Tiret profit des données d'activités de télécommunication pour optimiser la stratégie de rétention de la clientèle

Asma Draouil - Mohamed Samet - Mohamed Yassine KHALDI

Unsupervised Learning - Algèbre linéaire - Optimisation - Bases de données

ESPRIT School of Business

08 Février, 2025

Table des matières

1 Contexte général du projet **1**

1.1 Introduction 2

1.2 Problématique 2

1.3 Objectifs du Projet 2

1.4 Méthodologie Adoptée 2

2 Analyse des données **4**

2.1 Analyse univariée 5

2.1.1 **Objectif** 5

2.1.2 Préparation des données 5

2.1.3 Statistiques Descriptives 6

2.1.4 Visualisation des Distributions des Variables 7

2.2 Analyse bivariée 8

2.2.1 Objectif 8

2.2.2 Relation entre variables catégoriques 8

2.3 Analyse multivariée 9

2.3.1 Objectif 9

2.3.2 Encodage et Standardisation des Données 9

2.3.3 Analyse en Composantes Principales 10

2.3.4 Clustering 16

3 Intégration de l'algébre Linéaire dans l'analyse des données **21**

3.1 Introduction 22

3.2 Calcul des Distances Euclidiennes 22

3.3 Mise à Jour des Centroides dans K-means 22

3.4 Réduction Dimensionnelle par PCA : Valeurs Propres et Vecteurs Propres 23

3.5 Conclusion 24

4 Prédiction D'une serie temporelle **25**

4.1 Introduction 26

4.2 Contexte et objectif du projet : 26
4.3 Préparation des données : 26
4.3.1 Modélisation avec SARIMA : 31
4.3.2 Interprétation des Prédictions du Modèle SARIMA : 33
4.3.3 Transformation Logarithmique des Données : 35
4.3.4 Deuxième Transformation Logarithmique (Log1) : 38
4.4 Comparaison des modèles 42
4.5 Conclusion 43

5 Organisation des données dans une Base de données relationnelle 44
5.1 Introduction 45
5.2 Établissement d'un Diagramme de Classe : 45
5.3 Conclusion : 52

6 Optimisation 53
6.1 Introduction 54
6.2 Base de données en graph 55
6.3 Insights de fidélité 55
6.4 Conclusion 60

Conclusion générale 61

Table des figures

2.1 Scree plot 12
2.2 Scree plot 13
2.3 Corrélations PC1 PC2 14
2.4 Corrélations PC2 PC3 15
2.5 Corrélations PC1 PC3 15
2.6 Contribution des variables à PC1 16
2.7 Contribution des variables à PC2 16
2.8 Contribution des variables à PC3 16
2.9 Visualisation des clusters 17
2.10 Rapport de classification 18
2.11 Rapport de classification 18

4.1 exemple du ficher source 26
4.2 date 27
4.3 colonne target 27
4.4 visualisation des données : 28
4.5 Décomposition de notre série 29
4.6 test ADF 29
4.7 visualisation de l'act et pacf 30
4.8 Modelisation avec Sarima 32
4.9 Analyse du résidu 33
4.10 Visualisation du prédiction 34
4.11 Evaluation des performances du modele 35
4.12 Transformation logarithmique 36
4.13 Application du deuxieme test ADF 36
4.14 Application du deuxieme test ADF 37
4.15 Visualisation de l'ACF et PACF aprés diff 37
4.16 Visualisation de l'ACF et PACF aprés la deuxieme diff 38
4.17 Visualisation du deuxieme modele sarima 38

4.18 Visualisation du deuxieme transformation logarithmique 39
4.19 Visualisation du 3 eme modele Sarima 40
4.20 Visualisation du 4 eme modele Sarima 40
4.21 Visualisation du 4 eme modele Sarima 41
4.22 Visualisation du modele prophet 42

5.1 Diagramme de classe 45
5.2 Connexion entre Talend et PostgreSQL 46
5.3 Chargement de la table Client 47
5.4 Chargement de la table Recharges 48
5.5 Chargement de la table Contrat 48
5.6 Chargement de la table usage 48
5.7 Chargement de la table Client 49
5.8 Visualisation de la table Recharges 50
5.9 Visualisation de la table contrat 51
5.10 Visualisation de la table usages 52

6.1 Visualisation de la BD en graph 55
6.2 Slicing clients fidèles 56
6.3 Graph de localisation des entités 57
6.4 Application : Graph de localisation des entités 59

Liste des tableaux

4.1 Comparaison des modèles selon les métriques disponibles. 42

Chapitre 1

Contexte général du projet

Plan

1 Analyse univariée **5**

2 Analyse bivariée **8**

3 Analyse multivariée **9**

1.1 Introduction

Dans un monde de plus en plus régi par les données, la capacité à extraire des informations pertinentes à partir de vastes ensembles de données est devenue un enjeu majeur. Le machine learning, et en particulier l'apprentissage non supervisé, joue un rôle crucial dans cette dynamique en permettant de découvrir des structures cachées et des modèles sous-jacents sans recourir à des étiquettes prédéfinies.

Ce projet s'inscrit dans cette perspective et vise à appliquer des techniques avancées de machine learning pour analyser un ensemble de données clients comprenant des informations variées telles que l'activité des contrats, les usages des services, les revenus générés, ainsi que des indicateurs de résiliation.

1.2 Problématique

Dans un contexte où les entreprises collectent d'énormes volumes de données clients, comment peut-on exploiter efficacement ces données pour détecter des comportements, fidéliser la clientèle, optimiser la gestion des ressources et anticiper des tendances telles que la résiliation des contrats ?

1.3 Objectifs du Projet

L'objectif principal est d'explorer, d'analyser et de modéliser ces données afin d'identifier des patterns significatifs, de prédire des comportements futurs, et d'optimiser la gestion des ressources associées. Pour ce faire, plusieurs compétences interdisciplinaires seront mobilisées, englobant l'apprentissage non supervisé, l'algèbre linéaire, l'optimisation et la gestion de bases de données.

1.4 Méthodologie Adoptée

Le projet repose sur l'utilisation de plusieurs domaines et technologies clés qui permettent de traiter, analyser et modéliser les données efficacement. Les principales techniques et outils utilisés sont les suivants :

Analyse des données et exploration

- **Statistiques descriptives et exploratoires** : Techniques univariées et bivariées pour analyser la distribution des variables et leurs relations.
- **Outils** : Python (pandas, numpy), R, Excel pour la préparation et l'analyse exploratoire des

données.

Machine Learning (apprentissage non supervisé)

— **Segmentation des clients** : Utilisation d'algorithmes de clustering (comme K-means et DBSCAN) pour identifier des groupes de clients aux comportements similaires, sans étiquettes prédéfinies.

— **Réduction de la dimensionnalité** : Méthodes comme l'Analyse en Composantes Principales (PCA) pour simplifier les données tout en conservant l'essentiel de l'information.

— **Outils** : Python (scikit-learn, seaborn, matplotlib) pour implémenter les modèles de machine learning et visualiser les résultats.

Modélisation des séries temporelles

— **Prédiction des comportements futurs** : Utilisation de modèles statistiques comme ARIMA et des modèles plus avancés tels que les réseaux neuronaux récurrents (RNN) pour prédire le churn et d'autres comportements clients sur des périodes futures.

— **Outils** : Python (statsmodels, TensorFlow ou Keras pour les RNN), R.

Optimisation des modèles

—

Chapitre 2

Analyse des données

Plan

1 **Introduction** 22

2 **Calcul des Distances Euclidiennes** 22

3 **Mise à Jour des Centroides dans K-means** 22

4 **Réduction Dimensionnelle par PCA : Valeurs Propres et Vecteurs Propres** 23

5 **Conclusion** 24

2.1 Analyse univariée

2.1.1 Objectif

L'analyse univariée vise à comprendre les distributions des variables individuelles dans le jeu de données. Cette étape permet de détecter les anomalies, de décrire les tendances principales et de fournir un aperçu des caractéristiques statistiques de chaque variable pour orienter les analyses ultérieures.

2.1.2 Préparation des données

Les trois jeux de données distincts (juillet 2022, novembre 2022, et février 2023) ont été concaténés pour former un jeu de données unique sur lequel toutes les analyses seront effectuées. Cet ensemble final contient **708 908** lignes et **21** colonnes. Nous travaillons sur un total de **19** variables numériques et **2** variables catégoriques.

Aperçu des premières lignes du jeu de données

Une exploration préliminaire des données avec `data.shape` et `data.info()` a permis d'identifier les colonnes avec des valeurs manquantes et d'analyser les types de données pour guider les étapes de préparation et d'analyse.

2.1.2.1 Identification des Doublons

La vérification de l'existence de doublons a permis d'en identifier 3, qui ont été supprimés afin de garantir l'unicité des enregistrements dans le jeu de données.

2.1.2.2 Gestion des Valeurs Manquantes

Le jeu de données contient des valeurs manquantes dans plusieurs colonnes. La stratégie adoptée est la suivante :

— Suppression des colonnes avec plus de 30 % de valeurs manquantes, car elles apportent peu d'informations exploitables : `total_rev_sos`, `usage_op1`, `usage_op2`, `usage_op3`, `total_rev_option`.

— Suppression de la colonne `co_id` qui n'apporte pas de valeur analytique directe.

— Les colonnes catégoriques `entity_code` et `entity_type_name` ont été imputées avec leur mode respectif.

— Suppression des lignes contenant des NaN dans `activation_date` et `flag_churn`.

Pour les autres colonnes numériques : Division des colonnes selon leur type :

— Colonnes représentant des activités mesurables : `total_nb_recharge`, `total_recharge`, `total_u_data`, `total_u_out`, `total_u_in`, `nb_cont_out`, `nb_cont_in`, `nb_cell_visite_out`, `nb_cell_visite_in`.

Hypothèse : Une valeur manquante indique une absence d'activité, donc ces colonnes ont été remplies par 0.

— Colonnes représentant des états ou des comptes : `nbr_contrat`, `nbr_actif`.

Hypothèse : Ces colonnes ont été imputées par la médiane pour limiter l'impact des valeurs extrêmes.

2.1.2.3 Gestion des Valeurs Aberrantes

Des boxplots ont été utilisés pour détecter les valeurs aberrantes dans les variables numériques. Les valeurs situées au-delà de l'intervalle interquartile (IQR) pour certaines colonnes, telles que `total_nb_recharge` et `total_u_data`, ont été identifiées comme des outliers.

Ces anomalies, pouvant influer sur l'analyse statistique et les modèles prédictifs, ont été traitées en remplaçant les valeurs extrêmes par des seuils calculés selon la formule :

$$\textbf{Valeur_supérieure} = \textbf{Q3} + \textbf{1.5 IQR}$$

Cela permet de limiter l'influence disproportionnée de ces valeurs tout en conservant la variabilité des données.

Les graphiques suivants illustrent les anomalies détectées dans ces colonnes :

2.1.3 Statistiques Descriptives

Des statistiques descriptives ont été calculées pour les variables numériques et catégoriques afin de résumer les caractéristiques essentielles du jeu de données.

Pour les **variables numériques**, les mesures incluent la moyenne, la médiane, l'écart-type, la variance, les valeurs minimales et maximales, ainsi que les quartiles.

Pour les **variables catégoriques**, une analyse de fréquence a été réalisée afin de comprendre la répartition des catégories dans le jeu de données.

2.1.4 Visualisation des Distributions des Variables

Nous avons exploré les distributions des **variables numériques** en générant des histogrammes pour des variables clés telles que **total_recharge**, **total_u_data**, et **total_u_out**. Ces visualisations nous ont permis d'observer des comportements typiques et des valeurs extrêmes.

Les histogrammes des variables **total_nb_recharge** et **total_recharge** montrent une forte concentration autour de faibles valeurs, suivie d'une diminution rapide. Quelques pics indiquent des utilisateurs effectuant un nombre élevé de recharges, révélant un comportement dominant de faible activité et des cas isolés d'activité intensive.

Les variables **total_u_data**, **total_u_out**, et **total_u_in** présentent des distributions fortement asymétriques à droite. Cela indique qu'un petit groupe de clients consomme des volumes élevés de données ou utilise intensivement les services d'appel.

Pour les variables liées aux contacts (**nb_cont_out**, **nb_cont_in**) et aux visites d'antennes (**nb_cell_visite_out**, **nb_cell_visite_in**), nous avons noté des distributions avec une prédominance de faibles valeurs, reflétant un usage modéré pour la majorité des utilisateurs.

Les variables **nbr_contrat** et **nbr_actif** montrent une dispersion importante, mais les valeurs faibles sont les plus fréquentes, ce qui indique que la plupart des clients détiennent peu de contrats.

La variable **flag_churn** est fortement déséquilibrée, avec une majorité de clients n'ayant pas résilié leur contrat (valeur 0).

Pour les variables qualitatives, nous avons utilisé des diagrammes en barres :

Pour la variable **entity_code**, un histogramme montre que la catégorie **TRA001** domine largement avec environ 80 000 occurrences. Les catégories suivantes, telles que **MON0104** et **TUNAER04**, comptent entre 30 000 et 40 000 observations. Ces données mettent en évidence une distribution asymétrique où quelques codes sont très fréquents, tandis que d'autres restent rares.

Concernant la variable **entity_type_name**, la catégorie **INDIRECT** est majoritaire avec environ 250 000 observations, suivie par **AGENCE TRADE** avec près de 200 000. Les catégories restantes, comme **FRANCHISE** et **BOUTIQUE**, présentent des fréquences décroissantes, illustrant une structure hiérarchisée des entités.

2.2 Analyse bivariée

2.2.1 Objectif

L'analyse bivariée a pour objectif d'examiner les relations entre deux variables du jeu de données. Cette étape permet d'identifier d'éventuelles dépendances, corrélations ou interactions entre les variables, et de tester des hypothèses concernant ces relations. En explorant ces associations, l'analyse bivariée fournit des informations cruciales pour comprendre comment les variables interagissent entre elles, ce qui peut orienter les choix d'algorithmes et les stratégies d'analyse pour les étapes suivantes.

2.2.2 Relation entre variables catégoriques

Dans cette sous-section, on a analysé la relation entre les variables **entity_type_name** et **entity_code** pour examiner la dépendance entre ces deux attributs catégoriques. Une table de contingence a été construite afin de répertorier les fréquences d'apparition des combinaisons de valeurs des deux variables.

Ensuite, un **test du Chi² d'indépendance** a été appliqué sur le tableau de contingence afin d'évaluer l'existence d'une relation statistiquement significative entre ces variables. Ce test permet de vérifier si les occurrences observées diffèrent significativement des fréquences attendues en cas d'indépendance.

Une p-value inférieure à 0,05 indiquerait une relation statistiquement significative entre `entity_type_name` et `entity_code`, suggérant que le type d'entité pourrait être lié aux codes d'entité utilisés.

Pour visualiser cette relation, un **graphique à barres empilées** a été réalisé en se concentrant sur les 10 codes d'entité les plus fréquents. Ce graphique montre la répartition des occurrences de chaque type d'entité (entity_type_name) par rapport aux codes (entity_code), facilitant ainsi la compréhension des liens entre ces deux variables catégoriques.

Voici le résultat de la visualisation :

2.2.2.1 Relation entre une variable numérique et une variable catégorique

La relation entre **total_nb_recharge** et **entity_type_name** a été explorée à l'aide d'un boxplot, mettant en évidence des variations dans les recharges selon le type d'entité. Un test ANOVA a confirmé l'existence d'une différence significative entre les moyennes des recharges

pour les différentes catégories, indiquant que le type d'entité influence le nombre total de recharges.

2.2.2.2 Relation entre deux variables numériques

L'analyse de la relation entre **total_recharge** et **total_nb_recharge** a révélé une corrélation positive significative avec un coefficient de corrélation de Pearson de **0,61**. Cette relation indique qu'une augmentation du montant total des recharges est associée à une augmentation du nombre total de recharges. Le scatter plot illustre cette tendance linéaire positive.

2.3 Analyse multivariée

2.3.1 Objectif

L'analyse multivariée a pour objectif d'explorer les relations complexes entre plusieurs variables simultanément, afin d'identifier des patterns, des groupes ou des structures sous-jacentes dans le jeu de données. Cette approche inclut des techniques telles que le clustering, qui permet de segmenter les données en groupes homogènes, et le profiling, qui consiste à analyser les caractéristiques distinctives de chaque groupe. L'analyse multivariée est essentielle pour comprendre les interactions entre les variables, effectuer une segmentation fine des données et construire des modèles prédictifs. Elle permet ainsi d'extraire des insights significatifs qui orientent les décisions stratégiques et l'optimisation des processus.

2.3.2 Encodage et Standardisation des Données

2.3.2.1 Encodage des données catégoriques

Dans cette étape, nous avons transformé les variables catégoriques en données numériques pour les rendre exploitables par les algorithmes d'apprentissage automatique. Étant donné la taille importante du jeu de données et le nombre élevé de catégories pour certaines variables qualitatives comme **entity_code** et **entity_type_name**, nous avons utilisé la technique d'encodage par fréquence (frequency encoding). Cette méthode consiste à remplacer chaque catégorie par sa fréquence d'apparition dans la colonne correspondante.

L'encodage par fréquence a été choisi afin de :

— Réduire la complexité du modèle en évitant l'ajout de nombreuses colonnes supplémentaires

comme dans l'encodage one-hot, ce qui aurait considérablement augmenté la dimensionnalité du jeu de données.

— Préserver la structure des données tout en minimisant l'impact sur la mémoire et les performances.

Après encodage, les colonnes d'origine ont été supprimées pour éviter la redondance. Cette approche garantit une meilleure optimisation du traitement des données tout en maintenant une représentation pertinente des variables nominales.

2.3.2.2 Standardisation

Ensuite, nous utilisons la classe 'StandardScaler' depuis le module 'preprocessing' de la bibliothèque 'scikit-learn' pour standardiser les données en supprimant la moyenne et en mettant à l'échelle à l'unité de variance. Cette méthode calcule d'abord la moyenne et l'écart-type pour chaque caractéristique de 'x' (c'est l'étape de "fit"), puis transforme les données en soustrayant la moyenne et en divisant par l'écart-type (c'est l'étape de "transform"). Le résultat est stocké dans 'x scaled'. En résumé, ce code standardise les données 'x' de sorte que chaque caractéristique ait une moyenne de 0 et un écart-type de 1. Cela est souvent fait pour préparer les données pour des algorithmes d'apprentissage automatique qui fonctionnent mieux lorsque les données sont centrées et mises à l'échelle.

Cette standardisation est nécessaire pour assurer un clustering équilibré, où chaque dimension contribue équitablement à la formation des groupes. Les variables standardisées ont été utilisées directement pour la segmentation des clients.

2.3.3 Analyse en Composantes Principales

'PCA' est une technique utilisée pour réduire la dimensionnalité des données tout en conservant autant de variance que possible. Elle est souvent utilisée pour visualiser des données de haute dimension ou pour réduire le bruit dans les données. Nous utilisons la classe 'PCA' (Analyse en Composantes Principales) depuis le module 'decomposition' de la bibliothèque 'scikit-learn' pour réduire la dimensionnalité des données standardisées 'x scaled'. Le modèle PCA s'ajuste aux données (c'est l'étape de "fit") et les transforme en projetant les données originales dans le nouvel espace des composantes principales (l'étape de "transform"). Le résultat est stocké dans 'X pca'.

