Rapport Data Mining
Année 2024-2025

Composition du groupe et répartition :

Pour la réalisation de ce travail, nous sommes un groupe de 2 personnes :

- **Manal Bouffa** (p2313532) — Étudiante en Data Science
- **Alexandre Faure** (p2006157) — Étudiant en Intelligence Artificielle

Répartition des tâches :

Nous avons effectué l'ensemble des tâches de manière collaborative. Le travail a donc été réalisé de façon équitable. Nous avons utilisé Google Colab pour coder, ce qui explique l'absence de commits sur Git.

Les données

Description des données :

Notre jeu de données provient du site Kaggle et porte sur les liens d'"amitié" du réseau social Twitter (maintenant appelé X).
Adresse du dataset : Twitter Friends Dataset

Le dataset contient 10 colonnes, décrites comme suit :

- **id** : Identifiant unique pour chaque utilisateur.
- **screenName** : Nom d'utilisateur ou pseudonyme utilisé sur la plateforme.
- **tags** : Liste de hashtags associés aux tweets de l'utilisateur, reflétant les sujets ou tendances abordés.
- **avatar** : URL de l'image de profil de l'utilisateur.
- **followersCount** : Nombre total d'abonnés de l'utilisateur.
- **friendsCount** : Nombre total de comptes que l'utilisateur suit.
- **lang** : Langue préférée ou principale utilisée par l'utilisateur dans ses tweets.
- **lastSeen** : Timestamp indiquant la dernière activité en ligne de l'utilisateur.
- **tweetId** : Identifiant unique d'un tweet spécifique.
- **friends** : Liste des identifiants des amis ou comptes suivis par l'utilisateur.

Nous avons réaliser une visualisation de la data "brut" afin de voir ce qu'il en ressortait (cf. Premiere_vue_des_donnees.ipynb)

Problèmes liée au données :

Lors de notre travail sur les données issues de Twitter, plusieurs problèmes ont été identifiés. Tout d'abord, la colonne "friends" présentait des virgules pour séparer les identifiants des amis. Cette structure a conduit, lors de l'ouverture du fichier, à la création de nouvelles colonnes pour chaque identifiant, ce qui a dispersé les données et généré un nombre variable de colonnes pour chaque ligne. De plus, ces colonnes étaient dépourvues de titres, ce qui a compliqué davantage le traitement. Ensuite, la grande taille de notre jeu de données a posé des problèmes majeurs. Lorsque nous avons tenté d'appliquer directement des méthodes de clustering, les ressources limitées sur Colab ont provoqué des échecs : bien que l'exécution initiale fonctionne, une fois les méthodes de clustering appliquées, Colab ne reconnaissait plus les étapes précédentes. Sur VSCode, l'exécution a causé une surcharge du système, entraînant la fermeture complète de l'ordinateur et rendant son utilisation impossible pendant un certain temps. Enfin, nous avons constaté la présence de plusieurs colonnes inutiles. Certaines avaient des valeurs identiques pour toutes les lignes, comme la langue, tandis que d'autres contenaient des liens inaccessibles, comme les URLs des photos de profil. Ces problèmes ont nécessité un travail significatif de nettoyage et de préparation des données avant de poursuivre nos analyses.

Méthode 1 :

