Rapport Projet IC05

Analyse critique des données numériques

*W*hat will you PIC ?

Sommaire

Introduction

Dans le cadre de l'UV IC05 *Analyse critique des données numériques*, nous avons été amenés à réaliser un projet d'enquête sur les algorithmes de recommandation.

Après une mûre réflexion sur le choix du sujet que nous voulions aborder et nourris par l'ambition d'apprendre et de mettre en pratique des notions d'IA, nous avons décidé d'élaborer notre propre algorithme de recommandation. Le projet "What will you PIC ?" a alors vu le jour. L'objectif était d'étudier les habitudes de consommation des étudiants au PIC'asso, foyer étudiant de l'UTC, en utilisant des techniques d'analyse de données et d'intelligence artificielle, et proposer des nouvelles solutions pour guider les étudiants

Notre méthodologie s'est décomposée en plusieurs parties. Il était tout d'abord nécessaire d'obtenir l'accord du PIC'asso et de connaître les contraintes de confidentialité afin d'obtenir les données pseudo-anonymisées. Dans le même temps, nous avons commencé à nous former au Machine Learning. Étant seulement en GI02, nous n'avions pas encore abordé ces notions. Après cette étape de formation, nous avons pu commencer le projet en le découpant en trois parties :

1. Nettoyage et visualisation des données (preprocessing et EDA)
2. Détection de profils consommateurs (clustering)
3. Prédiction d'achat (classification)

Ce rapport détaille les différentes étapes de notre méthodologie et présente les résultats observés.

1 Préambule

1.1 Choix du sujet

Nos premières idées de sujet consistaient en l'analyse de célèbres algorithmes de recommandation déjà existants comme ceux de YouTube, Netflix ou Google. Après réflexion, nous avons eu l'ambition de nous challenger davantage pour ce projet en développant notre propre algorithme de recommandation. Quoi de plus intéressant pour comprendre leurs mécanismes que d'en créer un soi-même. Pourquoi ce choix ? Afin d'acquérir des connaissances pratiques en matière d'analyse de données et d'intelligence artificielle, mettre en pratique les notions qui nous seront utiles dans notre future filière IAD.

1.2 Formation

Pour nous former, nous avons suivi plus de 15 heures de vidéos et de formation proposées par Guillaume Saint-Cirgue, data scientist basé au Royaume-Uni, sur sa chaîne YouTube *Machine Learnia*. La playlist utilisée est référencée dans les sources. Ces vidéos nous ont permis de comprendre les différentes étapes nécessaires pour mener à bien un projet de machine learning.

Nous avons également visionné des vidéos plus approfondies de la chaîne StatQuest (également référencée dans les sources) afin de renforcer nos connaissances, notamment en matière d'encodage, ainsi que notre compréhension de certains algorithmes de machine learning.

1.3 Les données

Nous nous sommes tournés vers le PIC'asso pour obtenir les données nécessaires à notre projet. Étant donné l'importance de la confidentialité de ces données, qui pourraient permettre d'identifier des utilisateurs, elles ont été pseudo-anonymisées. Nous n'avons pas eu accès aux informations les plus sensibles, telles que l'âge et le sexe des consommateurs. Le PIC'asso nous a fourni un ensemble de fichiers Excel regroupant les ventes depuis sa création, que nous avons ensuite concaténés en un seul fichier.

2 Exploratory Data Analysis et preprocessing

2.1 Compréhension et préparation

Avant d'utiliser les données dans un modèle de machine learning, il est essentiel d'effectuer une **exploratory data analysis**, et de faire du **preprocessing**. Ces étapes sont essentielles pour comprendre, préparer et visualiser nos données. Dans cette phase, l'objectif est de nettoyer les données, d'acquérir une connaissance approfondie des données, notamment leur signification, les potentielles relations entre elles, les valeurs manquantes, les valeurs aberrantes...

2.1.1 Traitement sur les colonnes

Nous avons commencé par nous intéresser aux différentes colonnes (variables) dont nous disposions dans notre dataset. Nous avons remarqué que certaines contenaient plusieurs types de données. Nous nous sommes intéressés à ce défaut. Après quelques recherches, nous avons fini par comprendre que certaines données étaient de type "NaN" (not a number), ce qui correspond à des valeurs manquantes.

En traçant la heatmap des valeurs manquantes (en blanc sur le graphique), nous pouvons voir une visualisation pratique de l'ensemble de notre dataset, et nous pouvons ainsi facilement voir où se situent les valeurs manquantes. Nous avons remarqué que les colonnes *Nom point de vente* et *Activité* contenant des valeurs manquantes. Après avoir inspecté les différentes valeurs que comprenaient ces colonnes, nous avons remarqué que la colonne "activité" n'était pas utile pour notre étude, nous l'avons donc supprimé. Pour ce qui est de la colonne "point de vente", nous avons remarqué que certaines ventes ont été effectuées à des lieux qui ne correspondent pas au PIC'asso. Nous avons donc supprimé ces lignes (cf. traitement des lignes plus bas), pour ne garder que les lignes qui mentionnaient le PIC'asso ou dont le nom du point de vente était manquant (ces données sont nombreuses et une grande partie d'entre elles représentent des données du PIC'asso). Une fois ce traitement fait, nous avons supprimé la colonne "point de vente", qui n'était plus utile.