Nous affichons la variance expliquée par chaque composante principale. La propriété ‘explained variance ‘ de l'objet ‘pca‘ contient un tableau indiquant la quantité de variance expliquée par chaque composante principale. Différentes mesures liées à l'ACP ont été calculées et affichées, notamment les valeurs propres ajustées, les valeurs singulières, et les ratios de variance expliquée. Ces mesures aident à comprendre l'importance de chaque composante principale dans la représentation des données.

```
Variance expliquée par chaque composante :
 [5.3936582  1.96212524 1.36710504 1.09248856 0.97381745 0.78027051
 0.49446125 0.42731572 0.39527244 0.38172139 0.26639608 0.23196798
 0.15091843 0.08250145]
```

Différentes mesures liées à l'ACP ont été calculées et affichées, notamment les valeurs propres ajustées(eigval = (n-1)/n * pca.explained variance), les valeurs singulières (print(pca.singular values **2 / n)), et les ratios de variance expliquée (pca.explained variance ratio). Ces mesures aident à comprendre l'importance de chaque composante principale dans la représentation des données. En résumé, ces bouts de code fournissent différentes mesures liées à l'ACP : les valeurs propres ajustées, les valeurs singulières au carré ajustées, et les ratios de variance expliquée. Ces informations sont utiles pour comprendre la structure des données et l'importance relative de chaque composante principale. Il nous est avéré que La première composante accapare 38.5% de l'information disponible et ensemble, les deux premières composantes surpassent la moitié des informations.

2.3.3.1 Détermination du nombre de facteur à retenir

Nous avons expliqué la variance cumulée expliquée par les composantes principales ; Cela montre comment la variance totale des données est expliquée à mesure que le nombre de composantes principales augmente. Plusieurs seuils de variance expliquée (62%, 70%, 80%, et 90%) ont été définis pour évaluer à quel point les composantes principales expliquent la variance totale des données. Un graphique a été tracé pour visualiser la variance cumulée expliquée en fonction du nombre de composantes principales. Ce graphique montre comment l'ajout de composantes supplémentaires contribue à l'explication de la variance totale.Les points où la courbe bleue croise les lignes de seuil sont annotés avec le pourcentage de variance expliquée et le nombre de composantes nécessaires pour atteindre ce seuil. Par exemple : 62% de la variance

est expliquée par 3 composantes, 70 de la variance est expliquée par 4 composantes, 80% de la variance est expliquée par 6 composantes, 90% de la variance est expliquée par 9 composantes.

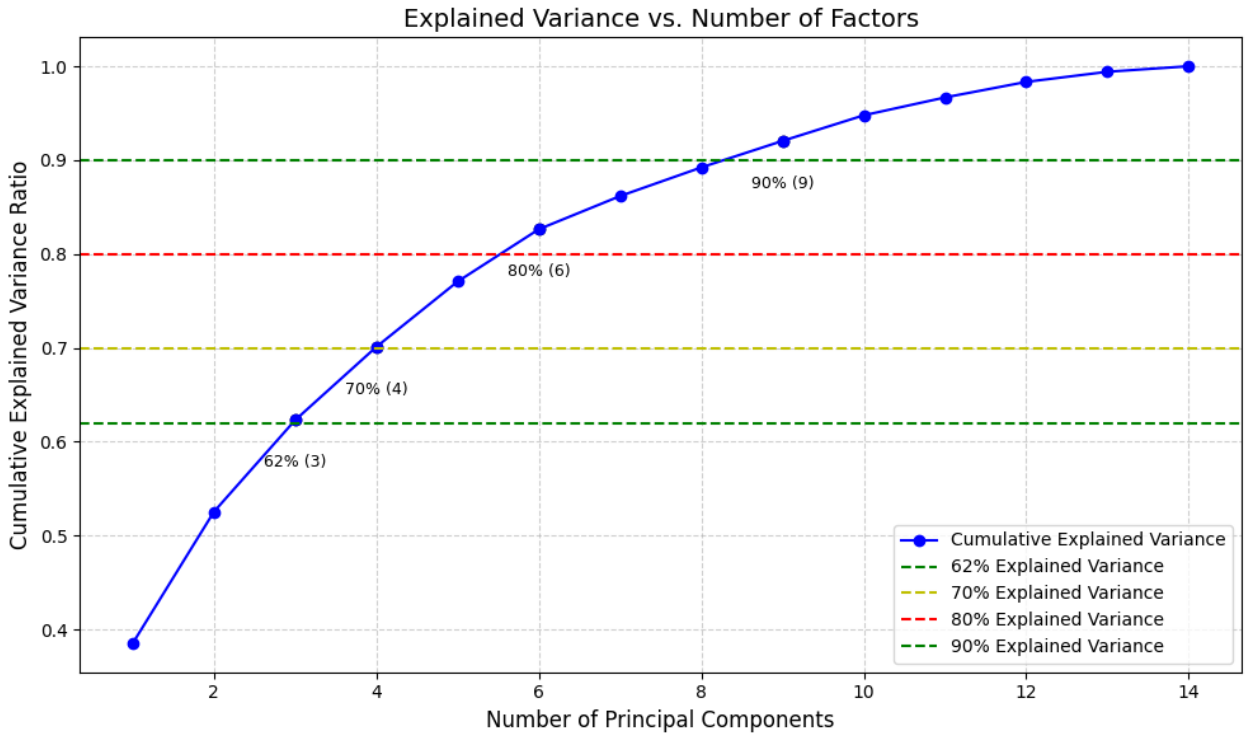

Figure 2.1 : Scree plot

Toutefois, L'approche Elbow est un outil utilisé pour identifier le point de coude dans une courbe représentant les valeurs propres des composantes principales. Le point de coude est l'endroit où la diminution de la variance expliquée commence à ralentir, indiquant que l'ajout de composantes supplémentaires apporte peu de bénéfices.

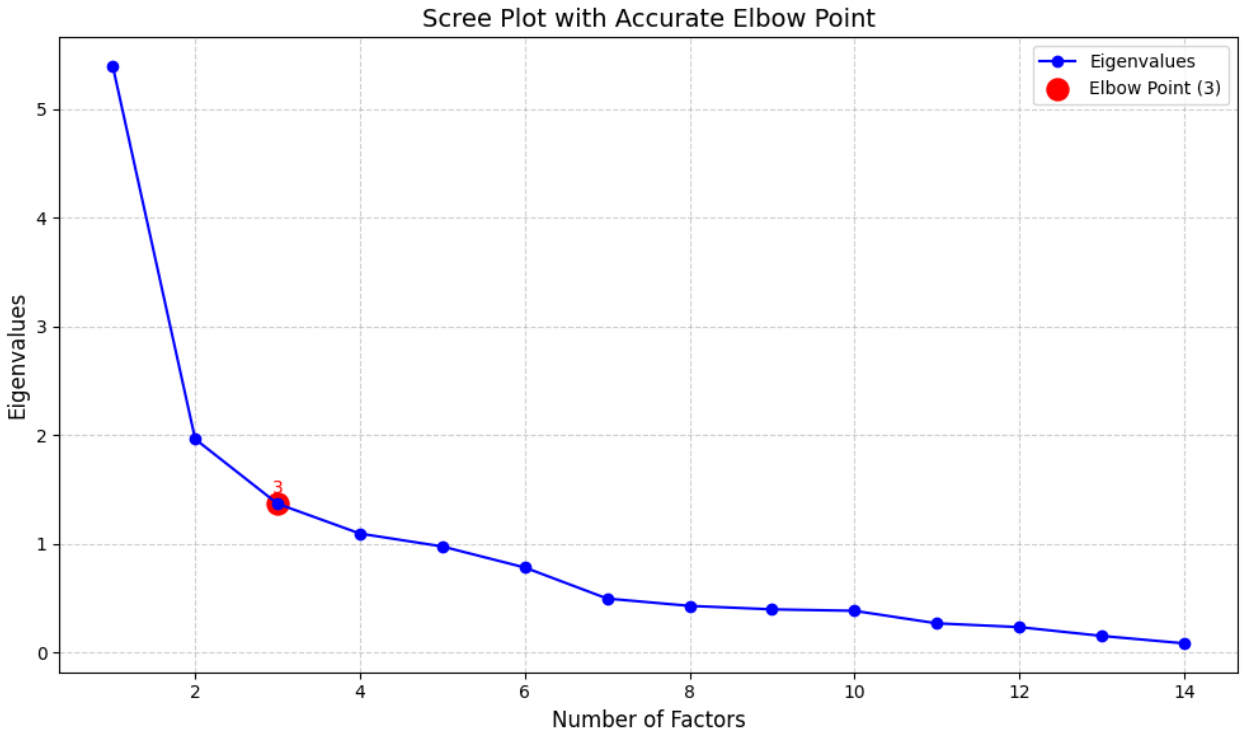

FIGURE 2.2 : Scree plot

Sur le graphique, le point de coude détecté a été mis en évidence pour le rendre facilement identifiable. Cela aide à déterminer visuellement le nombre optimal de composantes à retenir. Nous avons opté pour l'identification du point de coude dans un "scree plot" pour déterminer le nombre optimal de composantes principales à conserver dans notre analyse en composantes principales (ACP). Ainsi, la valeur de 3 sera sélectionnée.

2.3.3.2 Cercles de corrélations

Le graph ci dessous est utilisé pour visualiser les relations entre les variables originales et les deux premières composantes principales à travers un graphique appelé "cercle des corrélations". Les deux premières composantes principales sont extraites de l'analyse en composantes principales (ACP) effectuée précédemment. Ces composantes sont les plus importantes car elles expliquent la plus grande partie de la variance dans les données. Les coefficients des deux premières composantes principales sont obtenus. Ces coefficients représentent la contribution de chaque variable originale à ces composantes principales.

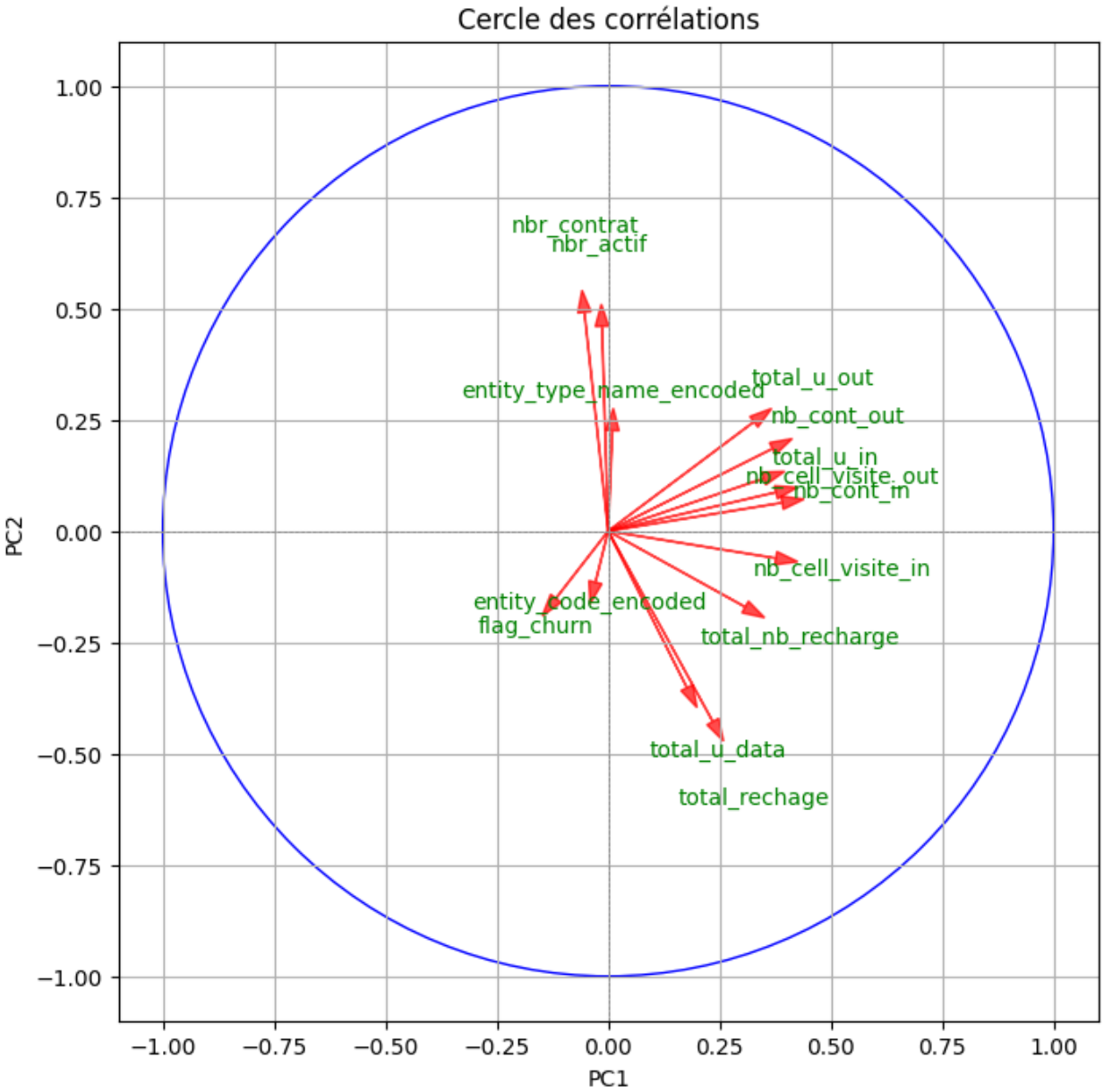

Figure 2.3 : Corrélations PC1 PC2

Un cercle de rayon 1 est tracé pour servir de référence visuelle représentant les limites de l'indice de corrélation compris entre -1 et 1. Ce cercle aide à visualiser la force et la direction des relations entre les variables et les composantes principales. Pour chaque variable originale, une flèche est tracée à partir de l'origine vers un point déterminé par les coefficients de la variable pour les deux premières composantes principales. Le même exercice a été réalisé entre les composantes PC2-PC3 et PC1-PC3.

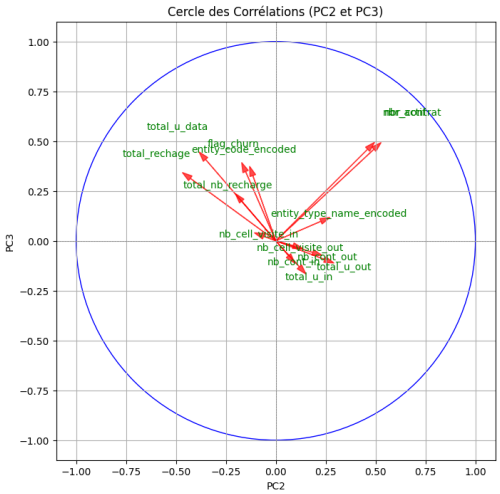

Figure 2.4 : Corrélations PC2 PC3

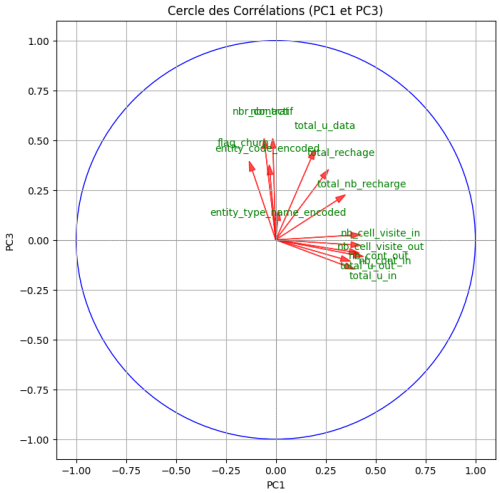

Figure 2.5 : Corrélations PC1 PC3

Dans notre analyse, nous avons procédé en plusieurs étapes pour comprendre la structure sous-jacente de nos données à l'aide de l'Analyse en Composantes Principales (ACP). Tout d'abord, nous avons standardisé les données pour garantir que chaque variable contribue de manière égale à l'analyse. Ensuite, nous avons appliqué l'ACP pour réduire la dimensionnalité des données tout en conservant autant de variance que possible. Nous avons calculé les valeurs propres et les vecteurs propres pour déterminer les composantes principales, et avons visualisé la variance expliquée par chaque composante à travers un scree plot. Pour mieux interpréter les résultats, nous avons tracé un cercle des corrélations, qui montre comment chaque variable originale est corrélée avec les deux premières composantes principales. Enfin, nous avons calculé la qualité de représentation (COS^2) et la contribution (CTR) de chaque variable pour évaluer l'importance de chaque variable dans la construction des composantes principales. L'objectif

de cette analyse était de simplifier la complexité des données tout en identifiant les variables les plus influentes, ce qui permet une meilleure compréhension et interprétation des relations sous-jacentes dans le jeu de données. Ci dessous se trouvent des visualisations représentant les contributions des Variables aux Composantes Principales.

