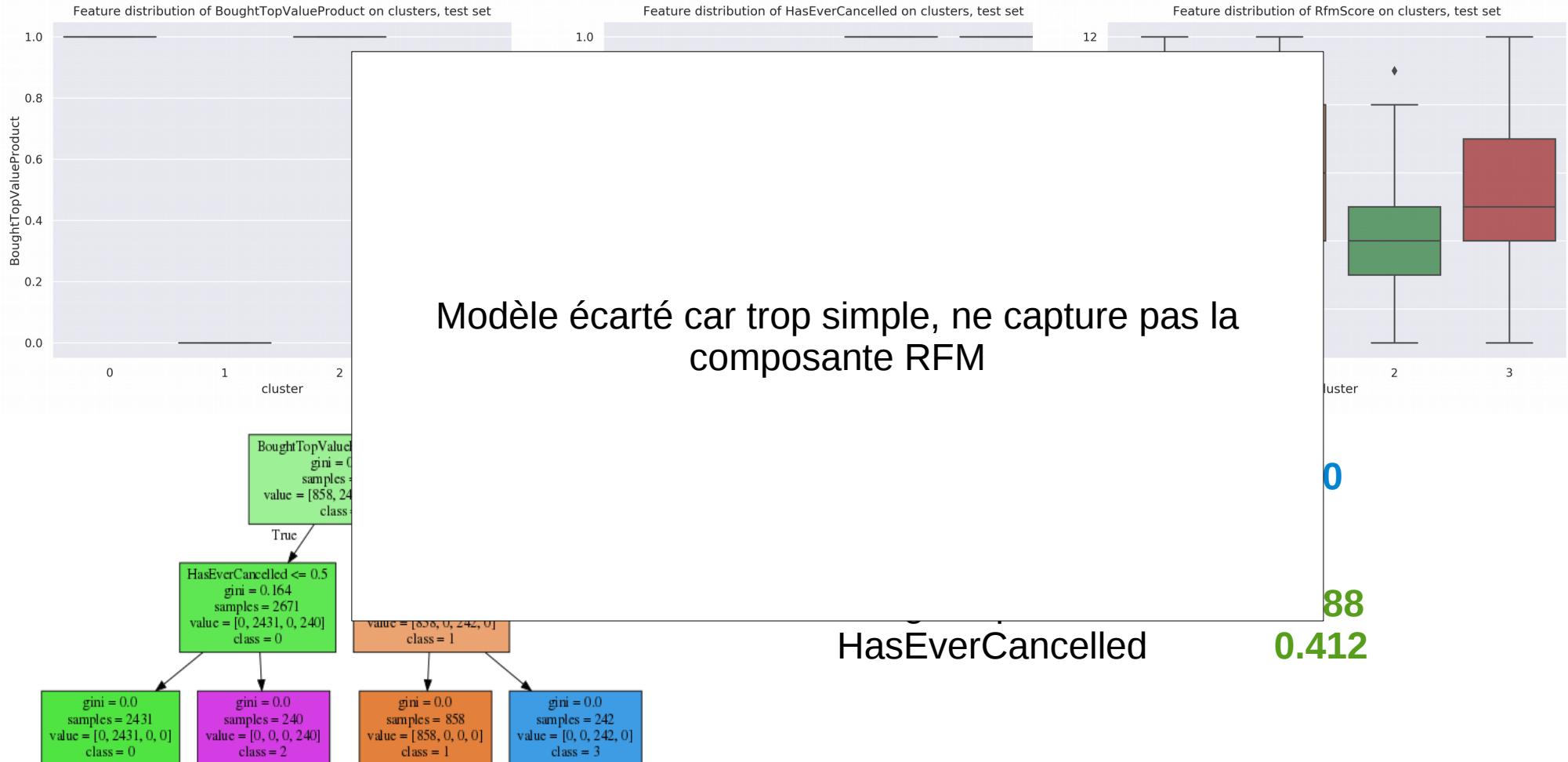
# Modèle 1



Silhouette score final **0.70**

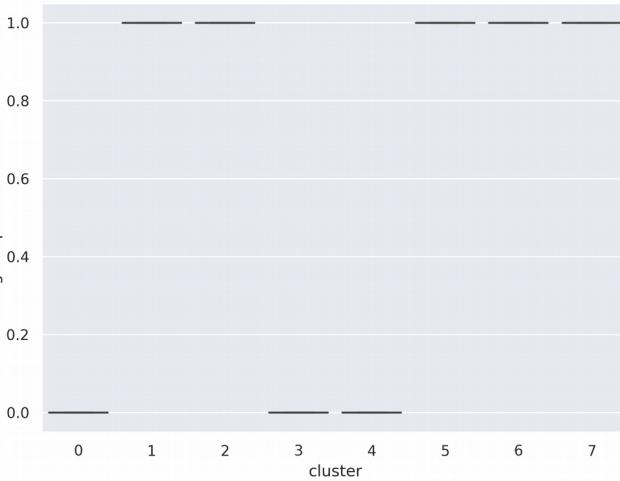
Feature importances :  
BoughtTopValueProduct **0.588**  
HasEverCancelled **0.412**