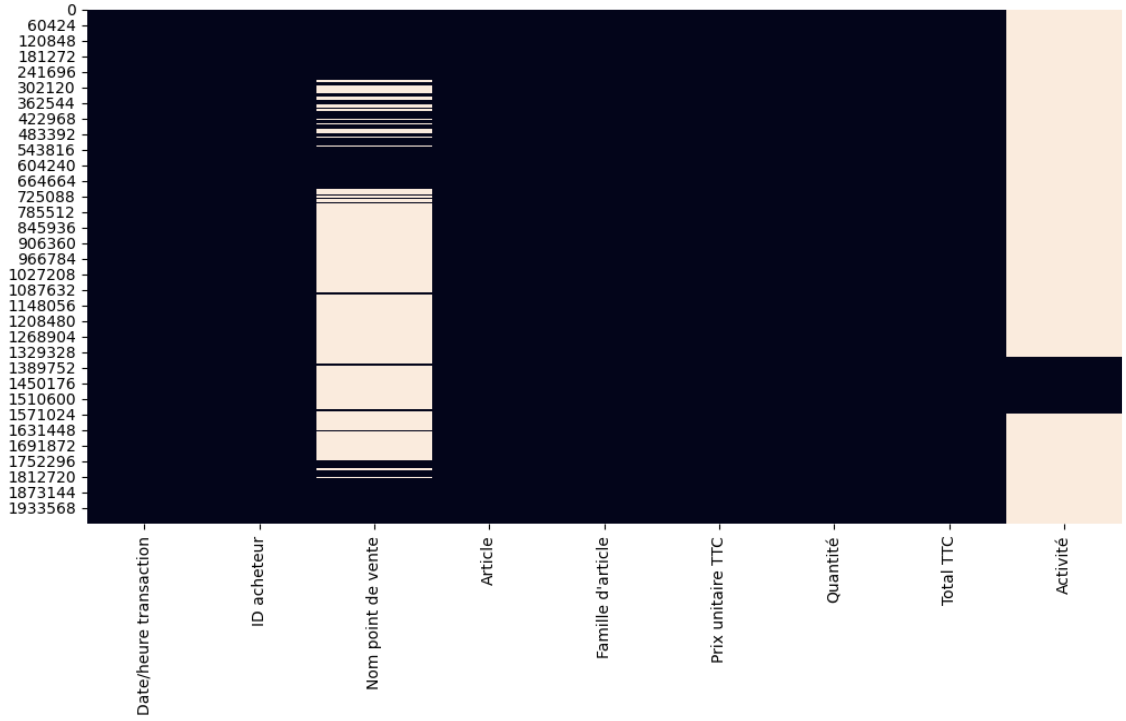

FIGURE 1 – Heatmap des valeurs manquantes de notre dataset initial

2.1.2 Modification sur le dataset restant

1. **Colonne date** : Le format de la colonne date n'était pas optimal pour la suite du projet : certaines cases comprenaient des millisecondes et d'autres non, de plus le format de la date n'était pas pratique pour être utilisé avec du code. Nous avons supprimé les millisecondes pour les cases concernées et transformé toutes les dates au format *datetime*, un format qui permet aux programmes de facilement comprendre et comparer des dates.
2. **Colonne famille d'article** : nous avons fusionné les familles d'articles ayant la même signification, par exemple « Bière bouteille » et « Bière pression » transformés en « Bière ».
3. **Colonne article** : Nous avons nettoyé tous les articles en supprimant les accents et en imposant la minuscule. Cela permet, comme pour les familles d'articles, de supprimer certains articles ont la même signification (« cuvée des trolls » et « Cuvee des trolls »). Cela n'a cependant pas suffi, car certains n'ont pas la même écriture selon le semestre, par exemple « peach mel bush » et « peche mel bush ». Un traitement à la main a donc dû être appliqué pour gérer ces cas.

2.1.3 Ajout de colonnes

Nous avons transformé la colonne date en des informations plus pertinentes au modèle. Une date au format datetime (date + heure) est une information brute qui contient des informations qui sont pertinentes et d'autres non. Ce qu'il est intéressant de savoir concernant un achat, relativement à la date, c'est le créneau de la journée (matin, midi, après-midi et soir), le jour (lundi à vendredi) et le semestre. Nous avons donc ajouté ces trois nouvelles colonnes en déduisant leurs valeurs pour chaque ligne en fonction de la colonne date.

2.1.4 Suppression des lignes inutiles

Dans le dataset récupéré, il existait beaucoup de lignes inutiles. Garder ces lignes auraient altéré les résultats de notre modèle. Avant d'expliquer en quoi elles modifieraient les résultats, il est préférable de vous présenter les suppressions effectuées.

Nous avons supprimé :

1. Toutes les ventes dont le nom de point de vente ne concerne pas le PIC'asso
2. Les familles d'articles trop peu présentes dans le fichier
3. Les articles trop peu présents dans le fichier
4. Les retours de produit - Ecocup (quantité négative ou prix négatif)
5. Les achats en grande quantité effectués par des associations (quantité ou prix excessif, ne représente pas le comportement d'un étudiant)
6. Les ventes faites le samedi et le dimanche (hors de notre cadre d'étude)
7. Les ventes faites pendant l'inter-semestre.

Notre domaine d'étude se limitant aux habitudes d'achat des étudiants au PIC'asso en semaine, les informations des inter-semestres, des évènements UTCéens et du week-end ne nous intéressaient pas. Quant aux autres données supprimées, elles faussent le comportement type d'un étudiant, étant des valeurs aberrantes.

Si nous les avions laissées, ces valeurs aberrantes auraient faussé les statistiques descriptives, comme la moyenne, l'écart-type, et nous auraient donné une image incorrecte de la distribution des données. Le risque de garder ces valeurs est aussi d'entraîner un sur-apprentissage (overfitting) du modèle, où celui-ci s'adapte trop aux données aberrantes spécifiques au lieu de généraliser correctement. Cela peut notamment entraîner une mauvaise performance du modèle lorsqu'il est confronté à de nouvelles données. Ainsi, un traitement approprié des valeurs aberrantes contribue à améliorer la qualité des prédictions.

2.2 Visualisation

Afin de bien comprendre les données, il est important de faire une visualisation des données. Par ce biais, on peut déjà découvrir certaines tendances. Voici donc les graphiques que nous avons jugés les plus pertinents à vous présenter.

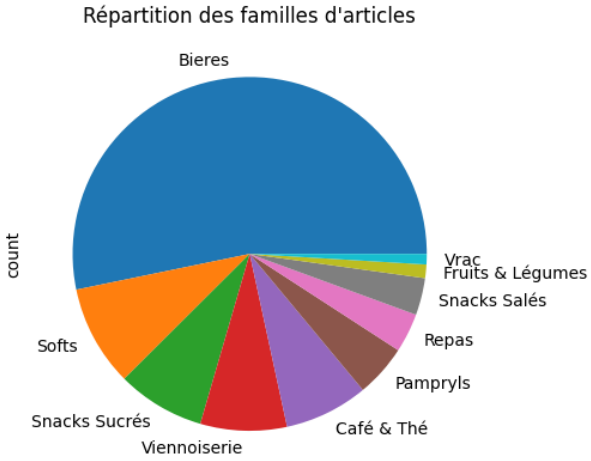

FIGURE 2 – Répartition générale des ventes au PIC'asso par famille d'article

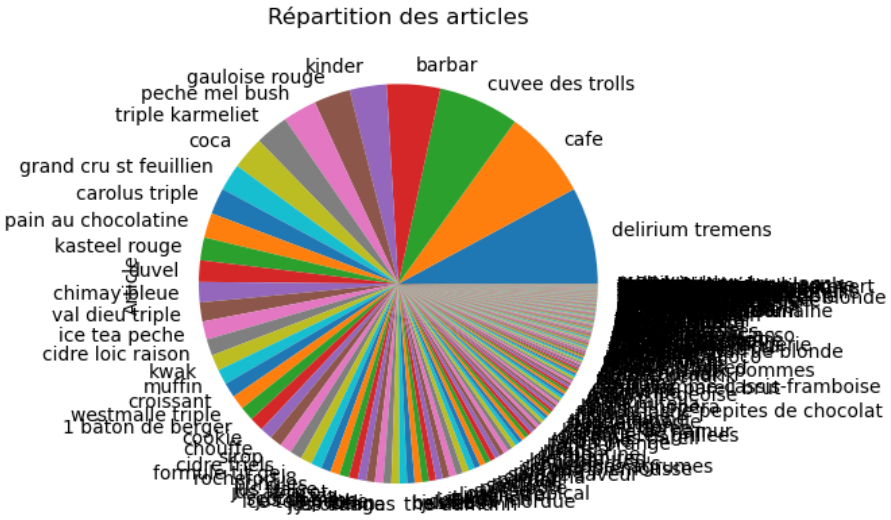

FIGURE 3 – Répartition générale des ventes au PIC'asso par article

Nous constatons que les articles phares du PIC'asso sont donc les bières, suivi de très loin par les softs et les snacks. Pour ce qui est des articles, on voit que l'on a une vingtaine d'articles phares qui se vendent beaucoup, suivis par une multitude d'articles un peu moins vendus. Il peut maintenant être intéressant de voir l'évolution des consommations par rapport au moment de la journée.