Pour résoudre les problèmes liés à la structure et à la taille des données TwitterFriends, nous avons suivi une approche méthodique combinant nettoyage, transformation et exploration. La colonne "friends" a été modifiée pour remplacer les virgules par des points-virgules, tandis que les colonnes non pertinentes comme "screenName", "tags", "avatar", "lang", "tweetId" et "lastSeen" ont été supprimées, simplifiant ainsi les données et réduisant leur complexité. Nous avons construit une matrice binaire où chaque colonne représente un identifiant unique d'ami et chaque ligne un utilisateur, avec une valeur de 1 indiquant une amitié. Cette transformation, bien qu'efficace pour structurer les données, a produit une matrice massive (100 lignes x 196 660 colonnes) qui, même après avoir testé des tailles supérieures comme 1000 lignes, s'est avérée trop lourde pour des méthodes comme le clustering. Afin d'optimiser la mémoire, nous avons utilisé des matrices sparse avec `lil_matrix` de SciPy et expérimenté les options de pandas comme `pd.DataFrame.sparse.from_spmatrix`, mais ces techniques n'ont pas suffi à réduire significativement la taille des données. La limitation à 100 lignes s'est avérée nécessaire pour garantir la stabilité des calculs. Des techniques comme la réduction dimensionnelle via PCA ou la segmentation préalable des données n'ont pas donné les résultats attendus. Enfin, nous avons visualisé les relations utilisateurs à l'aide d'un graphe NetworkX, où chaque nœud représentait un utilisateur et chaque lien une relation d'amitié, cette étape servant principalement à tester la cohérence des données et à vérifier la bonne représentation des connexions.

Méthode 2 :

Dans cette méthode de traitement des données TwitterFriends, une approche systématique a été adoptée pour nettoyer, transformer et exploiter les informations brutes. Les données ont été nettoyées et ciblées des colonnes essentielles. La colonne *friends*, contenant des listes d'identifiants encodées en JSON, a été convertie en listes Python exploitables via « [json.loads(x) for x in df.friends] », préparant ainsi les données relationnelles pour les analyses. La colonne *id* a été nettoyée pour convertir les identifiants en entiers avec « [int(x.replace("","")) for x in df.id] », et les noms d'utilisateurs dans *screenName* ont été uniformisés en supprimant les guillemets superflus. Afin de réduire la complexité, un filtre a été appliqué pour isoler les utilisateurs ayant moins de 300 amis, en utilisant « df[df.friendsCount < 300] ». Ce critère a permis de limiter le volume de données tout en conservant un sous-ensemble pertinent pour des analyses exploratoires.

Méthodes appliquer et résultats

UNSUPERVISED ML (Clustering)

Le but de cette méthode était de regrouper les utilisateurs de Twitter en segments homogènes selon leurs interactions et caractéristiques. En appliquant des techniques de réduction de dimensionnalité et des algorithmes de clustering, nous avons mis en évidence des comportements partagés, favorisant des analyses exploratoires approfondies.

Les données ont d'abord été prétraitées et normalisées pour équilibrer le rôle de chaque variable. Ensuite, des techniques de réduction de dimensionnalité, telles que PCA, MDS, Isomap et t-SNE, ont simplifié les données tout en conservant leur structure essentielle, permettant une visualisation en deux dimensions. La méthode du coude a servi à déterminer le nombre optimal de clusters en se basant sur l'inertie intra-cluster.

Plusieurs algorithmes de clustering ont été comparés : K-Means, K-Medoids, DBSCAN et le modèle de mélange gaussien (GMM). K-Means et K-Medoids ont bien séparé les clusters avec des scores de silhouette élevés (k=3). K-Means a été plus rapide, tandis que K-Medoids a mieux géré les outliers. DBSCAN a échoué à identifier plusieurs clusters significatifs, indiquant une faible densité des données ou des paramètres inadéquats. GMM, plus flexible pour des formes complexes, a capturé efficacement la structure probabiliste des clusters, bien que sa qualité dépend du type de covariance choisi.

Globalement, K-Means était le plus efficace, tandis que GMM convient mieux à des clusters non linéaires ou de formes variées. Cette analyse souligne l'importance de sélectionner l'algorithme en fonction des propriétés des données et des objectifs.

