UNIVERSITÉ DE LYON

Université Claude Bernard Lyon 1

Rapport synthèse du projet Data Mining

Découverte de Connaissances dans les Données d'e-commerce d'un détaillant du prêt à porter britannique

Source de données : *https://www.kaggle.com/carrie1/ecommerce-data*

Vidéo explicative : *https://www.youtube.com/watch?v=SM7a2jLKb2U*

Réalisé par :

- Nassim EL GHAZAZ
- Fallou DIENG
- Oussama BENALI
- Soufiane FADEL
- Victor RAZY

Professeur :

Pr. Marc PLANTEVIT

Année Universitaire 2018 – 2019

Table des figures

Table des matières

Introduction

La découverte de connaissances dans les données peut être utilisée dans plusieurs domaines (Finance, assurance, météorologie … etc). Nous avons choisi, dans le cadre de ce projet, d'explorer un jeu de données de transactions d'achats pour un détaillant britannique du prêt et porter.

Nous avons entre les mains une multitude de métadonnées qui vont nous servir à tirer des informations dans les objectifs de veille concurrentielle, c'est-à-dire anticiper la demande, corriger les défaillances (les retours fréquents et autres), et améliorer la santé de l'entreprise au fil du temps.

Ces mêmes métadonnées nous permettrons d'évaluer la performance du store, de prédire la demande et prévenir contre des futures menaces en regardant. Nous réaliserons d'abord une première approche présentant le jeu de données choisi, le prétraitement et quelques analyses statistiques. Les pistes explorées en matière d'évaluation de l'état actuel de l'entreprise et les algorithmes de data mining seront ensuite présentées, avec les différents résultats.

1. Description du Data Source

Notre projet traite la fouille de données dans un Data Set tiré de Kaggle.com (lien dans la page de couverture), nos données sont les différents tickets de transactions effectuées.

Ci-dessous la composition de notre Data Source (les champs dont on dispose) :

Champ	Libellé
Invoice No	Le numéro de la facture
StockCode	Code du produit
Description	La description du produit
Quantity	La quantité achetée
InvoiceDate	La date de la facture
UnitPrice	Le prix unitaire
CustomerID	Le Numéro du client
Country	Le pays

Figure 1 : Les métadonnées du Data Source étudié

2. Objectifs de fouille de données du Data Source

Comme il s'agit de données d'e-commerce, donc les principales activités de ce magasin en question sont : la vente en ligne, le Click & Collect et la vente sur place du prêt à porter.

Et donc les points à améliorer pour augmenter la valeur du chiffre d'affaire globale de cette structure étudiée sont :

- Extraire les règles des associations pour proposer des produits en fonction du panier.
- Savoir les sources des pertes du chiffre d'affaire pour ensuite les corriger, exemples : retour client remboursé, retour pour des raisons logistiques : produit qui n'a pas été physiquement transporter à sa destination finale, détérioration d'article. Notre objectif sera de savoir les pays qui minimisent la marge du gain du store et ensuite remédier à ce fléau.
- Etudier les relations entre les différentes métadonnées.
- Détecter les relations client-produit.

- Le Pattern Mining (chercher des structures qui se répètent dans nos données, par exemple voir s'il y a une saisonnalité, des effets calendrier propre à chaque pays

qui vont servir pour des campagnes de publicité mieux personnalisées), et la Trend Discovery (détection de la tendance du CA pour les tops 6 pays qui rapportent le plus d'argent, si cette dernière est baissière pour certains pays ou pour la globalité du store, cela va être un signal d'alarme pour changer de politique)

- Elasticité croisée (EC) : C'est-à-dire comprendre la variation des ventes d'un produit par rapport aux effets d'un changement de prix d'un autre produit. Ceci nous permettra d'identifier les produits complémentaires, les produits de substitution et les produits indépendants.
- Bonus : Faire du Forecasting (étude prévisionnelle, tant que nous avons une date pour chaque transaction, on peut utiliser ça pour prévoir la santé de ce vendeur dans le futur).

1 Approches suivies pour atteindre les objectifs

Nous avons utilisé Python dans ce projet pour résoudre notre problématique de remonter les connaissances de la (Pandas, Numpy, Sklearn, Stats …).

1.1 Pattern Mining et Trend Discovery dans la série temporelle du Chiffre d'Affaire journalier du store

1.1.1 Préparation et Nettoyage des données

Pour l'étude prédictive nous avons regroupé les donner pour tirer la recette journalière du store. (Nous avons remarqué qu'il y a 0£ de recette pour les samedis ainsi que pour les derniers jours du mois du décembre et donc les jours fériés).

Pour ne pas affecter notre étude nous avons procéder par le nettoyage, ceci en remplaçant les NaN par la moyenne.

1.1.2 Analyse statistique des ventes journalières

Tout d'abord, nous avons étudié la distribution des ventes :

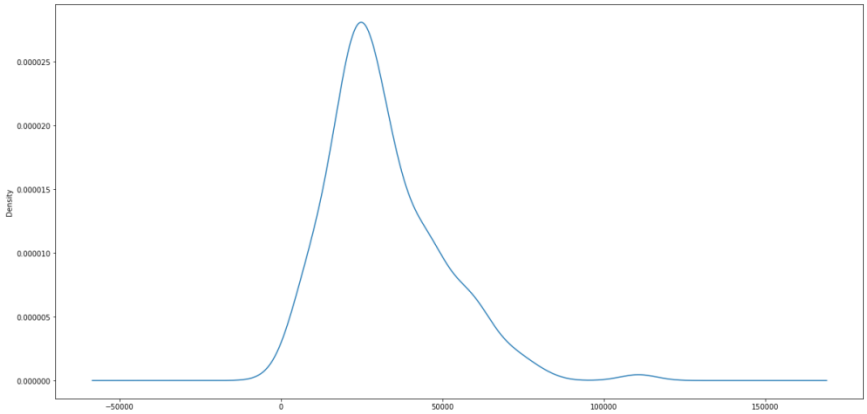

Figure 2 : La distribution des ventes journalières

On remarque que les ventes suivent une loi normale.