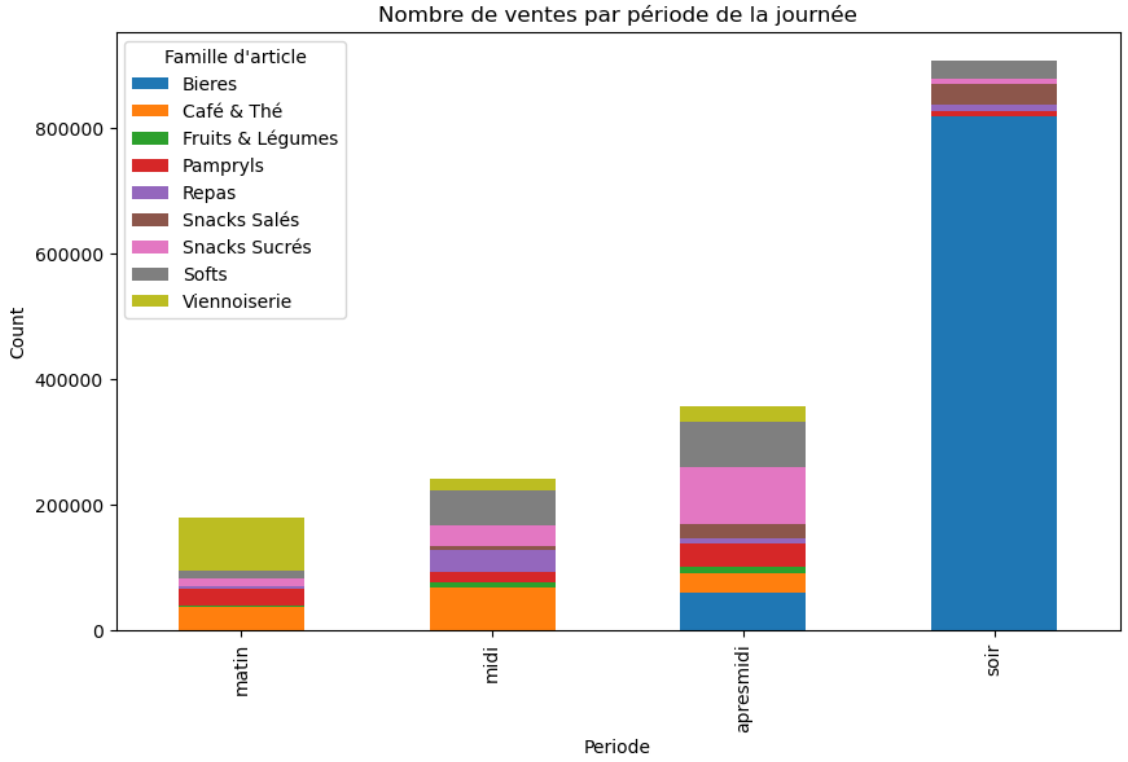

FIGURE 4 – Représentation du nombre de ventes par période

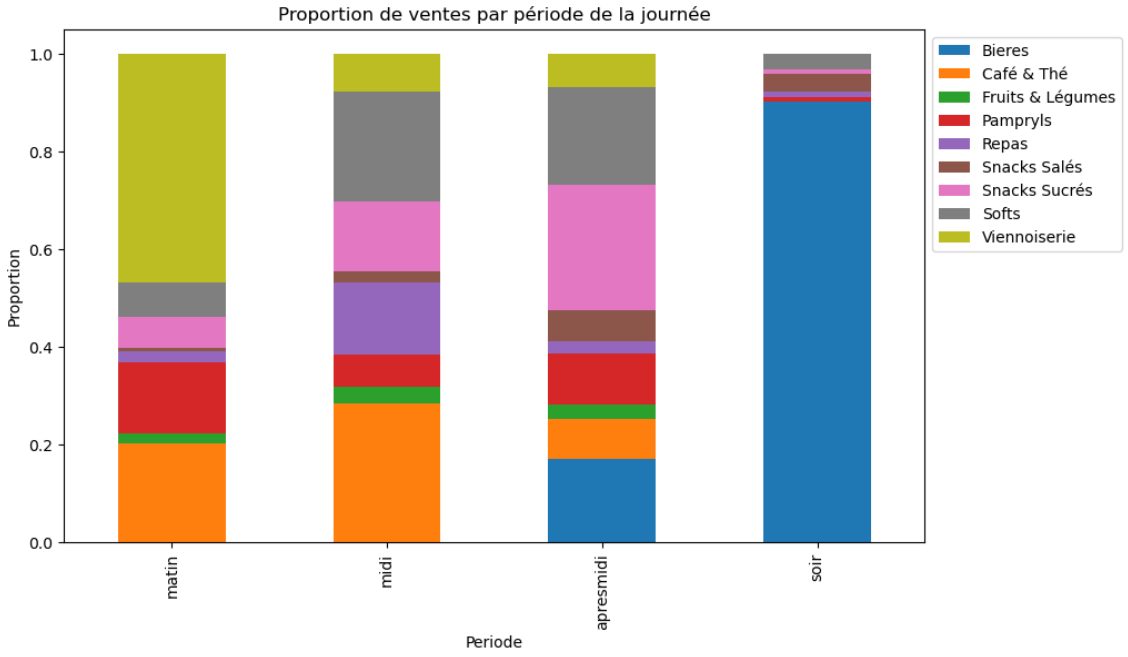

FIGURE 5 – Représentation de la proportion de ventes par période

Ici, on voit en premier une représentation des ventes totales par période de la journée et par catégorie, et en deuxième le même graphe, mais sur une même échelle afin de ne plus comparer la quantité, mais la proportion. En moyenne, 53.9% des achats sont effectués le soir, ce créneau se retrouve loin devant l'après-midi avec 21.2%, le midi avec 14.3% et le matin avec 10.6%. Le soir est donc la période de consommation principale. Nous pouvons remarquer que la proportion de consommation au fil de la journée est logique : le matin, les articles les plus consommés sont le

café, les viennoiseries et jus de fruits, le midi le café, les sodas et les repas, l'après-midi des snacks sucrés (goûter) et des softs, et enfin le soir quasi exclusivement des bières.

Nous pouvons maintenant nous intéresser à la consommation en fonction du jour de la semaine :

