UNIVERSITE GALATASARAY
FACULTÉ D'INGÉNIERIE ET DE TECHNOLOGIE

SEYREK ETİLEŞİMLİ VERİ İÇİN GRAF TABANLI HİBRİT ÖNERİ SİSTEMİ

(SYSTEME DE RECOMMANDATION HYBRIDE BASE SUR LES GRAPHES POUR DES DONNEES D'INTERACTION RARES)

PROJET DE FIN D'ETUDES

Doğa Yağmur YILMAZ

Département : GENIE INDUSTRIEL
Directrice du projet de fin d'études : Prof. Dr. S. Emre ALPTEKIN

MARS 2024

PRÉFACE

Je tiens à remercier mon professeur S. Emre Alptekin qui a contribué à la formation de mon projet de fin d'études.

Doğa Yağmur Yılmaz

Mars, 2024

TABLE DES METIÈRES

PRÉFACE 2
RESUME 5
ÖZET 7
1.INTRODUCTION 10
2. REVUE DE LITTERATURE 13
3. LES NOTIONS FONDAMENTALES 17
3.1. TYPES DE SYSTEMES DE RECOMMANDATION 17
3.1.1. Filtrage Collaboratif 17
3.1.2. Filtrage Basé Sur Le Contenu 17
3.1.3. Approches Hybrides 18
3.1.3.1. Modèles De Recommandation Basés Sur L'apprentissage Profond 18
3.1.3.1.1. La Recommandation Avec Les Blocs Neuronaux 18
3.1.3.1.2. La Recommandation Avec Les Modèles Hybrides Profonds 18
3.1.3.2. Modèles De Recommandation Basés Sur L'Autoencodeur 19
3.1.3.3. Système De Recommandation Hybride Basé Sur Les Graphes 19
3.1.3.3.1. Caractéristiques Basées Sur Les Graphes 20
3.2. CONCEPT CLES ET DEFIS 22
3.2.1. Démarrage à Froid 22
3.2.2. Rareté 22
3.2.3. Évolutivité 22
3.2.4. Diversité et Sérendipité 22
3.2.5. Confidentialité des Données Personnelles 22
4. METHODOLOGIE ET MODELE 24
4.1. ARCHITECTURE 24
4.2. ALGORITHME PROPOSE 26
4.3. TECHNOLOGIES A UTILISER 28
4.4. METHODES A UTILISER POUR LE GROUPEMENT DES UTILISATEURS 29
4.4.1. Algorithm de K-Means 29
4.4.2. Méthode d'Elbow 29
4.4.3. Méthode De La Silhouette Moyenne 30
5. APPLICATION DU MODELE PROPOSE 31
5.1. DATASET 31
5.2. CARACTERISTIQUES UTILISEES POUR L'UTILISATEUR 32
5.3. ETAPES APPLIQUE 33

RESULTATS 38
BIBLIOGRAPHIE 39

RESUME

Le succès dans le secteur des services est largement déterminé par l'expérience utilisateur, ce qui met en lumière l'importance des systèmes de recommandation (RS) personnalisés offrant des suggestions adaptées aux besoins et aux demandes des utilisateurs pour améliorer leur satisfaction. Les RS établissent des liens entre les produits et/ou les utilisateurs, offrant des recommandations en fonction des comportements ou des intérêts des utilisateurs, et contribuant ainsi à personnaliser l'expérience utilisateur. Ils se classent généralement en trois catégories : les modèles de Filtrage Collaboratif (CF), les systèmes de Filtrage Basé sur le Contenu (CBF) et les modèles hybrides combinant les deux. Les CF, exploitant l'historique de notation des utilisateurs pour fournir des recommandations personnalisées, peuvent être basés sur les utilisateurs ou les produits. Les CBF, en revanche, recommandent des produits en fonction de leurs caractéristiques et du profil créé par les interactions passées de l'utilisateur. Bien que les modèles CF dominent traditionnellement, les approches hybrides deviennent populaires pour combiner l'efficacité du CF avec les informations contextuelles du CBF. En ce qui concerne la gestion des données rares, un défi majeur dans les RS, l'apprentissage profond, en particulier le sous-domaine de l'apprentissage sur les graphes, montre des avantages significatifs. Les modèles de recommandation basés sur les graphes utilisent des autoencodeurs pour prédire les notations manquantes et combinent des techniques de CF avec des informations contextuelles pour offrir des recommandations plus précises. L'utilisation de l'apprentissage profond permet de capturer des relations non linéaires entre les utilisateurs et les produits, tandis que la représentation des données sous forme de graphes aide à comprendre les liens entre les nœuds. Des applications réussies, telles que les systèmes de recommandation basés sur les DNN pour YouTube et les RNN pour Yahoo News, illustrent cette approche.

Le projet vise à développer un système de recommandation hybride basé sur les graphes, utilisant l'apprentissage profond pour résoudre le problème du démarrage à froid et des données rares. En se concentrant sur un système de recommandation de films, il aspire à améliorer l'expérience utilisateur en offrant des recommandations personnalisées, en utilisant des techniques telles que les autoencodeurs et l'apprentissage des graphes pour prédire avec précision les préférences des utilisateurs. En combinant ces méthodes avec des algorithmes de CF, le projet vise à surmonter les défis liés à la rareté des données et à garantir une expérience utilisateur satisfaisante pour les nouveaux utilisateurs.