On suite nous avons analysé les LAGS (corrélations des observations successives)

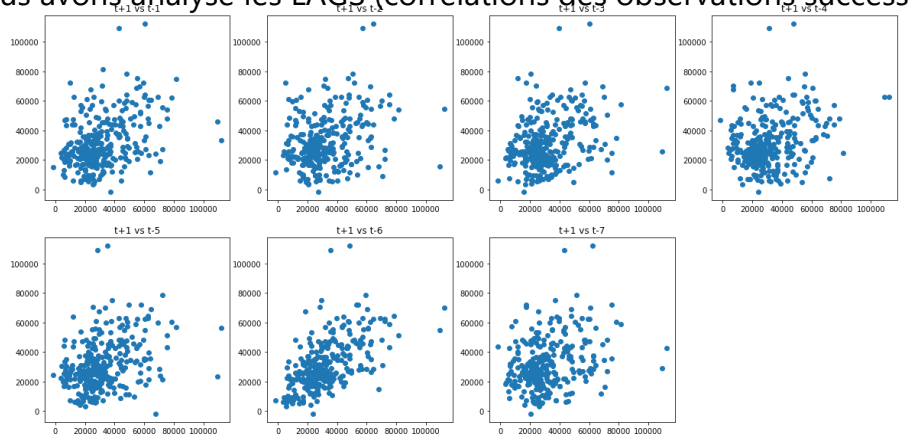

Figure 3 : Analyse des LAGS de ventes

On constate qu'il y a une forte dépendance entre les observations consécutives, comme la montre la figure ci-dessus.

Et après l'analyse des lags, du graphique d'autocorrélation ACF et de la série temporelle elle-même on constate qu'il y a une forte saisonnalité qui sera prédite par un modèle exponentiel, nous avons donc choisi d'appliquer Holt-Winters.

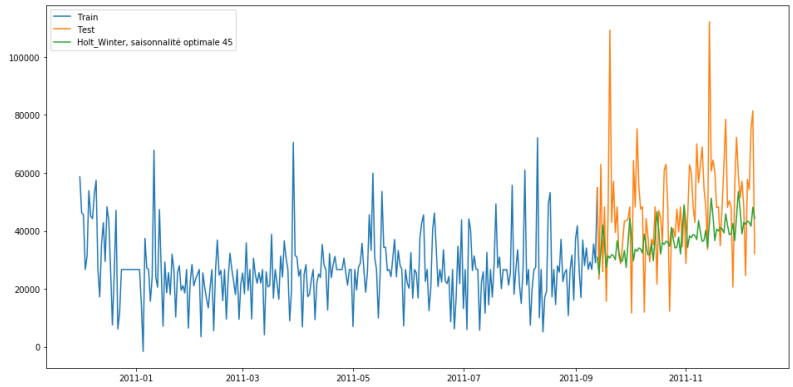

Figure 4 : Approche pour trouver le bon modèle prédictive

Ce qui donne le résultat suivant :

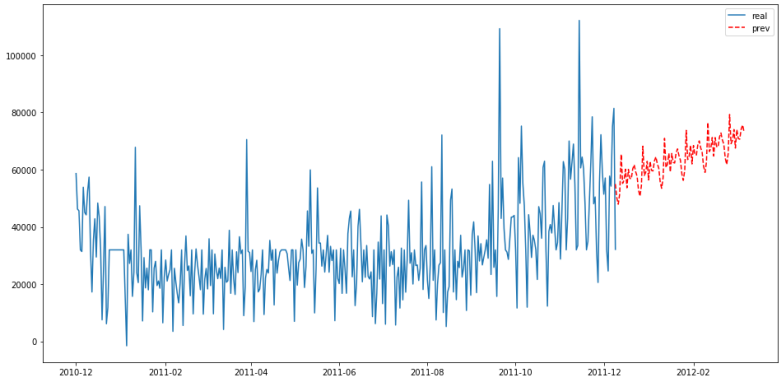

Figure 5 : Génération des prédictions des ventes

Ceci était une étude macro pour savoir la santé du magasin dans sa totalité, nous nous sommes intéressé aussi par le comportement des clients par pays, pour cela nous avons traité les tops 6 pays afin de tirer les aspects de saisonnalité et de tendance de la courbe d'évolution du chiffre d'affaire pour chacun de ces pays.

Nous avons tiré les 6 tops pays en termes de contribution en CA.

Exemple Royaume-Uni :

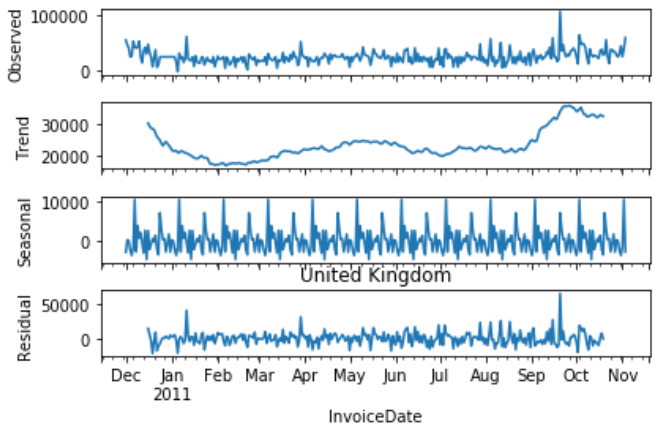

Figure 6 : Décomposition de la série temporelle du chiffre d'affaire journalier (client royaume-uni uniquement)

On voit clairement que la tendance et stationnaire dans l'ensemble avec une allure de croissance durant les 3 derniers mois, mais pour d'autres pays ce n'était pas le cas donc il faut revoir la politique des ventes pour mieux viser les clients de ces pays.