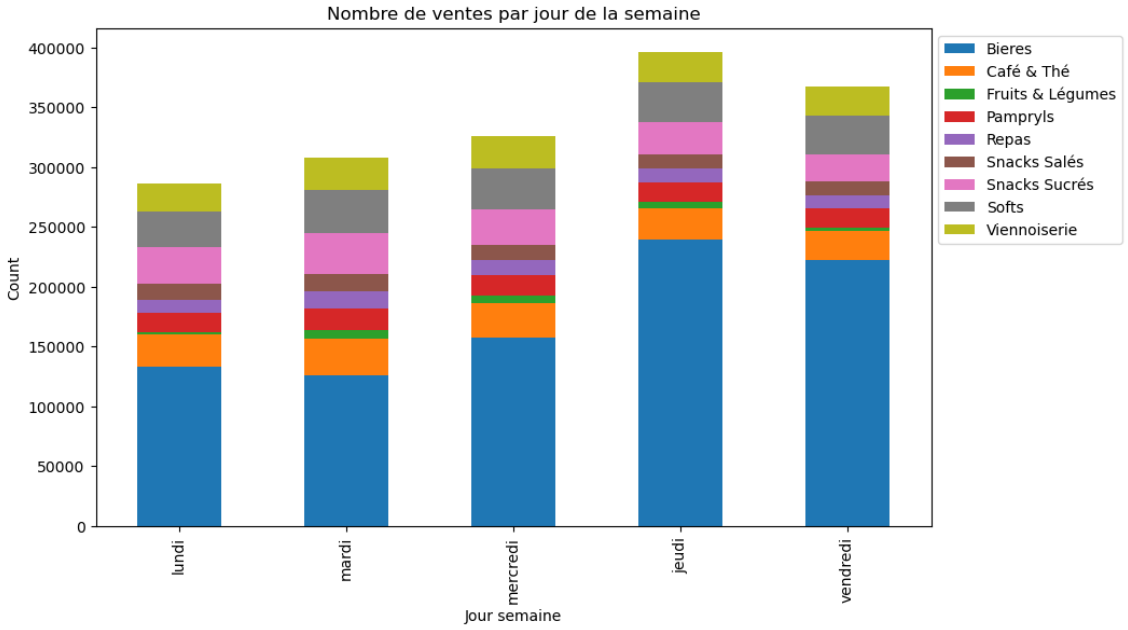

FIGURE 6 – Représentation du nombre de ventes par jour de la semaine

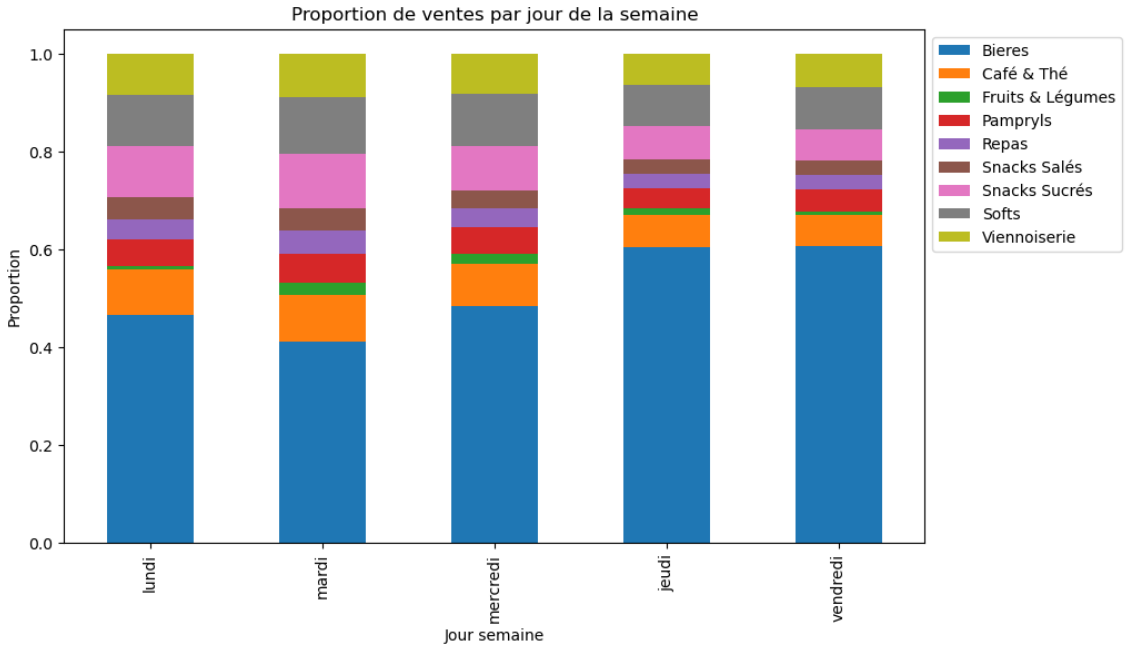

FIGURE 7 – Représentation de la proportion de ventes par jour de la semaine

C'est bien le jeudi qui devance les autres jours de la semaine. Ces graphiques correspondent bien à ce que l'on s'était imaginé, le jeudi étant le jour le plus animé au PIC'asso, suivi de peu par le vendredi. Quand on s'intéresse aux uniquement proportions des consommations pour chaque famille d'article, on peut voir que les

gens consomment de la même manière tout au long de la semaine, à l'exception des bières qui sont consommées légèrement plus en fin de semaine.

Nous nous sommes également demandés si le PIC'asso suivait ou non les tendances de société. Tout d'abord, regardons ce qu'il s'est passé durant la période de la Covid-19.

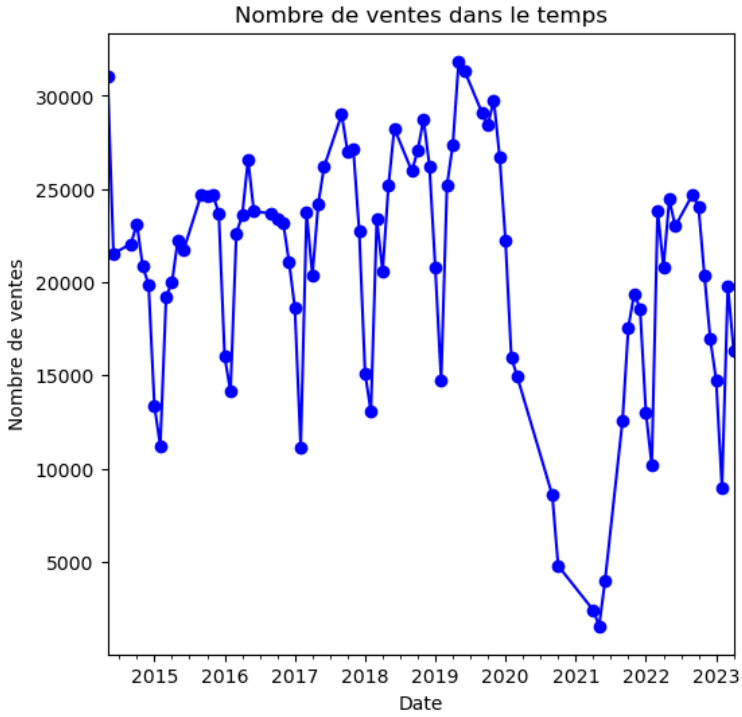

FIGURE 8 – Évolution du nombre de ventes dans le temps

On remarque effectivement une chute massive des ventes durant la période covid. L'UTC ayant fermé durant le semestre P20, puis rouvert avec le PIC'asso fermé jusqu'à la moitié du semestre de P21. Nous pouvons également voir des baisses régulières à chaque semestre, au niveau de la même période. En regardant les dates de plus près, nous avons remarqué que ces chutes représentaient les semaines d'examens (et non les inter-semestres qui ne se voient pas ici, car nous avons retiré les données correspondant à ces périodes).

Intéressons-nous maintenant aux prix des différents articles :

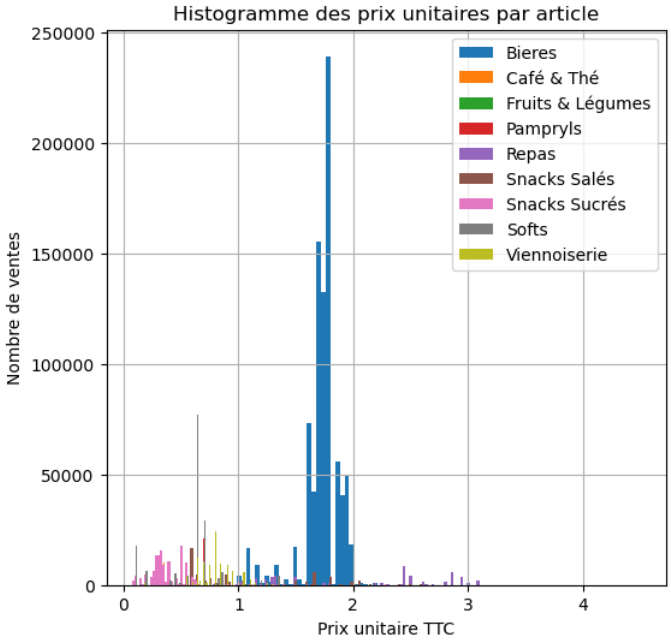

FIGURE 9 – Histogramme des prix par famille d'article

Au-delà du fait que ce graphique nous rappelle que les bières sont très vendues au PIC'asso, ce dernier nous montre que chaque famille d'article se vend dans un certain ordre de grandeur de prix : les bières coûtent à peu près toutes entre 1.6€ et 2€, les snacks sucrés autour de 40c, les softs 85c, etc. La seule exception est pour les repas dont le prix ne varie plus, on voit leur prix varier de 1.5€ à plus de 3€.