# Modèle 1

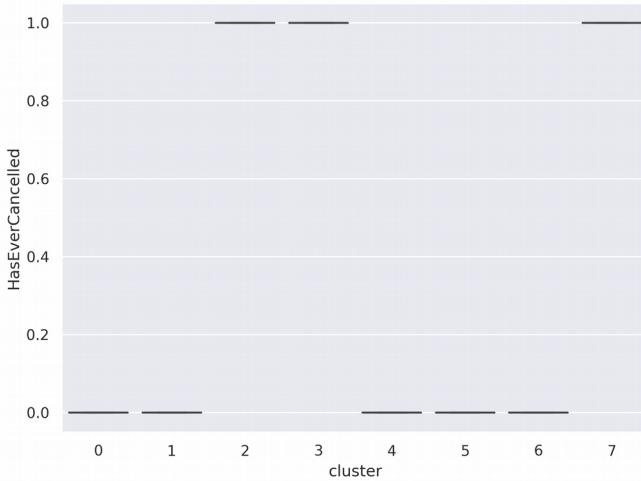


# Modèle 2

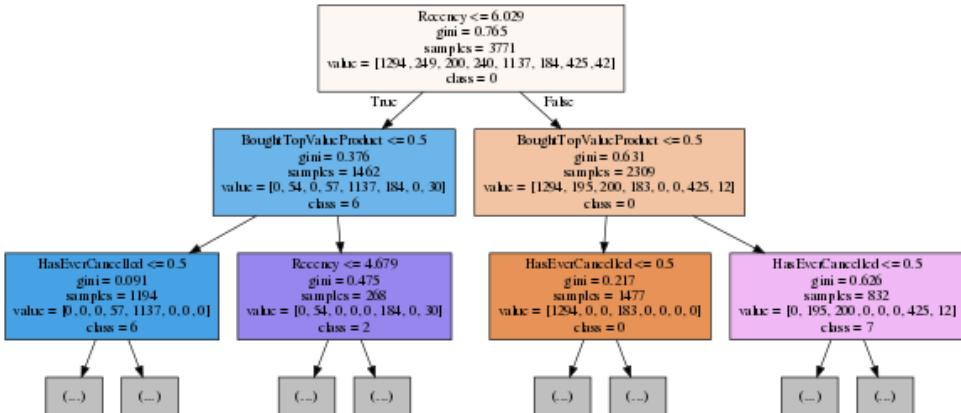
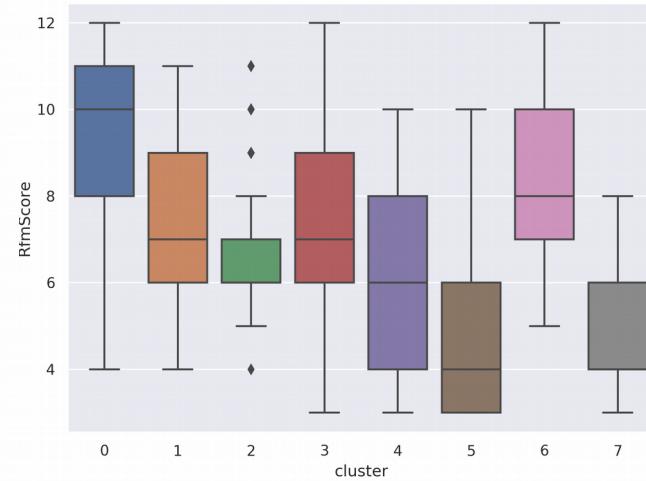
Feature distribution of BoughtTopValueProduct on clusters, test set