1.2 Les préférences d'achat de chaque pays

1.2.1 1er Approche - FP-Growth Algorithm:

Vu qu'Apriori peut subir deux coûts non négligeables lors de son exécution :

- Parfois, Il faudra peut-être encore générer un grand nombre d'ensembles de candidats. Par exemple, s'il y a 104 éléments 1 éléments fréquents, l'algorithme Apriori devra générer plus de 107 candidats 2-itemsets.
- Il est, peut-être, nécessaire d'analyser de manière répétée l'ensemble de la base de données et de vérifier un grand nombre de candidats. Il est coûteux de passer chaque transaction dans la base de données pour déterminer le support des itemsets.

Une méthode intéressante dans cette tentative est appelée croissance fréquente (Frequent pattern Growth), ou tout simplement FP- Growth, qui adopte une division et une conquête stratégie comme suit :

1. Premièrement, il compresse la base de données représentant les éléments fréquents en un arbre de modèle fréquent, ou FP-tree, qui conserve les informations d'association de jeu d'éléments.
2. Il divise ensuite la base de données compressée en un ensemble de bases de données conditionnelles (un type spécial de base de données projetée), chacune associée à un élément fréquent ou « fragment de motif », et mines chaque base de données séparément.

3. Pour chaque « fragment de modèle », seuls ses ensembles de données associés ont besoin d'être examiné.

Par conséquent, cette approche peut réduire considérablement la taille de l'espace de recherche, ainsi que la « croissance » des profils examinés. Pour implémenter ceci on utilise la bibliothèque « **pyfpgrowth** » de python (Jupyter) et on visualise les résultats.

1.2.2 2émr Approche – HFM Algorithm:

Pour avoir un bon résultat des préférences de chaque pays, Nous aurons besoin d'intégrer la quantité vendue et le profit de chaque « itemset » dans l'algorithme de recherche. Pour cela, Nous avons cherché une exploration plus rapide des jeux d'éléments à haute utilité avec Élagage estimé de la co-occurrence de l'utilité. L'algorithme balaye d'abord la base de données pour calculer le TWU (Total Weight Utility) de chaque article. Ensuite, l'algorithme identifie l'ensemble de tous les articles ayant un TWU pas moins que minutil (les autres éléments sont ignorés car ils ne peuvent pas faire partie d'un itemsets à haute utilité). Les valeurs des éléments TWU sont alors utilisées pour établir un ordre total sur les articles, qui est l'ordre croissant de TWU valeurs (trier les itemsets dans ordre croissant de la TWU). Une seconde analyse de la base de données est ensuite effectuée. Dans cette base de données, les éléments des transactions sont réorganisés en fonction de l'ordre. Une structure EUCS (Estimated Utility Co-Occurrence Structure) est construite. Après la construction de l'EUCS, l'exploration de recherche en profondeur des itemsets commence en appelant la procédure récursive Recherche avec le vide itemset ∅, l'ensemble d'éléments uniques I*, minutil et la structure EUCS. Voir l'algorithme en annexe.

Exemple de Visualisation:

Sur ce nuage de points, des points de données sont placés sur un axe horizontal et un axe vertical afin d'illustrer l'importance de l'influence de la confidence sur le support. Dans les motifs fréquents générés, chaque ligne est représentée par la confidence et le support. Ci-dessous un nuage de points pour les ventes en France.

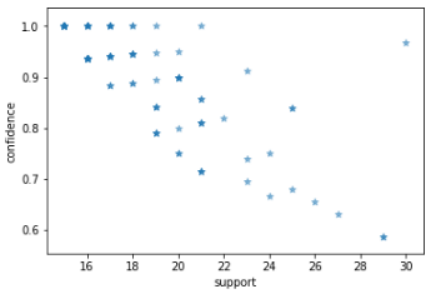

Figure 7 : scatter plot for support and confidence

Pour représenter les règles d'association sous forme de diagramme, On peut utiliser la bibliothèque NetworkX python.

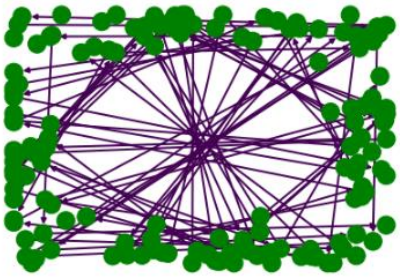

Fig ion.

On peut même faire un heat map pour visualiser les produits achetés ensemble. Les couleurs violets indiquent les produits les plus fréquent et achetés ensemble, tandis que les autres cellules ne sont jamais achetées ensemble.