Pour finir, étant donné que les bières ont l'air d'être très importantes au PIC'asso, nous avons décidé de tracer l'évolution de leur prix moyen, afin d'inspecter leur prix de plus près :

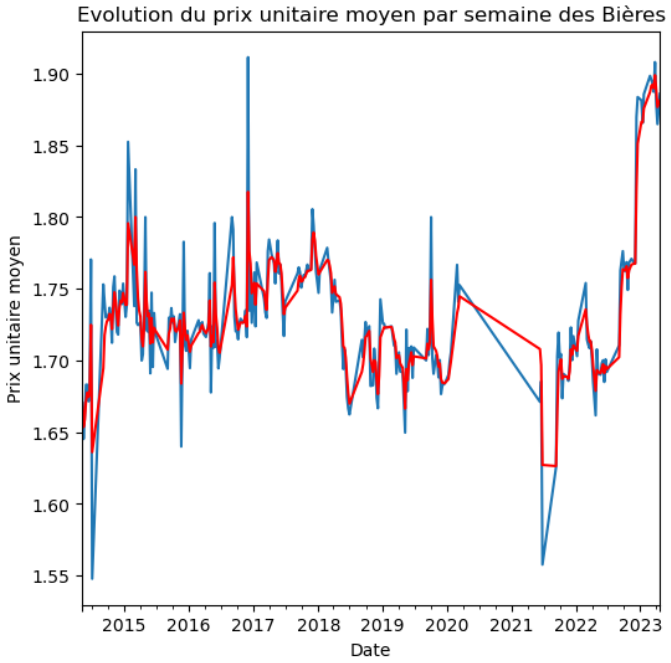

FIGURE 10 – Évolution du prix moyen des Bières

On peut voir en bleu la moyenne exacte sur une semaine et en rouge la moyenne

glissante sur trois semaines. Cette courbe en rouge, nous a permis de mieux comprendre la tendance du prix unitaire des bières du PIC'asso. On remarque une tendance fortement à la hausse du prix des bières au PIC'asso. depuis mi-2021. Les prix des bières suivent donc la tendance inflationniste de notre société, le PIC'asso étant contraint à augmenter également ses prix. Pour les variations pré-2021, les variations peuvent s'expliquer par l'évolution naturelle du prix des bières sur le marché, mais également aux bières que propose le PIC'asso. Cette courbe représentant une moyenne, si le PIC'asso arrête de vendre une bière qui était chère et se met à vendre une autre bière moins chère, alors la moyenne baisse, sans que les bières aient réellement changé de prix.

Ceci clôt le travail de preprocessing et d'EDA, passons maintenant à la phase de détection de profils consommateurs.

3 Clustering

3.1 Création d'un nouveau jeu de données

Pour faire de la détection de profil de consommateurs, nous devons utiliser une technique appelée le "clustering". Le clustering consiste à détecter des rassemblements de données et de créer des groupes en fonction de ces rassemblements. Le clustering peut s'apparenter à de la classification, à une différence majeure près : avec la classification, les groupes sont définis à l'avance et connus, c'est donc un problème d'apprentissage supervisé, tandis qu'avec du clustering, les groupes ne sont pas connus à l'avance, on ne sait ni combien il y aura de groupes ni quels seront les groupes. Par exemple, avant de mettre en pratique tout cela, nous pensions que les groupes de consommateurs qui allaient se former seraient en fonction du type de produits consommés (par exemple un profil "bière avec les amis le soir" et un profil "petit-déjeuner avant les cours"), mais nous avons fini par nous rendre compte que ce n'était pas le cas.

Avant de pouvoir faire du clustering, nous avons dû faire face à un problème. Quand on veut faire du clustering, on cherche à détecter des groupes dans les points (les lignes) que contient notre dataset. Sauf que pour cela, il faut se poser la question de ce que représente un point dans notre dataset. Dans le cas présent, chaque ligne représente une vente et non un client, on ne peut donc pas encore regrouper des points de clients si nos points représentent des ventes et non des clients. Pour ce faire, nous avons donc dû créer un nouveau dataset, où pour chaque client, nous avons récupéré l'ensemble de leurs achats pour les réunir en une seule ligne. Dans ce nouveau jeu de données, chaque ligne représente un client et l'ensemble des achats qu'il a effectués. Nous avons donc représenté un client par son nombre d'achats pour chaque famille d'article et pour chaque période de la journée, ainsi que la somme moyenne dépensée par période de la journée. On a ainsi des colonnes comme "soft_apresmidi" représentant pour un utilisateur donné le nombre d'achats de softs que ce dernier a effectué l'après-midi.

3.2 Traitement avant le clustering

Avant de faire du clustering, il reste quelques dernières étapes à effectuer. Tout d'abord, nous avons normalisé les données. Il est très important de normaliser les données afin de toutes les mettre sur la même échelle, ce sans quoi les algorithmes sont bien moins performants.

Ensuite, nous nous sommes rendus compte que l'étape précédente avait créé beaucoup de variables, dont certaines plus utiles que d'autres. Nous avons appliqué deux méthodes pour réduire le nombre de variables.

La première fut d'utiliser le "feature selection", méthode consistant à sélectionner les variables les plus pertinentes en fonction de certains critères. Un critère souvent utilisé est la variance : on cherche des variables qui ont un impact significatif sur la sortie du modèle. Si une colonne contient très peu de valeurs différentes, alors il est légitime de remettre en cause son utilité, car ce n'est pas sur cette colonne que le modèle pourra se baser pour prendre ses décisions. Nous avons donc éliminé les variables avec le moins de variance, ce qui nous a par exemple permis d'éliminer des colonnes comme "biere_matin". En effet, la consommation de bières le matin étant quasiment nulle pour tout le monde, inutile d'intégrer cette colonne.

La deuxième méthode que nous avons utilisée pour réduire le nombre de variables est la réduction de dimension. Après différents tests et étude des algorithmes qui nous permettaient de faire cela, nous avons finalement opté pour PCA (Analyse en composantes principales). Avec PCA, le but est généralement de réduire le nombre de variables (dimensions) au maximum tout en gardant un maximum de variance expliquée (voir cours dans les sources et page Wikipédia sur PCA). Nous nous sommes donc servis de cela pour savoir jusqu'où réduire la dimensionnalité.

Après ces étapes, nous avions donc des données normalisées, des variables significatives et un nombre de variables raisonnable.

3.3 Clustering

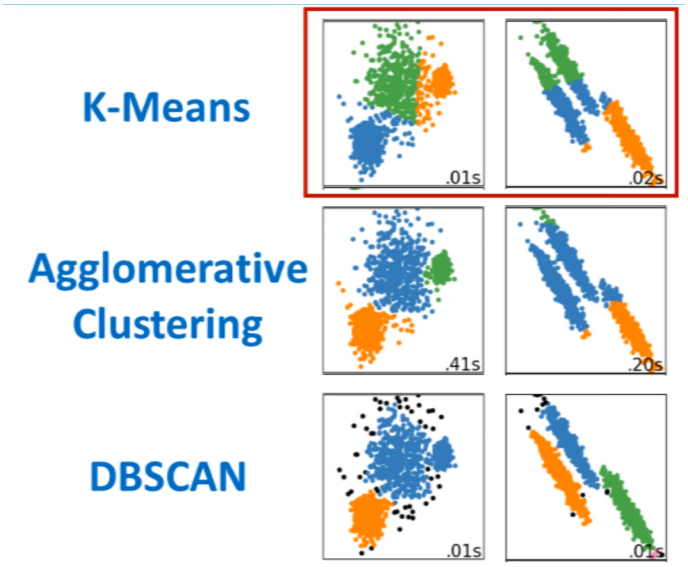

FIGURE 11 – Création de groupe par différents algorithmes de clustering

On peut ici avoir une intuition des modèles les plus pertinents. Cependant, cela n'est que possible grâce au fait que les données sur cette image soient en deux dimensions, cette visualisation devient impossible dès lors que l'on augmente la dimensionnalité.