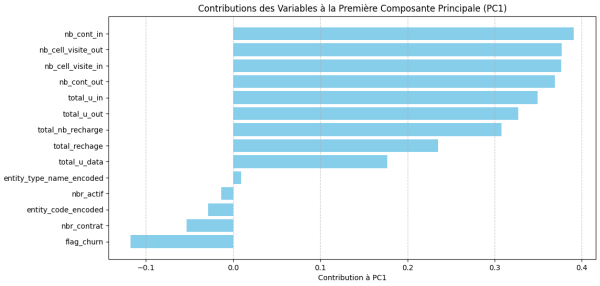

FIGURE 2.6 : Contribution des variables à PC1

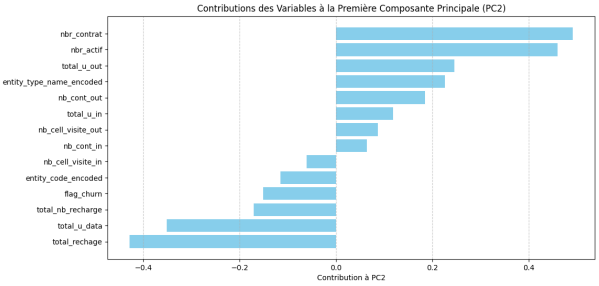

FIGURE 2.7 : Contribution des variables à PC2

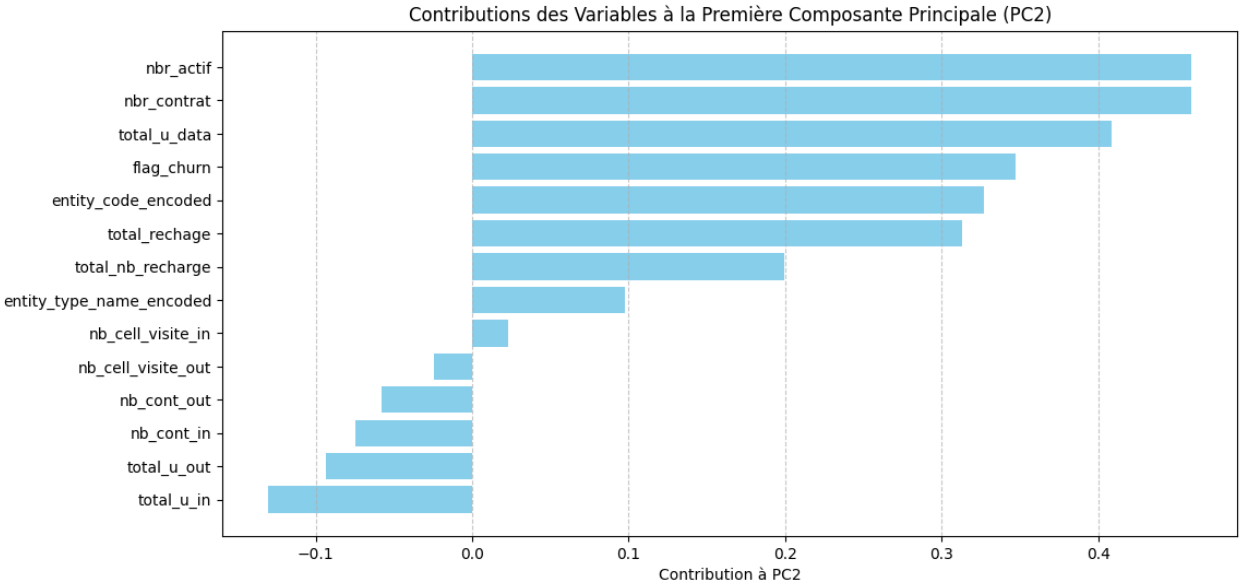

FIGURE 2.8 : Contribution des variables à PC3

2.3.4 Clustering

L'objectif de cette analyse était de segmenter les données en groupes homogènes basés sur les caractéristiques des observations. En utilisant K-means, nous avons pu identifier des structures sous-jacentes dans les données et regrouper les observations similaires ensemble. Cela permet une meilleure compréhension des relations entre les différentes observations et peut être utilisé pour des analyses ultérieures, telles que la personnalisation de stratégies ou l'identification de comportements spécifiques au sein de chaque cluster.

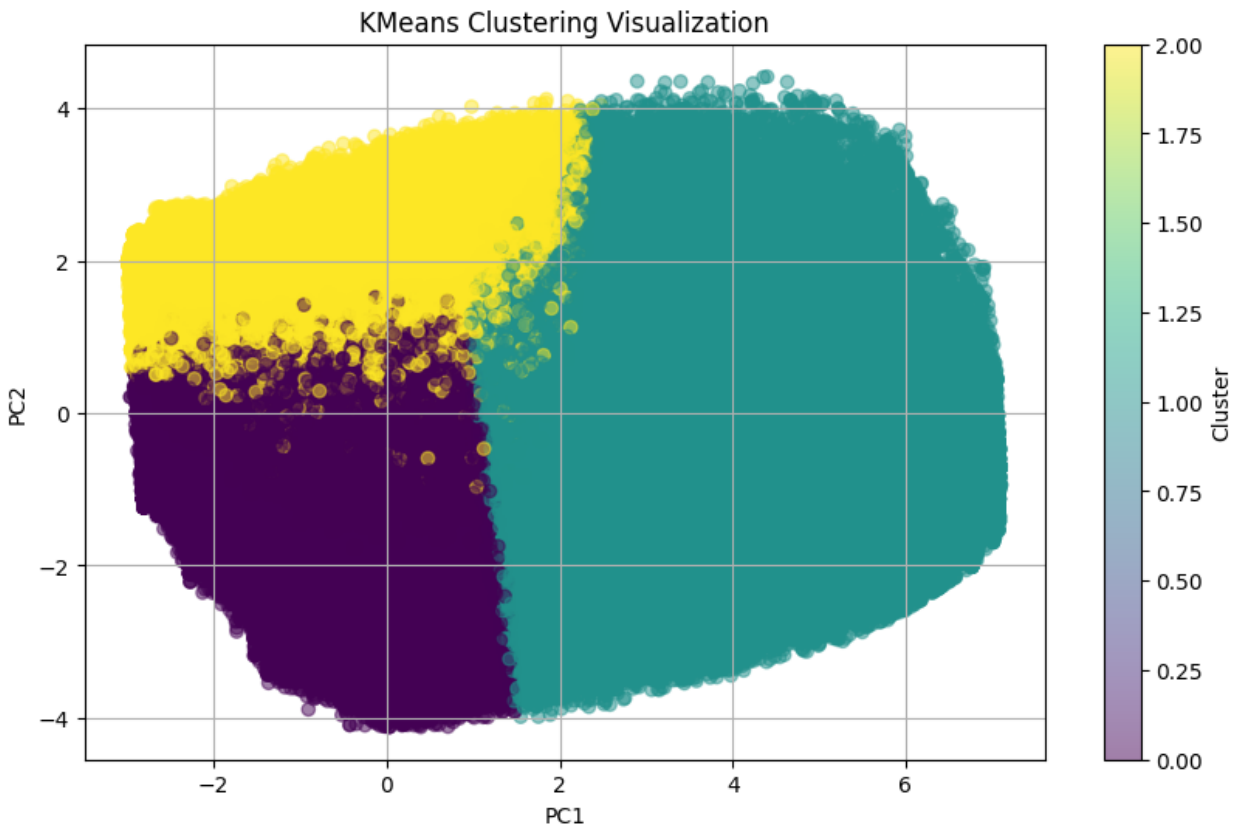

Figure 2.9 : Visualisation des clusters

2.3.4.1 Profiling

Dans cette analyse, nous avons procédé à la construction et à l'évaluation d'un modèle de classification en utilisant un arbre de décision. L'objectif de cette analyse était d'évaluer la capacité d'un modèle d'arbre de décision à prédire correctement les clusters auxquels appartiennent les observations, en utilisant les caractéristiques disponibles. Cela permettra facilement à l'entité de télécommunication de classer de nouveaux clients ayant de nouvelles entrées.

```
[495]
...                precision    recall  f1-score   support

           0       0.98      0.98      0.98     75996
           1       0.94      0.94      0.94     35328
           2       0.95      0.96      0.96     30457

    accuracy                           0.96    141781
   macro avg       0.96      0.96      0.96    141781
weighted avg       0.96      0.96      0.96    141781
```

Figure 2.10 : Rapport de classification

Ensuite, grâce à la heatmap ci-dessous, nous avons visualisé les caractéristiques moyennes de chaque cluster, ce qui nous permet de repérer les différences clés entre les groupes.

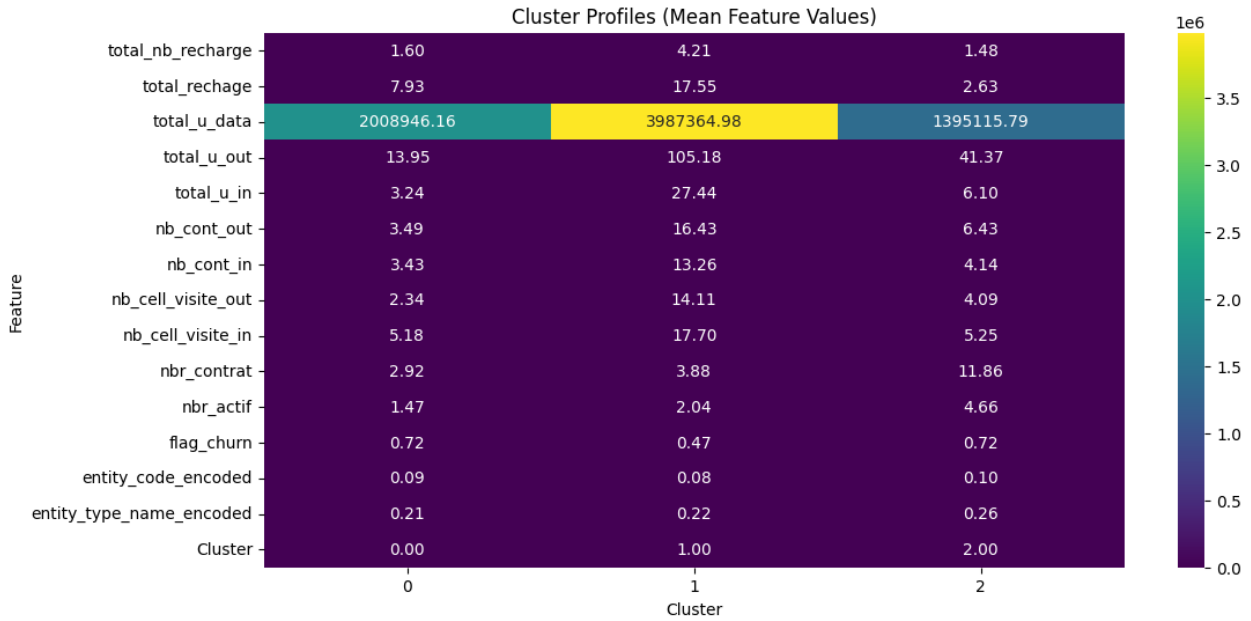

Figure 2.11 : Rapport de classification

2.3.4.2 Interprétations

Cluster 0 : Utilisateurs à Faible Engagement Ce cluster regroupe des clients qui utilisent très peu les services de l'opérateur. Le nombre de recharges ainsi que le montant total rechargé sont faibles, ce qui indique une faible rentabilité de ces clients. Leur consommation de données est modérée, mais ils effectuent très peu d'appels et ont peu d'interactions avec

leurs contacts. De plus, le taux de résiliation est élevé, suggérant que de nombreux clients de ce segment ont déjà quitté l'opérateur ou sont sur le point de le faire. Ce groupe peut inclure des utilisateurs occasionnels ou des clients insatisfaits qui envisagent un changement d'opérateur.

Cluster 1 : Utilisateurs à Forte Valeur & Engagement Élevé Ce cluster représente les clients les plus actifs et les plus rentables. Ils effectuent un nombre élevé de recharges et génèrent probablement des revenus supplémentaires grâce aux services optionnels. Leur consommation de données est la plus élevée parmi tous les clusters, et ils interagissent fréquemment avec leurs contacts. Leur engagement est renforcé par une forte connectivité aux antennes, suggérant une utilisation continue du réseau. De plus, leur faible taux de résiliation montre qu'ils sont plus fidèles à l'opérateur. Ce segment est stratégique pour les programmes de fidélisation et les offres premium.

Cluster 2 : Utilisateurs Multi-Contrats à Usage Modéré Ce groupe est composé de clients possédant plusieurs contrats, mais leur consommation individuelle reste modérée. Leur volume d'appels et leur utilisation de données sont supérieurs au Cluster 0 mais bien en dessous du Cluster 1. Le taux de résiliation est élevé, ce qui signifie qu'ils ont tendance à abandonner certaines lignes. La présence de nombreux contrats actifs et résiliés laisse penser qu'il s'agit peut-être de clients professionnels ou d'entreprises gérant plusieurs abonnements. Ce cluster nécessite des stratégies adaptées pour limiter la perte de clients et maximiser l'engagement.

2.3.4.3 Conclusions & Recommandations

Cluster 0 (Utilisateurs à Faible Engagement)

- Mettre en place des promotions incitant à l'utilisation des services.
- Offrir des bonus de données ou des réductions sur les recharges pour les fidéliser.

Cluster 1 (Utilisateurs à Forte Valeur)

- Cibler ces clients avec des offres VIP et des avantages exclusifs.
- Proposer des services premium pour maximiser leur satisfaction et leur fidélité.

Cluster 2 (Utilisateurs Multi-Contrats)

- Identifier les raisons des résiliations et proposer des solutions adaptées.

— Mettre en avant des offres groupées pour fidéliser ces clients multi-lignes.

Ces recommandations permettront d'optimiser la gestion des clients selon leur profil et d'améliorer la rétention et la rentabilité de chaque segment.

Chapitre 3

Intégration de l'algébre Linéaire dans l'analyse des données

Plan

1 Introduction 26

2 Contexte et objectif du projet : 26

3 Préparation des données : 26

4 Comparaison des modèles 42

5 Conclusion 43

3.1 Introduction

L'algèbre linéaire joue un rôle clé dans l'analyse des données et le machine learning, en offrant des outils pour manipuler et interpréter des ensembles de données complexes. Ce chapitre met en avant son application à travers le calcul des distances euclidiennes pour mesurer la similarité, la mise à jour des centroides dans le clustering K-means, et la réduction de dimensionnalité via l'Analyse en Composantes Principales (PCA). Ces concepts permettent d'optimiser l'analyse des données et d'améliorer la performance des modèles prédictifs.

3.2 Calcul des Distances Euclidiennes

Dans de nombreux algorithmes de machine learning, et en particulier dans le clustering avec K-means, la mesure de la similarité entre observations est essentielle. Une distance couramment utilisée est la distance euclidienne, qui correspond à la norme L_2 d'un vecteur.

Soient deux points $A = (x_1, x_2, \ldots, x_n)$ et $B = (y_1, y_2, \ldots, y_n)$ dans un espace à n dimensions, la distance euclidienne entre ces points est définie comme suit :

$$d(A, B) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

Cette formule, directement dérivée du théorème de Pythagore, permet de quantifier la ressemblance entre deux observations. Elle est particulièrement utile dans le clustering, où les points similaires doivent être regroupés.

Afin d'éviter que certaines variables ayant des échelles différentes ne dominent le calcul des distances, les données sont souvent standardisées avant d'appliquer cette métrique. Cela garantit que chaque variable contribue de manière équitable à la distance globale.

3.3 Mise à Jour des Centroides dans K-means

L'algorithme K-means repose sur un principe fondamental d'algèbre linéaire : le calcul de la moyenne vectorielle.

À chaque itération, après l'affectation des points à leurs clusters respectifs, le recalcul des centroides s'effectue en déterminant la moyenne de tous les vecteurs appartenant à un même cluster. Concrètement, pour un cluster donné, nous effectuons la somme de tous les points

(représentés comme des vecteurs) et nous divisons cette somme par le nombre total d'observations dans le cluster :

$$C_k = \frac{1}{N_k} \sum_{i=1}^{N_k} X_i$$

Où :

— C_k est le centroïde du cluster k,

— N_k est le nombre de points dans le cluster k,

— X_i est un vecteur appartenant à ce cluster.

Cette opération correspond au calcul de la moyenne arithmétique en algèbre linéaire, permettant d'obtenir un point central, ou "centre de gravité", du cluster. Cela minimise la somme des distances euclidiennes entre ce centroïde et les points du groupe.

Ce processus itératif, basé sur des concepts fondamentaux tels que la norme vectorielle et la moyenne, illustre parfaitement le lien entre la théorie mathématique et son application pratique dans le clustering.

3.4 Réduction Dimensionnelle par PCA : Valeurs Propres et Vecteurs Propres

L'Analyse en Composantes Principales (PCA) est une technique de réduction de dimensionnalité qui repose entièrement sur des concepts d'algèbre linéaire. Elle permet de simplifier un ensemble de données en identifiant les axes principaux qui capturent le plus de variance.

Le processus se déroule en trois étapes majeures :

Calcul de la Matrice de Covariance : la première étape consiste à calculer la matrice de covariance, qui mesure la relation entre les différentes variables. Si deux variables varient de manière similaire, leur covariance sera élevée, ce qui indique une redondance d'information.

Décomposition en Valeurs Propres et Vecteurs Propres : une fois la matrice de covariance obtenue, on procède à sa décomposition spectrale pour extraire les valeurs propres et les vecteurs propres.

— Les valeurs propres indiquent l'importance de chaque direction dans la variance totale des données. Une valeur propre élevée signifie qu'une composante principale explique une

part significative de la variance.

— Les vecteurs propres définissent les axes selon lesquels les données seront projetées. Ces axes sont perpendiculaires entre eux et représentent les directions optimales pour la réduction de dimension.

Projection dans un Espace de Dimension Réduite : en sélectionnant les vecteurs propres correspondant aux plus grandes valeurs propres, on projette les données dans un nouvel espace de dimension réduite. Cette transformation permet :

— de visualiser les données de manière plus simple (ex. projection en 2D ou 3D),

— d'éliminer le bruit et les redondances,

— d'améliorer la performance des modèles en conservant uniquement les dimensions essentielles.

3.5 Conclusion

L'algèbre linéaire est omniprésente dans l'analyse des données et le machine learning. La distance euclidienne est utilisée pour mesurer la similarité entre points, le calcul des centroides dans K-means repose sur la moyenne vectorielle, et la décomposition en valeurs propres dans la PCA permet d'extraire les axes les plus informatifs d'un jeu de données.

Ces concepts mathématiques jouent un rôle clé dans la structuration et l'interprétation des données, rendant possible des analyses avancées et optimisant les algorithmes de clustering et de réduction dimensionnelle.

Chapitre 4

Prédiction D'une serie temporelle

Plan

1 **Introduction** 45

2 **Établissement d'un Diagramme de Classe :** 45

3 **Conclusion :** 52

4.1 Introduction

Dans ce chapitre, nous nous intéressons à faire des prédctions en utilisant des modèles de séries temporelles. Contrairement aux approches globales qui cherchent à modéliser la tendance sur une période étendue, nous avons choisi d'analyser et de prédire chaque mois individuellement en appliquant des modèles SARIMA, SARIMAX et AutoSARIMA.