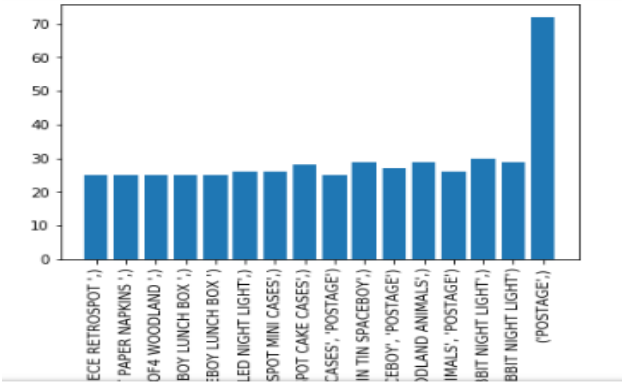

Figure 9: Plot des motifs fréquents en fonction de support

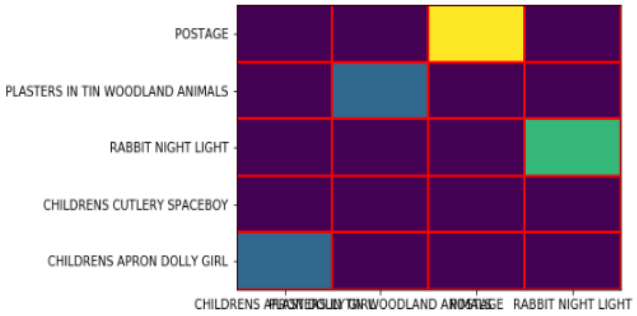

Figure 10 : HeatMap pour les motifs fréquents

1.3 Recommendation de Produit

Le but de cette partie est en fonction du panier d'un utilisateur lui proposer des items qui l'interesserait en fonction de l'historique des achats. Pour cela on a utilisé un algo Apriori.

Une étape préliminaire est de formater notre base de données sous forme de ticket. Les produits « ? » et « DotCom postage » ont étés supprimés car inutile, ainsi que les quantités <0 car ce sont des retours. Puis on fait un groupBy sur le numero de transaction pour regrouper en ticket.

1.3.1 Problématique:

Apriori prend du temps à s'exécuter s'il faut extraire toute les règles d'associations.

1.3.2 Solution:

Pour éviter cela on utilise des paramètres pour accélérer le temps d'exécution.
Ici on a choisi des paramètres tels que l'algorithme nous extrait 7800 règles d'associations pour un temps d'exécution d'environ 30 min. Le problème est qu'un utilisateur peut avoir un panier qui ne nous permettrait pas de lui proposer de produit en fonction de ces règles d'associations.

Une solution est de créer un échantillon de notre base, en récupérant seulement les tickets qui contiennent le panier. Puis de lancer l'algorithme Apriori sur cet échantillon. Un calcul du support minimum est obligatoire car même si on récupère seulement un échantillon le calcul peut prendre du temps, pour cela on calcul un support minimum en fonction de la taille de notre échantillon.

Il faut que le calcul se fasse très rapidement car il se fait au moment où l'utilisateur remplit son panier, donc on ne peut pas lancer un apriori de plusieurs secondes à chaque fois que l'ulisateur change son panier.

Pour une meilleure expérience utilisateur on a rajouté 2 fonctions **nameToCode** et **codeToName** qui permette de traduire les codes.

Finalement, la fonction **faitTout** Permet de récupérer les produits qui ont le plus de chance d'être acheté avec notre panier.

Et donc finalement de proposer les produits les plus intéressants
Exemple de résultat :

```
avec: DOORMAT MERRY CHRISTMAS RED
Il va acheter:

PAPER CHAIN KIT 50'S CHRISTMAS
avec une confiance de:
0.40939597315436244

CHRISTMAS CRAFT LITTLE FRIENDS
avec une confiance de:
0.30201342281879195

RECIPE BOX PANTRY YELLOW DESIGN
avec une confiance de:
0.33557046979865773

GREEN REGENCY TEACUP AND SAUCER
avec une confiance de:
0.3087248322147651

ROSES REGENCY TEACUP AND SAUCER
avec une confiance de:
0.30201342281879195

JAM MAKING SET PRINTED
avec une confiance de:
0.31543624161073824
```

1.4 L'élasticité prix de la demande

Dans cette partie, nous étudierons l'élasticité de la quantité vendue par rapport au prix.
En 1890, Alfred Marshall, le grand économiste néo-classique, élabora une mesure spéciale pour la réponse d'une variable, telle que la quantité demandée, afin de modifier une autre variable, telle que le prix. C'est ce qu'on appelle l'élasticité, qui est une mesure de la sensibilité de la demande au marché.

L'élasticité-prix nous permet donc de connaître l'évolution de la quantité vendue à la suite d'une variation des prix.

Nous allons d'abord segmenter nos données en clusters, avec comme coordonnées le prix et la quantité vendue, avant de calculer l'élasticité.

1.4.1 Le clustering

Avant d'étudier l'élasticité prix de la demande nous avons jugé nécessaire d'organiser les données brutes en silos homogène selon le prix et la quantité vendue. Mais la question fondamentale est la suite: comment choisir le nombre de classes. Pour répondre à cette question nous avons utilisé le dendrogramme pour trouver le nombre optimal de classes.

Un dendrogramme est un diagramme qui montre la relation hiérarchique entre les objets.

Le dendrogramme ci-dessous montre le regroupement de nos données, avec les coordonnées prix et la quantité vendue.