L'architecture du système de recommandation proposé repose sur plusieurs étapes. Tout d'abord, un graphe est construit en fonction du nombre d'utilisateurs, reliant les utilisateurs similaires en fonction de leurs interactions passées. Ensuite, diverses mesures de centralité, telles que le PageRank et la centralité de degré, sont calculées pour chaque utilisateur à partir de ce graphe de similarité. Ces informations sont ensuite combinées avec des données contextuelles, telles que l'âge et le genre, pour offrir des recommandations personnalisées aux utilisateurs. En parallèle, un autoencodeur est utilisé pour extraire de nouvelles caractéristiques des données et réduire leur dimensionnalité, avant d'effectuer une classification des utilisateurs en clusters à l'aide de l'algorithme K-means. Les nouveaux utilisateurs sont ensuite assignés à des clusters en

fonction de leurs caractéristiques encodées, et des prédictions de notation sont faites pour les nouveaux éléments en fonction de leur similarité avec d'autres éléments. Enfin, les taux estimés de tous les éléments pour chaque utilisateur sont calculés en fonction de la notation moyenne de leur cluster respectif.

ÖZET

Kullanıcı deneyimi, hizmet sektöründe başarıyı belirleyen en önemli faktörlerden biri olarak karşımıza çıkmaktadır. Bu bağlamda, kullanıcı memnuniyetini arttırmak için kullanıcıların ihtiyaç ve taleplerine uygun kişiselleştirilmiş öneriler sunan öneri sistemleri (RS), kullanıcı deneyimini iyileştirmede önemli bir rol oynamakta ve e-platformlarda sıkça kullanılmaktadır. Öneri sistemleri temelde ürünler ve/veya kullanıcılar arasında bağlantılar kurarak büyük ve karmaşık verilerle çalışabilen sistemlerdir. Kullanıcıların davranışlarına veya ilgi alanlarına göre çeşitli ilişkiler kurarak kullanıcılara önerilerde bulunur ve kullanıcı deneyimini kişiselleştirmeye yardımcı olurlar. Öneri sistemleri üç farklı kategoride incelenmektedir. İlk olarak İşbirlikçi Filtreleme (CF) modelleri, kişiselleştirilmiş bir öneri sağlamak için kullanıcıların ürünlere yönelik derecelendirme geçmişi hakkındaki bilgilerden yararlanmayı amaçlamaktadır. Geçmişte aynı fikirde olan kullanıcıların gelecekte tekrar aynı fikirde olma ihtimalinin yüksek olduğunu varsayar. Başka bir deyişle, benzer kullanıcıların tercihlerine dayalı olarak ürünler önerir. İki ana işbirlikçi filtreleme türü vardır. İlki kullanıcı tabanlı işbirlikçi filtrelemedir. Bir hedef kullanıcıya, o kullanıcıya benzer diğer kullanıcıların beğendikleri ürünleri önerir. İkincisi Ürün tabanlı işbirlikçi filtrelemedir. Hedef kullanıcının daha önce beğendiği veya etkileşimde bulunduğu ürünlere benzer ürünler önerir. İşbirlikçi filtreleme, ürünler veya kullanıcılar hakkında tanımlayıcı bilgi gerektirmez, yalnızca kullanıcı-ürün etkileşimlerine dayanır. İkinci kategori olan içerik tabanlı filtreleme (CBF), ürünlerin özelliklerine göre ve kullanıcının tercihlerinin yarattığı profile dayalı olarak kullanıcılara ürünler önerir. Ürünlerin özelliklerine ve kullanıcının geçmiş etkileşimlerine odaklanır. Bu sistem, kullanıcının olumlu etkileşimde bulunduğu ürünlerin özelliklerine dayalı bir kullanıcı profili oluşturur. Özellikle çok kullanıcılı platformlarda daha verimli sonuçlar elde edildiğinden işbirlikçi filtreleme yöntemi içerik tabanlı filtreleme yönteminden daha fazla tercih edilmektedir çünkü CF yöntemi yalnızca kullanıcı derecelendirmelerine odaklanırken CBF yöntemi iyi sonuçlar vermek için ek(yan) bilgi gerektirir. Her iki yöntemde de çeşitli ortak problemlerle karşılaşılmaktadır. Sık karşılaşılan problemlerden birisi eldeki verinin seyrek etkileşimli veri olmasıdır. Yani her kullanıcının her bir ürünü veya yeterli sayıda ürünü derecelendirmemiş olması sebebiyle veri setinde hiç derecelendirilmemiş ürünler kalırken tüme yakın kullanıcının değerlendirdiği ürünler olması durumudur. Bu durumda eşit olarak dağılmamış (not uniformly distributed) bir veri seti ile baş edilmelidir. Elde edilen gerçek hayat veri setlerinden kullanıcı-ürün matrisleri oluşturulduğunda eksik veri noktalarından bu problem net bir şekilde görülmektedir. Bir diğer problem soğuk başlangıç olarak adlandırılan yeni kullanıcı geldiği durumda kullanıcının tanımlanarak kullanıcıya öneri yapıldığı senaryodur. Bu sorun birçok platformda yeni gelen kullanıcıya gösterilen ürün kümelerinden beğendiklerini seçmesi istenerek aşılmaya çalışılsa da idare edilir bir başlangıç çözümü sunmaktan öteye gidemez. Bu noktada işbirlikçi filtreleme ile içerik tabanlı filtreleme yöntemlerini birleştiren hibrit modeller üçüncü kategori olarak karşımıza çıkmaktadır. Bu sayede işbirlikçi filtrelemenin kullanıcı-ürün etkileşimi kullanılırken içerik filtrelemede kullanılan tanımlayıcı yani yan bilgiler de modelin eğitim sürecine dahil edilir. Şimdiye kadar olasılıksal matris faktorizasyonunu genişleten bazı başarılı yaklaşımlar uygulansa da genel çerçeveye bakıldığında geleneksel yaklaşımları domine eden bir yöntem