Feature distribution of HasEverCancelled on clusters, test set



Feature distribution of RfmScore on clusters, test set



Silhouette score final

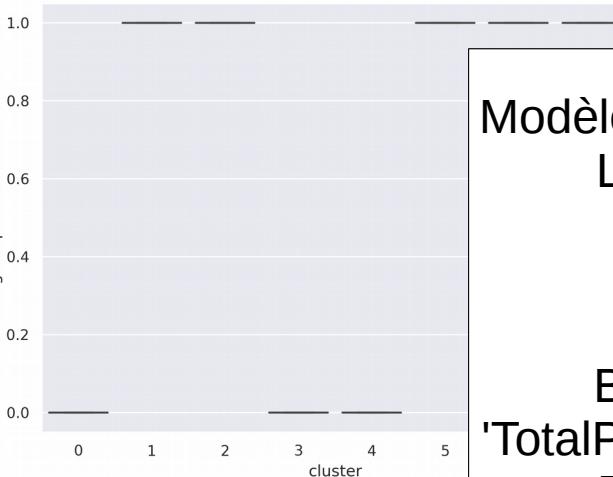
**0.66**

**Feature importances :**  
**Recency**  
**BoughtTopValueProduct**  
**HasEverCancelled**

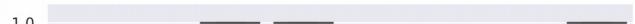
**0.43**  
**0.32**  
**0.25**

# Modèle 2

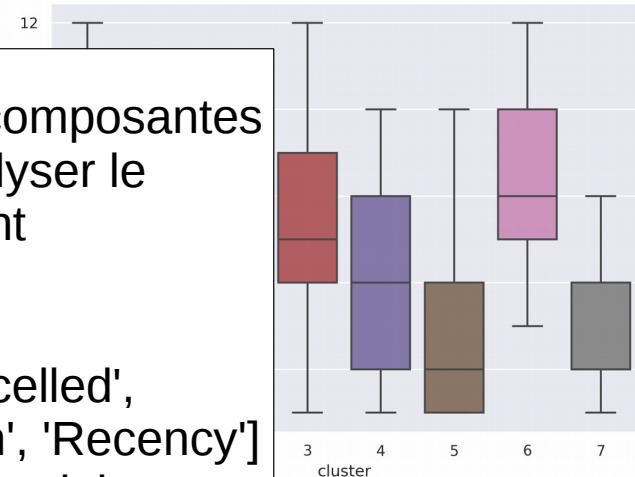
Feature distribution of BoughtTopValueProduct on clusters, test set



Feature distribution of HasEverCancelled on clusters, test set



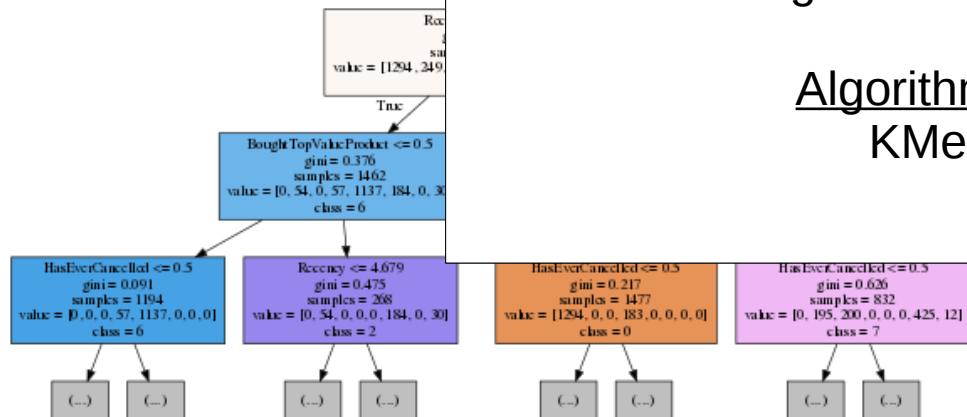
Feature distribution of RfmScore on clusters, test set



Modèle retenu car bien équilibré entre les 3 composantes  
Les features sont importantes pour analyser le comportement et la valeur du client

## Features

'BoughtTopValueProduct', 'HasEverCancelled',  
'TotalPricePerMonth', 'TotalQuantityPerMonth', 'Recency']  
Pas de bag of words / description des produits



