

Parcours DataScientist Projet 5:

Segmentation des comportements de clients



Jérôme d'Harveng

Mentor: Pierre Comalada

OpenClassrooms avril 2019

Table des matières

PART I: Contexte de l'analyse de données

PART II: Nettoyage des données

PART III: Analyse exploratoire

PART IV : Pistes de modélisation

PART V: Modèle final et performances

PART VI: Conclusions



Part I:

Contexte de l'analyse



Problématique



- *AirData:*

* *But final:*

à partir des paramètres d'une ou plusieurs commandes

=> *classer le(s) client(s)*

* *Informations:*

une **Base de données**

reprenant les transactions commerciales du 01/12/2010 au
09/12/2011 (~540 000 lines)

* *Points importants:*

Segmentations (clustering) et évaluation des modèles
(Non-supervisé + supervisé)

Interprétation et déductions



- ***Brève analyse de la littérature*** (annexes)
- ***Etude exploratoire*** nécessaire => se focaliser sur les paramètres clés
- **Feature engineering** => spécifique à la segmentation commerciale
- **Clustering Non-supervisé**
 - * K-means,...
- **Méthodes ensemblistes (supervisé)**
 - utilisation des labels du clustering non-supervisé*
 - * Random Forest / Gradient Boosting
- **Evaluation du modèle**
 - * Validation croisée
 - * Precision / Recall / F-score / Matrice de confusion

Part II:

Nettoyage des données



Nettoyage de la Base (CLEANING : 1)

- **Valeurs manquantes** => différentes options

* **Supprimer certaines lignes**

« CustomerID » = 25% de valeurs manquantes

* **Supprimer certaines variables**

« Description »

- **Suppression des doublons**

- **Conversion type de données:**

* float vers int (ex. : 'CustomerID')

- **Valeur erronée:**

* ex. « Unspecified » pour Country



Nettoyage de la Base (CLEANING : 2)

- *Pour la partie EXPLORATION*

- * On garde également les commandes adaptées / annulées

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
440983	C574558	22946 WOODEN ADVENT CALENDAR CREAM	-6	2011-11-04 15:48:00	12.75	15187.0	United Kingdom
491887	C578077	22910 PAPER CHAIN KIT VINTAGE CHRISTMAS	-20	2011-11-22 16:18:00	2.55	12936.0	United Kingdom
391862	C570683	23203 JUMBO BAG VINTAGE DOILY	-2	2011-10-11 16:09:00	2.08	16161.0	United Kingdom

- *Pour la partie SEGMENTATION*

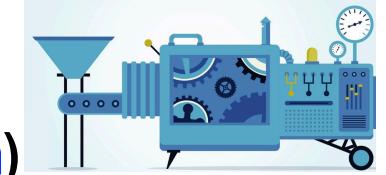
- * Suppression des lignes avec Quantity < 0 et/ou UnitPrice<0
- * Suppression des StockCode spéciaux (moins de 5 digits)

Ce qui correspond principalement aux annulations de commandes

```
: df_segmentation.query('StockCode.str.len() < 5').Description.value_counts()
```

POSTAGE	1099
Manual	279
CARRIAGE	133
DOTCOM POSTAGE	16
PADS TO MATCH ALL CUSHIONS	3

Feature Engineering (pour l'exploration)



- **Variable :** **Montant** (*ligne commande*)

= Quantity x UnitPrice

- **Variables temporelles:** « **InvoiceDate** »

- * Month
- * DayOfMonth
- * DayOfWeek
- * Hour



Part III:

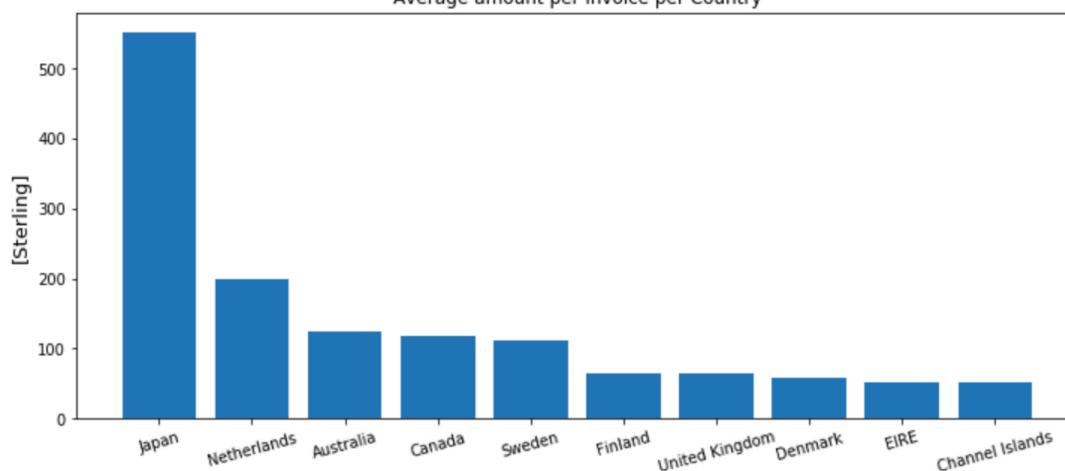
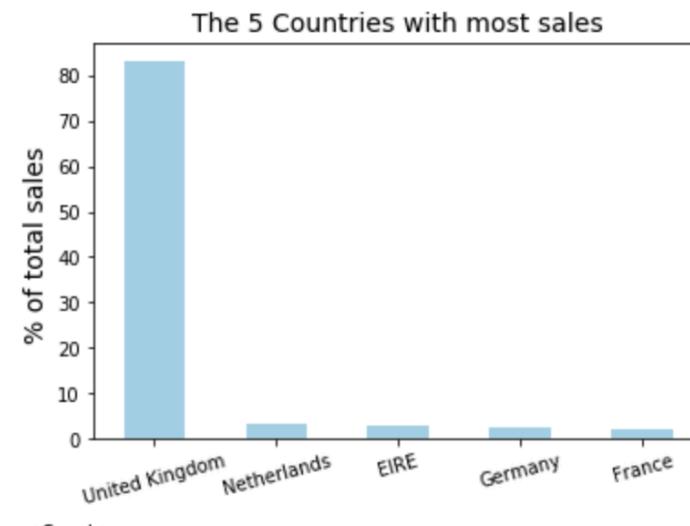
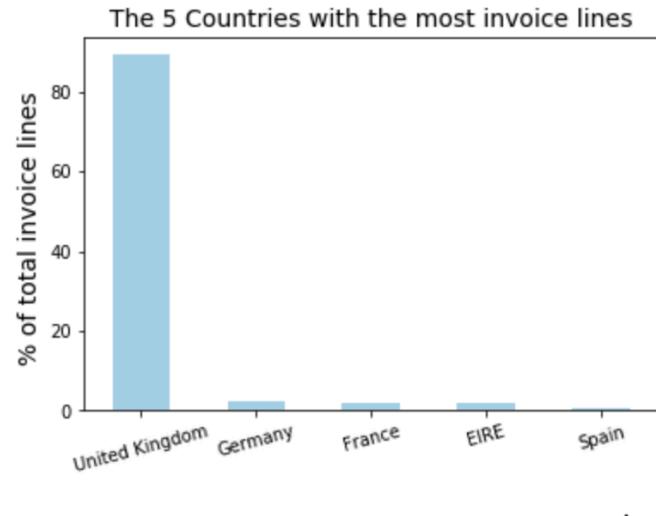
Analyse Exploratoire