bulunmamaktadır. Öneri sistemlerinde kullanıcı-ürün etkileşimleri arasındaki karmaşık ilişkiyi yakalayabilmek için derin öğrenme algoritmaları sıklıkla denenmektedir. Derin öğrenme, doğrusal olmayan kullanıcı-ürün ilişkilerini yakalayabilir ve görsel, metinsel ve bağlamsal gibi farklı veri kaynaklarından gelen verilerin kendi içindeki karmaşık ilişkileri tanımlayabilmektedir. Son dönemlerde yapılan çalışmalarda öneri sistemleri geliştirilirken derin öğrenmenin alt dalı olan graf öğrenmesi kullanıldığı ve bu yöntemin verideki seyreklik probleminin etkisini azaltmaya pozitif etki sağladığı görülmüştür. Verilerin graf şeklinde yapılandırılması ve düğümler arasında oluşan bağlantıların sebebinin keşfedilmesi öğrenmeye katkı sağlamaktadır. Aynı zamanda seyrek veri sorunuyla baş etmek için derin öğrenme sağlayan autoencoder'ların eksik derecelendirmeleri tahmin etmek amacıyla işbirlikçi filtreleme yöntemiyle birlikte kullanıldığı modeller de karşımıza çıkmaktadır. Youtube için DNN ve Yahoo News için RNN tabanlı başarılı öneri sistemi uygulamaları bulunmaktadır. Tüm bu modeller geleneksel modeller üzerinde bir gelişme göstermiş olsa da varolan derin öğrenme modelleri kullanıcı derecelendirmesiyle yüksek korelasyona sahip kullanıcı veya ürüne ait yan (tanımlayıcı) bilgi özelliğini göz ardı etmektedir. Bu nedenle bu proje çalışmasında yan(tanımlayıcı) bilgileri ve bir derin öğrenme yöntemi olan Autoencoder kullanarak seyrek veri ve soğuk başlangıç problemlerine çözüm ürettiğimiz ve kullanıcı ile ürün arasındaki doğrusal olmayan ilişkiyi tanımladığımız bir öneri sistemi geliştirmeyi planlamaktayız. Bu bağlamda kullanıcıların tercihlerini, benzerliklerini ve tanımlayıcı (yan) bilgilerini benzersiz bir matriste birleştirerek çok kullanıcılı e-platformlarda sık kullanılan işbirlikçi filtrelemeden daha iyi sonuçlar elde etmeyi hedeflemekteyiz.

Projenin odak noktası, derin öğrenme kullanılarak soğuk başlangıç problemine çözüm sağlanan seyrek etkileşimli veri üzerinde graf tabanlı bir hibrit öneri sistemi geliştirmektir. Bu amaç doğrultusunda uygun veri setini elde etmek ve ilgi çekici bir alanda fikirlerimizi uygulamak için film öneri sistemi oluşturmayı tercih ettik. Sürekli gelişmekte olan film endüstrisine hizmet eden popüler film ve dizi yayınlama platformları geleneksel öneri sistemi yöntemlerini çeşitli yenilikler deneyerek kullanmakta ve bizim gibi geliştiriciler için belirli oranlarda verilerini paylaşmaktadırlar. Toplanan film ve kullanıcı verileri istenen şekilde kullanılmak üzere veri ön işleme ve veri hazırlığı aşamalarından geçirilerek modelde kullanılır.

Çalışmanın amacı kullanıcı deneyimini geliştirmek için seyrek etkileşimli veri üzerinde graf tabanlı bir hibrit öneri sistemi geliştirmektir. Projenin ana hedefi, kullanıcılara kişiselleştirilmiş doğru öneriler sunarak kullanıcı memnuniyetini artırmaktır. Bu doğrultuda proje; seyrek etkileşimli veri üzerinde derin öğrenme kullanarak öneri sistemi geliştirme, graf tabanlı yaklaşımların kullanıcı-ürün ilişkilerini yakalamak için etkili bir şekilde uygulanması, soğuk başlangıç problemine çözüm sunarak yeni kullanıcıların da kişiselleştirilmiş öneriler alabilmesini sağlama hedeflerine odaklanmaktadır. Derin öğrenme altında yer alan autoencoder ve graf öğrenmesi tekniklerini ve aynı zamanda işbirlikçi filtreleme ile tanımlayıcı bilgilerin kullanıldığı içerik tabanlı filtreleme yöntemlerini birleştirerek kullanıcıların tercihlerini daha doğru bir şekilde tahmin etmeyi ve öneri sistemlerinin performansını artırmayı amaçlamaktadır. Bu unsurlar projenin özgün değerini oluşturmaktadır. Ayrıca, soğuk başlangıç problemine çözüm sunarak yeni kullanıcıların da sistemi etkin bir şekilde kullanabilmesini sağlanır.

Öneri sistemi mimarimiz, graf tabanlı hibrit öneri sistemi adımlarını içeren bir yapıdan oluşmaktadır. İlk adımda kullanıcı sayısına göre bir graf oluşturulur ve benzerliklere dayalı olarak kullanıcılar birbirine bağlanır. Daha sonra her kullanıcı için benzerlik grafiğinden çeşitli bilgiler çıkarılır. Örneğin düğümlerin PageRank'ı, derece merkeziliği ve diğer merkezilik ölçüleri hesaplanır. Üçüncü adımda bu bilgilere yan bilgiler eklenir ve kullanıcıların en uygun filmleri alması sağlanır. Ardından Autoencoder kullanılarak yeni özellikler çıkarılır ve boyut azaltma gerçekleştirilir. Sonrasında yeni özellikler kullanılarak kullanıcı kümeleme yapılır ve her veri kümesi için uygun küme sayısı belirlenir. Daha sonra kodlanmış özelliklere dayalı olarak yeni kullanıcılar kümelere atanır ve diğer ögelerle benzerliklerine göre yeni öge derecelendirmesi yapılır. Son adımda ise her kullanıcı için tüm ögelerin tahmini oranları, kümenin ortalama derecelendirmesine göre hesaplanır. Bu uygulama adımlarını Graf ve Kümeleme olarak iki başlık altında topladığımızda kullanılacak yöntemlerin ana hatlarını şu şekilde özetleyebiliriz.