4.2 Contexte et objectif du projet :

Ce projet vise à analyser et prédire des séries temporelles horaires à partir de trois fichiers de données correspondant à différents mois (février, juillet et novembre). Ces fichiers contiennent des données similaires, mais présentent des variations saisonnières et des tendances spécifiques. L'objectif est d'appliquer des techniques de préparation, de visualisation et de modélisation pour identifier des modèles prédictifs robustes et fiables. Le rapport est organisé en plusieurs sections allant de la préparation des données à l'évaluation des modèles et aux conclusions finales.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	co_id	activation_date	total_nb_rec	total_rechag	total_u_data	total_rev_option	total_rev_sos	total_u_out	total_u_in	usage_op1	usage_op2	usage_op3	nb_cont_out	nb_cont_in	nb_cell_visite_out	nb_cell_vis
2	41084377	2023-02-24 14:00:28	1	5	6,879882813	NULL	NULL	31,433	12,467	NULL	31,433	NULL	7	4	7	
3	569522	2023-02-03 17:51:54	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	40987858	2023-02-13 17:34:09	1	5	NULL	NULL	NULL	0,067	NULL	NULL	NULL	NULL	2	NULL	1	NULL
5	40843509	2023-02-03 09:43:32	2	2	179641,6309	NULL	NULL	6,8	15,016	NULL	NULL	2,4	4	6	1	
6	6418540	2023-02-24 02:32:15	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
7	5569804	2023-02-24 05:59:16	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	5548168	2023-02-22 09:20:34	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9	5389343	2023-02-28 01:59:08	NULL	NULL	6857228,496	NULL	NULL	324,766	275,349	NULL	NULL	51,133	37	67	39	
10	38313237	2023-02-28 08:08:50	NULL	NULL	10240	NULL	NULL	15,733	10,5	NULL	3,433	5	12	11	4	
11	40916483	2023-02-06 16:29:08	NULL	NULL	794040,9697	NULL	NULL	33,316	16,133	NULL	NULL	12,883	16	8	8	
12	41053859	2023-02-23 10:36:08	1	1	36688,49805	0.75630250	NULL	18,433	32,784	NULL	NULL	NULL	2	8	1	
13	16873023	2023-02-28 11:01:58	14	95	116370083,2	75.63025210	1,596638736	3326,1	808,566	NULL	923,067	938,933	68	70	95	
14	40796200	2023-02-11 13:31:51	NULL	NULL	9296066,946	NULL	NULL	376,935	362,617	NULL	70,934	258,234	34	47	37	
15	40604637	2023-02-11 10:43:28	NULL	NULL	11071423,17	NULL	NULL	384,867	128,1	NULL	NULL	259,834	25	25	36	
16	41015710	2023-02-17 16:59:07	1	1	NULL	NULL	NULL	96,401	36,716	NULL	NULL	32,967	13	7	4	
17	40990951	2023-02-28 14:33:01	3	3	8119556,835	2.02881183	NULL	139,067	140,734	NULL	NULL	NULL	12	13	7	
18	41015708	2023-02-17 16:57:36	NULL	NULL	NULL	NULL	NULL	3,4	2,4	NULL	NULL	0,2	3	3	1	
19	41037286	2023-02-21 10:54:11	NULL	NULL	1395135,066	NULL	NULL	6,017	12,817	NULL	NULL	5,617	4	4	3	
20	40982313	2023-02-13 16:35:04	NULL	NULL	NULL	NULL	NULL	20,3	43,933	NULL	NULL	8,4	11	31	6	
21	36206841	2023-02-28 01:04:25	1	2	7599050,172	NULL	NULL	2615,584	1039,216	NULL	391,267	882,367	177	210	137	
22	40796306	2023-02-01 17:19:02	2	2	1352712,229	0.84033613	NULL	20,683	10,167	NULL	NULL	0,15	8	13	10	
23	41016749	2023-02-20 09:13:42	NULL	NULL	10240	NULL	NULL	34,317	16,183	NULL	NULL	16,25	9	8	7	
24	41016752	2023-02-20 09:14:38	1	1	684986,5547	NULL	NULL	27,3	14,55	NULL	NULL	10,55	19	12	17	
25	40799699	2023-02-01 13:35:16	6	41	2922496	32.77310924	NULL	106,467	74,766	NULL	NULL	29,317	32	54	29	
26	40925679	2023-02-07 16:10:51	NULL	NULL	2746368	NULL	NULL	1,583	NULL	NULL	NULL	1,583	3	1	2	
27	41017183	2023-02-17 14:23:34	1	5	NULL	NULL	NULL	54,25	13,767	NULL	NULL	10,183	10	9	12	
28	40795979	2023-02-01 09:50:57	NULL	NULL	1189888	NULL	NULL	0	0,617	NULL	NULL	NULL	1	3	0	
29	40965566	2023-02-10 12:23:54	NULL	NULL	1189888	NULL	NULL	0,967	33	NULL	NULL	NULL	4	2	2	

Figure 4.1 : exemple du ficher source

4.3 Préparation des données :

Dans cette section, nous décrivons les étapes nécessaires pour préparer les données avant leur analyse et modélisation. Les fichiers fournis contiennent des données similaires pour trois mois différents (février, juillet et novembre).

Tout d'abord, la colonne de date a été convertie au format datetime pour faciliter les opérations temporelles.

```
[ ]
    # Conversion de la colonne date en datetime
    feb['activation_date'] = pd.to_datetime(feb['activation_date'])

    # Tri des données par date
    feb = feb.sort_values(by='activation_date')

[ ]  print(feb.activation_date.dtypes)

datetime64[ns]
```

Figure 4.2 : date

Ensuite, nous avons extrait et transformé les données pour les rendre prêtes à l'analyse. Un nouveau jeu de données a été créé, contenant uniquement deux colonnes principales : la colonne date, représentant le temps, et la colonne cible total_recharge, correspondant au total des recharges effectuées. Afin de simplifier l'analyse et de capturer les variations temporelles pertinentes, un regroupement des données a été effectué par heure. Cette agrégation permet de mieux observer les tendances et les comportements horaires dans les séries temporelles.

```
[ ] # Regrouper les données par heure et sommer les valeurs de total_re
    df_hourly = df.resample('H')['total_rechage'].sum()

    # Renommer les colonnes pour clarté
    df_hourly.columns = ['date_hour', 'total_rechage']
    df_hourly.head()
```

activation_date	total_rechage
2023-02-01 00:00:00	804.161000
2023-02-01 01:00:00	567.256000
2023-02-01 02:00:00	364.256000
2023-02-01 03:00:00	610.116002
2023-02-01 04:00:00	362.940002

dtype: float64

Figure 4.3 : colonne target

4.3.0.1 Visualisation des données :

Pour mieux comprendre la distribution et les tendances de notre série temporelle, nous avons réalisé une visualisation du total des recharges pour le mois de février. Ce graphique permet d'identifier les variations horaires, les éventuelles tendances globales, ainsi que les cycles ou saisonnalités présents dans les données. Cette étape est essentielle pour détecter visuellement les comportements anormaux ou les motifs récurrents qui guideront les étapes suivantes de l'analyse.

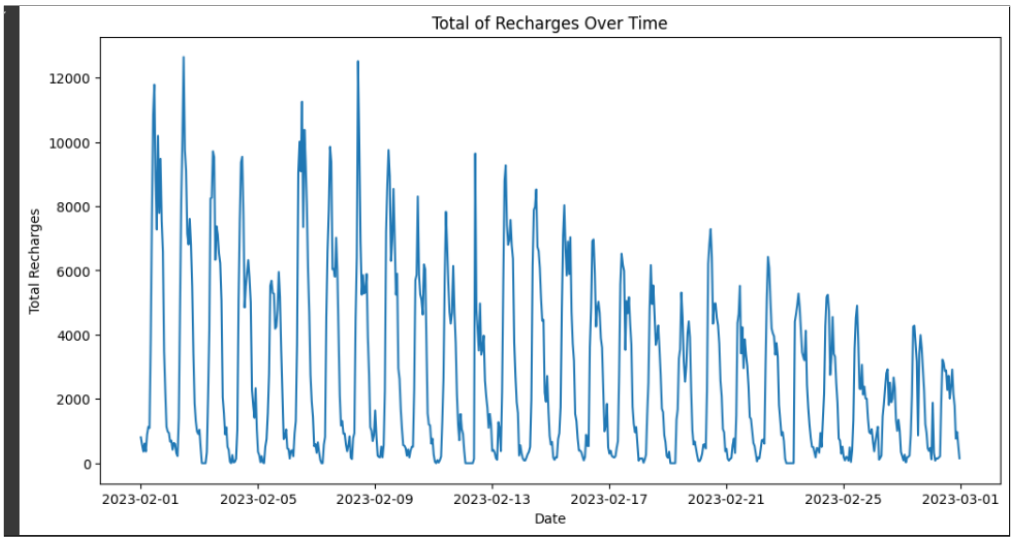

FIGURE 4.4 : visualisation des données :

Random Forest classifier :

4.3.0.2 Décomposition de la série temporelle :

Dans cette étape, nous avons procédé à la décomposition de notre série temporelle pour mieux comprendre ses composantes. La décomposition nous a permis de distinguer trois éléments principaux :

- `Tendance` : Reflétant l'évolution globale des recharges sur le mois de février.
- `Saisonnalité` : Montrant les motifs cycliques récurrents, typiques des comportements horaires ou quotidiens
- `Résidu` : Capturant les variations non expliquées, correspondant aux irrégularités ou bruits dans les données.

Cette analyse a été cruciale pour valider la présence de saisonnalité et de tendance dans les données, ce qui a orienté le choix du modèle prédictif à utiliser.

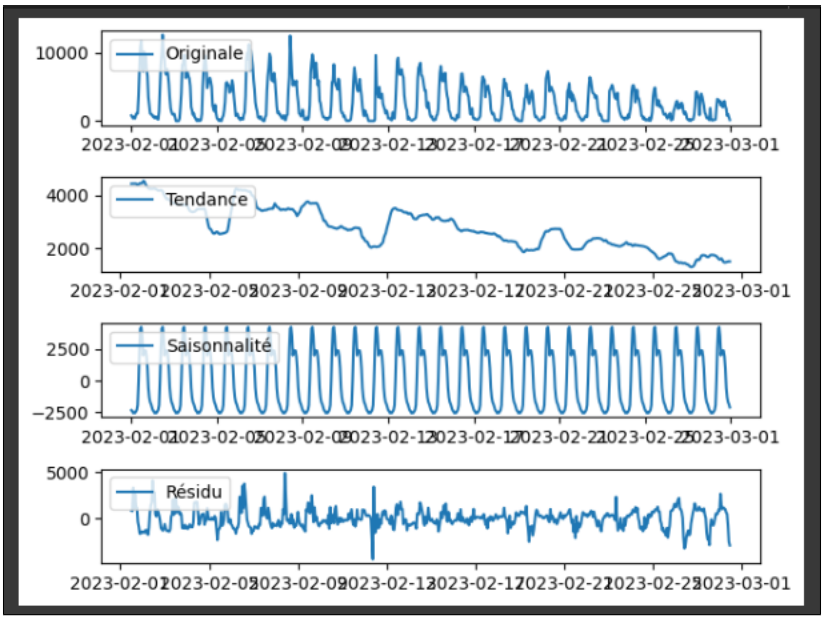

FIGURE 4.5 : Décomposition de notre série

4.3.0.3 Application d'un test ADF :

Pour évaluer si les données étaient stationnaires (c'est-à-dire si leurs propriétés statistiques restent constantes au fil du temps), nous avons appliqué le test ADF (Augmented Dickey-Fuller). Cette étape est cruciale pour déterminer si une différenciation supplémentaire est nécessaire avant de modéliser les données.

```python
from statsmodels.tsa.stattools import adfuller

# Application du test ADF
result = adfuller(df_hourly.dropna())  # Assurez-vous qu'il n'y a pas de valeurs NaN

# Extraction des résultats
print("Statistique ADF : {:.4f}".format(result[0]))
print("P-valeur : {:.4f}".format(result[1]))
print("Valeurs critiques :")
for key, value in result[4].items():
    print(f"  {key}: {value:.4f}")

# Interprétation
if result[1] <= 0.05:
    print("\nLa série est stationnaire (on rejette l'hypothèse nulle).")
else:
    print("\nLa série n'est pas stationnaire (on ne peut pas rejeter l'hypothèse nulle).")
```

```
Statistique ADF : -3.0582
P-valeur : 0.0298
Valeurs critiques :
  1%: -3.4404
  5%: -2.8660
  10%: -2.5691

La série est stationnaire (on rejette l'hypothèse nulle).
```

FIGURE 4.6 : test ADF

4.3.0.4 Visualisation de l'ACF et du PACF :

Après avoir décomposé et préparé les données, nous avons analysé les graphiques de l'autocorrélation (ACF) et de l'autocorrélation partielle (PACF).

ACF (Autocorrelation Function) : Ce graphique montre la corrélation entre les valeurs actuelles et leurs valeurs décalées sur différentes périodes. On observe des pics significatifs autour des multiples de 24, indiquant une forte saisonnalité horaire dans les données.

PACF (Partial Autocorrelation Function) : Ce graphique identifie les relations directes entre une observation et ses décalages en éliminant l'effet des autres décalages intermédiaires. Les premiers pics significatifs suggèrent les ordres possibles des paramètres AR (auto-régressifs) pour modéliser les données.

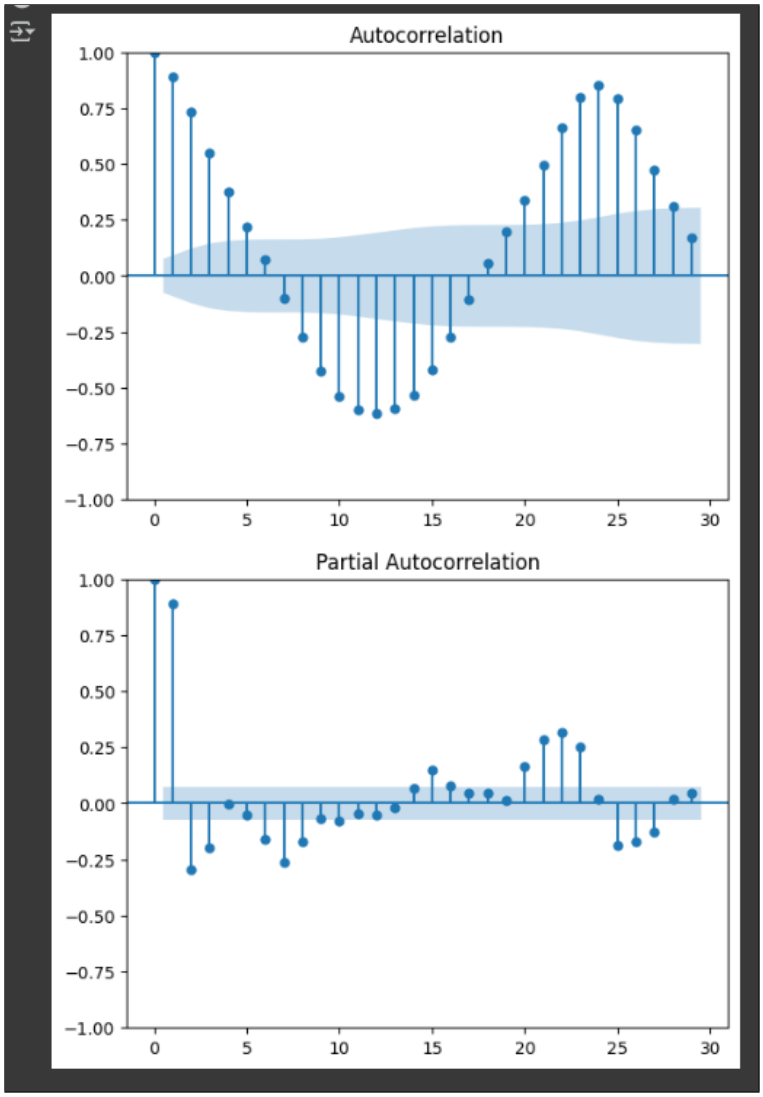

Figure 4.7 : visualisation de l'act et pacf

Ces visualisations ont été utilisées pour déterminer les paramètres p,q,P,Q de notre modèle SARIMA en tenant compte de la saisonnalité identifiée avec une période de 24 heures. Cette étape est cruciale pour construire un modèle prédictif robuste.

4.3.1 Modélisation avec SARIMA :

Suite à l'analyse des graphiques ACF et PACF, une saisonnalité horaire de période 24 heures a été identifiée dans les données. Pour capturer cette composante saisonnière ainsi que les relations auto-régressives et de moyenne mobile, nous avons utilisé un modèle SARIMA (Seasonal AutoRegressive Integrated Moving Average).

Les résultats ajustés montrent les coefficients estimés, leurs erreurs standards et leurs tests de significativité. Les statistiques telles que l'AIC (10947.692) et le BIC (10970.244) permettent d'évaluer la qualité du modèle. La faible valeur des erreurs (sigma^2) et la significativité des coefficients confirment la pertinence du modèle pour capturer les dynamiques temporelles du total des recharges.

Ce modèle SARIMA sera utilisé pour effectuer des prédictions et analyser les performances sur l'ensemble des données.

SARIMA

```python
from statsmodels.tsa.statespace.sarimax import SARIMAX

# Définir les paramètres SARIMA
p, d, q = 1, 0, 1  # Non saisonnier
P, D, Q, s = 1, 0, 1, 24  # Saisonnière

# Ajuster le modèle
model = SARIMAX(df_hourly, order=(p, d, q), seasonal_order=(P, D, Q, s))
results = model.fit()

# Résumé des résultats
print(results.summary())
```

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                      total_rechage   No. Observations:                  672
Model:             SARIMAX(1, 0, 1)x(1, 0, 1, 24)   Log Likelihood               -5468.846
Date:                            Fri, 07 Feb 2025   AIC                          10947.692
Time:                                    09:14:58   BIC                          10970.244
Sample:                                02-01-2023   HQIC                         10956.426
                                     - 02-28-2023
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.8688      0.019     46.674      0.000       0.832       0.905
ma.L1         -0.3690      0.028    -13.355      0.000      -0.423      -0.315
ar.S.L24       0.9912      0.002    583.545      0.000       0.988       0.995
ma.S.L24      -0.7910      0.021    -37.271      0.000      -0.833      -0.749
sigma2      6.098e+05   1.62e+04     37.655      0.000    5.78e+05    6.42e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):              1494.63
Prob(Q):                              0.99   Prob(JB):                         0.00
Heteroskedasticity (H):               0.36   Skew:                             0.62
Prob(H) (two-sided):                  0.00   Kurtosis:                        10.20
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

Figure 4.8 : Modelisation avec Sarima

4.3.1.1 Analyse des Résidus :

Afin de valider notre modèle SARIMA, nous avons analysé les résidus pour vérifier les hypothèses fondamentales. La figure ci-dessus présente quatre graphiques de diagnostic :

Résidus standardisés (en haut à gauche) : Les résidus sont répartis de manière homogène autour de zéro, ce qui indique une absence de tendance ou de structure non capturée par le modèle.

Histogramme avec densité estimée (en haut à droite) : La distribution des résidus est proche d'une loi normale (courbe verte), confirmant que les erreurs suivent une distribution normale, comme attendu.

Graphique Q-Q (en bas à gauche) : Les quantiles des résidus suivent approximativement la ligne rouge (diagonale), ce qui soutient l'hypothèse de normalité des erreurs.

Correlogramme (en bas à droite) : Les autocorrélations des résidus sont proches de zéro et se situent dans l'intervalle de confiance, indiquant que les résidus sont indépendants.

Ces diagnostics confirment que notre modèle SARIMA est bien ajusté aux données et respecte les hypothèses statistiques nécessaires pour des prévisions fiables.