Par pays

Observations: (*sur commandes non-annulées*)

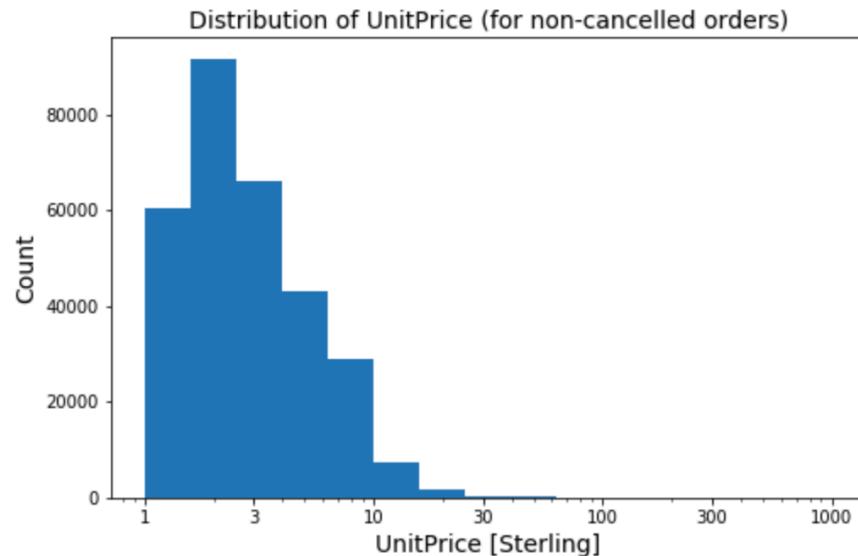
- **Country:** 36 pays



Info par ligne de commande

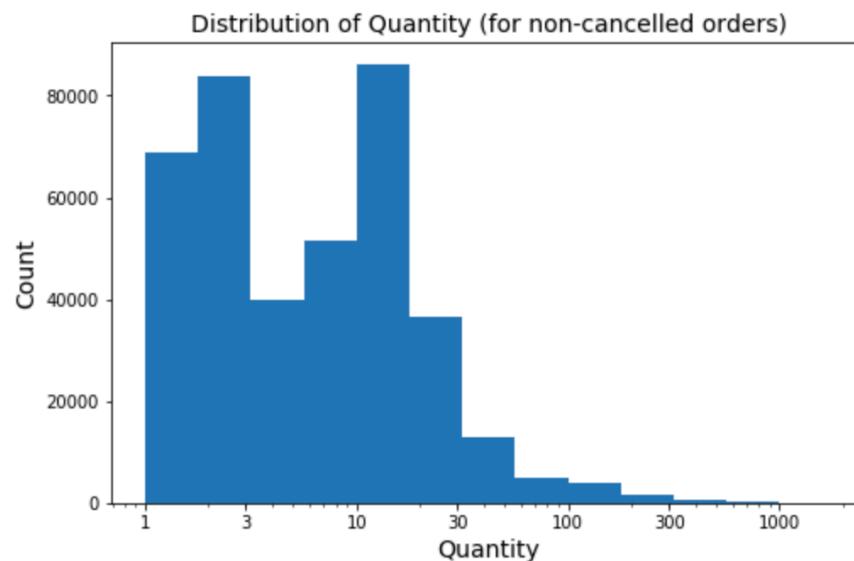
- UnitPrice :

- * 75% des prix unitaire < 3,75 Sterling
- * max UnitPrice = 649,5 Sterling



- Quantity :

- * 50% < 6 unités
- * 75% < 12 unités
- * max Quantity = 80995



Par commande et par client

- ***Montant PAR Commande:***

- * 50% < 301,6 Sterling
- * 75% < 463 Sterling
- * Montant max/commande = 168469 Sterling

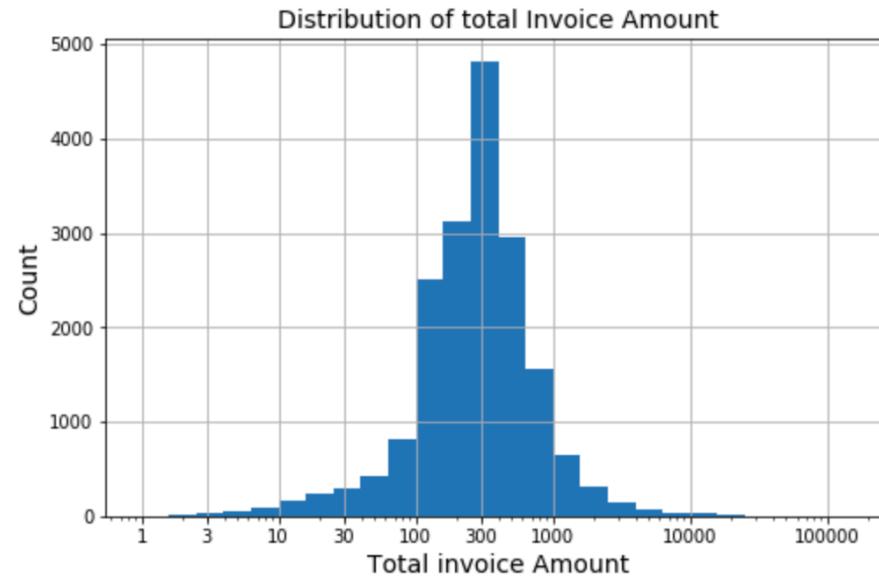
- ***Meilleurs clients :***

- * min = 3,75 Sterling
- * 50% < 663,61 Sterling
- * 75% < 1632,7 Sterling
- * max = 279138 Sterling