Graf bölümünde yer alan Page Rank, bir düğümün geçişsel etkisini veya bağlantısını ölçen bir algoritmadır ve genellikle web sayfalarını sıralamak için tasarlanmıştır. Derece Merkeziliği, bir düğümden gelen ve giden ilişkilerin sayısını ölçer ve bireysel düğümlerin popülerliğini bulmak için kullanılır. Yakınlık Merkeziliği, bir grafikte bilgiyi etkili bir şekilde yayılabilen düğmeleri tespit etmek için bir yol sunar. Aracılık Merkeziliği, bir düğümün bir grafikteki bilgi akışı üzerindeki etkisini tespit etmek için kullanılır ve genellikle bir grafikteki bir bölgeden diğerine köprü görevi gören düğümleri bulmak için kullanılır. Yük Merkeziliği, bir düğümün üzerinden geçen tüm en kısa yolların oranını ifade eder. Ortalama Komşu Derecesi, her düğümün komşuluğunun ortalama derecesini döndürür.

Kullanıcı Kümeleme bölümünde ise, her kullanıcının belirli bir kümelere ait olduğunu ve bir öge için küme oranının kullanıcı-öge çifti için tahmini dereceyi oluşturacağını belirttik. Önerilen yöntemde, Autoencoder tarafından çıkarılan özelliklere dayanarak kullanıcıları kümelemek için K-Means algoritması kullanılır. Bu tür algoritmaları kullanmanın önemli bir konusu performans faktörleri göz önünde bulundurularak uygun küme sayısını bulmaktır. Küme sayısını seçmek için Elbow ve Average Silhouette algoritmaları kullanılır. Elbow yöntemi, açıklanan varyasyonu küme sayısının bir fonksiyonu olarak çizmek ve çıktı eğrisinin dirseğini uygun küme sayısı olarak seçmek üzerine dayanır. Ortalama Siluet yöntemi, bir kümenin kalitesini ölçer ve her nesnenin kendi kümesi içinde ne kadar iyi yer aldığını belirler. Optimal küme sayısı, k için farklı değerler için gözlemlerin ortalama siluetinin maksimize edildiği değerdir.

1.INTRODUCTION

Les systèmes de recommandation (RS) sont des systèmes capables de travailler avec des données volumineuses et complexes afin d'améliorer l'expérience de l'utilisateur et d'établir des liens entre les produits ou/et les utilisateurs. Ils font des recommandations aux utilisateurs en établissant diverses relations basées sur leur comportement ou leurs intérêts et contribuent à personnaliser l'expérience de l'utilisateur. Ce domaine, baptisé en 1995, ont parcouru un long chemin avec l'émergence de divers nouveaux problèmes et le développement continu de la technologie.[1] Les méthodes RS sont principalement classées en trois catégories : Filtrage Collaboratif (CF), Filtrage Basé sur le Contenu (CBF) et Système Hybride.

Les modèles de Filtrage Collaboratif visent à exploiter les informations relatives à l'historique de notes des utilisateurs pour les produits afin de fournir une recommandation personnalisée.[2] Il suppose que les utilisateurs qui ont été d'accord par le passé ont tendance à être d'accord à nouveau à l'avenir. En d'autres termes, il recommande des éléments en fonction des préférences d'utilisateurs similaires. Il existe deux types principaux de filtrage collaboratif. Le filtrage collaboratif basé sur l'utilisateur recommande des éléments à un utilisateur cible en trouvant d'autres utilisateurs similaires à cet utilisateur et en recommandant des éléments qu'ils ont aimés. Il calcule la similarité entre les utilisateurs en fonction de leurs évaluations ou interactions passées. Le filtrage collaboratif basé sur les éléments identifie des éléments similaires à ceux que l'utilisateur cible a déjà aimés ou avec lesquels il a interagi. Il recommande des éléments similaires à ceux que l'utilisateur a appréciés dans le passé. Le filtrage collaboratif ne nécessite pas de connaissance explicite des éléments ou des utilisateurs, il repose uniquement sur les interactions utilisateur-élément.[3] Le filtrage basé sur le contenu recommande des éléments aux utilisateurs en fonction des caractéristiques ou des attributs des éléments et d'un profil des préférences de l'utilisateur. Il se concentre sur les caractéristiques des éléments et les interactions passées de l'utilisateur. Ce système construit un profil utilisateur en fonction des caractéristiques des éléments avec lesquels ils ont interagi positivement. Le CBF utilise les informations relatives à l'élément de l'utilisateur pour estimer une nouvelle note.[6] La méthode CF est plus appliquée que la méthode CBF car elle ne s'intéresse qu'à l'évaluation des utilisateurs, alors que la méthode CBF nécessite des informations complémentaires pour donner de bons résultats.[4] Les systèmes de recommandation hybrides combinent plusieurs techniques de recommandation, telles que le filtrage collaboratif et le filtrage basé sur le contenu, pour surmonter les limitations des méthodes individuelles et fournir des recommandations plus précises et diversifiées.