Nous avons alors testé tous ces différents algorithmes afin de les comparer. Il existe de nombreuses manières de mesurer la performance des modèles (métriques), mais toutes semblaient indiquer que Kmeans était le plus pertinent pour notre étude, suivi de près par Agglomerative Clustering. Voici la courbe d'une des métriques qui nous a permis d'arriver à cette conclusion (les courbes pour les autres métriques sont similaires).

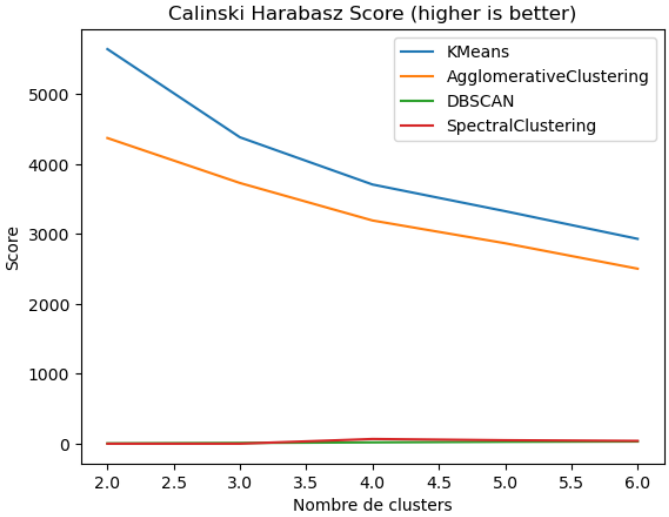

FIGURE 12 – Efficacité des modèles de clustering

Une fois le modèle choisi, nous avons optimisé les "hyper paramètres", qui sont des paramètres qui définissent davantage un détail comment le modèle se comporte. Un exemple d'hyper paramètre est le nombre de groupes, de "clusters" que le modèle doit chercher. Pour le nombre de clusters, nous avons utilisé une technique très populaire appelée "elbow method", consistant à chercher un coude dans les courbes des différentes métriques.

Une fois le modèle prêt, nous pouvons enfin visualiser la manière dont le modèle a regroupé les données. Nous ne pouvons malheureusement pas tout visualiser en une seule fois en raison de la dimensionnalité de nos données. En revanche, nous pouvons visualiser ces groupes sur certains axes, en prenant les axes deux à deux (voir une variable par rapport à une autre). En nous basant sur l'EDA, nous avons donc sélectionné les variables les plus importantes telles que "viennoiserie_matin", "biere_soir", etc, pour visualiser sur ces variables le comportement des données :

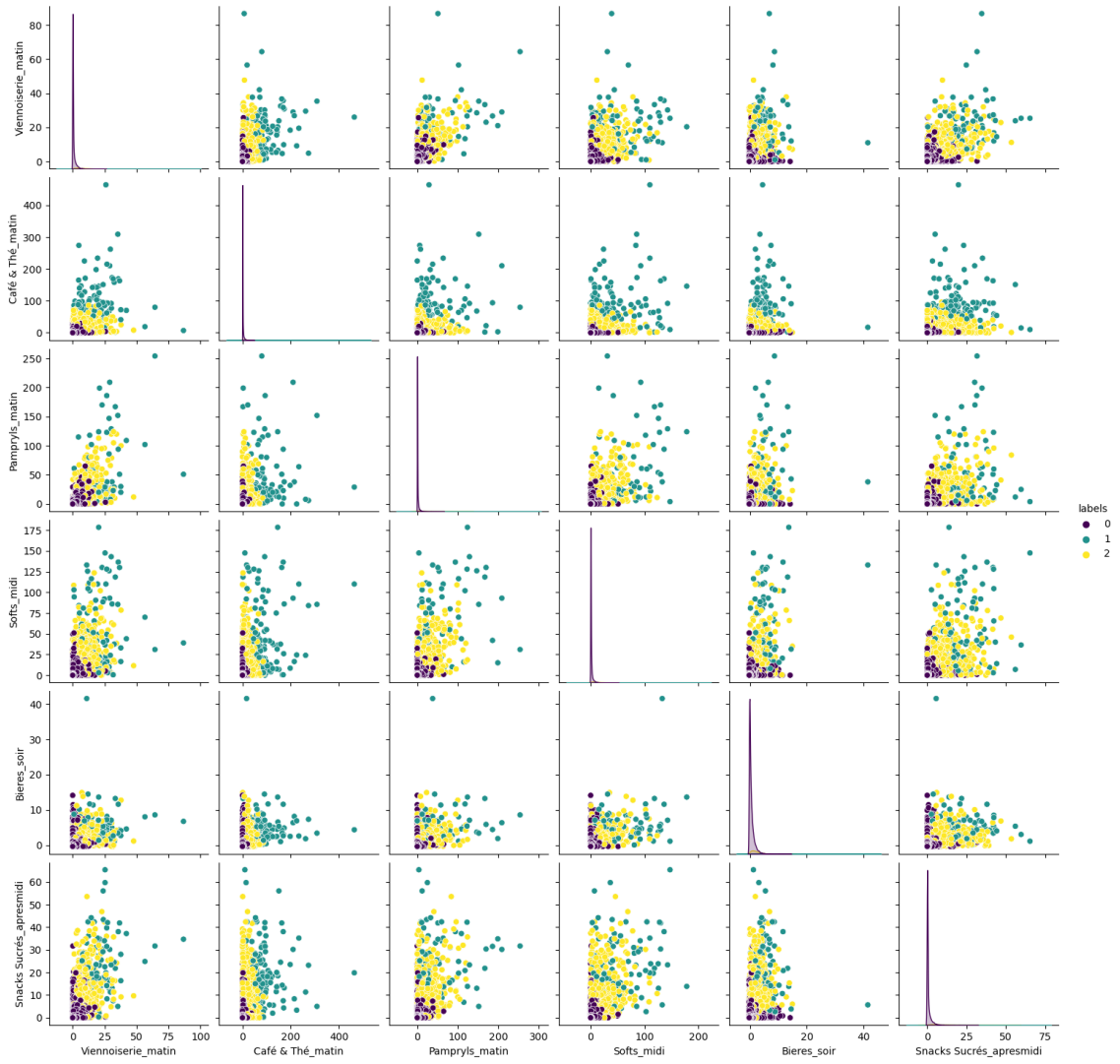

FIGURE 13 – Nombre d'achats d'une catégorie par rapport à une autre

Malgré le nombre limité de variables que nous avons gardé pour l'affichage, il y a beaucoup de graphes au vu du nombre élevé de combinaisons de deux variables possibles. Ces graphiques sont cependant très intéressants. Pour chaque graphique, on a en abscisse le nombre d'achats pour une famille d'articles, et en ordonnée le nombre d'achats pour une autre famille d'articles. Un point en bas à gauche représente alors un client qui ne consomme ni l'un ni l'autre, en haut à gauche beaucoup de l'un, mais très peu de l'autre, vice-versa pour en bas à droite, et enfin un point en haut à droite représente un client qui consomme beaucoup de produits des deux familles d'articles.