* **10 meilleurs clients** = 17,4% ventes

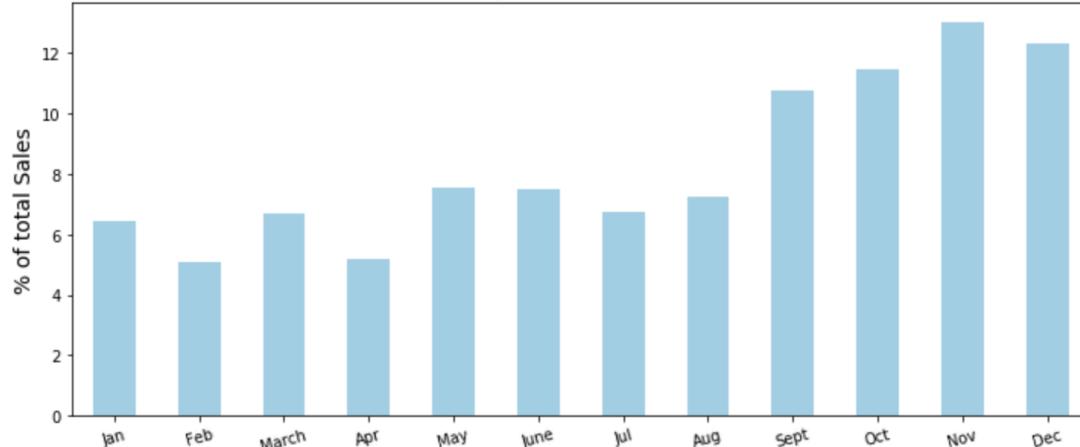
* **26% des clients** = 80% ventes

(~Pareto : 20-80)

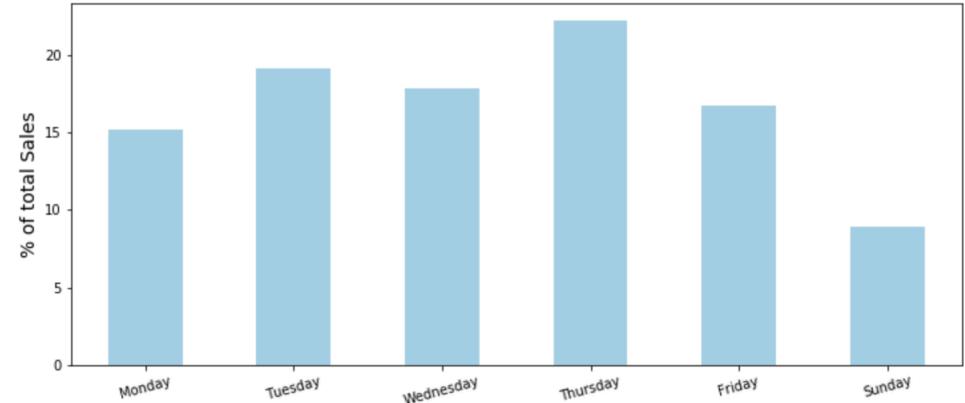


Information temporelle

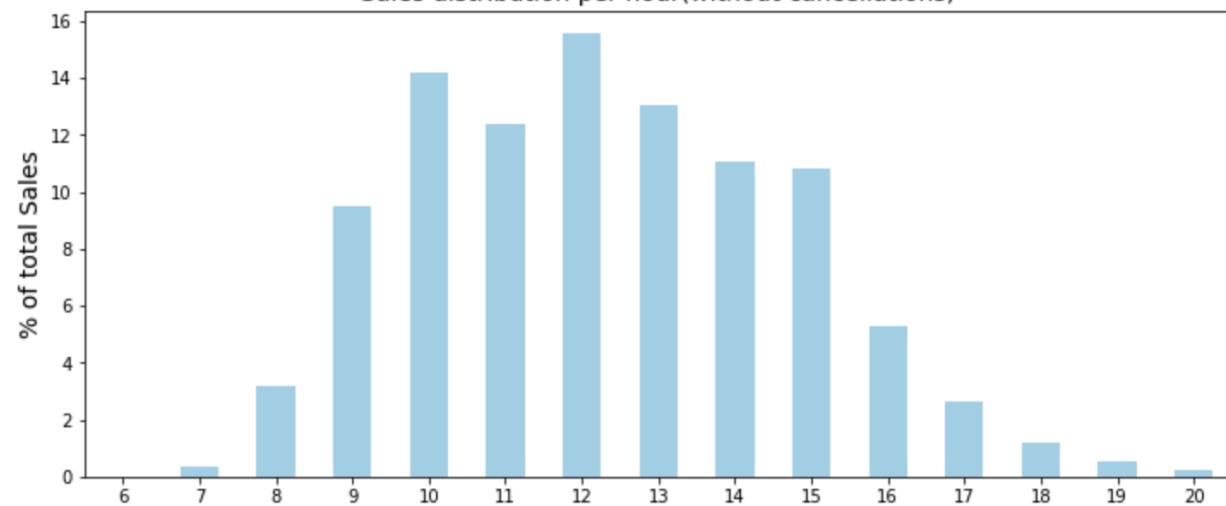
Sales distribution per month (without cancellations)



Sales distribution per day of the week (without cancellations)