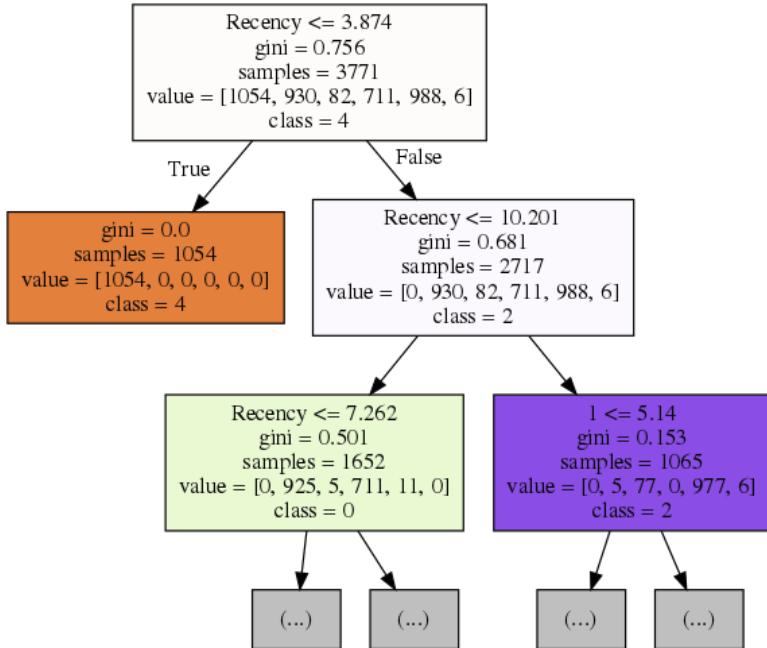
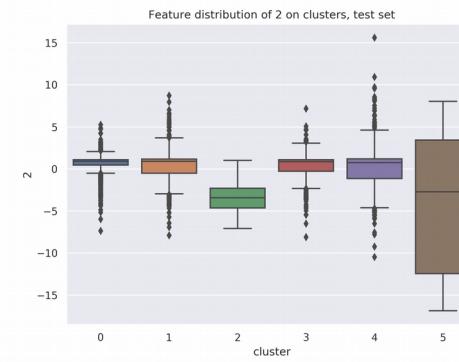
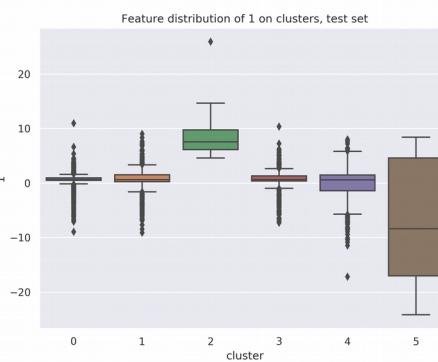
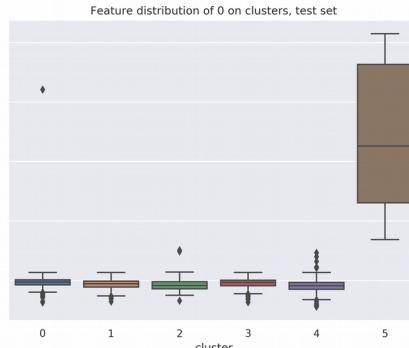
## Algorithme de clustering : KMeans 8 clusters