Avant notre étude, on s'attendait à ce que les utilisateurs se différencient en fonction de la nature de ce qu'ils consomment d'un côté ceux qui boivent une bière le soir et de l'autre ceux qui prennent leur petit déjeuner. Si cela avait été le cas, alors on aurait vu un groupe (une couleur) s'étendre sur un axe et l'autre groupe s'étendre sur l'autre axe, or ce n'est pas ce que l'on peut observer. À la place, on voit plutôt des groupes qui se forment en strates, on peut voir en violet le groupe de ceux qui consomment très peu, peu importe le produit, en jaune les consommateurs modérés, pour tous les produits, et en vert les gros consommateurs, encore une fois tous produits confondus. Le fait que ce schéma se retrouve sur tous les graphes nous montre qu'il existe bien des groupes, mais ces derniers ne sont pas basés sur les critères que nous attendions. Les étudiants au PIC'asso ne se différencient pas ce qu'ils

consomment, mais par combien ils consomment. Il n'y a pas de tendance significative à consommer plus un produit qu'un autre pour des groupes d'étudiants, chaque groupe consommera à peu près la même chose, juste pas dans les mêmes proportions.

Nous avons donc réussi à détecter différents profils de consommateurs, nous avons pu en apprendre plus sur les comportements de consommations et nous avons pu voir que les hypothèses et intuitions de départ ne sont pas toujours les bonnes.

4 Classification

Maintenant que nous avions réussi à détecter des profils consommateurs, nous voulions mettre en place un algorithme de recommandation de produits. Ce problème n'est pas un problème de clustering mais de classification, car ici, on connait à l'avance quelles sont les catégories que l'on cherche à prédire. Dans le contexte de notre étude, notre objectif principal est de prédire les articles qui seront consommés par un client spécifique.

4.1 Première idée : Prédiction sur les articles

Nous avons entrepris une modélisation de la recommandation d'articles en utilisant les données précédemment présentées. Dans cette optique, nous avons cherché à prédire l'article en nous basant sur plusieurs variables, à savoir :

- le moment de la journée : *Matin*, *Midi*, *Après-midi*, *Soir*.
- le jour de la semaine du *lundi* au *vendredi*.
- le type de client : *Profil consommateur* correspondant au profil détecté durant la phase de clustering.

Pour commencer, nous avons procédé à l'encodage de nos données, car les algorithmes de classification ne peuvent pas traiter directement les chaînes de caractères. Par défaut, nous avions utilisé l'ordinal encoding, méthode d'encodage qui permet d'encoder n classes par des nombres entre 0 et n-1, mais cela entraîne un ordre arbitraire entre les classes qui, dans notre cas, a réduit la précision de notre score. Par conséquent, nous avons opté pour le One-Hot encoding, qui transforme n variables catégorielles en n colonnes booléennes. Par exemple, au lieu d'une colonne "Période" qui prend comme valeur "matin", "midi" "après midi" ou "soir", on remplace par quatre colonnes, "matin" qui prend comme valeur 0 ou 1, idem pour les autres. Cette technique est très utilisée, car elle permet d'encoder les données non numériques sans leur donner d'ordre arbitraire. Son défaut est qu'elle va créer autant de colonne que de catégorie, ce qui peut faire beaucoup lorsqu'on a beaucoup de valeurs possibles. Dans notre cas, cela ne pose pas vraiment un problème, car nous n'avions que peu de valeurs possibles (5 pour les jours, 4 pour le moment de la journée et 3 pour les clusters). Nous avons donc appliqué cette méthode pour l'encodage.

Nous avons également dû encoder notre target (la colonne que l'on essaie de prédire) : l'article. Nous avons ici pu utiliser le label encoding, qui est une méthode d'encodage qui effectue la même transformation que l'ordinal encoding, mais est faite pour la target. Cela ne pose pas un problème d'encoder la colonne à prédire de cette manière. En effet, l'ordre des valeurs de la target n'est pas pris en compte dans les algorithmes d'apprentissage que nous utilisons et ne risque donc pas de biaiser nos résultats.

Pour le choix de l'algorithme, au vu de la quantité de donnée, nous devions porter notre choix sur des modèles paramétriques. Il existe en effet deux types de modèles en machine learning, les modèles paramétriques et les modèles non paramétriques. Les modèles paramétriques sont légers, car ils résument la modélisation qu'ils représentent en une équation, il n'y a donc qu'à stocker des paramètres. Une régression linéaire, par exemple, est un modèle paramétrique, ou l'on a juste à stocker les différents coefficients de l'équation. Un exemple de modèle non paramétrique est le

KNeighbors classifier, qui est un modèle de classification non paramétrique qui fonctionne de la manière suivante : lorsqu'on veut déterminer la catégorie d'un nouveau point, on regarde en fonction de ses caractéristiques les K points les plus proches, et l'on assigne à ce point la classe majoritaire de ses voisins. Voici un exemple visuel de ce que cela peut donner :

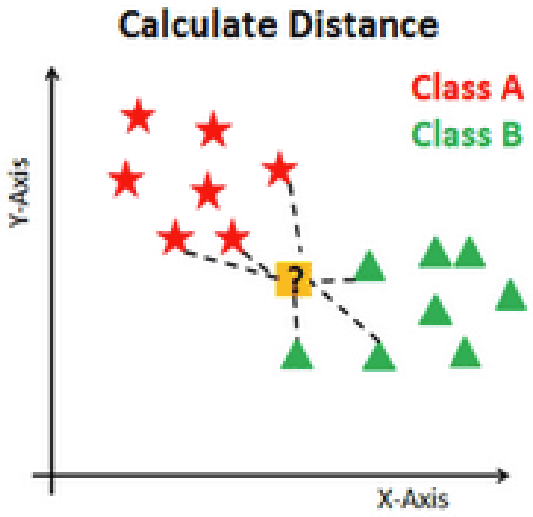

FIGURE 14 – Représentation visuelle du Kneighbors classifier

Ce type de modèle ne fait donc pas ses prédictions en fonction de paramètres qu'il a calculé afin de les mettre dans une équation, mais directement à partir des données qui ont servi à son entraînement. Bien que ce type de modèle puisse s'avérer pratique dans certaines situations, il trouve vite ses limites quand on se trouve en présence de beaucoup de données. En effet, lorsque l'on veut prédire un nouveau point, on doit calculer la distance entre ce point et tous les autres du jeu de données, afin de savoir quels sont ses K plus proches voisins. Cela reste faisable lorsque l'on n'a que quelques dizaines ou centaines de milliers de points. En revanche, ce genre d'algorithme arrive vite à ses limites lorsque l'on est en présence de beaucoup de données, ce qui est notre cas. Les fournisseurs de librairies de machine learning, telles que Scikit-Learn, conseillent eux-mêmes de se tourner vers des modèles paramétriques lorsque l'on est en présence de beaucoup de données.

Nous avons alors comparé les résultats obtenus avec plusieurs algorithmes de classification paramétriques, notamment la régression logistique (LogisticRegression) et le classifieur stochastic gradient descent (SGDClassifier), un classifieur basé sur la descente de gradient. Nous avons obtenu des scores très similaires pour les deux algorithmes de l'ordre de 5 à 10% de précision. L'algorithme de SGDClassifier prenant nettement moins de temps à être entraîné, en raison de sa méthode d'apprentissage, nous avons donc opté pour ce modèle.