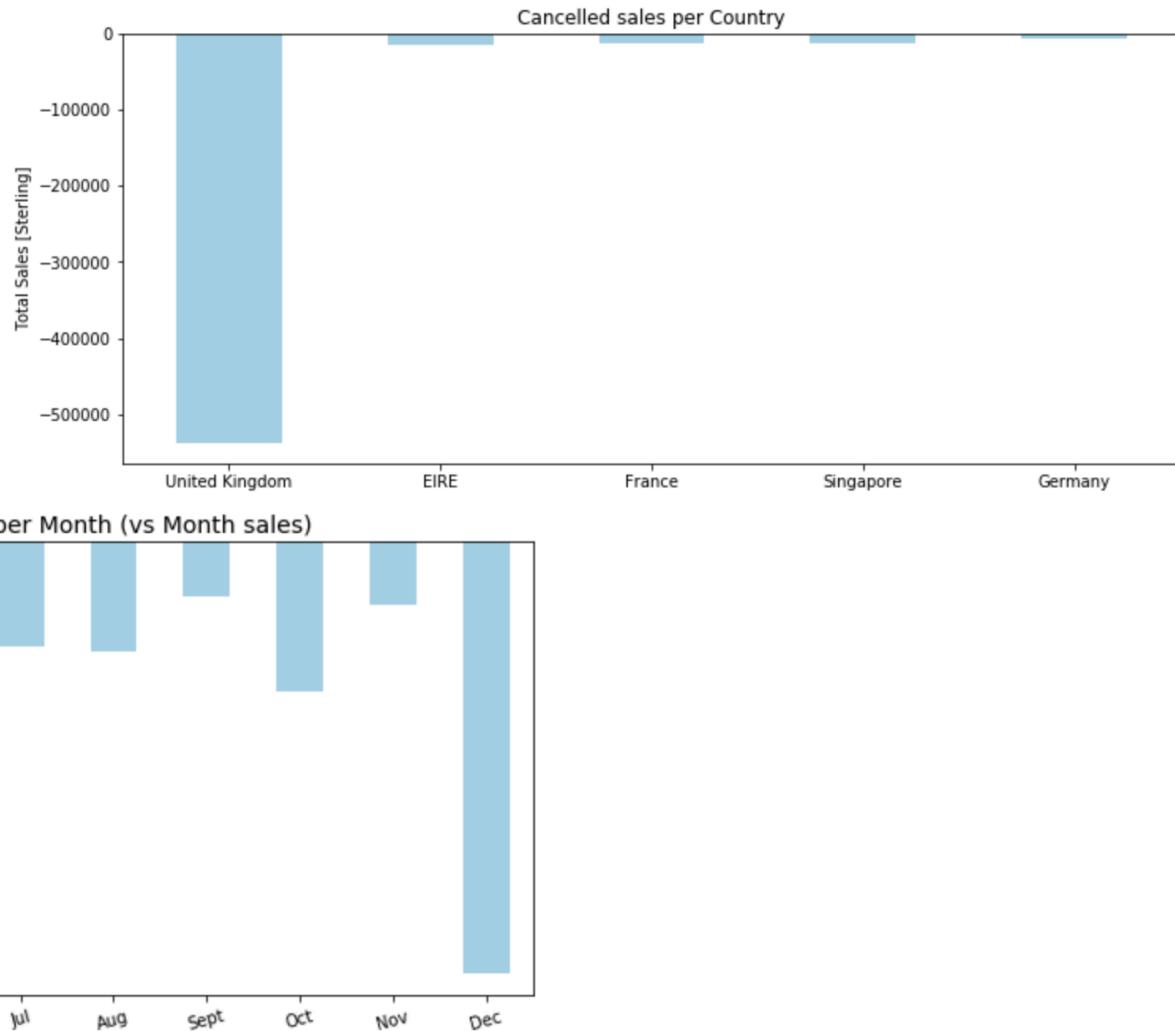


Sales distribution per hour (without cancellations)



Commandes annulées

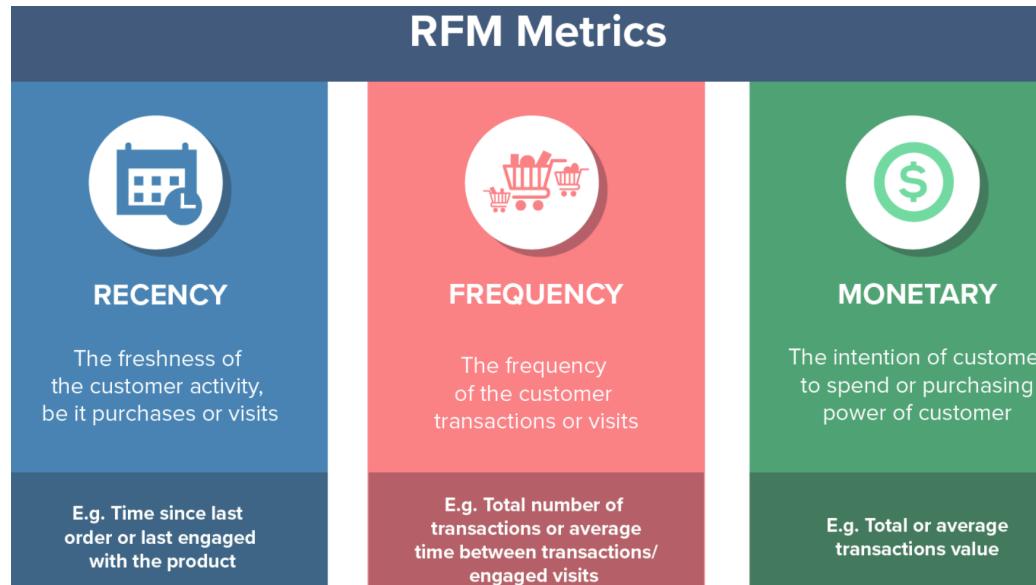
Total amount of cancelled sales = ~610 000 Sterling



Part IV:

Pistes modélisation

Feature Engineering : RFM



- *Utilisation des quantiles (25%,50%,75%) => un label de 1(best) à 4(worst)*
- En Marketing:
 - **Best Customers: 111**, Bought most recently and most often, and spend the most
 - **Loyal Customers: X1X**, Buy most frequently
 - **Big Spenders: XX1**, Spend the most
 - **Almost Lost: 311**, Haven't purchased for some time, but purchased frequently and spend the most
 - **Lost Customers: 411**, Haven't purchased for some time, but purchased frequently and spend the most
 - **Lost Cheap Customers: 444**, Last purchased long ago, purchased few, and spend little

Feature Engineering : Country /products

- Variable : **Country** (d'après l'exploration)

= [« United Kingdom », « Other »]

- Nombre de produits différents achetés/ clients

CustomerID	R	F	M	NuniqueProd
13052	212	18	348.15	18
14234	22	147	754.12	122
16918	49	128	1068.82	97
17677	1	303	16345.38	178
15214	1	110	1661.44	109

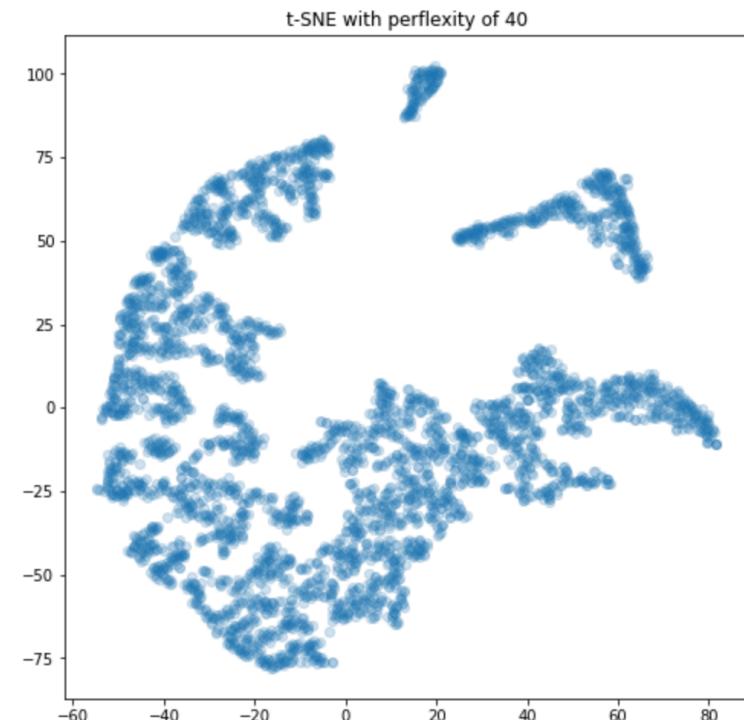
Réduction dimensionnelle

- Utilisation des différentes variables agrégées par client et du feature engineering

CustomerID	Other	UK	R	F	M	NuniqueProd
12346	0	1	325	1	77183.60	1
12347	1	0	2	182	4310.00	103
12348	1	0	75	27	1437.24	21

- **t-SNE** avec différentes valeurs de perplexité entre 5 et 50 (pour visualisation)