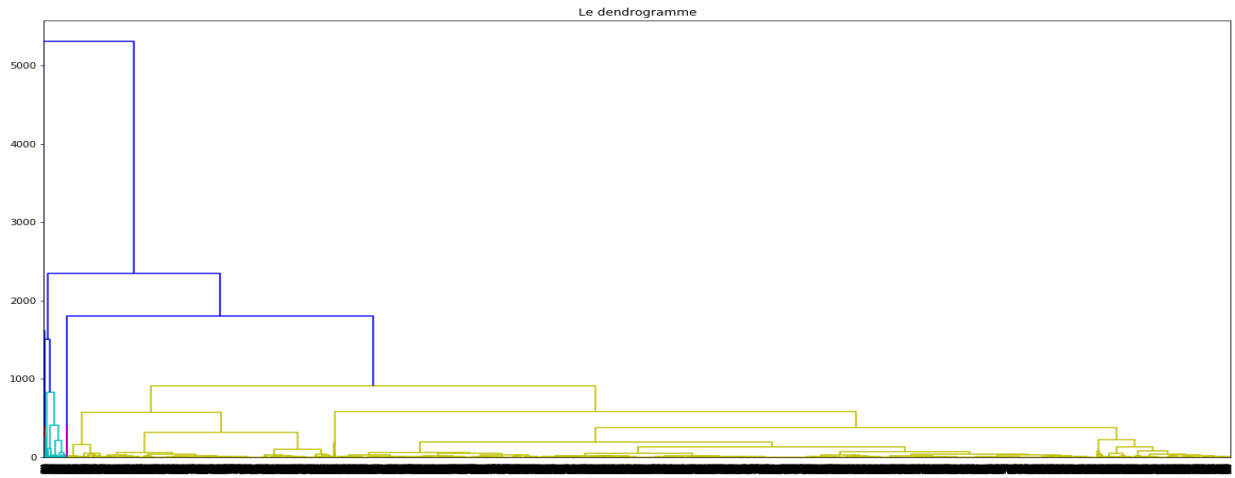

Figure 11: le dendrogramme

Notre dendrogramme nous donne trois principales couleurs, donc le nombre optim
al de classes est de 3.

Nous allons donc faire une classification ascendante hiérarchique pour 3 classes en utilisant la distance euclidienne comme mesure de similarité interclasse.

Visualisation des clusters:

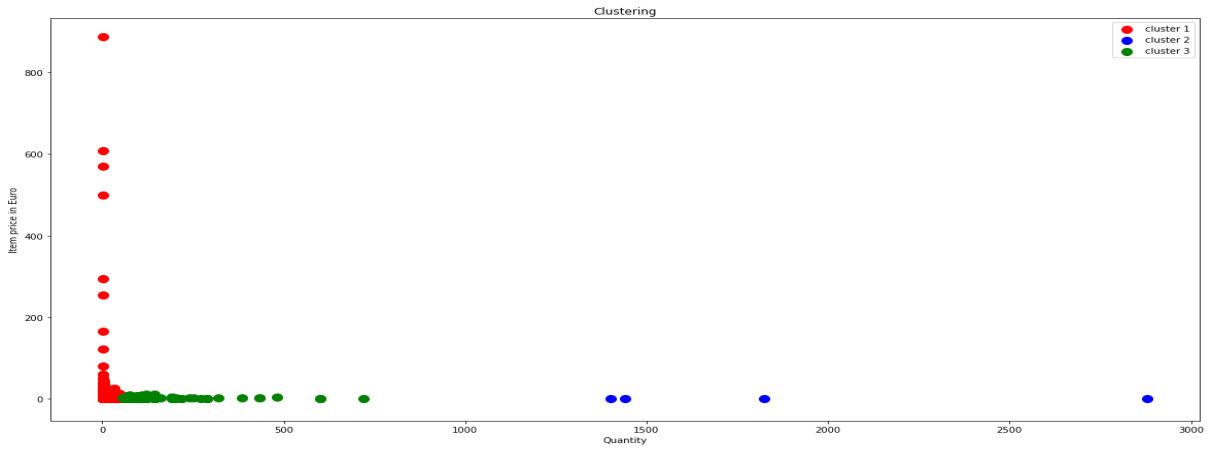

Figure 12: Visualisation des clusters

1.4.2 L'élasticité prix directe

L'élasticité prix directe nous permet de connaître l'évolution de la quantité vendue à la suite d'une variation des prix.

La formule de calcul est la suivante : Variation de la consommation / variation des pr ix. Pour calculer chaque variation, on utilise la formule du taux de variation.

$$\text{Price elasticity of demand} = \frac{\text{Proportionate change in quantity demanded}}{\text{Proportionate change in price}} = \frac{\frac{\Delta Q}{Q} \times 100\%}{\frac{\Delta P}{P} \times 100\%} = \frac{\frac{\Delta Q}{Q}}{\frac{\Delta P}{P}}$$

Figure 13: La formule de calcul de l'élasticité

Nous avons récupéré les « StockCode » des produits qui ont connu un changement de prix et nous avons calculé leurs élasticités.

L'élasticité sur produit « BLUE GLASS GEMS IN BAG » :

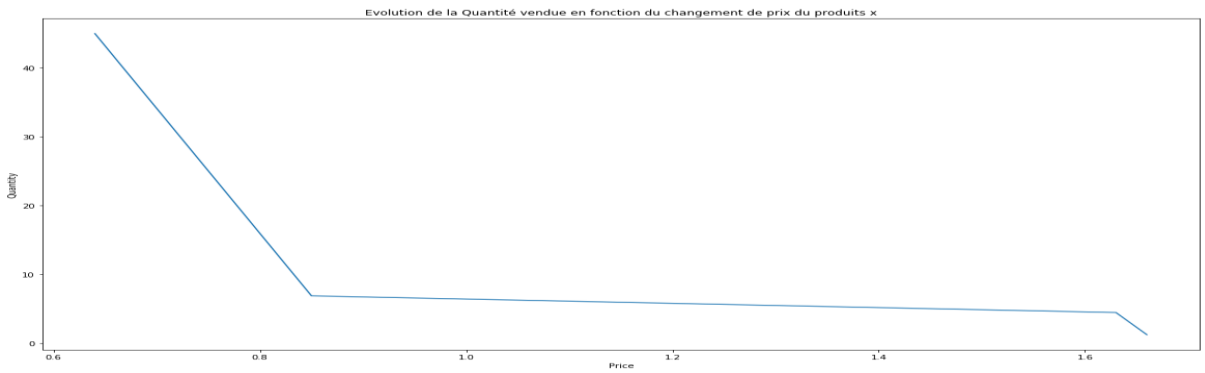

Figure 14: La variation de la quantité vendue en fonction du changement de prix du produit BLUE GLASS GEMS IN BAG

Une augmentation du prix entraine une baisse de la quantité demandée, donc l'élasticité est négative ici.