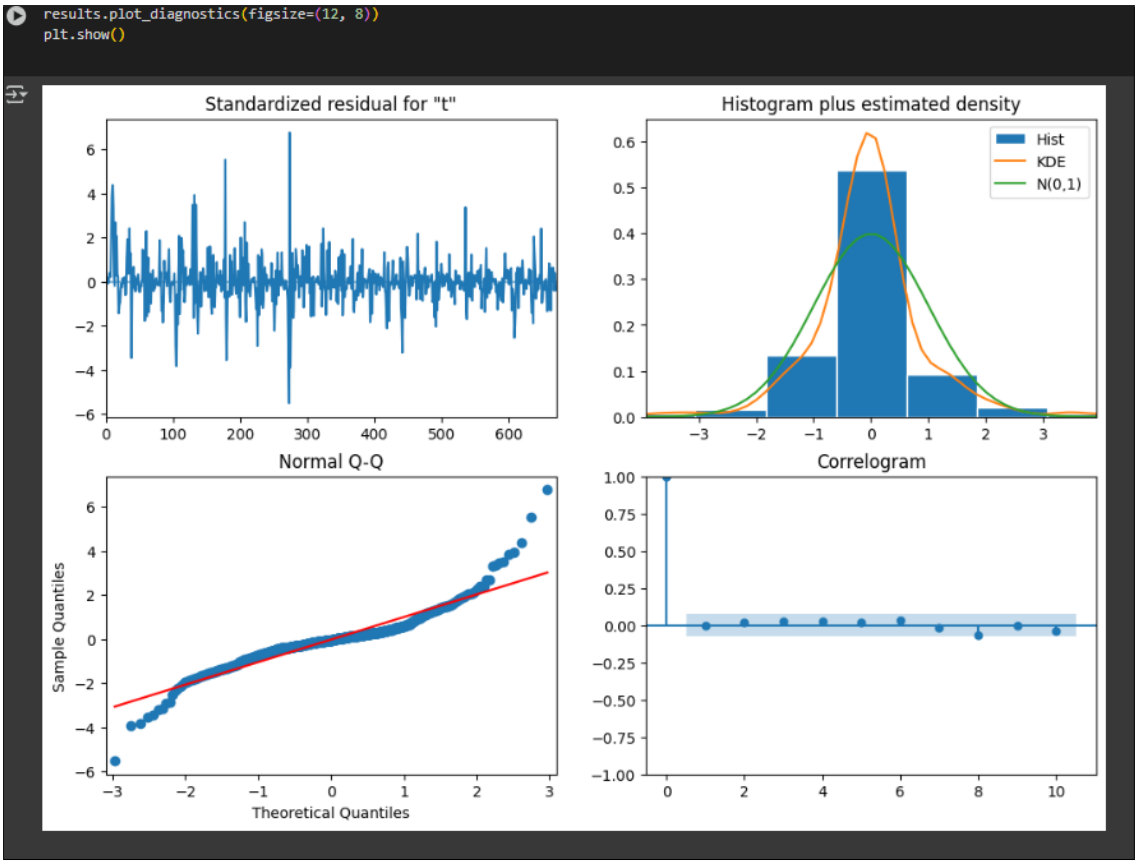

Figure 4.9 : Analyse du résidu

4.3.2 Interprétation des Prédictions du Modèle SARIMA :

La capture ci-dessus présente les données d'entraînement (Train Data), les données de test (Test Data) et les prévisions (Forecast) générées par le modèle SARIMA.

Données d'entraînement : Représentées en bleu, elles couvrent 80 % de l'ensemble des données, soit la période utilisée pour ajuster le modèle. Données de test : Représentées en orange, elles couvrent les 20 % restants des données et servent à évaluer la performance du modèle. Prédictions : Représentées en vert, elles montrent les valeurs prédites par le modèle pour la période correspondant aux données de test.

Observation :

Le modèle SARIMA parvient à capturer la saisonnalité et les tendances principales des données, avec une bonne correspondance entre les prévisions (vert) et les données de test (orange). Les pics et creux du total des recharges sont correctement prévus, bien que de légers écarts soient observés dans certaines périodes. La continuité et la proximité des courbes de prévisions et des données réelles démontrent la capacité du modèle à fournir des résultats précis.

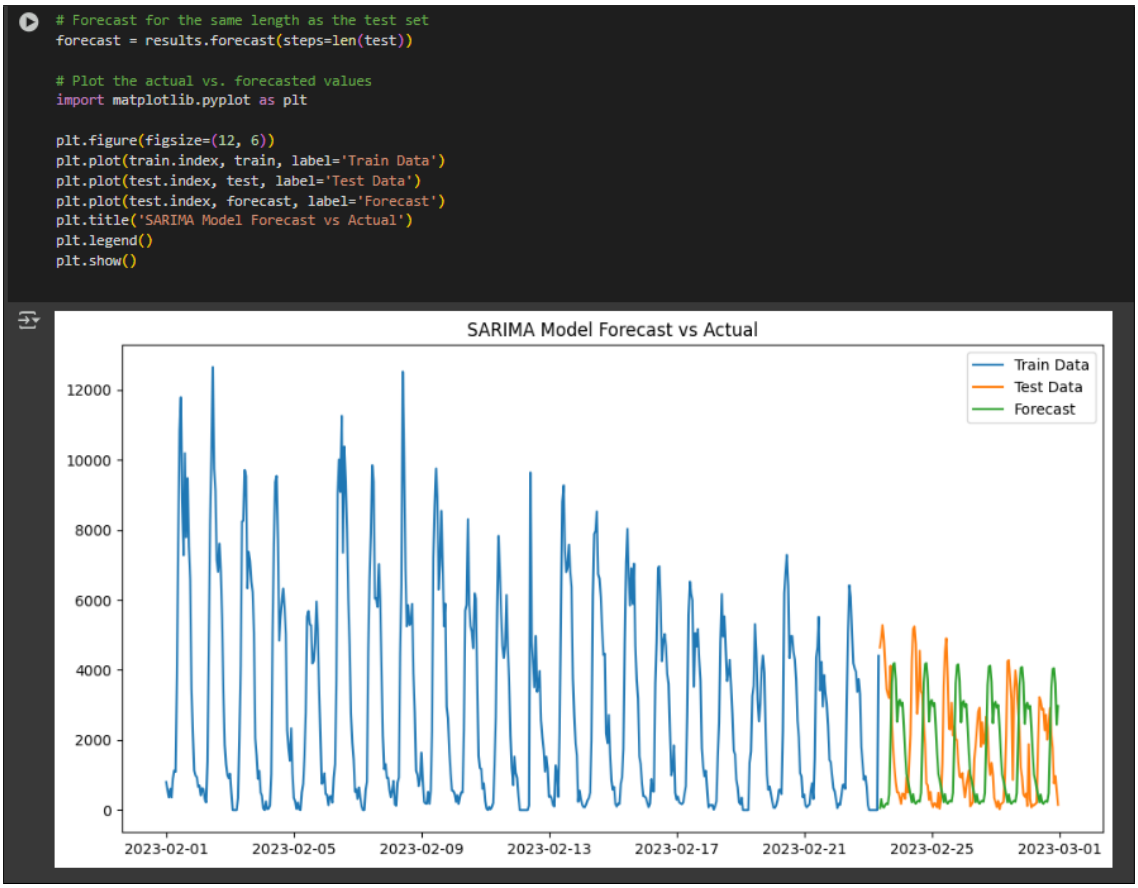

Figure 4.10 : Visualisation du prédiction

4.3.2.1 Évaluation des Performances du Modèle à l'aide des Métriques :

Pour évaluer la précision des prévisions effectuées par le modèle SARIMA, nous avons calculé les métriques suivantes :

Mean Absolute Error (MAE) : Cette métrique mesure l'erreur moyenne absolue entre les valeurs prédites et les valeurs réelles. Dans notre cas, le MAE est de 2293.68, ce qui représente l'erreur moyenne en termes absolus dans les prédictions.

Mean Squared Error (MSE) : Cette métrique mesure l'erreur quadratique moyenne, pondérant davantage les erreurs importantes. La valeur obtenue est de 6,554,626.52, indiquant qu'il existe des écarts notables dans certaines périodes.

Root Mean Squared Error (RMSE) : La racine carrée du MSE fournit une mesure des erreurs dans la même unité que les données. Ici, le RMSE est de 2560.20, ce qui signifie que l'erreur moyenne en termes de prévision est environ 2560 unités.

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np

mae = mean_absolute_error(test, forecast)
mse = mean_squared_error(test, forecast)
rmse = np.sqrt(mse)

print(f'Mean Absolute Error (MAE): {mae:.2f}')
print(f'Mean Squared Error (MSE): {mse:.2f}')
print(f'Root Mean Squared Error (RMSE): {rmse:.2f}')
```

```
Mean Absolute Error (MAE): 2293.68
Mean Squared Error (MSE): 6554626.52
Root Mean Squared Error (RMSE): 2560.20
```

Figure 4.11 : Evaluation des performances du modele

Interprétation :

Ces métriques montrent que le modèle SARIMA est globalement performant, avec une erreur relativement faible compte tenu des variations importantes dans les données horaires. Cependant, des ajustements supplémentaires du modèle ou l'utilisation de méthodes de prétraitement des données pourraient encore améliorer sa précision.

4.3.3 Transformation Logarithmique des Données :

Pour réduire la variance et stabiliser la série temporelle initiale, nous avons appliqué une transformation logarithmique. Cette transformation est particulièrement utile lorsque les données présentent des fluctuations importantes, comme observé dans la série initiale (graphique de gauche).

Dans le graphique de droite, après transformation logarithmique, la série devient plus homogène, réduisant les écarts extrêmes tout en conservant les caractéristiques essentielles du comportement saisonnier et des tendances. Cette étape permet d'améliorer l'ajustement du modèle SARIMA, en facilitant une meilleure modélisation et en réduisant l'impact des valeurs aberrantes.

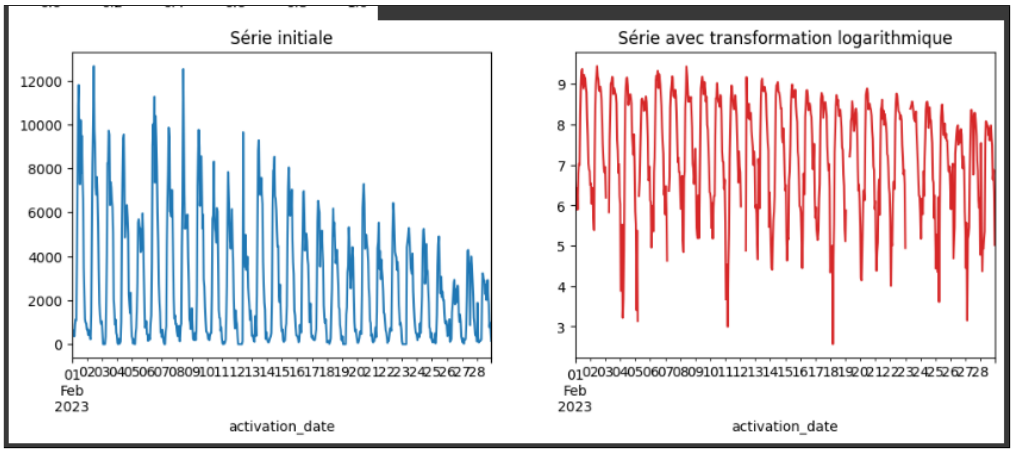

Figure 4.12 : Transformation logarithmique

Un test de Dickey-Fuller Augmenté (ADF) a été réalisé pour vérifier la stationnarité de la série temporelle après transformation. Les résultats indiquent une statistique ADF de -3.98 et une p-valeur de 0.0015, inférieure au seuil de 5%. De plus, la statistique ADF est inférieure à toutes les valeurs critiques (1%, 5%, et 10%). Par conséquent, nous pouvons conclure que la série est stationnaire, ce qui est une condition essentielle pour appliquer correctement le modèle SARIMA.

```
from statsmodels.tsa.stattools import adfuller

result = adfuller(df_hourly1.dropna())
print("ADF Statistic:", result[0])
print("p-value:", result[1])
print("Critical Values:", result[4])

if result[1] <= 0.05:
    print("La série est stationnaire.")
else:
    print("La série n'est pas stationnaire.")
```

```
ADF Statistic: -3.9836355934957894
p-value: 0.0014986662397180864
Critical Values: {'1%': -3.440890045708521, '5%': -2.8661904001753618, '10%': -2.569246579178572}
La série est stationnaire.
```

Figure 4.13 : Application du deuxieme test ADF

4.3.3.1 Différenciation pour Éliminer la Saisonnalité :

Pour supprimer la composante saisonnière et rendre la série plus stationnaire, une différenciation d'ordre 1 a été appliquée à la série temporelle. La comparaison entre la série initiale (à gauche) et la série différenciée (à droite) montre une réduction significative des variations saisonnières. Cette transformation est essentielle pour stabiliser la moyenne et préparer la série à une modélisation SARIMA efficace.

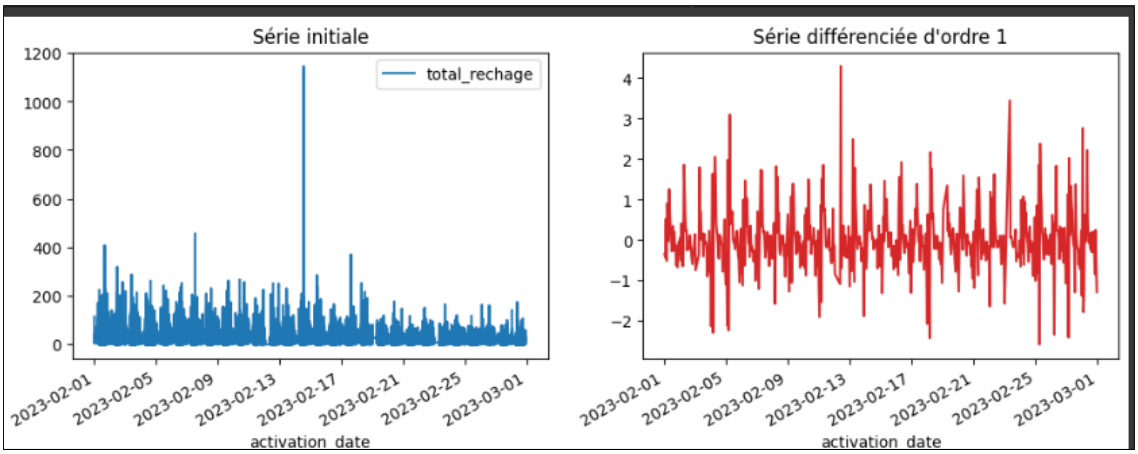

Figure 4.14 : Application du deuxieme test ADF

Visualisation de l'acf et pacf aprés diff :

Après avoir appliqué la différenciation pour tenter d'éliminer la saisonnalité, les graphiques de la fonction d'autocorrélation (ACF) et de la fonction d'autocorrélation partielle (PACF) révèlent que des éléments saisonniers persistent encore dans la série. Ces motifs sont visibles par des pics réguliers dans les deux graphiques, indiquant que des composantes saisonnières ou des dépendances temporelles n'ont pas été complètement supprimées.

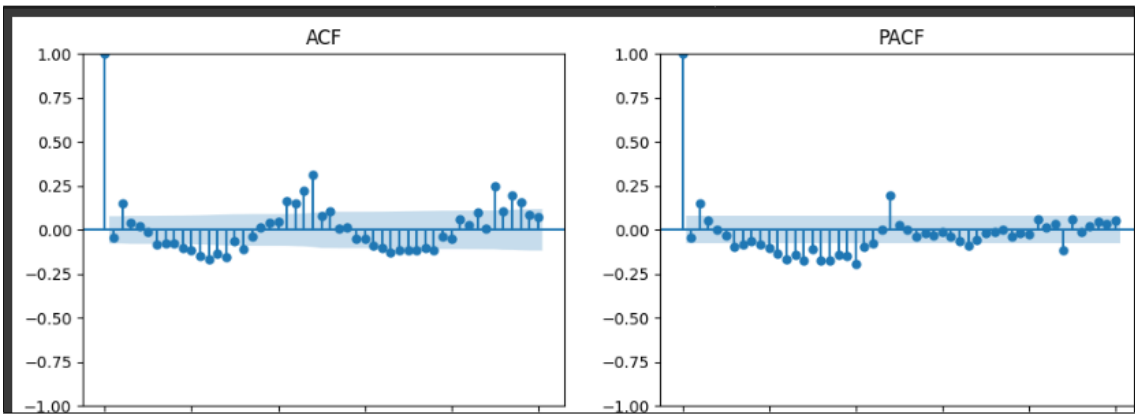

Figure 4.15 : Visualisation de l'ACF et PACF aprés diff

Application d'une deuxieme diffenciation :

Après avoir appliqué une deuxième différenciation saisonnière, les graphiques de l'ACF et du PACF montrent que les motifs saisonniers ont été éliminés. Les autocorrélations significatives observées précédemment ont disparu, et les valeurs sont maintenant majoritairement dans les intervalles de confiance, indiquant que la série est devenue stationnaire sans composantes saisonnières évidentes. Cette étape permet de mieux préparer la série pour le modèle SARIMA en s'assurant que la saisonnalité ne biaise plus les résultats.

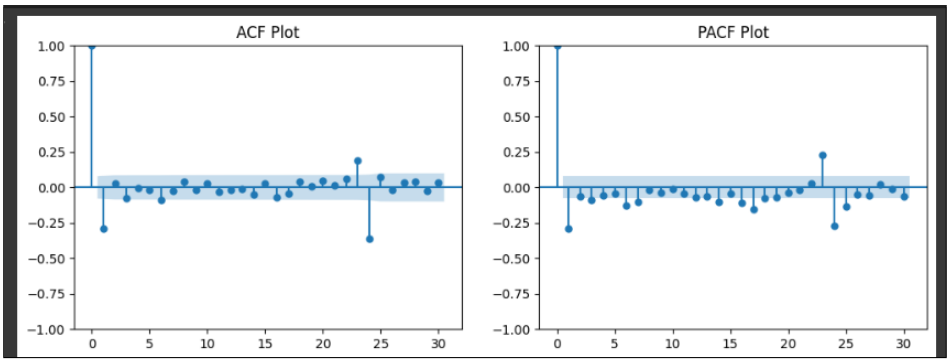

FIGURE 4.16 : Visualisation de l'ACF et PACF aprés la deuxieme diff

Après avoir confirmé l'absence de saisonnalité dans la série grâce aux graphiques ACF et PACF, nous avons appliqué le modèle SARIMA pour modéliser les données. Le modèle SARIMA a été choisi pour sa capacité à gérer les composantes saisonnières, les tendances, et les autocorrélations dans les séries temporelles. Cette étape marque l'utilisation d'un modèle robuste pour prédire les valeurs futures tout en tenant compte des propriétés stationnaires de la série.

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                        total_rechage   No. Observations:                  537
Model:             ARIMA(1, 2, 1)x(1, 1, [1, 2], 24)  Log Likelihood               -4198.782
Date:                               Fri, 07 Feb 2025  AIC                           8409.563
Time:                                       10:32:42  BIC                           8434.981
Sample:                                   02-01-2023  HQIC                          8419.528
                                        - 02-23-2023
Covariance Type:                                 opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.3854      0.022    -17.243      0.000      -0.429      -0.342
ma.L1         -1.0000      0.234     -4.277      0.000      -1.458      -0.542
ar.S.L24      -1.0000      1.257     -0.795      0.426      -3.465       1.465
ma.S.L24       0.1340      1.491      0.090      0.928      -2.788       3.056
ma.S.L48      -0.8659      1.292     -0.670      0.503      -3.398       1.666
sigma2      7.532e+05   3.71e-07   2.03e+12      0.000    7.53e+05    7.53e+05
===================================================================================
Ljung-Box (L1) (Q):                   2.86   Jarque-Bera (JB):              1332.76
Prob(Q):                              0.09   Prob(JB):                         0.00
Heteroskedasticity (H):               0.50   Skew:                             0.53
Prob(H) (two-sided):                  0.00   Kurtosis:                        10.84
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 2.29e+30. Standard errors may be unstable.
```

FIGURE 4.17 : Visualisation du deuxieme modele sarima

4.3.4 Deuxième Transformation Logarithmique (Log1) :

À cette étape, une deuxième transformation logarithmique a été appliquée à la série temporelle, comme illustré dans le graphique. Cette transformation permet de stabiliser davantage la variance et d'atténuer les fluctuations importantes dans les données. En réduisant l'impact des valeurs extrêmes, cette approche facilite l'identification des tendances et des motifs sous-jacents tout en améliorant la performance des modèles prédictifs, notamment pour les séries présentant

une grande amplitude de variation.