*Choix perplexité selon coefficient de silhouette
Le plus élevé = 40*

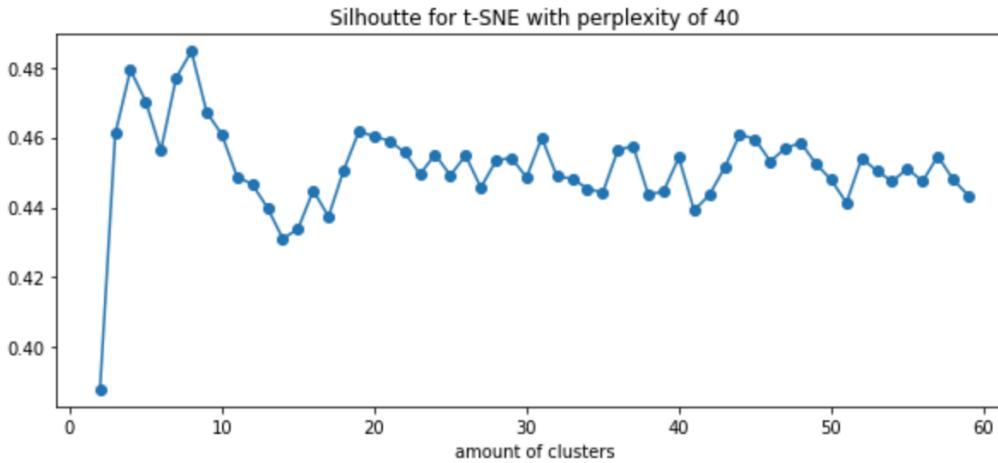


Clustering non-supervisé

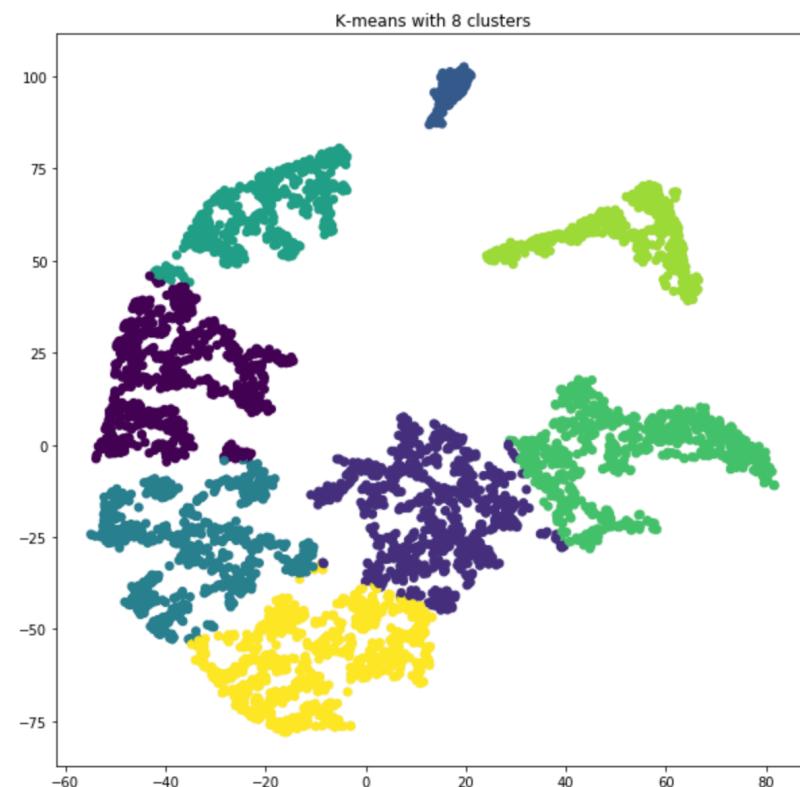
- Idée:

utiliser **K-means++** pour déterminer le *nombre de segments et leur label*.

Ensuite utiliser ces labels comme **variable cible** pour les **méthodes ensemblistes (supervisées)**

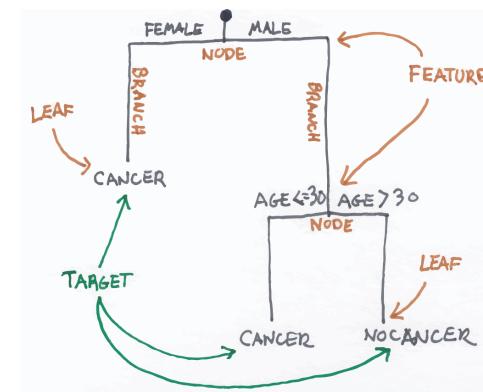


CustomerID	Other	UK	R	F	M	NuniqueProd	Kmeans
12346	0	1	325	1	77183.60	1	5
12347	1	0	2	182	4310.00	103	6
12348	1	0	75	27	1437.24	21	6



Random Forest(1)

= méthode **ensembliste** (//) supervisée utilisée pour la **classification** et la **régression**

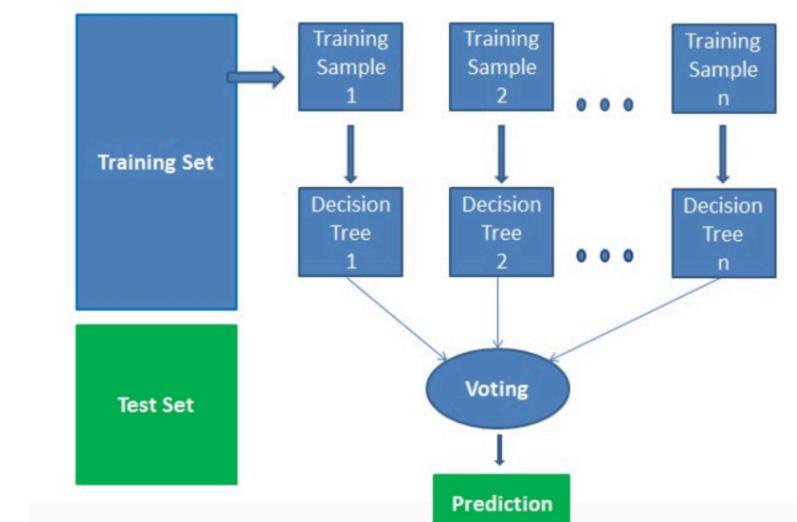


- création d'**arbres de décision** sur des échantillons aléatoirement choisis
 - * générés en utilisant un indicateur pour la sélection des features
Par ex. Index d'impureté de Gini
- chaque **arbre de décision** fait sa prédiction, meilleure solution choisie par vote

Algorithm

1. Sélection aléatoire de sous-ensembles de données et variables (features)
2. Construction d'un arbre de décision par sous-ensemble et obtention résultat de prédiction pour chaque arbre
3. Vote pour chaque prédiction
4. Sélection de la prédiction ayant le plus de votes

Sources: <https://www.datacamp.com/community/tutorials/random-forests-classifier-python>
Chris Albon : Flashcards



Random Forest(2)

NB:

Bagging: technique ensembliste, construisant plusieurs modèles indépendants et les combinants en utilisant une technique d'agrégation (moyenne pondérée, vote,...)