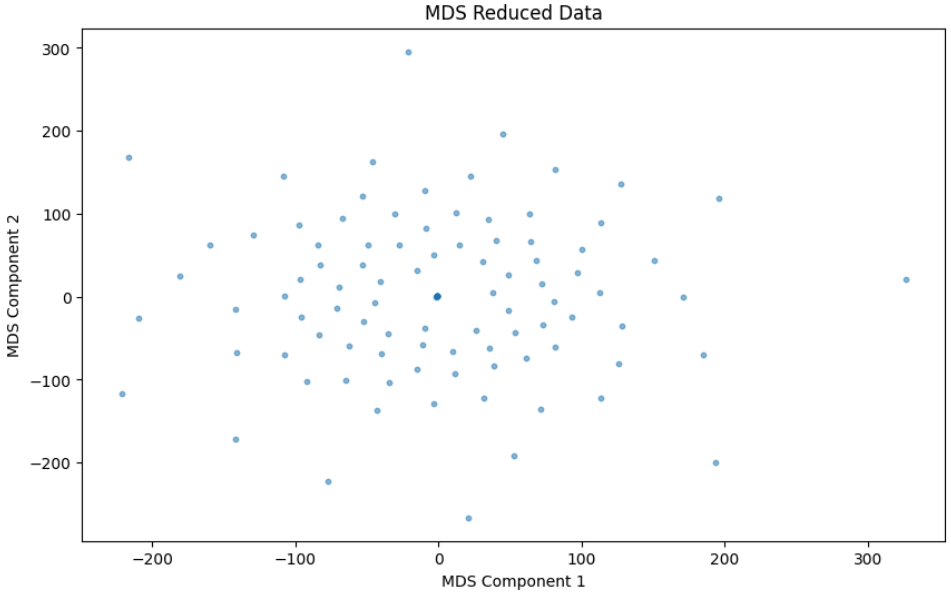

Figure 1 : Réduction des données avec MDS

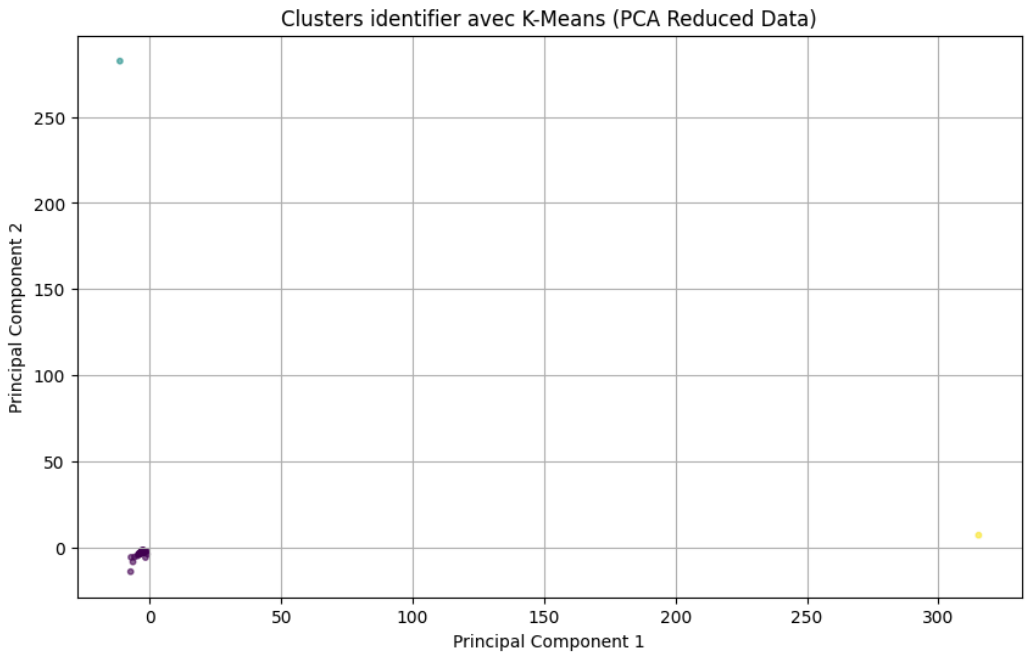

figure 2 : Cluster avec K-Means

NETWORK DATA MINING

Avec cette méthode, nous avons exploré la structure relationnelle des données en construisant un graphe social où chaque utilisateur était représenté comme un nœud et leurs amitiés comme des liens. Cela a permis d'analyser la connectivité du réseau, d'identifier les utilisateurs influents et de comprendre les dynamiques communautaires.