1.4.3 L'élasticité croisée

L'élasticité croisée est la variation des ventes d'un produit par rapport aux effets d'un changement de prix d'un autre produit. Ceci nous permettra d'identifier les produits complémentaires, les produits de substitution et les produits indépendants.

- Quand l'élasticité-prix croisée est nulle, cela signifie que l'évolution du prix d'un bien ou d'un service n'a aucune influence sur la consommation d'un autre bien ou service. On dit alors que les biens ou services en question sont indépendants.
- Quand l'élasticité-prix croisée est positive, cela signifie que l'augmentation du prix d'un bien ou d'un service entraîne l'augmentation de la consommation d'un autre bien ou service. On dit alors que les biens ou services en question sont substituables : quand le prix d'un bien augmente, cela nous pousse à en consommer un autre, que l'on considère alors comme équivalent.
- Quand l'élasticité-prix croisée est négative, cela signifie que l'évolution du prix d'un bien ou d'un service entraîne une diminution de la consommation d'un autre bien ou service. On dit alors que les biens ou services en question sont complémentaires.

Nous avons calculé l'élasticité croisée au sein de chaque cluster par rapport à un produit du cluster.

- L'élasticité croisée du produit « PAPER CHAIN KIT 50'S CHRISTMAS » au sien de son cluster.

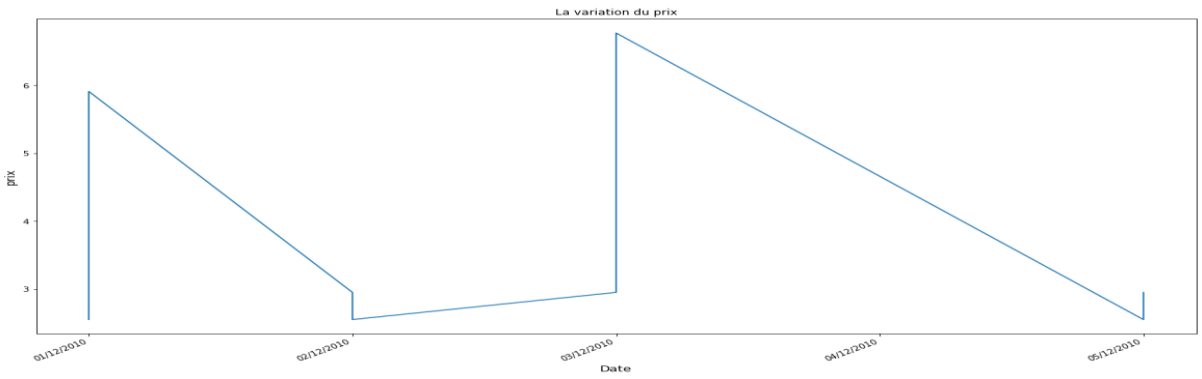

Figure 15: La variation du prix du "PAPER CHAIN KIT 50'S CHRISTMAS"

- La variation de la quantité vendue des autres produits du cluster en fonction du changement de prix du « PAPER CHAIN KIT 50'S CHRISTMAS »

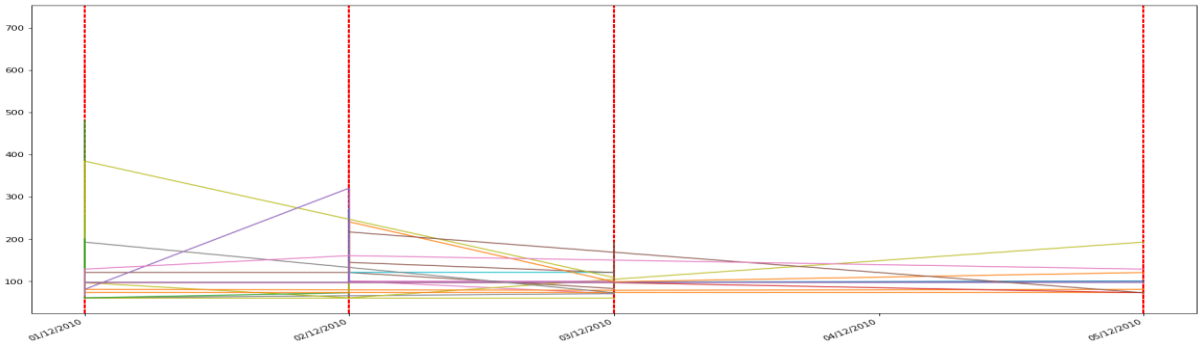

Figure 16: La variation de la quantité vendue des produits du cluster

- La distribution de l'élasticité croisée au sein du cluster par rapport au changement de prix du « PAPER CHAIN KIT 50'S CHRISTMAS »