La recherche sur les systèmes de recommandation a intégré une grande variété de techniques d'intelligence artificielle, notamment l'apprentissage automatique. Donc, les approches algorithmiques les plus récentes et les plus innovantes dans le domaine des systèmes de recommandation sont en constante évolution et sont fortement liées aux progrès d'intelligence artificielle. Parmi les dernières tendances, il existe plusieurs

méthodes remarquables visant spécifiquement à améliorer la précision et l'efficacité des systèmes de recommandation personnalisés. Citons le filtrage collaboratif neuronal (NCF), qui analyse plus en profondeur les interactions entre l'utilisateur et les éléments; les réseaux neuronaux convolutifs (CNN) et les réseaux neuronaux récurrents (RNN), qui traitent les données non structurées telles que le texte et les images; les réseaux neuronaux graphiques (GNN), qui peuvent modéliser des relations complexes; les mécanismes d'attention qui déterminent l'importance des interactions entre les utilisateurs; l'apprentissage fédéré, qui protège la confidentialité des données; et l'apprentissage par renforcement, qui vise la satisfaction de l'utilisateur à long terme.[2] [3] Bien que ces techniques aient le potentiel d'améliorer la précision et l'efficacité des systèmes de recommandation, elles nécessitent de grands ensembles de données et présentent des défis tels que le coût de calcul et la complexité du modèle.

Dans le monde d'aujourd'hui où la transformation digitale prend de l`importance, les recommandations personnalisées est devenu un élément important de nombreuses applications de commerce électronique en ligne telles qu'Amazon.com, Netflix et Spotify. Ces plateformes offrent des options presque illimitées aux consommateurs grâce à leur large choix de contenus. Donc, le besoin de systèmes de recommandation efficaces pour aider les utilisateurs à découvrir les contenus qui correspondent à leurs intérêts dans cette abondance est aussi en augmentation.[5]

Bien que la plupart des systèmes de recommandation existants reposent soit sur une approche basée sur le contenu, soit sur une approche collaborative, il existe des approches hybrides qui peuvent améliorer la précision de la recommandation en utilisant une combinaison des deux approches. Même si de nombreux algorithmes sont proposés en utilisant de telles méthodes, il est encore nécessaire d'apporter des améliorations supplémentaires. [6]

Notre projet vise à développer un système de recommandation hybride basé sur des graphes, utilisant l'apprentissage profond pour résoudre le problème du démarrage à froid sur des données d'interaction peu fréquentes. Pour cela, nous avons choisi de créer un système de recommandation de films en utilisant les données fournies par les plateformes de diffusion populaires. L'objectif est d'améliorer l'expérience utilisateur en fournissant des recommandations personnalisées et précises, en particulier pour les nouveaux utilisateurs. Pour y parvenir, nous combinons des techniques l'apprentissage profond (les approches basées sur les graphes et l'autoencodeur) et les méthodes de filtrage basées sur le contenu (l'utilisation d'informations secondaires descriptives) afin de prédire avec précision les préférences des utilisateurs et d'améliorer les performances du système de recommandation. Notre architecture de système de recommandation comprend plusieurs étapes, notamment la construction d'un graphe basé sur la similarité des utilisateurs, l'extraction d'informations pertinentes et l'utilisation d'un autoencodeur pour réduire la dimensionnalité des données. A la fin, les nouveaux utilisateurs sont affectés à des groupes en fonction de leurs caractéristiques, et les prévisions de recommandation sont calculées en fonction des notes moyennes de chaque groupe. Si nous regroupons les étapes de l'application sous deux titres : Graphe et Regroupement, nous utilisons des algorithmes basés sur les graphes tels que le Page Rank, la centralité de degré, la centralité

de proximité, la centralité de médiation, la centralité de charge et la moyenne des degrés de voisinage dans la section graphe et nous regroupons les utilisateurs à l'aide de l'algorithme K-Means sur la base des caractéristiques obtenues avec l'autoencodeur dans la section regroupement des utilisateurs. Nous utilisons aussi les algorithmes Elbow et Average Silhouette pour déterminer le nombre de groupes ce qui nous aide à déterminer le nombre optimal de groupes.