Un réseau complexe de 19 757 nœuds et 21 394 arêtes a été construit, modélisant les relations étudiées. La visualisation d'un sous-graphe a offert une première compréhension graphique des connexions locales. L'étude des distributions des degrés entrants et sortants a mis en évidence une forte asymétrie, typique des réseaux réels. L'analyse du coefficient de clustering a révélé une absence de triangles et triades, avec des coefficients faibles, signalant une densité locale et globale réduite. Les mesures de centralité, comme celles de degré, proximité et médiation, ont montré une faible connectivité et influence globale. Les calculs de distances, du diamètre et des longueurs moyennes de chemins ont confirmé un graphe partiellement connecté, avec une grande composante principale entourée de petites composantes isolées.

Les k-cores ont révélé une faible densité globale avec un maximum de coreness de 5. Les algorithmes de détection de communautés, Girvan-Newman et Louvain, ont identifié des groupes de nœuds fortement connectés, clarifiant la structure du réseau. L'analyse du "rich-club" a mis en lumière des tendances de connectivité entre nœuds à haut degré. En testant la résilience via des attaques aléatoires et ciblées, la suppression de nœuds critiques a rapidement fragmenté le réseau, soulignant leur rôle dans sa stabilité. Enfin, des mesures comme le PageRank et la centralité des vecteurs propres ont identifié les nœuds les plus influents.

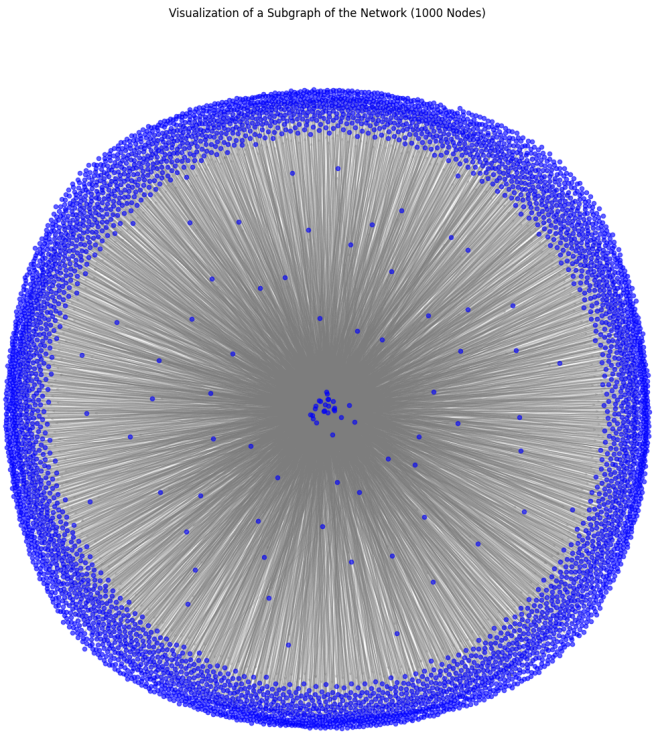

Figure 3 : sous-graphe de 1 000 noeuds

COMMUNITY DETECTION (GRAPH CLUSTERING)

Cette méthode a été utilisée pour identifier des sous-groupes d'utilisateurs fortement connectés dans le graphe social, visant à découvrir des communautés naturelles et mieux comprendre les regroupements et interactions des utilisateurs dans cet écosystème numérique.