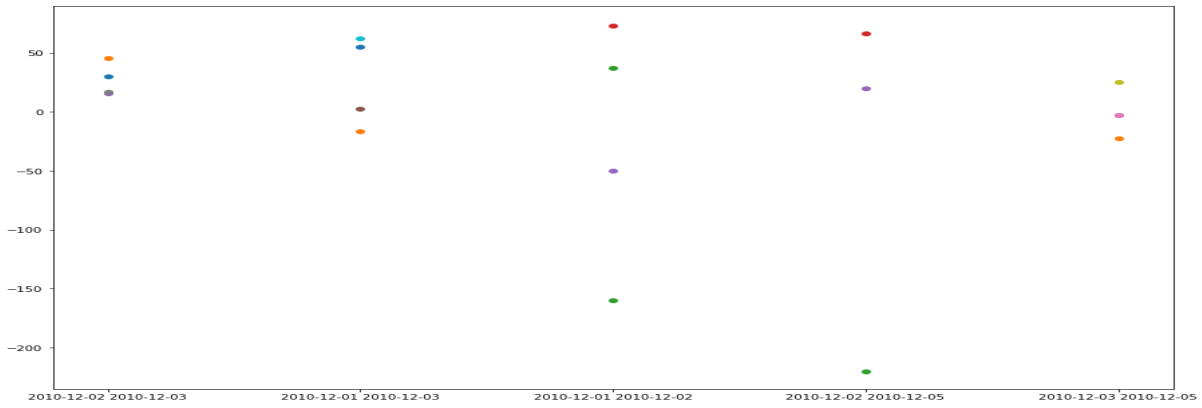

Figure 17: Elasticité croisée

Si on fixe le seuil à -50 pour les produits complémentaires, on remarque que le produit 22086 a trois produits complémentaires. Et si le seuil est de 50 pour les produits de substitution, il en a 4. Le reste des produits sont des produits indépendants.

1.5 Le graphe client-produits

1.5.1 Le top k des clients

Nous avons calculé le top 5 des clients selon le montant et la quantité de leurs achat
s.

```
#######top_k1#########
            Quantity     Montant
CustomerID
14646.0       196915   280206.02
18102.0        64124   259657.30
17450.0        69993   194550.79
16446.0        80997   168472.50
14911.0        80265   143825.06
```

1.5.2 Visualisation du graphe client-produits

La figure ci-dessous montre le graphe de liens du top 5 des clients et les produits.

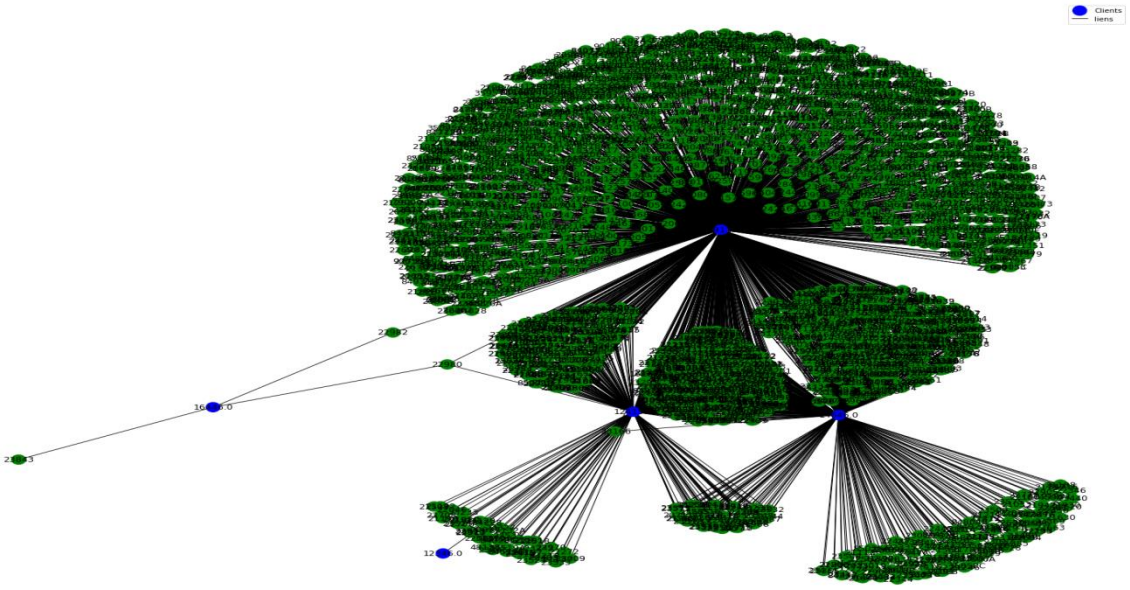

Figure 18: Le graphe de lien du top k des client et les produits

Les points blues représentent les clients et les points verts représentent les produits.

Nous avons remarqué que le client 16446 qui n'achète que 3 produits contrairement au client 14646

1.6 Corrélation des variables et Clustering

1.6.1 Corrélation des principales métadonnées :

Afin d'étudier l'intensité de la liaison qui peut exister entre les variables de notre Dat a set, nous étudions la corrélation entre ces variables. On calcule la corrélation des v ariables : puis on trace son graphe thermique :

Figure 19 : Corrélation des variables

Nous constatons que les variables UnitPrice et CustomerID ont de fortes valeurs de corrélation avec la variable Quantity, tandis que les variable Quantity et unitPrice sont faiblement corrélées à CustumerID.

On en déduit que dans nos données : La quantité des produits varie fortement en fonction de leurs prix unitaires ainsi que des consommateurs.

1.6.2 Clustering :

On s'intéresse à la distribution des commandes lancées. On applique l'algorithme K-means pour tenter d'identifier la répartitions des dates des commandes en fonction des prix unitaires.