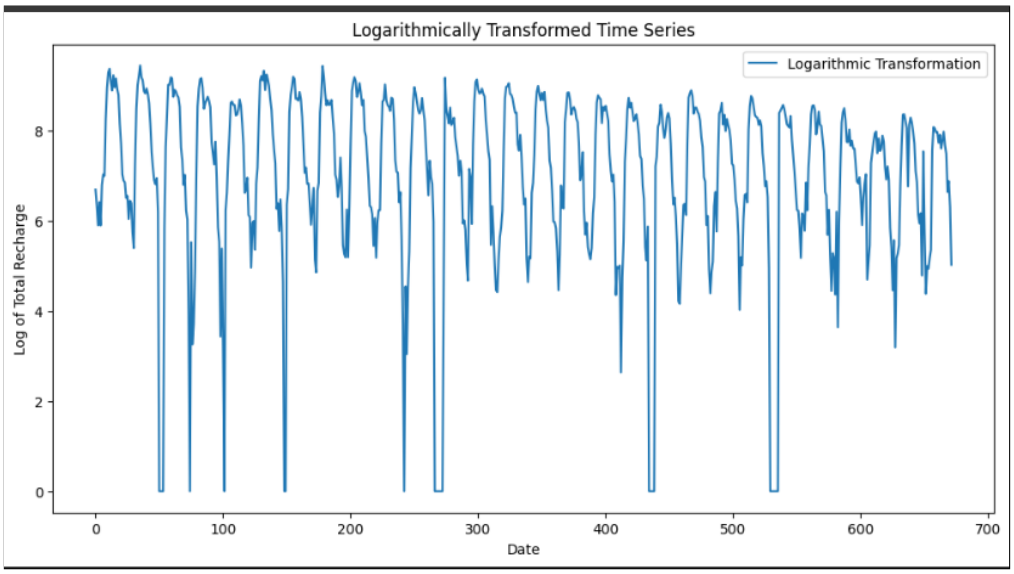

Figure 4.18 : Visualisation du deuxieme transformation logarithmique

Après avoir appliqué une transformation logarithmique sur la série temporelle pour stabiliser la variance, deux modèles SARIMA ont été testés avec des paramètres différents pour ajuster la série et capturer sa dynamique.

Dans le premier modèle SARIMA (1, 1, 1) × (1, 1, 1, 24), les résultats montrent des coefficients significatifs pour certaines composantes, avec des valeurs d'AIC et BIC relativement basses (AIC = 1772.464, BIC = 1794.826), indiquant une bonne adéquation au modèle. Cependant, les tests statistiques, comme la normalité des résidus (Jarque-Bera), signalent des déviations.

Dans le second modèle SARIMA (1, 0, 1) × (1, 0, 1, 24), un ajustement différent a été réalisé avec une configuration sans différenciation sur le terme D, pour évaluer une alternative. Les métriques d'évaluation (AIC = 1868.380, BIC = 1890.931) sont moins optimales comparées au premier modèle, suggérant un ajustement légèrement inférieur.

Ces essais montrent que le choix des paramètres influence directement la performance du modèle. Le premier modèle semble mieux convenir à la série après la transformation logarithmique.