BoughtTopValueProduct **0.32**  
HasEverCancelled **0.25**

56

13

# Modèle 5



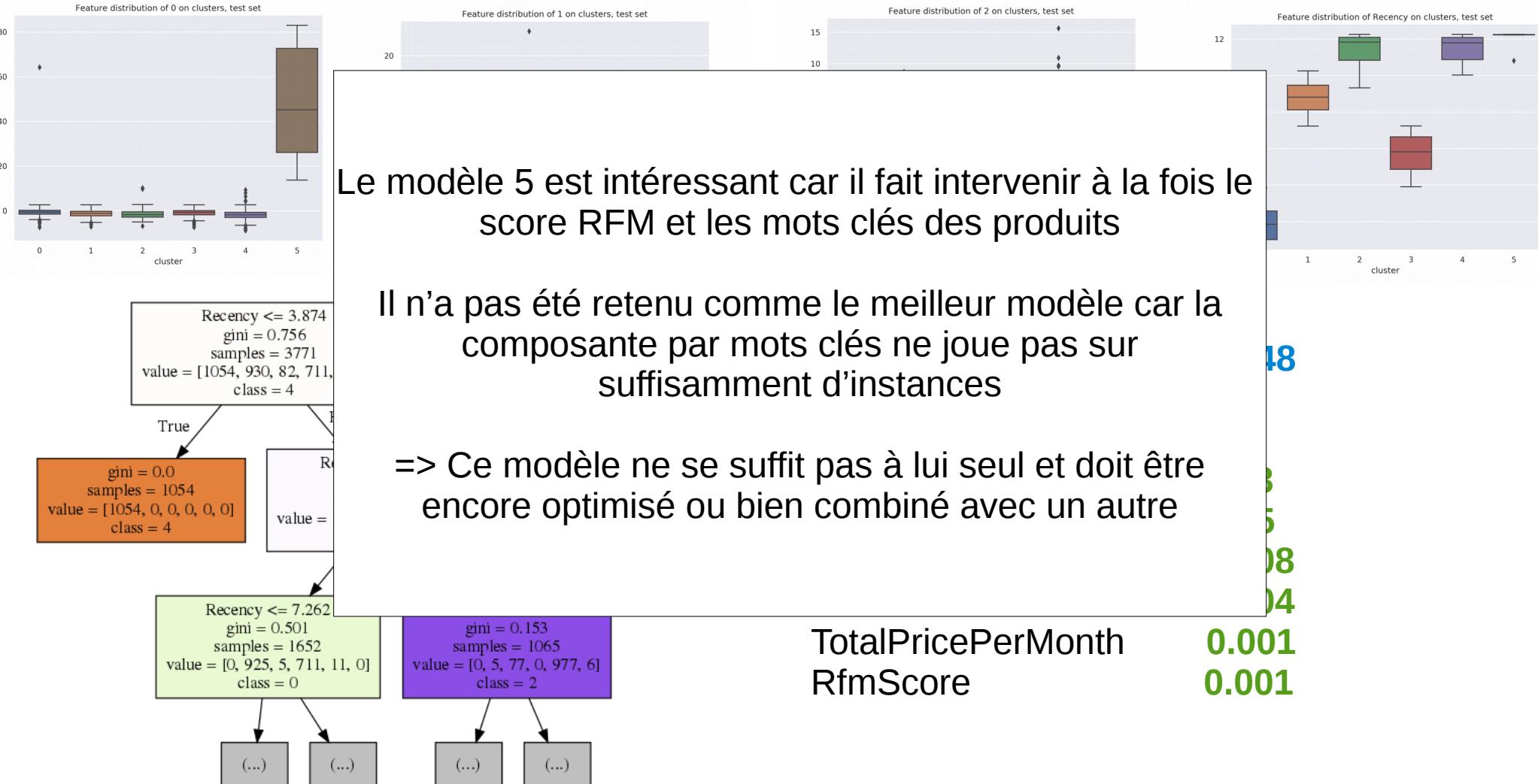
Silhouette score final **0.48**

Feature importances :

Recency	<b>0.93</b>
1	<b>0.05</b>
2	<b>0.008</b>
3	<b>0.004</b>
TotalPricePerMonth	<b>0.001</b>
RfmScore	<b>0.001</b>



# Modèle 5





Réalisation d'un programme de prédiction (avec en entrée une série de commandes)

# Interface hébergée sur AWS

<https://pj5.analysons.com/>

x

≡

### Order characteristics

By default, predictions are made from template input data

To change input data: download template file, update downloaded file with CSV/text editor, and import

[Download template file](#)

Import CSV input file

template\_test2.csv  
[browse files](#)

### Model analysis

Clusters/features distribution

Interpretation tree

Display debug data

## Openclassrooms Data Science training project 5 : e-commerce clients segmentation

François BOYER



Input data for clustering : data loaded by user

Assigned clusters for input data :