Avantages

- **Robuste** : # d'arbres de décision
- Réduit fortement le risque de **OVERRFITTIN**
vote/moyenne parmi les différentes prédictions
=> annule le Biais
- Ok pour problèmes de **Régression** et de **Classification**
- Indique également l'**importance relative des features**

Inconvénients

- plus lent, tous les arbres de décision doivent faire une prédiction
- plus difficile à interpréter qu'un arbre de décision

Source: <https://www.datacamp.com/community/tutorials/random-forests-classifier-python>

Random Forest(3)

Hyperparamètres

- ***n_estimators***:

d'arbres de décision construit avant de voter

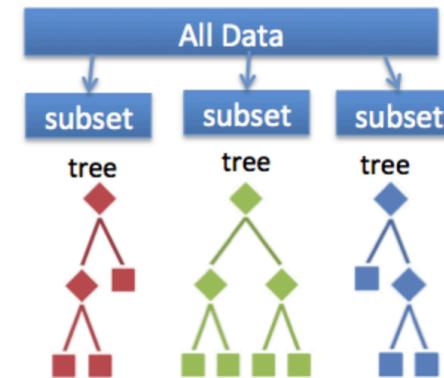
Augmente performance + stabilité (mais modèle plus lourd)

- ***max_features***:

max de features que le Random Forest considère pour faire splitter un noeud.

- ***min_sample_leaf***:

minimum de feuilles nécessaire pour splitter un noeud interne



Tuning des hyperparamètres:

à l'aide de *GridSearchCV* avec **Précision** comme score

= ratio entre # prédictions correctes et # total d'entrée

n_estimators = 100 / ***max_features*** = 5 / ***min_sample_leaf*** = 3

Gradient Boosting(1)

Boosting: technique ensembliste, où les modèles/apprenants ne sont pas indépendants mais utilisés **séquentiellement**.
Un modèle apprend des erreurs du précédent.

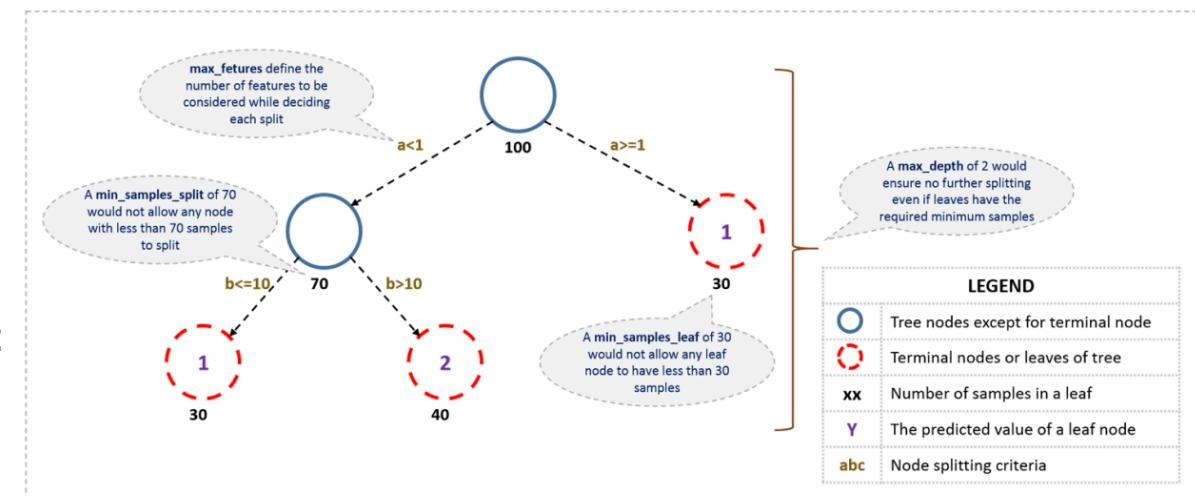
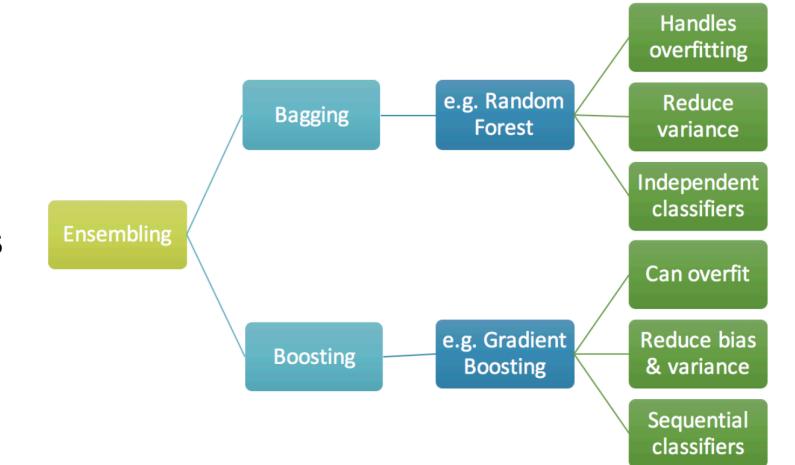
- Les apprenants faibles peuvent être choisis apd différents modèles (arbres de décision, régression, clustering,...)

- Hyperparamètres:

* **n_estimators** : # d'arbres

* Paramètres spécifique aux arbres:

- **max_depth** et **num_samples_split**
- **min_samples_leaf**
- **max_features**

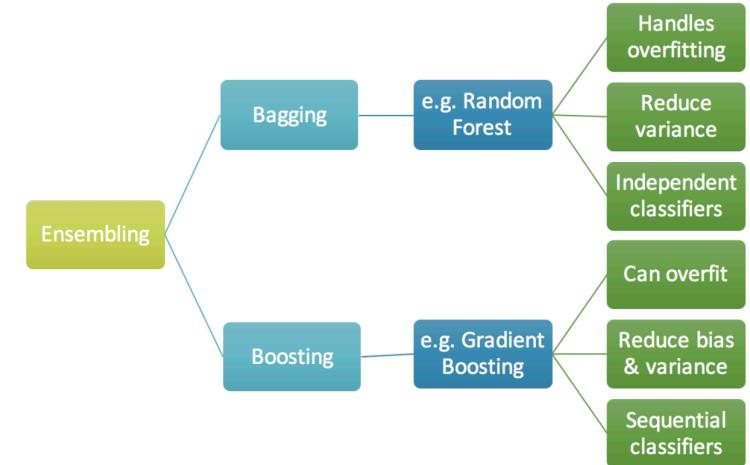


Sources: <https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d>
<https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/>

Gradient Boosting(2)