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                  log_total_rechage   No. Observations:                  672
Model:             SARIMAX(1, 1, 1)x(1, 1, 1, 24)   Log Likelihood                -881.232
Date:                            Fri, 07 Feb 2025   AIC                           1772.464
Time:                                    10:38:38   BIC                           1794.826
Sample:                                         0   HQIC                          1781.140
                                            - 672
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.6907      0.020     34.352      0.000       0.651       0.730
ma.L1         -0.9999      0.738     -1.354      0.176      -2.447       0.447
ar.S.L24       0.0684      0.033      2.090      0.037       0.004       0.132
ma.S.L24      -0.8828      0.024    -36.514      0.000      -0.930      -0.835
sigma2         0.8370      0.618      1.355      0.175      -0.374       2.048
===================================================================================
Ljung-Box (L1) (Q):                   0.19   Jarque-Bera (JB):              2095.41
Prob(Q):                              0.66   Prob(JB):                         0.00
Heteroskedasticity (H):               0.94   Skew:                            -0.12
Prob(H) (two-sided):                  0.67   Kurtosis:                        11.81
===================================================================================
```

Figure 4.19 : Visualisation du 3 eme modele Sarima

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                  log_total_rechage   No. Observations:                  672
Model:             SARIMAX(1, 0, 1)x(1, 0, 1, 24)   Log Likelihood                -929.190
Date:                            Fri, 07 Feb 2025   AIC                           1868.380
Time:                                    10:39:44   BIC                           1890.931
Sample:                                         0   HQIC                          1877.114
                                            - 672
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.8319      0.017     48.174      0.000       0.798       0.866
ma.L1         -0.1209      0.033     -3.680      0.000      -0.185      -0.057
ar.S.L24       0.9961      0.002    559.530      0.000       0.993       1.000
ma.S.L24      -0.8566      0.022    -39.763      0.000      -0.899      -0.814
sigma2         0.8545      0.024     35.881      0.000       0.808       0.901
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):              3178.41
Prob(Q):                              0.98   Prob(JB):                         0.00
Heteroskedasticity (H):               0.77   Skew:                             0.11
Prob(H) (two-sided):                  0.06   Kurtosis:                        13.65
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

Figure 4.20 : Visualisation du 4 eme modele Sarima

Visualisation de la prédiction du modele sarima :

Cette étape consiste à visualiser les performances du modèle SARIMA en comparant les valeurs réelles du total des recharges aux prédictions générées par le modèle. Le graphique montre les données observées (en bleu) et les prédictions SARIMA (en orange) sur la même échelle de temps, permettant de vérifier la correspondance entre les deux. Une bonne adéquation entre les deux courbes indique que le modèle capture efficacement les tendances et les variations de

la série temporelle, validant ainsi son utilité pour des prévisions futures. Cette visualisation est essentielle pour évaluer qualitativement la précision du modèle.

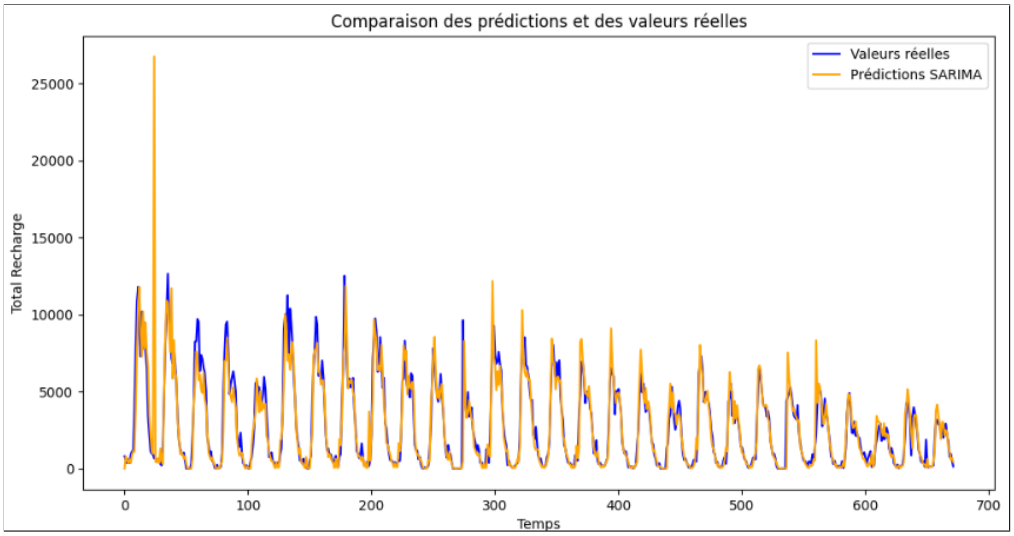

FIGURE 4.21 : Visualisation du 4 eme modele Sarima

Utilisation du Modele Prophet :

Prophet est un outil développé par Facebook pour effectuer des prévisions temporelles. Il est conçu pour être flexible, robuste et simple à utiliser, particulièrement pour des séries chronologiques contenant des tendances non linéaires, des saisons, et des jours fériés. Prophet décompose les données en trois composantes principales : tendance, saisonnalité, et événements exceptionnels, ce qui permet de modéliser et prévoir des données complexes.

Les prévisions de Prophet semblent capturer correctement la tendance globale et les fluctuations saisonnières, mais des écarts sont observés à certains pics et creux

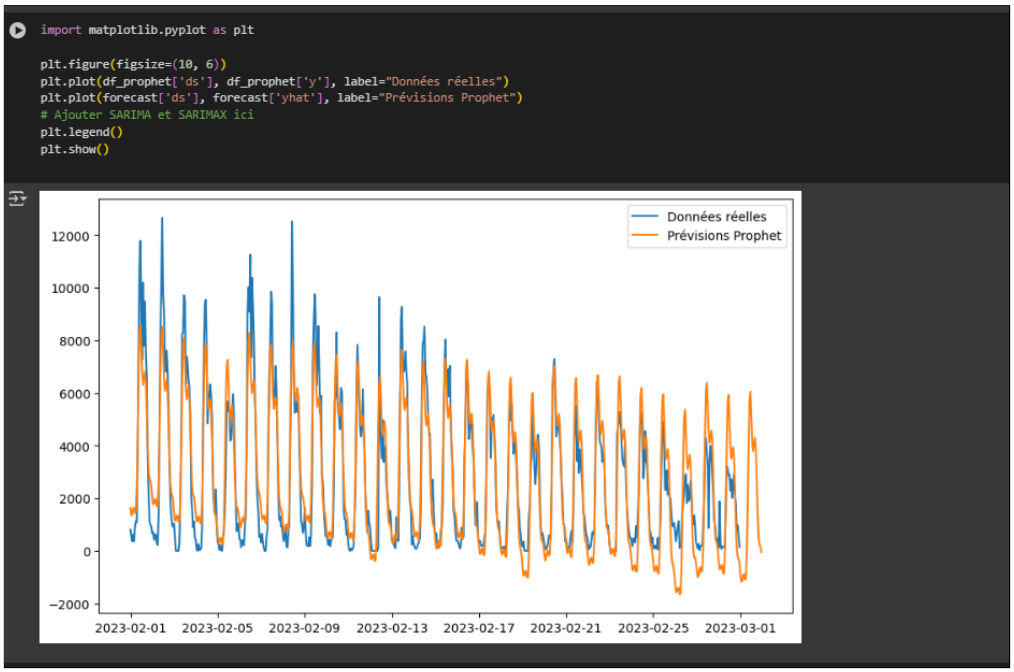

Figure 4.22 : Visualisation du modele prophet

4.4 Comparaison des modèles

Modèle	Log-Likelihood	AIC	BIC	MAE	RMSE
SARIMA (sans diff)	-929.190	1868.380	1890.931	645.07	1440.26
SARIMA (avec diff et transf.)	-881.23	1772.46	1794.83	-	-
Prophet	-	-	-	1249.05	1523.58

Tableau 4.1 : Comparaison des modèles selon les métriques disponibles.

En analysant les résultats :

— Le modèle **SARIMA (avec différenciation)** offre la meilleure performance globale avec une **loglikelihood** de -881,23 et une **AIC** de 1772.46.

— Le modèle **Prophet**, bien que facile à paramétrer, présente des erreurs plus élevées (**MAE** de 1249.05 et **RMSE** de 1523.58).

— Le modèle **SARIMA (sans différenciation)** est intéressant pour son approche log-transformée, mais les résultats des erreurs ne sont pas fournis directement.

Recommandation : Le modèle **SARIMA AVEC différenciation** est le plus performant pour les prédictions sur ces données.

4.5 Conclusion

Dans ce chapitre, nous avons étudié et comparé différents modèles de prévision des séries temporelles appliqués aux données des recharges totales. L'objectif principal était d'évaluer leurs performances et d'identifier le modèle le plus adapté pour des prédictions fiables et précises.

Chapitre 5

Organisation des données dans une Base de données relationnelle

Plan

1 Introduction 54

2 Base de données en graph 55

3 Insights de fidélité 55

4 Conclusion 60

5.1 Introduction

Dans le cadre de ce projet, nous avons pour objectif de structurer et d'organiser des données dans une base de données relationnelle afin de faciliter leur gestion, leur exploitation et leur analyse. Pour ce faire, nous avons utilisé deux outils principaux : Talend Open Studio for Data Integration et pgAdmin.

Talend a été utilisé pour extraire, transformer et charger (ETL) les données , tout en assurant leur nettoyage et leur formatage afin de garantir leur qualité et leur cohérence. Les données ainsi préparées ont été importées dans une base de données PostgreSQL, gérée via pgAdmin. .

5.2 Établissement d'un Diagramme de Classe :

Avant de procéder à la création des tables dans la base de données, il est essentiel de structurer les données de manière logique et cohérente. Pour ce faire, un diagramme de classe a été élaboré. Ce diagramme permet de modéliser les entités principales (tables) et les relations entre elles, conformément aux règles de normalisation des bases de données relationnelles.

Chaque classe du diagramme représente une entité clé, correspondant à une table dans la base de données, avec ses attributs et ses relations.

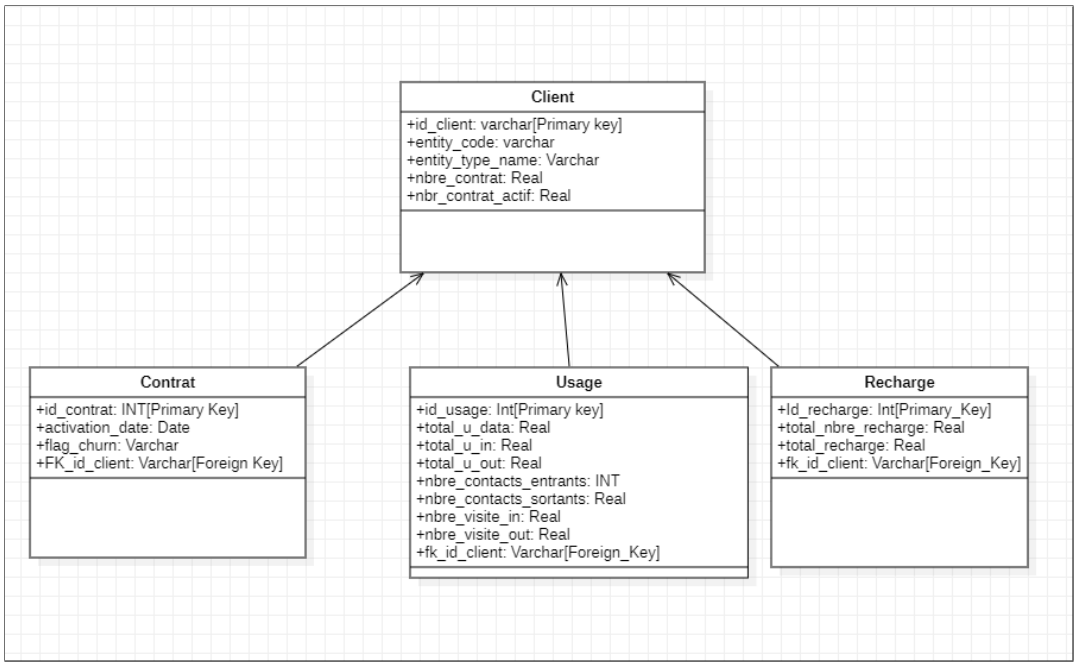

Figure 5.1 : Diagramme de classe

Voici une brève description des entités identifiées :

Client : Représente les informations générales sur les clients, comme leur identifiant unique et leur type.

Contrat : Contient les détails des contrats signés par les clients, tels que les dates et les statuts . Recharges : Stocke les informations sur les transactions de recharge effectuées par les clients.

Usage : Rassemble les données sur l'utilisation des services, comme le volume des données consommées ou le nombre d'appels effectués.

Les relations entre ces entités ont été définies pour refléter les liens logiques dans les données : On a 3 Relations :

Relation entre Client et Recharge : Un client peut effectuer plusieurs recharges, ce qui reflète une relation de type 1 :N.

Relation entre Client et Contrat :

Un client peut être lié à plusieurs contrats, ce qui représente également une relation de type 1 :N.

Relation entre Client et usages :

Un client peut avoir plusieurs enregistrements d'usage associés, ce qui traduit encore une relation de type 1 :N.

La clé étrangère fk_id_client dans la table Usage, contrat et recharges fait référence à la clé primaire id_client de la table Client.

Connexion entre Talend et PostgreSQL :

Dans cette étape, nous avons configuré la connexion entre Talend et PostgreSQL afin de permettre le transfert et la manipulation des données. Pour cela, nous avons vérifié et ajusté les paramètres de configuration du serveur PostgreSQL. Le job présenté ci-dessus illustre l'exemple du chargement des données dans la table Client :

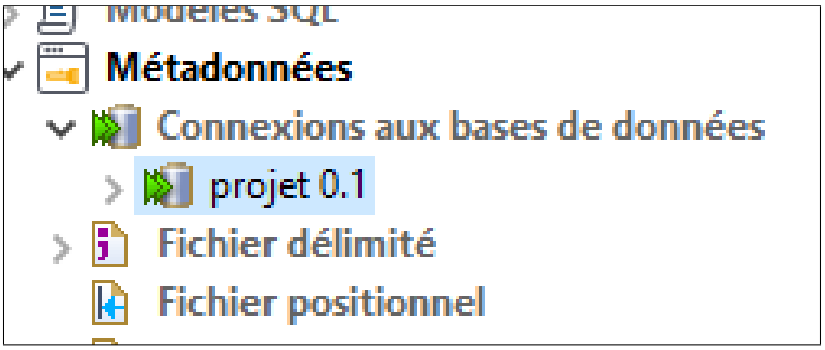

Figure 5.2 : Connexion entre Talend et PostgreSQL

Création des Jobs pour Charger les Tables :

Une fois la connexion établie entre Talend et PostgreSQL, nous avons créé des jobs Talend pour extraire, transformer, et charger les données dans chaque table de la base de données relationnelle. Ces jobs permettent d'assurer une migration structurée et propre des données depuis les fichiers sources vers PostgreSQL.

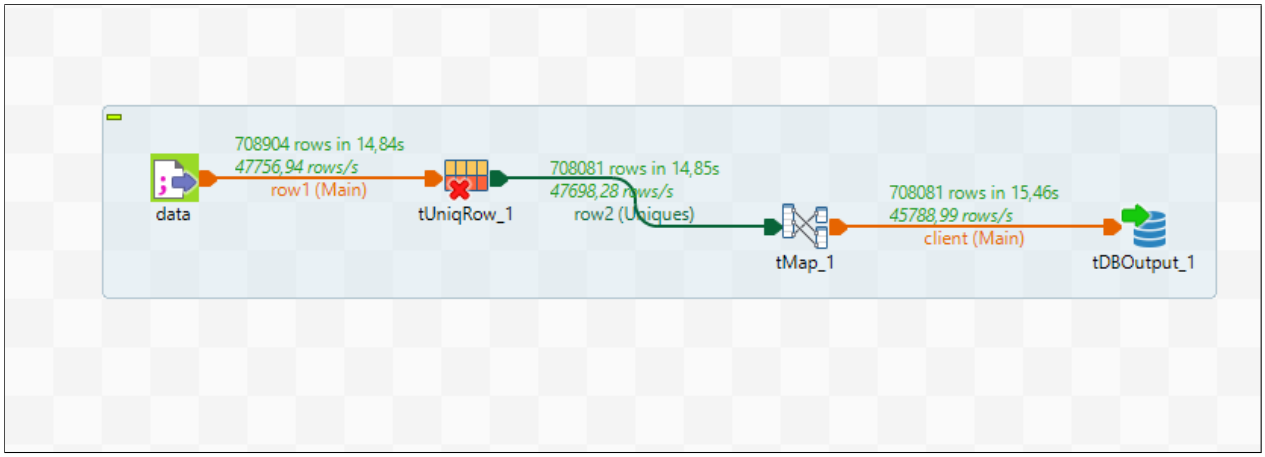

Figure 5.3 : Chargement de la table Client

Lecture des Données Sources :

Utilisation d'un composant tFileInputDelimited pour lire les données brutes à partir d'un fichier source.

Filtrage des Doublons :

Un composant tUniqRow a été utilisé pour éliminer les doublons dans les données, garantissant l'intégrité et la qualité des informations chargées dans la table.

Transformation avec tMap :

Le composant tMap a permis de manipuler les données en appliquant des transformations spécifiques, telles que la suppression de caractères inutiles (StringHandling.TRIM) et la conversion des formats. Ce composant a également permis de mapper les colonnes des données sources aux colonnes de la table cible.

Chargement dans PostgreSQL :

Le composant tDBOutput a été utilisé pour insérer les données traitées dans la table Client. Le processus a été configuré pour effectuer des insertions en masse, garantissant des performances optimales lors du chargement de volumes importants de données.

En répétant cette méthode pour chaque table (Contrat, Recharge, Usage), nous avons assuré

une structuration et une intégration cohérente des données dans la base relationnelle. Ce processus garantit que les relations entre les tables soient respectées et que les données soient prêtes pour les prochaines étapes d'analyse et de visualisation.

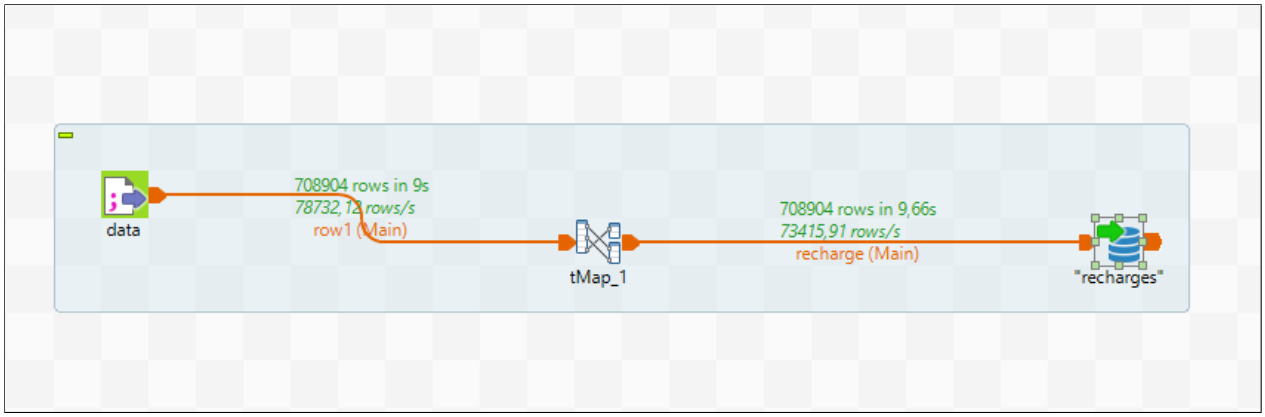

Figure 5.4 : Chargement de la table Recharges

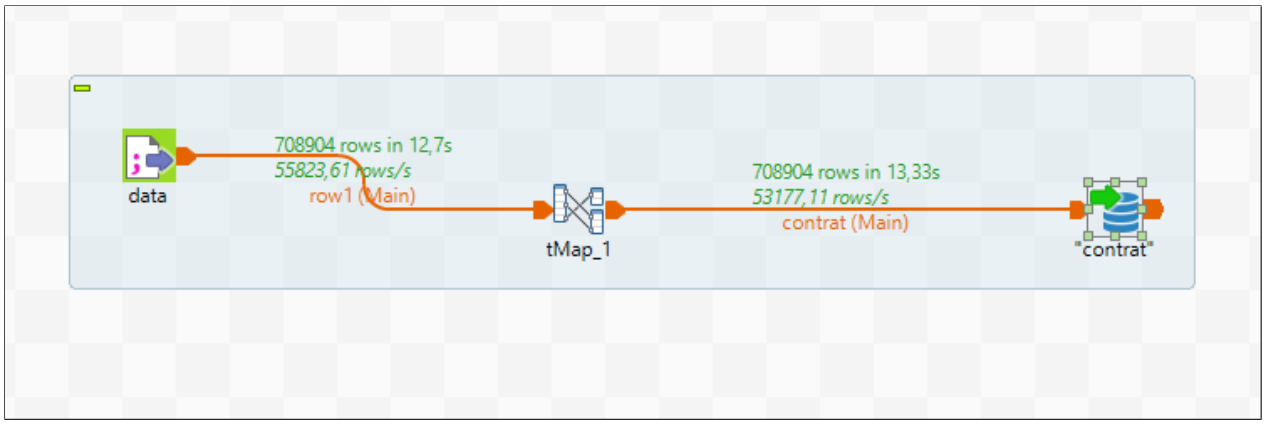

Figure 5.5 : Chargement de la table Contrat

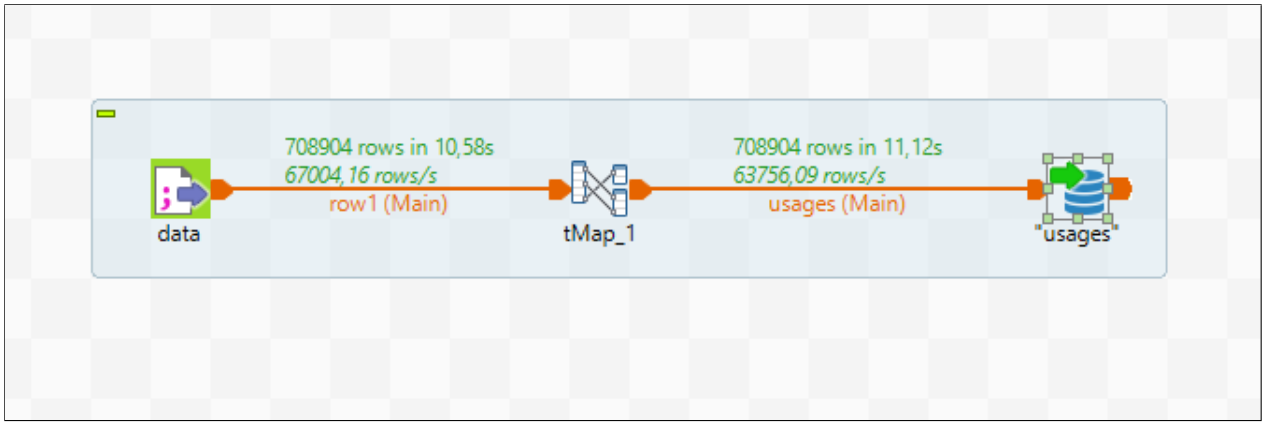

Figure 5.6 : Chargement de la table usage

Visualisation des Tables Remplies dans PostgreSQL :

Après avoir chargé les données dans les différentes tables à l'aide des jobs Talend, nous avons vérifié la réussite de cette opération en visualisant le contenu des tables dans l'interface de pgAdmin. Cette étape permet de s'assurer que les données ont été intégrées correctement et correspondent aux attentes initiales. En explorant chaque table (Client, Contrat, Recharge, Usage), nous avons validé la structure des données, les relations définies, ainsi que leur cohérence et leur intégrité. Cette vérification est essentielle avant de passer aux étapes d'analyse ou de visualisation des données.

	entity_code character varying (200)	entity_type_name character varying (200)	nbre_contrat real	nbr_contrat_actif real	id_client [PK] character varying (200)
1	TUN	BOUTIQUE	3	2	10121829
2	TRA	INDIRECT	3	2	10149522
3	TRA	INDIRECT	3	0	10239415
4	TRA	INDIRECT	3	2	10271203
5	TRA	INDIRECT	3	1	10384448
6	TUN	INDIRECT	2	2	10386195
7	TRA	INDIRECT	3	2	10414812
8	TRA	INDIRECT	3	1	10420041
9	TRA	INDIRECT	3	2	10464171
10	TRA	INDIRECT	3	1	10520218
11	TRA	INDIRECT	3	0	10525583
12	TRA	INDIRECT	3	1	10536871
13	TRA	INDIRECT	3	2	10545610
14	TRA	INDIRECT	3	1	10566866
15	TRA	INDIRECT	3	0	10606230
16	TRA	INDIRECT	3	2	10609306
17	TRA	INDIRECT	3	2	10614352
18	SOU	INDIRECT	4	2	10632925
19	EBU	AUTRES BOUTIQUES	449	390	10633681
20	EBU	AUTRES BOUTIQUES	449	390	10633682
21	EBU	AUTRES BOUTIQUES	449	390	10634381
22	EBU	AUTRES BOUTIQUES	449	390	10635604
23	EBU	AUTRES BOUTIQUES	449	390	10635608
24	EBU	AUTRES BOUTIQUES	449	390	10635663

Figure 5.7 : Chargement de la table Client

	total_nbre_recharge real	total_recharge real	recharges_id [PK] integer	fk_id_client character varying (200)
1	2	1.878	1	"36799963.0
2	1	0.878	2	"36800104.0
3	25	28.386	3	"34763697.0
4	1	0.878	4	"36822172.0
5	1	4.386	5	"34745291.0
6	0	0	6	"36436235.0
7	1	2	7	"38482347.0
8	1	5	8	"38318140.0
9	1	1	9	"38495525.0
10	1	4.386	10	"34742253.0
11	1	5.264	11	"37603739.0
12	1	4.386	12	"37633404.0
13	1	4.386	13	"34745638.0
14	1	0.878	14	"36814569.0
15	0	0	15	"37611171.0
16	4	24	16	"38420673.0
17	1	0.878	17	"36807878.0
18	9	9	18	"37858879.0
19	1	5.264	19	"34745561.0
20	2	15	20	"38297822.0
21	1	4.386	21	"37633459.0
22	2	9.386001	22	"37640295.0
23	2	6	23	"37604677.0

Figure 5.8 : Visualisation de la table Recharges

	activation_date date	flag_churn character varying (200)	id_contrat [PK] integer	fk_id_client character varying (200)
1	2022-07-20	0.0"	1	"36799963.0
2	2022-07-20	1.0"	2	"36800104.0