Afin de rendre la lecture de la clusterisation plus pratique, nous avons appliqué l'algorithme pour k=2. Chaque cluster a également une couleur spécifique, pour rendre la carte plus lisible

L'algorithme que nous avons utilisé s'est donc basé ce principe :

1. Regrouper chaque objet autour du centroïde le plus proche.

2. Replacer chaque centroïde selon la moyenne des descripteurs de son groupe.

Après quelques itérations, l'algorithme converge après avoir trouvé un découpage stable de notre jeu de données

```
Out[110]: <matplotlib.colorbar.Colorbar at 0x192d65d0>
```

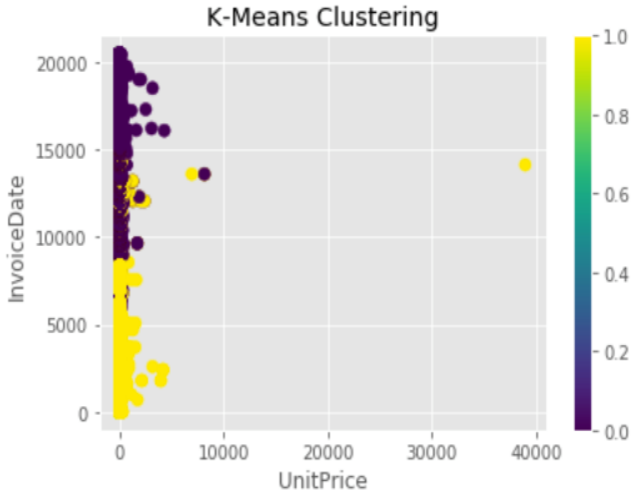

Figure 20 : k-means avec k=2

A partir du graphe, on constate la coexistence de deux clusters :

Le premier en violet : représente une valeur comprise entre 0 et 0.2 de densité, et caractérise des dates de commandes récurrentes où le prix unitaire du produit est peu variable.

Le deuxième cluster en jaune : représente une forte densité approximativement égale à 1, et caractérise des dates de commandes peu récurrentes où le prix unitaire du produit est quasiment stable.

Dans ce cluster, Un point aberrant figure et peut être expliqué par la hausse des prix unitaires en période de forte demande à un certain moment de l'année.

Conclusion :

Pour conclure, nous tenons à préciser qu'il s'agit d'un projet très intéressant car il permet de tirer autant de connaissances que possible à partir d'une base de données dans le but de supporter le processus métiers d'une structure donnée dans notre cas c'était le store du détaillant du prêt à porter.

Nous avons conclu que le Data Mining est un domaine ou interfère les statistiques, la data visualisation ainsi que les mathématiques appliquées, tous cela dans l'unique objectif d'aider à la décision.

Une vaste partie du temps consacrée à ce projet a été dédié au BENCHMARKING de multiples méthodes pour tirer les approches qui sont adaptées à notre cas.

Les approches que nous avons utilisées pour remonter de la connaissance à partir de notre data set sont : Analyses statistiques des données, Pattern Mining et Trend Discovery macro (sur l'ensemble des clients) et micro (pour les clients de chaque pays des tops 6 en termes d'apport en chiffre d'affaire), études des corrélations des variables, élaboration du graphe client-produit, tirage des règles d'associations et clusetering.

Annexe

Pseudo-Code - Algortihme FHM :

input : D: a transaction database,
minutil: a user-specified threshold
output: the set of high-utility itemsets
1 Scan D to calculate the TWU of single items;
2 I ∗ ← each item i such that TWU(i) < minutil;
3 Let be the total order of TWU ascending values on I ∗;
4 Scan D to built the utility-list of each item i ∈ I ∗ and build the EUCS structure;
5 ***Search*** *(∅, I ∗ , minutil, EUCS);*

Search Procedure :

input : P: an itemset, ExtensionsOfP: a set of extensions of P, the minutil threshold, the EUCS structure output: the set of high-utility itemsets 1
foreach itemset P x ∈ ExtensionsOfP do
2 if SUM(P x.utilitylist.iutils) ≥ minutil then
3 output P x;
4 end
5 if SUM(P x.utilitylist.iutils)+SUM(P x.utilitylist.rutils) ≥ minutil then
6 ExtensionsOfPx ← ∅;
7 foreach itemset P y ∈ ExtensionsOfP such that y x do
8 if ∃(x, y, c) ∈ EUCS such that c ≥ minutil) then
9 P xy ← P x ∪ P y;
10 P xy.utilitylist ← ***Construct*** *(P, P x, P y);*
11 ExtensionsOfPx ← ExtensionsOfPx ∪ P xy;
12 end
13 end
14 Search (P x, ExtensionsOfPx, minutil);
15 end
16 end

Construct procedure:

input : P: an itemset, P x: the extension of P with an item x, P y: the extension of P with an item y output: the utility-list of P xy
1 U tilityListOfP xy ← ∅;
2 foreach tuple ex ∈ P x.utilitylist do
3 if ∃ey ∈ P y.utilitylist and ex.tid = exy.tid then
4 if P.utilitylist != ∅ then
5 Search element e ∈ P.utilitylist such that e.tid = ex.tid.;

6 exy ← (ex.tid, ex.iutil + ey.iutil − e.iutil, ey.rutil);
7 end
8 else
9 exy ← (ex.tid, ex.iutil + ey.iutil, ey.rutil);
10 end
11 U tilityListOfP xy ← U tilityListOfP xy ∪{exy};
12 end
13 end
14 return UtilityListPxy;

Bibliographie

http://www.philippe-fournier-viger.com/spmf/index.php?link=algorithms.php

http://data-mining.philippe-fournier-viger.com/introduction-high-utility-itemset-mining/

https://www.displayr.com/what-is-dendrogram/

https://www.economicshelp.org/blog/195/economics/calculating-price-elasticity-of-demand/

Data Mining,Concepts and Techniques,Third Edition,Jiawei Han,University of Illinois at Urbana–Champaign,Micheline Kamber,Jian Pei,Simon Fraser University