L'analyse se concentre sur un sous-graphe de 10 000 nœuds pour limiter les temps de calcul. Pour la détection de communautés, l'algorithme de Louvain a révélé 9 900 communautés avec une modularité élevée de 0.9329 en 3.1 secondes, démontrant son efficacité pour détecter des structures complexes. Infomap, avec un temps d'exécution plus rapide (0.25 seconde), a identifié un nombre similaire de communautés (9 902), confirmant sa performance dans des environnements exigeant une exécution rapide. Les visualisations colorées associées clarifient les partitions et les relations entre nœuds.

Pour la prédiction des liens, les algorithmes Common Neighbors, Adamic-Adar Index et Jaccard Coefficient ont été utilisés. Common Neighbors a prédit des connexions potentielles avec un score de 1 pour les meilleures prédictions en 847 secondes. L'index d'Adamic-Adar a atteint un score maximal de 1.4427 pour ses meilleures prédictions, avec des temps d'exécution très rapides (0.23 seconde dans certains cas). Le coefficient de Jaccard a obtenu un score de 1, mais avec un calcul plus long (660 secondes).

Cependant, les métriques d'évaluation comme AUC-ROC et AUC-PR n'ont pas pu être calculées, faute de diversité suffisante dans les données de test. Cela met en lumière une limite dans les configurations actuelles pour évaluer quantitativement ces méthodes.

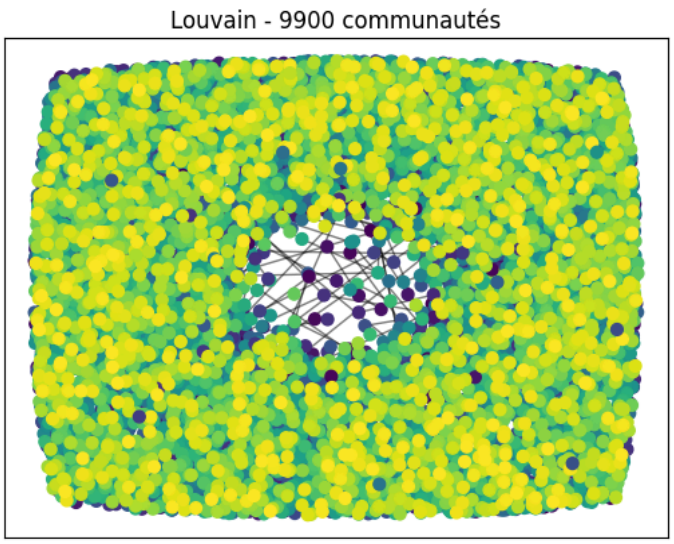

Figure 4 : détection de communautés avec Louvain

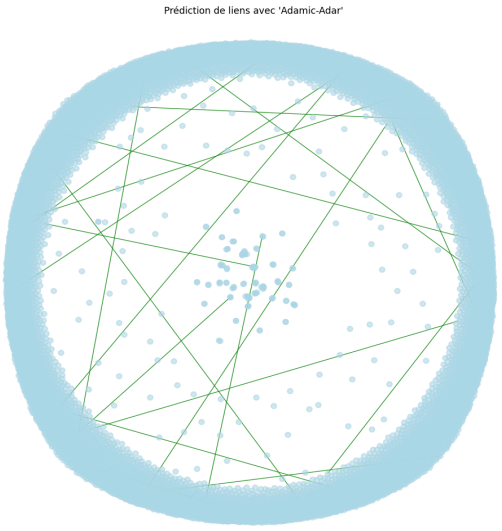

Figure 5 : prédiction de lien avec Adamic-Adar

MATRIX-FACTORIZATION (RECOMMENDER-SYSTEMS BI-CLUSTERING)

L'objectif de cette méthode était de développer un système de recommandation adapté à notre jeu de données TwitterFriends. En exploitant les relations entre utilisateurs et amis, des techniques comme la factorisation matricielle et le bi-clustering ont été utilisées pour proposer des recommandations pertinentes, tout en explorant les liens implicites.