CustomerID	Cluster number
0	1
1	2
2	0

## Distribution of features accross clusters

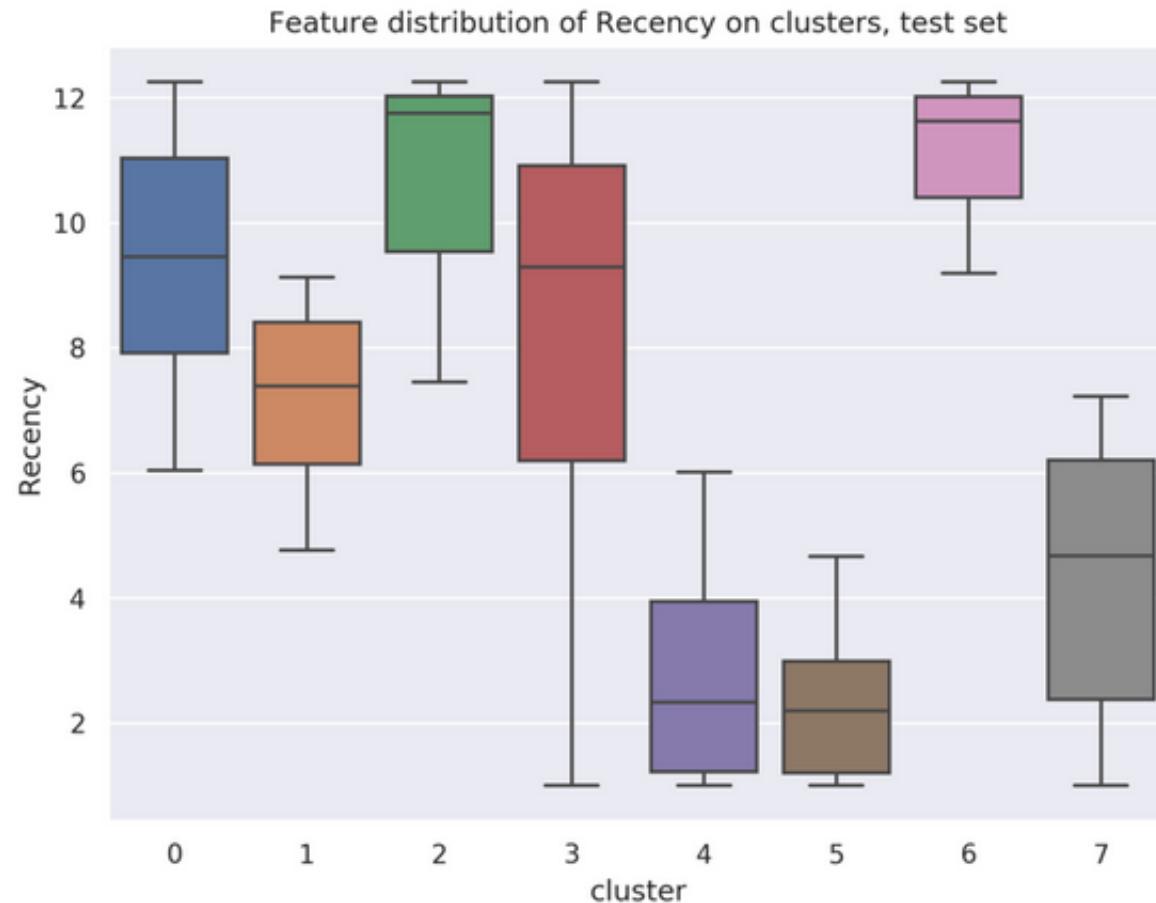
25-75% of the values are located in colored part of the boxes below