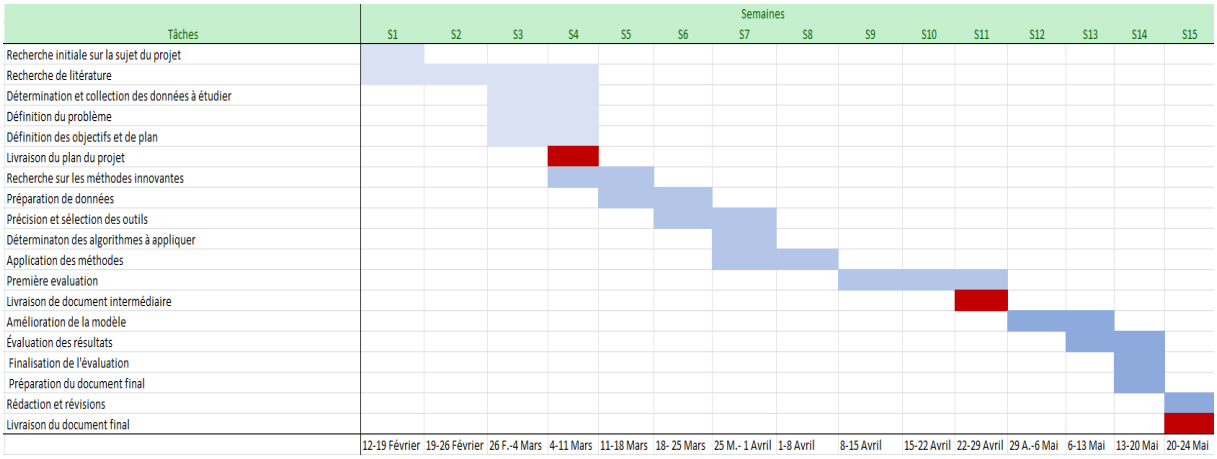

Figure 1. Calendrier De Tâches À Effectuer

2. REVUE DE LITTERATURE

Les systèmes de recommandation sont généralement classés en trois catégories : le filtrage collaboratif (CF), le filtrage basé sur le contenu (CBF) et les Systèmes Hybrides. Le filtrage collaboratif utilise une technique de filtrage des informations basée sur l'évaluation antérieure des éléments par l'utilisateur. [7] Il existe deux types principaux de filtrage collaboratif. En premier, le filtrage collaboratif basé sur l'utilisateur recommande des éléments à un utilisateur cible en trouvant d'autres utilisateurs similaires à cet utilisateur et en recommandant des éléments qu'ils ont aimés. En deuxième, le filtrage collaboratif basé sur les éléments recommande des éléments similaires à ceux que l'utilisateur cible a appréciés dans le passé. Toutefois, cette technique est connue avec deux problèmes majeurs : le problème de rareté et le problème de scalabilité. [8] Le filtrage basé sur le contenu (CBF) utilise le contenu des éléments pour déduire un profil d'utilisateur. Il recommande en fonction des caractéristiques des éléments et du profil des préférences de l'utilisateur. Cependant, la nature syntaxique du CBF, qui détecte les similitudes entre les éléments qui partagent le même attribut ou la même caractéristique, entraîne des recommandations surspécialisées qui n'incluent que des éléments très similaires à ceux que l'utilisateur connaît déjà.[9] Un autre problème est celui du démarrage à froid, courant dans les méthodes CF et CBF. En réponse à ces défis, les chercheurs ont exploré des approches hybrides combinant CF et CBF pour offrir des solutions plus efficaces, notamment en exploitant les capacités de l'apprentissage profond pour une personnalisation plus précise et une meilleure gestion de la rareté des données. L'apprentissage profond, une sous-discipline de l'intelligence artificielle, se distingue par sa capacité à analyser des données complexes grâce à des architectures de réseaux neuronaux profonds. Ces réseaux peuvent apprendre des représentations de données hiérarchiques à partir de données brutes ce qui est particulièrement bénéfique dans le contexte des systèmes de recommandation où les interactions entre les utilisateurs et les éléments sont souvent complexes et multidimensionnelles.

La classification des méthodes de systèmes de recommandation obtenue à partir de la recherche littéraire menée dans le cadre du projet est présentée à la figure 1. Il en ressort que l'approche de l'apprentissage profond est fréquemment utilisée pour renforcer le système de recommandation dans les trois classes. Dans ce contexte, les articles sélectionnés pour la présentation examinent l'approche de l'apprentissage profond sous différentes perspectives en comparant les méthodes traditionnelles. Donc, cette revue de littérature explore l'intégration de l'apprentissage profond dans les systèmes de recommandation mettant en évidence son rôle crucial dans la personnalisation et la gestion de la surcharge d'informations. Les études examinées soulignent l'efficacité de l'apprentissage profond pour découvrir des relations complexes entre l'utilisateur et l'élément, améliorant ainsi la qualité des recommandations par rapport aux méthodes traditionnelles. Les approches comprennent l'utilisation de réseaux neuronaux profonds pour le filtrage collaboratif, la prédiction des évaluations pour les nouveaux éléments et la résolution des problèmes de démarrage à froid. Des techniques telles que les

autoencodeurs et les réseaux neuronaux graphiques sont utilisées pour améliorer la performance des systèmes de recommandation, en particulier dans le contexte du traitement des données rares et des interactions complexes entre l'utilisateur et l'élément.