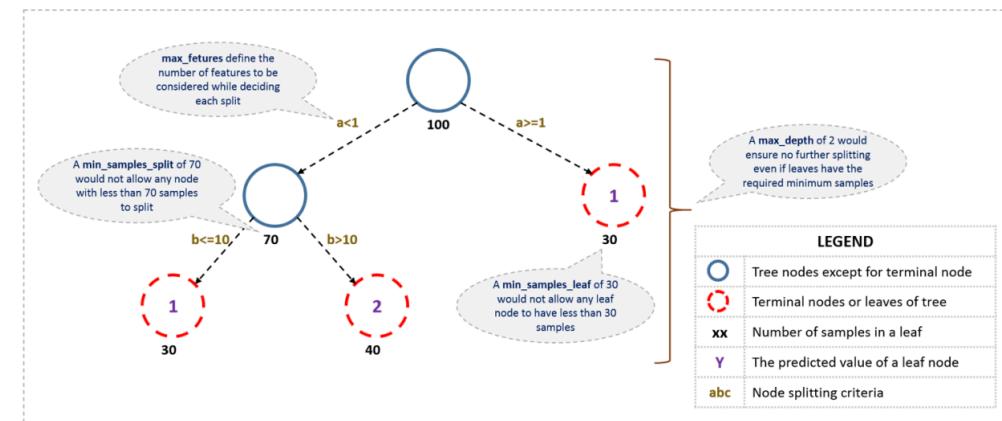
Tuning les hyperparamètres:

Utilisation de `GradientSearchCV()`



Differentes étapes :

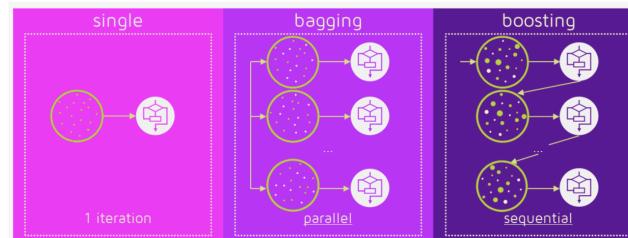
1. Tuner **n_estimators** => 60
2. Tuner **max_depth** et **min_samples_split** => 8 et 30
3. Tuner **min_samples_leaf** => 30
4. Tuner **max_features** => 6



Sources: <https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d>
<https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/>

Part V:

Modèle final et performances



Performances des prédictions(1)

Utilisation de différents critères d'évaluation avec **validation croisée**

		Classe réelle	
		-	+
Classe prédictée	-	True Negatives (vrais négatifs)	False Negatives (faux négatifs)
	+	False Positives (faux positifs)	True Positives (vrais positifs)

- Faux positifs (fausses alarmes) = **erreurs de type I.**
- Faux négatifs (non-détection) = **erreurs de type II.**

$$\text{Précision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{Rappel} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

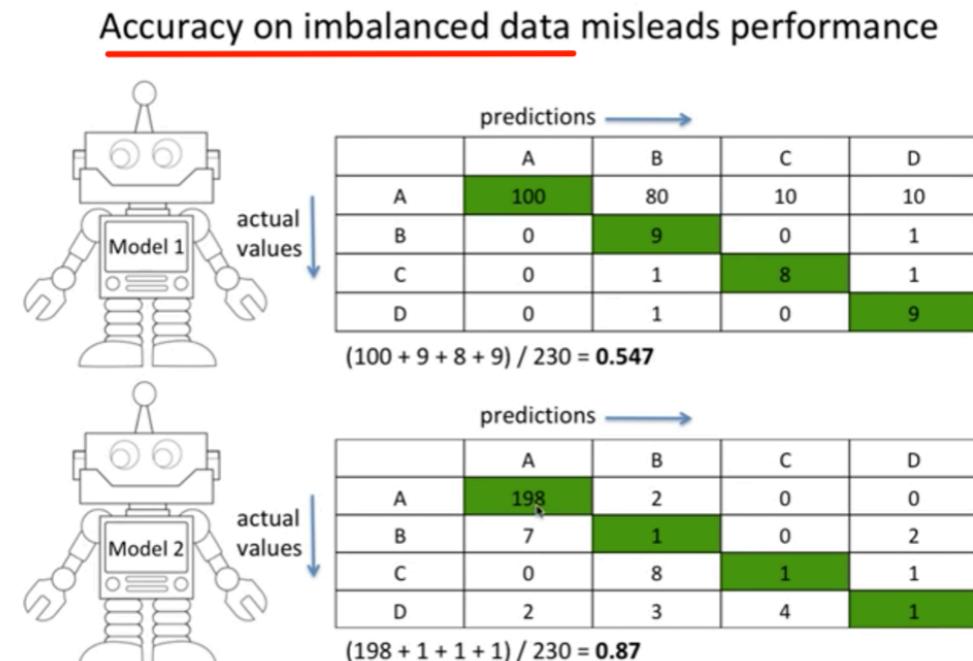
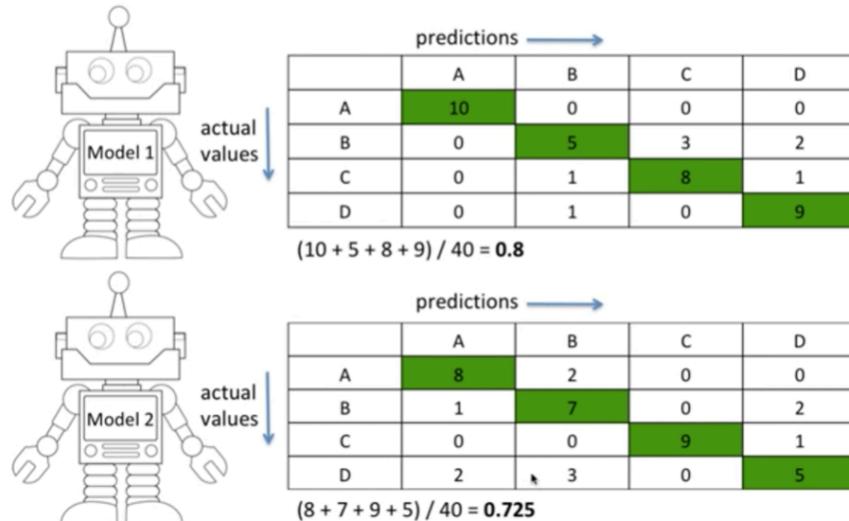
F-mesure (F-score) = moyenne harmonique du rappel et de la précision

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Rappel}}{\text{Precision} + \text{Rappel}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

Source: Openclassroom

Performances des prédictions(2)