### Model analysis

Clusters/features distribution

Interpretation tree

Display debug data



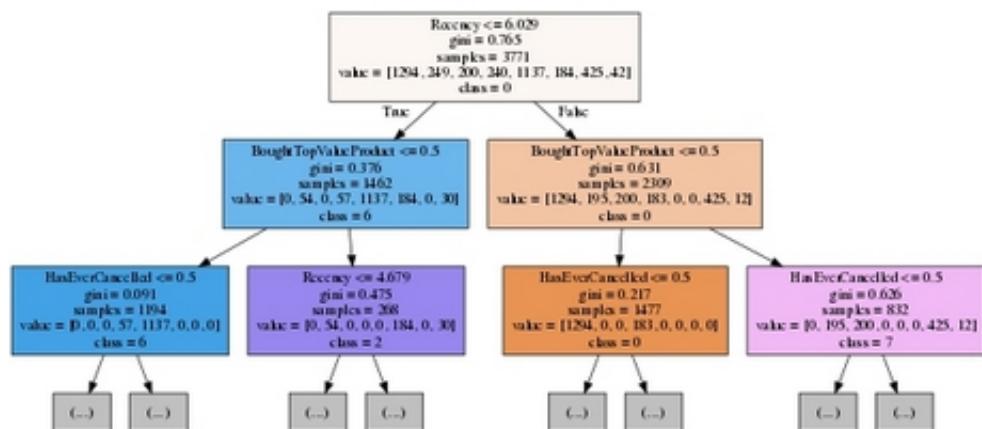
## Model analysis

- Clusters/features distribution
- Interpretation tree
- Display debug data



## Tree interpretation of clusters (simplified to depth 3)

This tree helps to interpret which main steps are processed by the model in most cases when it guesses which cluster to assign to a client



## Model analysis

- Clusters/features distribution
- Interpretation tree
- Display debug data



Step 0 : Input data

	InvoiceNo	StockCode	Quantity	InvoiceDate	CustomerID	TotalPrice	DescriptionNormalized	InvoiceMonth
0	C541433	23166	-74215	2011-01-18 10:17:00	12346	-77,183.6000	MEDIUM CERAMIC TOP STORAGE JAR	2011-01-01 00:00:00
1	541431	23166	74215	2011-01-18 10:01:00	12346	77,183.6000	MEDIUM CERAMIC TOP STORAGE JAR	2011-01-01 00:00:00
2	544203	21718	12	2011-02-17 10:30:00	12362	15	RED METAL BEACH SPADE	2011-02-01 00:00:00
3	99999	21718	12	2022-02-17 10:30:00	1	15	MY PRODUCT	2022-02-01 00:00:00

Step 1 : Data aggregated at client level (Unscaled feature)

	CustomerID	TotalPricePerMonth	HasEverCancelled	BoughtTopValueProduct	Recency	TotalQuantityPerMonth	RfmScore
0	1	15	0	0	1	12	7
1	12346	7,225.7833	1	1	10.6817	6,947.8685	6
2	12362	1.5471	0	0	9.6954	1.2377	10

Step 2 : Data before clustering, at client level (scaled features)

	CustomerID	BoughtTopValueProduct	HasEverCancelled	Recency	TotalPricePerMonth	TotalQuantityPerMonth
0	1	0	0	0	0.0006	0.0008
1	12346	1	1	0.8598	0.2890	0.4444
2	12362	0	0	0.7722	0.0001	0.0001



Perspectives pour  
aller plus loin



Supprimer les outliers sur TotalPricePerMonth et TotalQuantityPerMonth pour que ces features soient mieux prises en compte par les modèles

Utiliser un meilleur embedding pour mieux capturer le sens des mots clés des produits

Combiner plusieurs modèles spécialisés sur l'analyse de différents axes

Possibilité de réentraîner un autre modèle spécialisé sur le premier achat : ne conserver dans le training set que la première commande : voir ensuite le résultat sur le test set qui lui a l'ensemble des commandes. De plus ce modèle ne disposera pas de la composante Référence ni fréquence (1 commande ne suffit pas pour que ce soit représentatif)



Respect de bonnes  
pratiques de  
développement sur la  
base de la norme  
PEP8

# Bonnes pratiques de développement

**Utilisation de variables en majuscules en début de code pour les constantes**

**Déport des traitements les plus complexes et les plus réutilisables dans functions.py**

**Un certain standard dans ma façon de coder au niveau des espaces**

Exemple: laisser un espace après les virgules dans les paramètres.

```
st.sidebar.markdown(get_table_download_link(df_template), unsafe_allow_html=True)
```

**Pas d'espace entre les paramètres, le signal égal et leur assignation quand ce sont des paramètres à l'intérieur des fonctions.**

Mais un espace quand ce sont des assignations de variables :

```
st.sidebar.markdown(get_table_download_link(df_template), unsafe_allow_html=True)
```

```
df_input = df_template
```

**Commenter les fonctions importantes du code avec une description au début**

**Utiliser des noms de variables explicites :**

Exemple : model\_before\_clustering = model\_object['model\_before\_clustering']

**En général mes variables de dataframe commencent par df\_ et les series par series\_ et je les suffixe par \_train ou \_test selon le type de jeu de données**

```
df_grid_search_results.to_csv(GRIDSEARCH_FILE_PREFIX + save_file_suffix + '.csv')
```