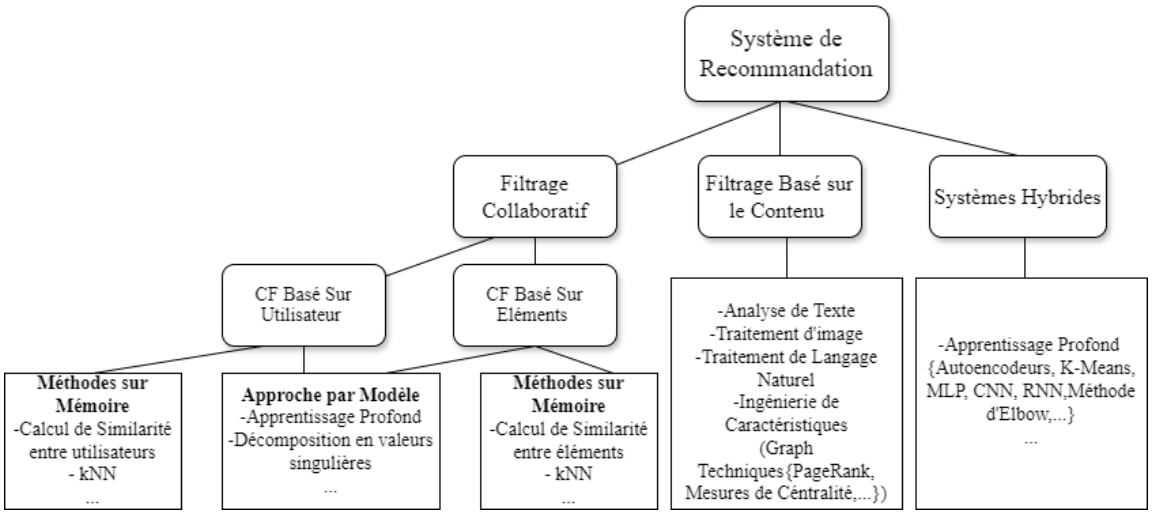

Figure 2. Classification des méthodes de Système de Recommandation

L'article « Deep Learning Based Recommander System : A Survey and New Perspectives » examine en détail l'intégration des techniques d'apprentissage profond dans les systèmes de recommandation en soulignant leur rôle essentiel dans la gestion de la surcharge d'informations par la personnalisation. Il souligne la transition des systèmes de recommandation vers l'apprentissage profond, attribuée à sa capacité supérieure de représentation des caractéristiques et à son succès avéré dans des domaines tels que la vision par ordinateur et le traitement du langage naturel. L'étude souligne la capacité de l'apprentissage profond à découvrir des relations complexes entre l'utilisateur et l'article, ce qui améliore considérablement la qualité des recommandations par rapport aux méthodes traditionnelles. Elle classe les modèles de recommandation basés sur l'apprentissage profond en filtres collaboratifs, filtres basés sur le contenu, modèles hybrides et autres modèles basés sur le type d'architecture d'apprentissage profond utilisé, tels que MLP, CNN et RNN. [10] Dans un autre exemple d'étude nommé « Collaborative Filtering and Deep Learning Based Recommendation System for Cold Start Items » présente une approche innovante pour résoudre le problème du démarrage à froid dans les systèmes de recommandation grâce à l'intégration du filtrage collaboratif (CF) et de l'apprentissage profond. Plus précisément, il propose deux modèles qui tirent parti d'un réseau neuronal profond, SADE, pour extraire les caractéristiques du contenu des éléments, et modifie le modèle timeSVD++ pour incorporer ces caractéristiques dans la prédiction des évaluations pour les nouveaux éléments (CCS) et les éléments avec des

évaluations limitées (ICS). L'étude démontre la performance supérieure des modèles dans la prédiction des évaluations pour les éléments à démarrage rapide par le biais d'expériences approfondies sur un grand ensemble de données Netflix. Elle relève des défis majeurs dans les systèmes de recommandation, tels que la rareté des interactions entre l'utilisateur et l'élément et l'évolutivité, en montrant des améliorations significatives par rapport aux modèles de base. Ce travail met en évidence le potentiel de la combinaison des approches de CF avec l'apprentissage profond pour améliorer la qualité des recommandations, en particulier pour les éléments nouveaux ou moins bien notés, et jette les bases de futures avancées dans les systèmes de recommandation personnalisés. [11] Puis, l'article « Neural Collaborative Filtering » décrit la technique du filtrage collaboratif neuronal (NCF), qui exploite les réseaux neuronaux profonds pour améliorer les performances des systèmes de recommandation, en particulier dans le contexte du filtrage collaboratif basé sur un retour d'information implicite. L'approche traditionnelle du filtrage collaboratif utilise principalement des techniques de factorisation de la matrice, qui représentent de manière linéaire les interactions entre l'utilisateur et l'élément, ce qui risque de négliger des schémas complexes dans les données. Les auteurs proposent une méthode plus nuancée qui remplace le produit intérieur par une architecture neuronale capable d'apprendre une fonction arbitraire à partir des données, ce qui permet une représentation plus sophistiquée des interactions entre l'utilisateur et l'élément. Ils introduisent un cadre général appelé NCF, qui est capable d'exprimer et de généraliser la factorisation matricielle dans son cadre. Pour saisir plus efficacement les non-linéarités et les relations complexes entre l'utilisateur et l'élément, ils suggèrent d'utiliser un perceptron multicouche dans le cadre de la NCF. Grâce à des expériences approfondies sur des ensembles de données réels, l'étude démontre des améliorations significatives dans la performance des recommandations avec l'approche NCF proposée par rapport aux méthodes de pointe, soulignant ainsi l'efficacité des techniques d'apprentissage profond pour capturer la dynamique nuancée des tâches de filtrage collaboratif. [12]

Comme les données sur les films ayant été choisies pour réaliser l'étude, deux documents sur les systèmes de recommandation de films ont été examinés. En premier, l'étude du « Design of an Unsupervised Machine Learning-Based Movie Recommender System » se concentre sur l'utilisation d'algorithmes d'apprentissage automatique non supervisés tels que K-Means, Birch et d'autres pour classer les utilisateurs en groupes en fonction de leurs préférences cinématographiques afin d'optimiser la sélection du nombre de clusters et d'améliorer la précision du système de recommandation.[13] En revanche, l'article « Movie Recommendations Using the Deep Learning Approach » présente une approche d'apprentissage profond basée sur des autoencodeurs pour le filtrage collaboratif, montrant une amélioration significative des performances en termes d'erreur quadratique moyenne (RMSE) par rapport aux techniques traditionnelles telles que le k-nearest-neighbor et la factorisation matricielle. Ces deux recherches soulignent l'importance des systèmes de recommandation pour les services de streaming et mettent en lumière le potentiel des techniques d'apprentissage automatique pour améliorer la précision et la satisfaction des utilisateurs dans ce domaine en plein essor. [5] En outre, l'article « An Efficient Deep Learning Approach for Collaborative Filtering Recommender System se penche sur l'application de l'apprentissage profond à l'amélioration des techniques de