Des algorithmes comme le filtrage collaboratif, la factorisation matricielle, le clustering et des méthodes hybrides ont été utilisés. La préparation des données a inclus la vectorisation des balises avec TF-IDF pour analyser les similarités et la similarité cosinus pour identifier des préférences relatives dans les recommandations basées sur le contenu.

Le filtrage collaboratif a utilisé une matrice utilisateur-item avec deux variantes : une approche utilisateur (User-based KNN) s'appuyant sur des distances cosinus, et une approche item (indice de Jaccard) évaluant la similarité entre items. Ces méthodes exploitent les préférences collectives pour enrichir les suggestions.

La factorisation matricielle par SVD a révélé des facteurs latents influençant les interactions utilisateur-item. Les matrices reconstruites permettent de prédire des recommandations visualisées en cartes thermiques. Les prédictions ont été évaluées avec des métriques telles que précision, rappel, RMSE et AUC, complétées par des courbes ROC et précision-rappel.

Pour traiter le problème du « cold start », une méthode hybride combine des recommandations populaires avec celles issues du filtrage collaboratif, améliorant couverture et robustesse. Des techniques de clustering et co-clustering (K-Means) ont segmenté la matrice utilisateur-item en groupes similaires, visualisés par des histogrammes et graphiques de distribution.

Ce cadre méthodologique a produit des recommandations concrètes et diversifiées, telles que « #narcos » ou « #backtohogwarts », illustrant l'efficacité des techniques déployées pour répondre aux défis des systèmes de recommandation.

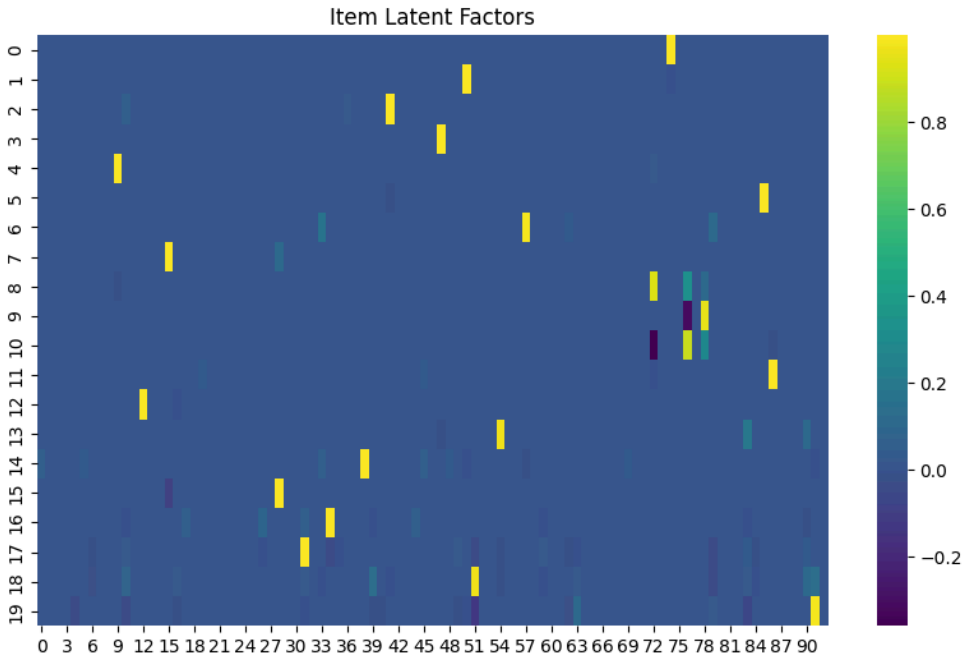

Figure 6 : Matrice des Facteurs Latents des Items

DATA TRANSFORMATION

Cette méthode vise à préparer et transformer les données issues de Twitter pour les rendre exploitables. En structurant les informations complexes (listes d'identifiants d'amis, hashtags) et en appliquant des techniques de réduction dimensionnelle comme le PCA et le t-SNE, nous avons extrait des caractéristiques significatives tout en simplifiant la représentation des données, facilitant des analyses telles que le clustering ou la détection de communautés.