3	2022-07-06	0.0"	3	"34763697.0
4	2022-07-17	0.0"	4	"36822172.0
5	2022-07-04	0.0"	5	"34745291.0
6	2022-07-31	1.0"	6	"36436235.0
7	2022-07-18	0.0"	7	"38482347.0
8	2022-07-07	0.0"	8	"38318140.0
9	2022-07-19	0.0"	9	"38495525.0
10	2022-07-04	1.0"	10	"34742253.0
11	2022-07-06	1.0"	11	"37603739.0
12	2022-07-14	1.0"	12	"37633404.0
13	2022-07-11	1.0"	13	"34745638.0
14	2022-07-18	1.0"	14	"36814569.0
15	2022-07-17	0.0"	15	"37611171.0
16	2022-07-15	1.0"	16	"38420673.0
17	2022-07-20	1.0"	17	"36807878.0
18	2022-07-05	1.0"	18	"37858879.0
19	2022-07-13	1.0"	19	"34745561.0
20	2022-07-04	0.0"	20	"38297822.0
21	2022-07-14	1.0"	21	"37633459.0
22	2022-07-18	1.0"	22	"37640295.0
23	2022-07-02	1.0"	23	"37604677.0
24	2022-07-20	0.0"	24	"36820915.0

Figure 5.9 : Visualisation de la table contrat

	total_u_data real	total_u_in real	total_u_out real	nbre_contacts_entrants integer	nbre_contacts_sortans real	nbre_visite_in real	nbre_visite_out real	usages_id [PK] integer	fk_id_client character varying
1	1.0895603e+07	0.633	14.95	4	10	5	6	1	"36799963.0
2	9.5208e+06	17.617	14.983	13	12	10	4	2	"36800104.0
3	8.024546e+06	86.034	240.1	56	69	28	22	3	"34763697.0
4	6.366023e+06	4.183	13.083	5	5	6	4	4	"36822172.0
5	0.25878906	5.784	1.967	11	2	7	1	5	"34745291.0
6	7.779721e+06	23.684	144.766	37	35	41	49	6	"36436235.0
7	0	3.333	0	2	1	1	0	7	"38482347.0
8	43.512695	36.2	59.766	22	17	24	21	8	"38318140.0
9	0	8.934	17.933	15	10	21	10	9	"38495525.0
10	6.2157365e+06	0	5.184	4	6	12	5	10	"34742253.0
11	5.242983e+06	5.15	0.25	10	3	7	1	11	"37603739.0
12	4.795809e+06	0	0	1	1	3	0	12	"37633404.0
13	5.24291e+06	0.1	3.717	5	4	7	2	13	"34745638.0
14	6.784794e+06	0	0.366	7	3	5	2	14	"36814569.0
15	0	0	0	7	1	3	0	15	"37611171.0
16	6.6093395e+06	15.567	37.167	7	5	17	9	16	"38420673.0
17	1.0692617e+07	10.95	13.833	11	11	9	3	17	"36807878.0
18	1.1189319e+06	45.183	644.234	11	15	6	9	18	"37858879.0
19	744711.3	0	7	2	2	3	1	19	"34745561.0
20	0	279.233	299.517	81	48	18	8	20	"38297822.0
21	5.24288e+06	0	0	1	1	2	0	21	"37633459.0
22	7.395328e+06	3.283	6	5	5	13	7	22	"37640295.0
23	5.0395135e+06	46.582	91.984	17	30	11	8	23	"37604677.0

Figure 5.10 : Visualisation de la table usages

5.3 Conclusion :

Dans ce chapitre, nous avons établi les bases essentielles pour la structuration et le chargement des données dans une base de données relationnelle. Après avoir conçu un diagramme de classe pour identifier les entités principales et leurs relations, nous avons implémenté ces structures dans PostgreSQL à l'aide de l'outil pgAdmin. Ensuite, nous avons utilisé Talend pour créer des jobs permettant de charger les données dans les différentes tables tout en assurant leur transformation et leur intégrité. Enfin, la visualisation des données dans pgAdmin nous a permis de valider le succès de ces opérations. Ces étapes posent une base solide pour les phases suivantes, notamment l'analyse et la visualisation des données, garantissant ainsi une meilleure exploitation des informations disponibles.

Chapitre 6

Optimisation

6.1 Introduction

Pour aller plus loin dans l'exploitation des données, nous avons créé une base de données en graph, une approche particulièrement optimisée pour l'implémentation de chatbots et la découverte de nouveaux insights et interprétations. Cette structure permet de modéliser les relations complexes entre les différents éléments de manière plus naturelle et efficace, facilitant ainsi la gestion des interactions et des processus décisionnels. De plus, nous avons représenté un graph des différentes localisations des entités, permettant aux utilisateurs de sélectionner des régions spécifiques et de calculer le plus court chemin entre elles. Cette fonctionnalité peut s'avérer particulièrement utile pour les acteurs de la télécommunication, qui doivent souvent optimiser leurs réseaux et leurs services en fonction des connexions entre différentes zones géographiques. Par exemple, cela peut aider à déterminer les itinéraires les plus efficaces pour le déploiement de nouvelles infrastructures, améliorer la gestion des flux de données entre régions ou encore optimiser les coûts d'entretien des réseaux en fonction des distances entre les antennes et les centres de données. Pour mener à bien cette analyse des données clients, nous avons utilisé **Neo4j Aura**, une plateforme de graphes basée sur la technologie de base de données **graphique Neo4j**. Neo4j est une base de données NoSQL spécialement conçue pour gérer des relations complexes entre les données. Contrairement aux bases de données relationnelles traditionnelles, Neo4j permet de représenter les données sous forme de graphes, où les entités (comme les clients, les contrats ou les services) sont reliées par des arêtes, représentant les relations entre elles. Cette approche est particulièrement adaptée pour des analyses exploratoires et prédictives dans des contextes où les connexions entre les données jouent un rôle central, comme dans le cas de la segmentation des clients ou de la détection des comportements de churn. En utilisant **Neo4j Aura**, une version cloud de Neo4j, nous avons pu bénéficier d'une solution scalable et flexible, permettant d'effectuer des requêtes puissantes et des analyses en temps réel, tout en simplifiant la gestion des infrastructures sous-jacentes. Cette technologie nous a donc permis de mieux comprendre et modéliser les relations entre les clients et leurs comportements au sein des différents clusters.

6.2 Base de données en graph

Dans le graphique que nous présentons, les **nœuds initiaux** représentent les différents **clusters** identifiés dans l'analyse des données clients. Chaque **cluster** regroupe les clients qui partagent des caractéristiques similaires en termes d'activité, d'utilisation des services, et d'autres paramètres comportementaux. Ces clusters permettent ainsi de segmenter les clients selon leurs habitudes, et chaque **nœud de cluster** contient les **clients respectifs** qui lui sont attribués. Cette représentation visuelle facilite la compréhension des relations entre les différents groupes de clients et permet d'identifier des tendances ou comportements spécifiques à chaque segment.

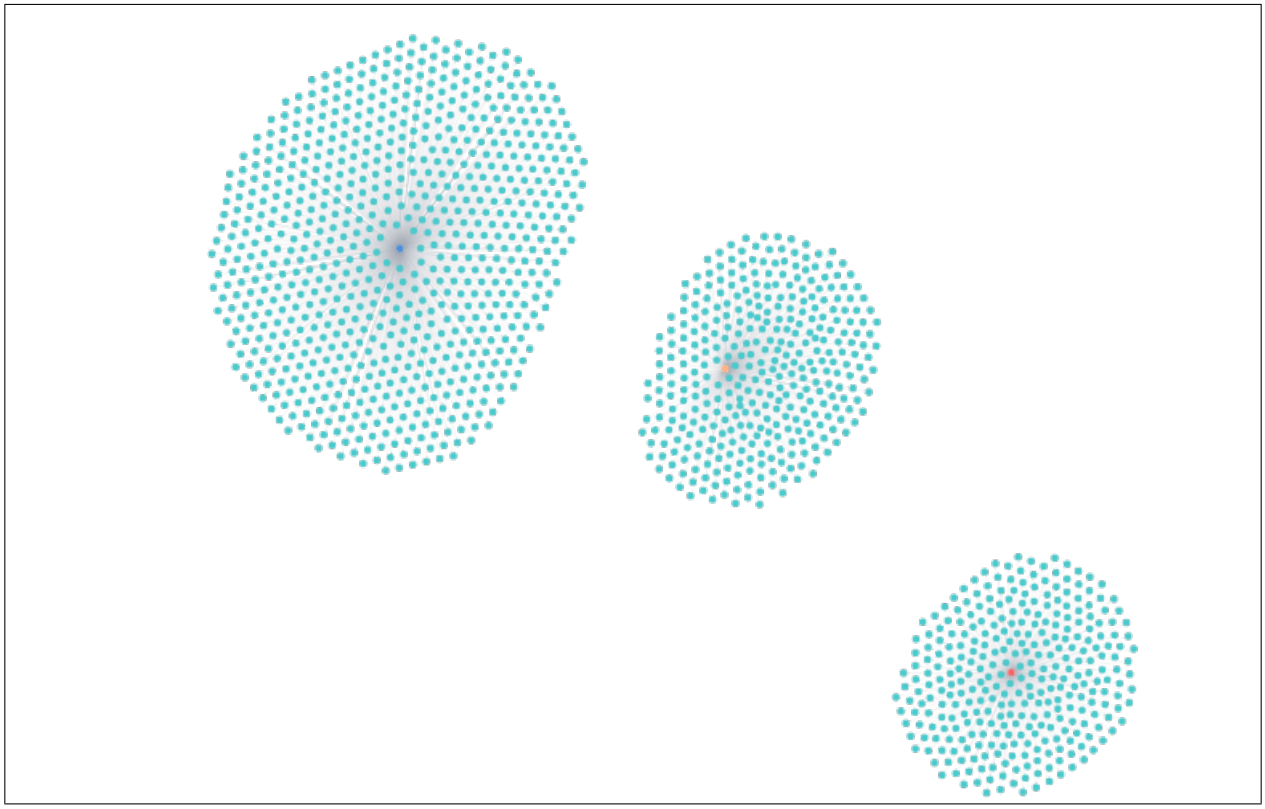

Figure 6.1 : Visualisation de la BD en graph

6.3 Insights de fidélité

Nous avons utilisé la fonctionnalité de slicing pour cibler spécifiquement le Cluster 2, celui qui est le plus actif et qui consomme le plus de données. Nous avons affiné cette segmentation en nous concentrant sur les clients ayant les indicateurs les plus remarquables, à savoir un flag churn actif, un total recharge élevé et un total de données utilisées (total u data) important.

Ces clients, présentant un fort potentiel de consommation, sont des cibles idéales pour des programmes de fidélisation. En leur offrant des récompenses personnalisées, nous pouvons encourager leur rétention, maximiser leur engagement et renforcer leur loyauté à long terme. Cette approche permet non seulement de valoriser les utilisateurs les plus impliqués, mais aussi d'optimiser les stratégies de fidélisation en s'appuyant sur des données précises et pertinentes.

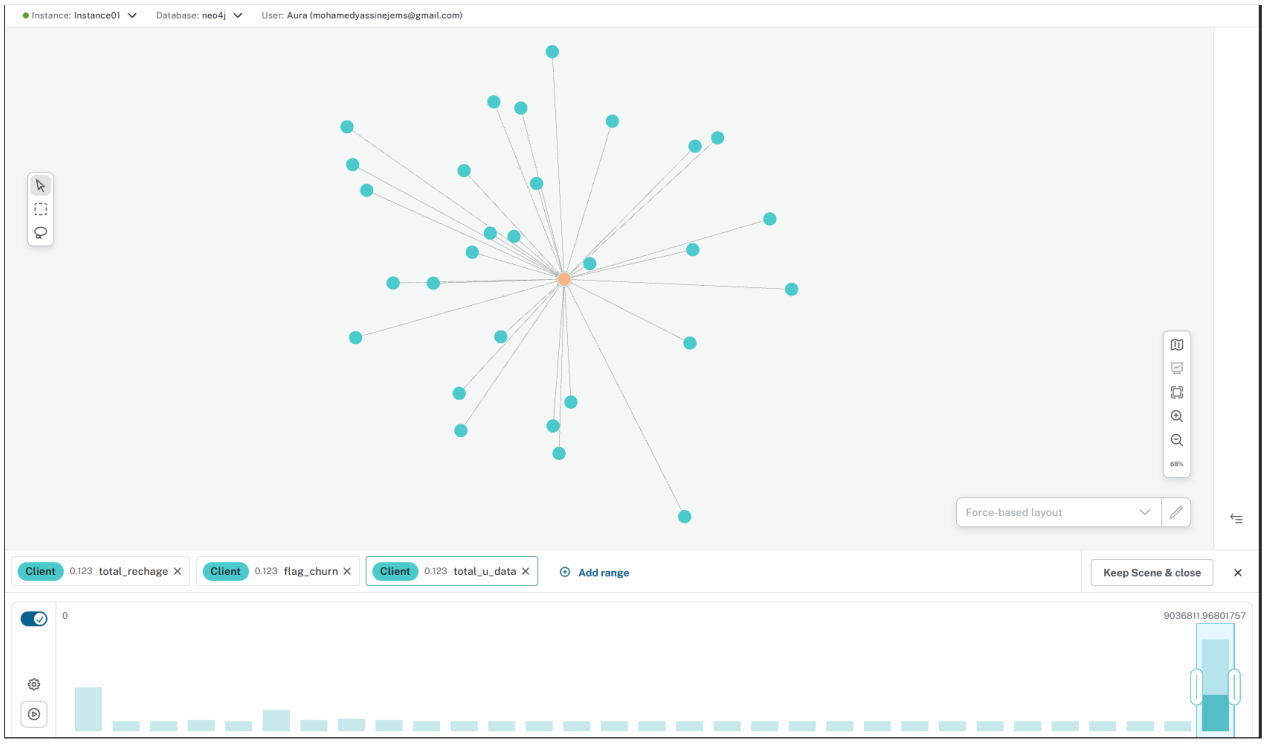

Figure 6.2 : Slicing clients fidèles

Nous avons développé un graph représentant les différentes localisations des entités, permettant aux utilisateurs de sélectionner des régions spécifiques et de calculer le plus court chemin entre elles. Cette approche offre une vision géospatiale dynamique et interactive des connexions entre différentes zones, facilitant ainsi la prise de décisions stratégiques en matière de gestion des infrastructures. Les acteurs de la télécommunication peuvent tirer un grand bénéfice de cette fonctionnalité en optimisant l'architecture de leur réseau et en ajustant leurs services aux besoins spécifiques de chaque région.

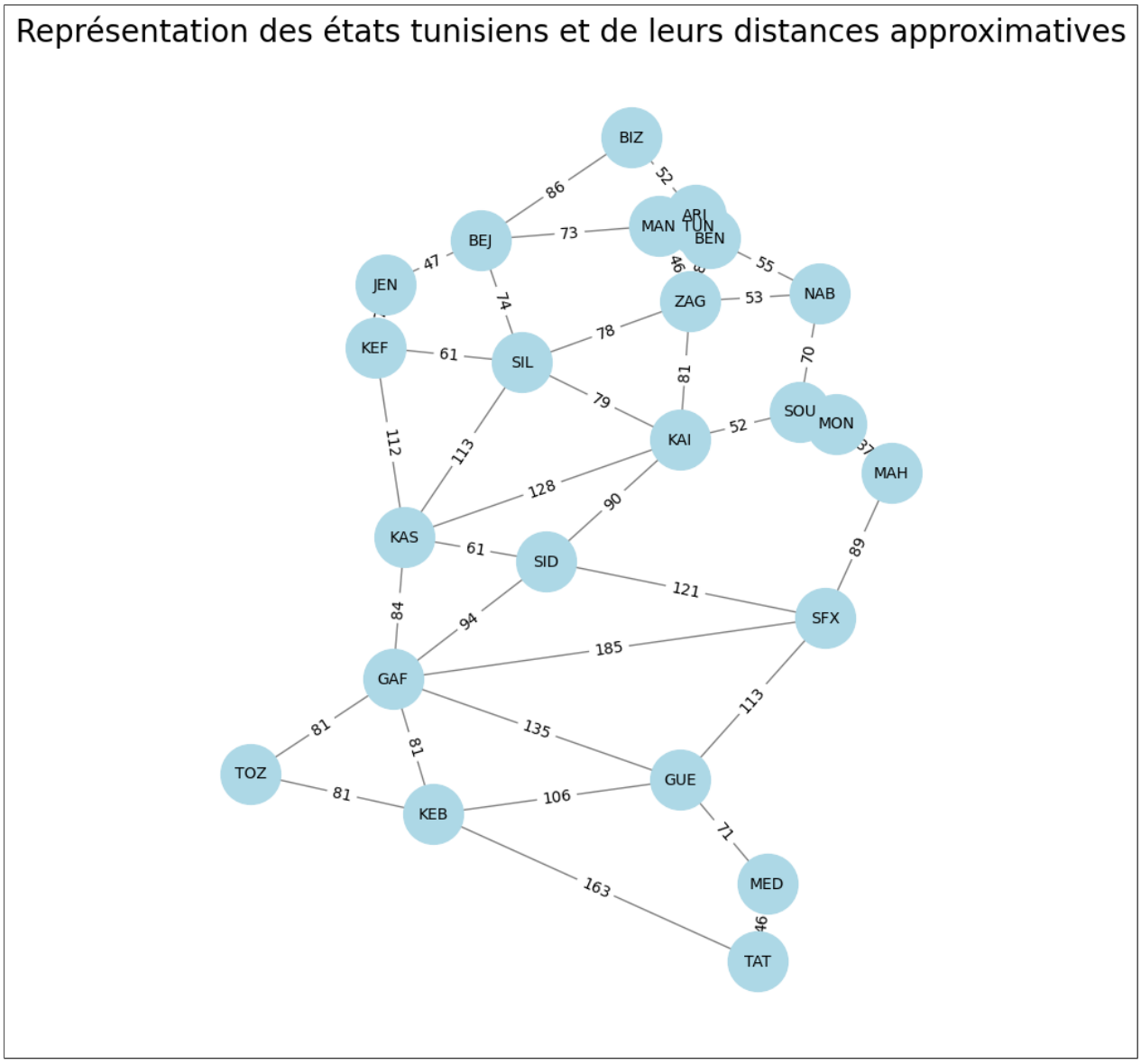

Figure 6.3 : Graph de localisation des entités

Par exemple, pour le déploiement de nouvelles infrastructures, cette fonctionnalité permet d'identifier les itinéraires les plus efficaces pour la construction de nouvelles antennes ou centres de données, en prenant en compte la distance entre les sites existants et les nouvelles zones à couvrir. Cela permet de réduire les coûts liés au transport des équipements et à la mise en place des nouvelles installations. De plus, en optimisant les trajets entre les différentes entités, les opérateurs peuvent également minimiser les délais d'installation, ce qui permet de répondre plus rapidement aux demandes croissantes des clients.

Dans la gestion des flux de données, cette fonctionnalité aide à analyser les connexions entre régions et à optimiser la distribution du trafic pour éviter les goulets d'étranglement. Par

exemple, si une région subit une augmentation de la consommation de données, les opérateurs peuvent utiliser ces informations pour réorganiser les flux et renforcer les capacités des infrastructures existantes, en redirigeant le trafic vers des canaux moins saturés. Ci-dessous se trouve un exemple d'utilisation.

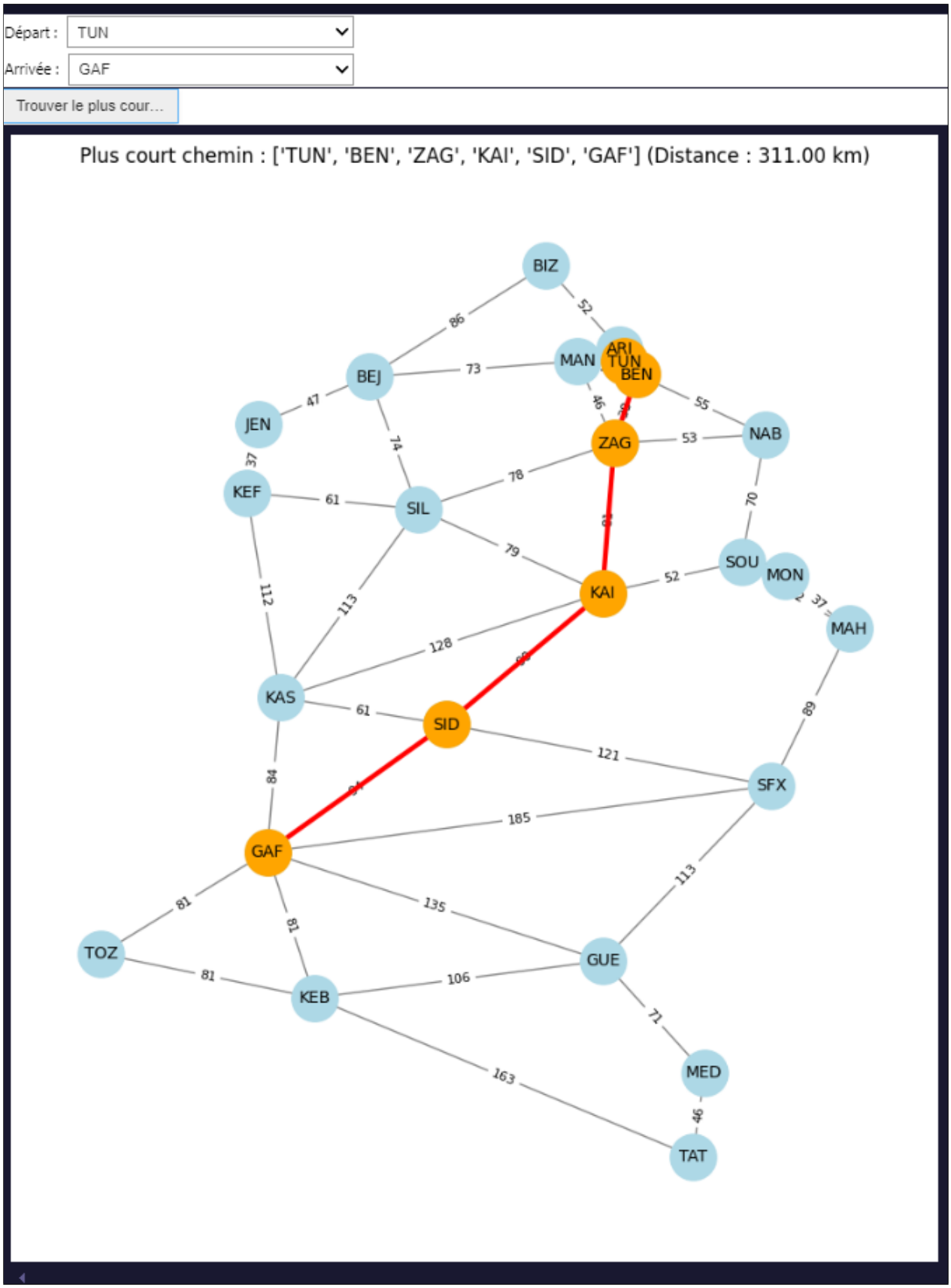

Figure 6.4 : Application : Graph de localisation des entités

Enfin, cette fonctionnalité peut être utilisée pour optimiser les coûts d'entretien des réseaux en identifiant les distances entre les antennes et les centres de données. Les opérateurs peuvent ainsi prioriser les zones qui nécessitent le plus d'attention en fonction de leur proximité géographique, réduisant ainsi les coûts logistiques et les délais de maintenance. En résumé, cette fonctionnalité géospatiale permet aux entreprises de télécommunications d'améliorer considérablement l'efficacité de leurs réseaux, en facilitant la planification, l'optimisation et l'entretien des infrastructures.

6.4 Conclusion

En conclusion, l'implémentation d'une base de données en graph et la représentation des localisations géographiques des entités offrent une multitude de bénéfices pour les acteurs du secteur des télécommunications. Cette approche permet non seulement d'optimiser la gestion des infrastructures et des flux de données, mais aussi d'améliorer les stratégies de fidélisation des clients en ciblant de manière précise les utilisateurs les plus actifs. L'optimisation des itinéraires pour le déploiement de nouvelles infrastructures, la gestion des coûts d'entretien et l'amélioration de l'efficacité des réseaux sont autant d'avantages qui découlent de l'exploitation de ces données géospatiales. En combinant ces outils avec des stratégies de récompense adaptées, les opérateurs peuvent renforcer la loyauté de leurs clients et maximiser la performance de leur réseau. Cette approche innovante constitue ainsi un levier stratégique essentiel pour répondre aux enjeux actuels du secteur.

Conclusion générale

Ce projet a permis de mener une analyse approfondie des données clients dans l'industrie de la télécommunication, un secteur particulièrement sensible aux enjeux de rétention de la clientèle. L'objectif principal de cette étude était d'explorer les relations et patterns cachés dans ces données afin de mieux comprendre les comportements des clients et d'optimiser les stratégies de fidélisation. Pour cela, nous avons structuré les informations de manière optimale, ce qui a permis de créer des modèles prédictifs fiables pour anticiper les actions des clients, telles que la résiliation ou l'engagement dans des services spécifiques.

En utilisant des techniques d'analyse avancées, telles que l'analyse univariée, bivariée et multivariée, nous avons pu identifier des corrélations importantes entre divers paramètres, comme le volume des recharges, l'utilisation des services, les contacts entrants et sortants, ainsi que les revenus générés. Ces analyses ont permis de mettre en évidence les comportements typiques des clients susceptibles de résilier leurs contrats, ce qui constitue une information clé pour anticiper et contrer les risques de churn. De plus, en appliquant des méthodes de séries temporelles pour la prédiction, nous avons pu observer des tendances évolutives dans le temps, telles que la diminution progressive de l'utilisation des services ou une faible fréquence de recharge, qui peuvent être des indicateurs précoces de désengagement.

En extrayant ces **insights** significatifs, nous avons enrichi la compréhension des comportements des clients, ce qui permet aux entreprises de mieux cibler leurs efforts pour **maintenir la fidélité des clients**. Par exemple, des offres personnalisées, des notifications de réengagement ou des promotions ciblées peuvent être mises en place pour les clients à risque de churn. Cela permet d'optimiser non seulement la rétention, mais aussi l'**expérience client**, en offrant une attention plus ciblée et des services adaptés aux besoins spécifiques de chaque segment.

En conclusion, ce projet met en lumière l'importance de l'analyse des données dans le secteur de la télécommunication pour **renforcer la rétention des clients**. La capacité à identifier des signes précoces de désengagement et à anticiper les besoins des clients constitue un atout stratégique majeur pour toute entreprise souhaitant non seulement fidéliser ses utilisateurs, mais aussi offrir des services de qualité, adaptés aux attentes évolutives des consommateurs.