Multiclass: Accuracy

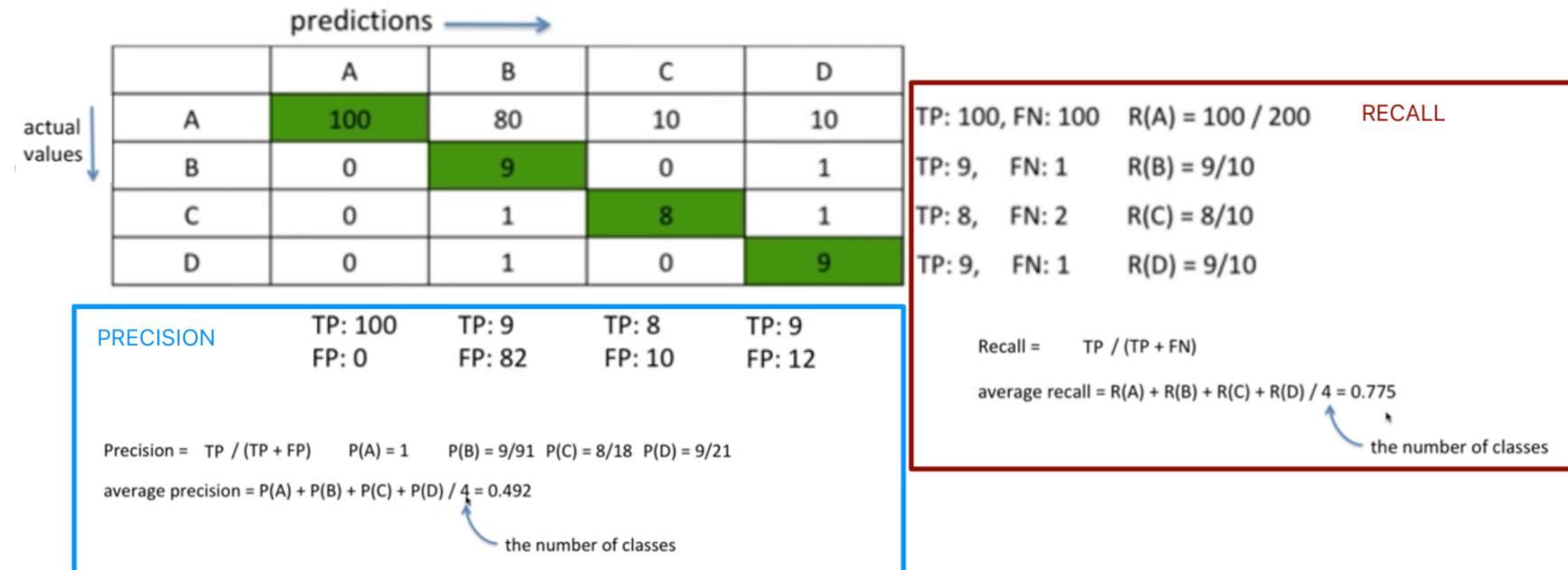


Source: <https://www.youtube.com/watch?v=HBi-P5j0Kec>

Performances des prédictions(3)

Multiclass: Precision and Recall

On vs all



Source: <https://www.youtube.com/watch?v=HBi-P5j0Kec>

Performances des prédictions(1)

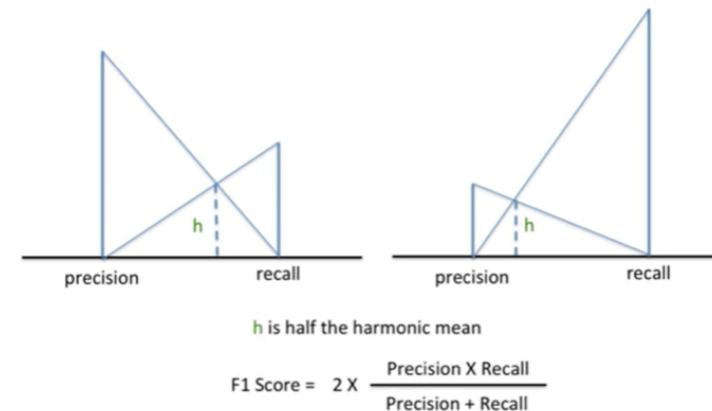
F-score for unbalanced classes

= moyenne harmonique

Given a class, will the classifier detect it ? (recall)

	A	B	C	D
A	100	80	10	10
B	0	9	0	1
C	0	1	8	1
D	0	1	0	9

Given a class prediction from the classifier,
how likely is it to be correct? (precision)



Source: <https://www.youtube.com/watch?v=HBi-P5j0Kec>

Performances des prédictions(2)

Random Forest (5-folds)

Mean Accuracy: 0.9314668456704785
Mean Precision: 0.9340263323162235
Mean Recall: 0.9697564736053501
Mean F1_score: 0.942913158476627

Predicted Segments	0	1	2	3	4	5	6	7
Actual segments	125	1	0	1	0	0	0	0
0	1	124	0	1	0	4	0	1
1	0	0	28	0	0	0	0	0
2	0	4	0	105	0	0	0	3
3	0	0	0	0	96	0	0	0
4	0	2	0	0	0	133	0	0
5	0	0	0	0	0	0	99	0
6	0	0	0	0	0	0	0	134
7	0	1	0	4	0	0	0	0

Performances des prédictions(3)

Gradient Boosting (5-folds)

Mean Accuracy: 0.9817573273026097
Mean Precision: 0.9825055191542498
Mean Recall: 0.9861615863709675
Mean F1_score: 0.9841605809203889

Predicted Segments	0	1	2	3	4	5	6	7
Actual segments	0	125	1	0	1	0	0	0
0	1	124	0	1	0	4	0	1
1	0	0	28	0	0	0	0	0
2	0	1	0	108	0	0	0	3
3	0	0	0	0	96	0	0	0
4	0	1	0	0	0	134	0	0
5	0	0	0	0	0	0	99	0
6	0	0	0	0	0	0	0	135
7	0	1	0	3	0	0	0	0

Python file

	df_test5								
1	CustomerID	Country	InvoiceDate	InvoiceNo	StockCode	Description	Quantity	UnitPrice	
2	16483	United Kingdom	2011-05-31 13:08:00	555092	22619	SET OF 6 SOLDIER SKITTLES	3	3.75	
3	16483	United Kingdom	2011-05-31 13:08:00	555092	22382	LUNCH BAG SPACEBOY DESIGN	2	1.65	
4	16483	United Kingdom	2011-05-31 13:08:00	555092	22606	WOODEN SKITTLES GARDEN SET	1	15.95	
5	16483	United Kingdom	2011-05-31 13:08:00	555092	21932	SCANDINAVIAN PAISLEY PICNIC BAG	1	1.65	
6	16483	United Kingdom	2011-05-31 13:08:00	555092	22384	LUNCH BAG PINK POLKADOT	3	1.65	

➔ **PEP8** `python3 cust_segmentation.py`

Customer with ID 16686, from United Kingdom is predicted to be in segment nb 3

Your code has been rated at 9.49/10 (previous run: 9.49/10, +0.00)

Part VI:

Conclusions

Conclusions



- Un des points clés de ce projet = **Feature Engineering**
(RFM, # produits distincts/client)
- Nous avons ensuite utilisé une **méthode non-supervisée de clustering** afin de définir le nombre idéal de segments.
(K-means++ après une t-SNE)
- Le résultat du K-means a été utilisé comme variable cible
afin d'appliquer des **méthodes ensemblistes**.
(Random Forest et Gradient Boosting)
- Différents critères d'évaluation on été utilisés.
(precision, recall, F-score, Confusion Matrix)

Pistes

Le tuning des différents *hyperparamètres* pourrait être poussé un peu plus loin.

La *stabilité des segments* pourrait être analysée en fonction des critères temporels.

D'un point de vue *marketing*, on pourrait analyser plus en détail les *différents segments* trouvés par K-means++.