Les données ont été nettoyées et standardisées avec "StandardScale" pour garantir leur homogénéité. Le PCA a réduit la dimensionnalité tout en préservant l'essentiel de l'information, et l'analyse de la variance expliquée a permis de sélectionner les composantes principales les plus significatives. Parallèlement, le t-SNE a réduit les données à deux dimensions pour une visualisation simplifiée. Un clustering avec K-Means (3 clusters) a été effectué, avec un score de silhouette satisfaisant, démontrant une bonne séparation. Les clusters ont été visualisés en deux dimensions, montrant des groupes distincts.

Une analyse de similarité a généré un graphe relationnel des données, révélant des connexions importantes pour un seuil de similarité >0.7. En complément, des embeddings textuels avec Word2Vec ont mis en évidence des relations entre termes, interprétées via une visualisation en espace réduit.

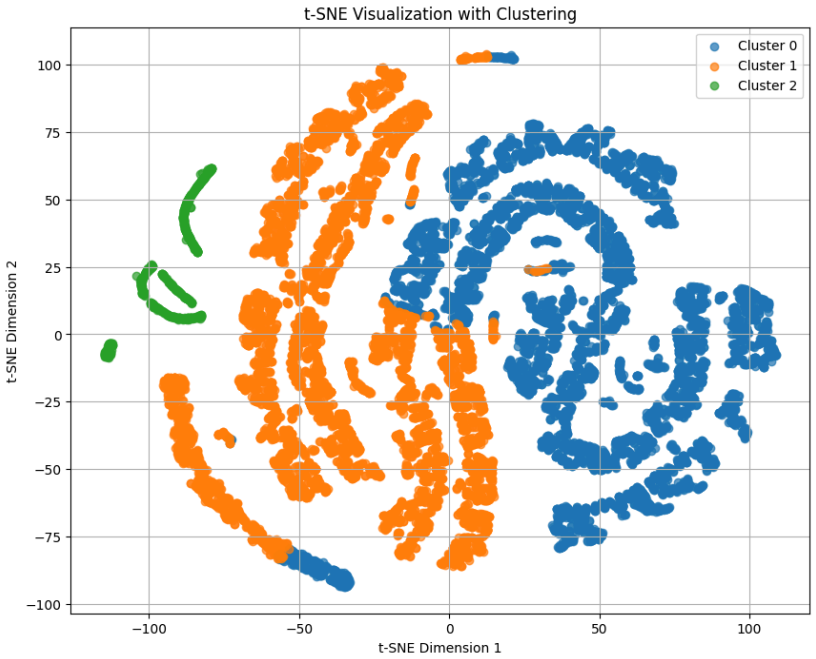

Figure 7 : Clustering des données avec la méthode t-SNE

Analyse des Patterns Fréquents

Dans cette méthode, nous avons cherché à identifier des associations récurrentes dans les données Twitter, que ce soit entre hashtags ou entre listes d'amis. En révélant des motifs significatifs, cette analyse visait à enrichir notre compréhension des comportements et des connexions sociales des utilisateurs.
Pour explorer les données et identifier des patterns récurrents,nous avons réaliser un prétraitement comme expliqué dans la méthode 2 ci-dessus, et nous avons appliqué la méthode "apriori" sur deux dimensions principales : les tags (avec 10000 ligne) et les amis (friends) (avec 5000 ligne).