Ce score est assez faible et nous avons poussé l'étude pour comprendre pourquoi. Le fait que le score soit trop faible est appelé "underfitting". "L'overfitting" est le phénomène qu'un modèle se spécialise trop sur les données qu'il a reçues en entraînement et n'arrive pas à généraliser sur de nouvelles données, tandis que "l'underfitting" est le phénomène ou un modèle n'arrive pas à capturer la complexité de données, que ce soit sur de nouvelles données ou ses données d'entraînements.

L'underfitting peut avoir de nombreuses causes, mais nous avons fini par trouver ce qui causait de l'underfitting dans notre cas : le manque de variables explicatives. Tout l'intérêt d'un algorithme de machine learning est de prédire une valeur sur une entité, valeur qui peut être déduite en fonction d'autres caractéristiques de cette entité. Cependant, si l'on n'a pas assez de caractéristiques sur cette entité, faire des prédictions précises devient moins possible. On essaye ici de faire des prédictions d'articles, on cherche à prédire quel article un client donné va consommer, en nous basant uniquement sur le moment de la journée, de la semaine, et son profil (s'il consomme beaucoup). Avec si peu de données de prédictions, il est difficile de prédire quelque chose d'aussi précis qu'un "Coca cola". On pourrait prédire avec facilité qu'un consommateur a besoin d'un "soda", mais en prédisant des articles, les algorithmes doivent par exemple savoir que recommander entre un "Coca cola", un "7up" et un "Orangina", ce qui est peu faisable, d'autant plus qu'il y a des centaines d'articles différents possibles. Des informations plus précises sur les clients telles que genre, l'âge, et autres informations personnelles pourraient permettre de meilleures prédictions. L'accès à de telles données poserait néanmoins de sérieuses questions en termes d'éthique étant donné que cela implique une réelle utilisation commerciale des données personnelles des utilisateurs.

Afin de ne pas en rester là, nous avions poussé l'étude afin de savoir si nos données pseudo anonymisées nous permettaient de faire de bonnes prédictions sur le type d'article dont un client a besoin (la famille d'article donc). Cela donnerait une information moins précise, on prédit par exemple qu'un client a besoin d'une viennoiserie au lieu de prédire qu'il a besoin d'un pain au chocolat, mais cela a au moins le mérite d'être plus respectueux de la vie privée.

De plus, lors de l'EDA, nous avons pu voir que nos variables avaient des résultats significatifs sur les familles d'articles. Nous nous sommes alors tournés vers la prédiction des familles d'articles.

4.2 Seconde idée : les familles d'articles

Ainsi, nous avons exploré une deuxième approche consistant à prédire les familles d'articles plutôt que les articles. Pour ce faire, nous avons suivi les mêmes étapes que précédemment et simplement changé la target. Cette approche s'est révélée fructueuse, nous permettant d'atteindre initialement 60% de précision.

Une fois le modèle choisi, nous avons cherché les meilleurs hyper-paramètres en utilisant la méthode de recherche par grille avec validation croisée (GridSearchCV). Cette méthode nous a permis d'explorer un large éventail de combinaisons de paramètres et d'identifier ceux qui offraient les meilleures performances. Finalement, nos efforts ont fini par payer et nous avons réussi à améliorer notre score à 65% de précision.

4.3 Troisième idée : prédiction des articles en utilisant la famille d'article

Dans notre volonté initiale de prédire les articles, nous avons à nouveau essayé de prédire un article à recommander, mais nous avons décidé d'utiliser la famille d'article en plus des variables précédentes. On cherche donc à recommander un article en fonction d'un moment dans la journée, un moment dans la semaine, un profil consommateur et une famille d'article. Cela s'éloigne légèrement de notre volonté ini-

tiale. En effet, cela suppose que l'utilisateur saura à l'avance la famille d'article qu'il souhaite consommer avant que l'on soit en mesure de prédire l'article, par exemple le jeudi après-midi, un consommateur modéré voudra un "soda" et l'algorithme se chargera de trouver lequel. Dans ce cas, contrairement à la première idée, le but n'est plus de recommander un produit parmi toute la liste des produits disponibles.

Cette méthode nous a permis de tripler nos résultats précédents et nous avons obtenu un score de 28% cette fois-ci, après optimisation du modèle. Bien que ces résultats ne soient pas extrêmement précis, savoir prédire l'article correct correspondant à une situation près de 30% du temps reste une bonne performance, surtout au vu du nombre d'articles disponibles.

Conclusion

En conclusion, notre projet d'enquête sur les algorithmes de recommandation a été une expérience enrichissante. Nous avons réussi à mettre en pratique nos connaissances en matière d'analyse de données et d'intelligence artificielle en développant notre propre algorithme de recommandation. Notre objectif était d'étudier les habitudes de consommation des étudiants au PIC'asso et de proposer de nouvelles solutions pour les guider dans leurs choix.

Au cours de notre projet, nous avons réalisé une phase d'*EDA* et de *preprocessing* pour comprendre et préparer les données, un modèle de clustering pour regrouper les étudiants selon leurs habitudes de consommation, et des modèles de classification pour prédire les futurs achats des étudiants.

Cependant, notre projet présente certaines limites. Tout d'abord, les données utilisées étaient pseudo-anonymisées et ne contenaient pas toutes les informations pertinentes pour une classification plus précise. Cela nous a permis de comprendre que des algorithmes de recommandation tels que ceux utilisés par Netflix, YouTube, etc. Avec plus de données personnelles, on aurait sûrement pu prédire avec plus de précisions les articles, au prix d'un usage commercial et peu éthique des données personnelles (ce que l'on peut comparer aux pratiques de Netflix, Google, etc. qui utilisent nos données personnelles, mais offrent des recommandations remarquables), tandis que sans données personnelles, nous avons pu faire des précisions correctes, mais moins ciblées, en prédisant notamment la famille d'article, une information moins précise (ce que l'on peut comparer aux pratiques ceux qui respectent la vie privée en n'utilisant pas ou peu de données d'utilisateurs, mais qui offrent alors des recommandations moins précises, comme DuckDuckGo).

Pour aller plus loin, il serait intéressant d'approfondir l'analyse en utilisant des données plus complètes et en explorant d'autres modèles d'apprentissage automatique plus avancés. De plus, il serait bénéfique de collecter des retours des étudiants pour évaluer l'efficacité de l'algorithme de recommandation et l'adapter en conséquence. Ces retours pourraient également nous aider à identifier les lacunes de notre approche actuelle et à proposer des améliorations.

En résumé, malgré les limites rencontrées, notre projet a permis d'approfondir notre compréhension des algorithmes de recommandation et de mettre en pratique nos connaissances. Nous croyons fermement que la recherche et le développement dans ce domaine peuvent engendrer des améliorations dans notre consommation au PIC'asso. Qui sait, un jour, nos travaux pourraient même être inclus dans une étude plus exhaustive visant à recommander de manière précise les articles du PIC'asso aux consommateurs.

Table des figures

SOURCES

1. Machine Learnia : sa chaîne YouTube nous a permis d'apprendre les algorithmes de machine learning, leur fonctionnement et comment les mettre en place. Cela nous a également permis de comprendre comment mettre en place un projet de machine learning.
2. Playlist Machine Learnia : la playlist de Machine Learnia que nous avons suivie
3. Statquest : sa chaîne YouTube nous de comprendre les notions en profondeur
4. Scikit-learn : Librairie de machine learning que nous avons utilisé, contenant beaucoup de documentation sur le fonctionnement des différents algorithmes de machine learning