- Étude des Tags :
  Nous avons d'abord cherché à analyser les associations fréquentes entre différents tags pour déterminer si certains étaient régulièrement utilisés ensemble. Cependant, les résultats obtenus n'ont révélé aucun pattern significatif. Les taux de support étaient très faibles, ce qui suggère une faible corrélation entre les tags.
- Analyse des Amis (Friends) :
  Nous nous sommes ensuite concentrés sur les données relatives aux amis pour identifier si certains individus apparaissent fréquemment ensemble dans les transactions. En appliquant la méthode "apriori" sur cette dimension, des groupes d'amis récurrents ont été détectés, mettant en évidence des associations intéressantes :
  - Certains amis sont régulièrement trouvés ensemble, formant des groupes potentiellement interconnectés ou des communautés sociales denses.
  - Les résultats ont permis d'extraire des règles d'association robustes avec des métriques de confiance et de lift élevées, indiquant des relations fortes entre ces individus.

Ces analyses montrent que l'exploration des relations entre amis fournit des insights plus significatifs que celle des tags. Cela met en lumière la valeur des relations sociales dans la structuration des données étudiées.

SPATIAL DATA ANALYSIS

Bien que les données ne soient pas explicitement géographiques, cette méthode visait à explorer les relations spatiales implicites entre utilisateurs, offrant une perspective complémentaire pour identifier des corrélations liées à la localisation ou à d'autres dimensions inexploitées.

Un graphe statique non orienté a modélisé les connexions entre utilisateurs, permettant d'extraire des métriques comme le nombre de nœuds, d'arêtes, la densité et le coefficient de clustering moyen, et d'analyser la distribution des degrés à travers des visualisations détaillées.

Les réseaux dynamiques ont été construits en décomposant le graphe en snapshots avec des intervalles fixes et des fenêtres glissantes, afin de capturer l'évolution des interactions. Des métriques dynamiques, telles que la densité moyenne, le coefficient de clustering moyen et la taille des voisinages temporels, ont été calculées pour chaque intervalle, révélant des tendances significatives à travers des visualisations. La détection de communautés, grâce à un algorithme basé sur la modularité, a identifié des clusters de nœuds. Une analyse des interactions a révélé la fréquence des connexions entre nœuds, visualisée par un histogramme, avec une densité des arêtes de 0,039.

L'analyse des flux d'information, réalisée avec l'algorithme PageRank, a identifié les 10 nœuds les plus influents, avec un score maximal de 6,37e-6, les résultats étant présentés sous forme graphique pour illustrer les dynamiques d'influence. Cette étude a permis de caractériser les propriétés statiques, les interactions et l'évolution temporelle du réseau, offrant une compréhension détaillée et visuelle de ses structures et de ses corrélations implicites.

Conclusion et Discussion :

Cette étude a permis d'explorer les données issues du réseau social Twitter, en mettant en lumière à la fois leur potentiel analytique et leurs limites intrinsèques. Les données disponibles, bien qu'intéressantes dans leur concept, se sont révélées imparfaites, avec de nombreux problèmes de structure, de pertinence et de diversité. Ces contraintes ont parfois compliqué leur exploitation et limité la portée des analyses. Cependant, malgré ces défis, des observations utiles sur les relations et les interactions entre utilisateurs ont pu être dégagées, offrant un aperçu partiel mais significatif des dynamiques sociales présentes dans ces données.

Il apparaît clairement que la qualité des données joue un rôle déterminant dans la réussite des analyses. Les lacunes constatées, telles que des colonnes redondantes ou peu informatives, ainsi que des formats inadaptés, soulignent l'importance d'un prétraitement rigoureux et d'une collecte de données plus ciblée. Ces enseignements mettent en avant la nécessité de travailler sur des bases de données mieux structurées pour obtenir des résultats plus exploitables et des interprétations plus robustes. Enfin, cette étude offre des pistes pour aborder les défis similaires dans le cadre d'analyses de réseaux sociaux, tout en posant les bases d'un travail futur sur des données mieux adaptées aux objectifs d'exploration et d'interprétation.