filtrage collaboratif (CF) pour les systèmes de recommandation. Reconnaissant les limites posées par les méthodes traditionnelles de CF, telles que la rareté des données et l'évolutivité, les auteurs proposent un système de recommandation collaboratif d'apprentissage profond (DLCRS) qui évite le besoin d'informations supplémentaires, en s'appuyant uniquement sur les interactions entre l'utilisateur et l'élément. Le DLCRS s'avère plus performant que les approches conventionnelles, avec une erreur quadratique moyenne (RMSE) plus faible sur les ensembles de données MovieLens 100K et 1M. Cette recherche souligne le potentiel de transformation de l'apprentissage profond dans les systèmes de recommandation, en particulier pour relever les défis de la rareté des données et de l'évolutivité, ouvrant ainsi la voie à des modèles de recommandation plus précis et plus fiables à l'avenir.[14] Pour finir, l'article « A Deeper Graph Neural Network For Recommender Systems » étudie l'application d'un cadre de réseau neuronal graphique (GNN) pour relever les défis du filtrage collaboratif (CF) dans les systèmes de recommandation, en se concentrant particulièrement sur l'atténuation du problème de la rareté des données. L'étude présente un modèle qui intègre le GNN avec un mécanisme d'attention, ce qui lui permet de gérer efficacement des entrées de taille variable pour différents nœuds et d'améliorer la précision des recommandations. Grâce à des expériences approfondies sur des ensembles de données réels tels que MovieLens et Taobao, l'étude démontre que son approche est plus performante que plusieurs méthodes de référence, en particulier dans le traitement des données éparses, grâce à l'apprentissage d'une intégration de nœuds plus profonde via GNN qui capture plus efficacement les modèles d'interaction complexes entre l'utilisateur et l'élément. Les auteurs concluent en proposant des améliorations futures potentielles, telles que l'incorporation d'informations latérales dans le cadre GCF et l'exploration de la parallélisation pour améliorer l'efficacité. [15]

En conclusion, les études examinées mettent en évidence l'importance croissante de l'intégration de l'apprentissage profond dans les systèmes de recommandation. L'adoption de techniques d'apprentissage profond offre de nouvelles perspectives pour résoudre les défis persistants tels que la gestion de la surcharge d'informations et le problème du démarrage à froid. Les résultats montrent que les modèles basés sur l'apprentissage profond surpassent souvent les méthodes traditionnelles en termes de précision et de qualité des recommandations, en particulier dans des domaines tels que la prédiction des évaluations pour les éléments nouveaux ou moins bien notés. En outre, les études soulignent la diversité des approches basées sur l'apprentissage profond, allant de l'utilisation de réseaux de neurones profonds pour extraire des caractéristiques des éléments à l'intégration de réseaux neuronaux graphiques pour gérer la rareté des données. Ces avancées ouvrent la voie à des recommandations plus personnalisées et efficaces, contribuant ainsi à améliorer l'expérience utilisateur dans divers domaines tels que le divertissement cinématographique.

3. LES NOTIONS FONDAMENTALES

Les systèmes de recommandation sont exploités par une variété d'algorithmes, chacun étant conçu pour prédire les préférences des utilisateurs et suggérer des éléments à l'aide de méthodologies spécifiques.

3.1. TYPES DE SYSTEMES DE RECOMMANDATION

3.1.1. Filtrage Collaboratif

Cette méthode permet de faire des prédictions automatiques sur les intérêts d'un utilisateur en recueillant les préférences de nombreux utilisateurs (collaboration). L'hypothèse sous-jacente est que si un utilisateur A a la même opinion qu'un utilisateur B sur un sujet, A est plus susceptible d'avoir l'opinion de B sur un sujet différent que celle d'un utilisateur aléatoire.

3.1.1.1. Filtrage Collaboratif Basé Sur L'utilisateur

Cet algorithme recommande des éléments en trouvant des utilisateurs similaires. Cette approche prend en compte les utilisateurs qui ont des préférences similaires et recommande des éléments que les utilisateurs similaires ont aimés.

3.1.1.2. Filtrage Collaboratif Basé Sur Les Eléments

Cet algorithme recommande des éléments similaires à ceux que l'utilisateur a aimés ou avec lesquels il a interagi dans le passé. Cette approche calcule les similitudes entre les éléments sur la base des évaluations des utilisateurs.

3.1.2. Filtrage Basé Sur Le Contenu

Cette méthode recommande des éléments sur la base d'une description de l'élément et d'un profil des préférences de l'utilisateur. Les recommandations sont faites en faisant

correspondre les éléments préférés de l'utilisateur avec des éléments qui ont un profil de contenu similaire.

3.1.3. Approches Hybrides

Ces systèmes combinent le filtrage collaboratif, le filtrage basé sur le contenu et d'autres approches pour améliorer la qualité des recommandations. Les approches hybrides peuvent aider à surmonter certaines limites des systèmes individuels, telles que les problèmes de démarrage à froid et de rareté.

3.1.3.1. Modèles De Recommandation Basés Sur L'apprentissage Profond

L'apprentissage profond est un domaine de recherche de l'apprentissage automatique. Il apprend de multiples niveaux de représentations et d'abstractions à partir des données et peut résoudre des tâches d'apprentissage supervisées et non supervisées. Nous pouvons classer les modèles de recommandation existants en fonction des types d'approches d'apprentissage profond employées dans les deux classes suivantes : La recommandation avec les blocs neuronaux et La recommandation avec les modèles hybrides profonds [10]

3.1.3.1.1. La Recommandation Avec Les Blocs Neuronaux

Cette technique d'apprentissage profond détermine l'applicabilité du modèle de recommandation. Par exemple, les MLP peuvent simplement modéliser les interactions non linéaires entre les utilisateurs et les éléments, les CNN peuvent extraire des représentations locales et globales à partir de sources de données hétérogènes telles que le texte et l'image ; les systèmes de recommandation peuvent modéliser la dynamique temporelle et l'évolution séquentielle des informations sur le contenu en utilisant des RNN. [7]

3.1.3.1.2. La Recommandation Avec Les Modèles Hybrides Profonds

Certains modèles utilisent plus d'une technique d'apprentissage profond. La flexibilité des réseaux neuronaux profonds permet de combiner plusieurs blocs neuronaux pour se compléter mutuellement et former un modèle hybride plus puissant. Il existe de nombreuses combinaisons possibles de ces techniques d'apprentissage profond. [7] Par exemple, un modèle hybride peut utiliser des CNN pour extraire des caractéristiques visuelles à partir d'images et des RNN pour modéliser les séquences temporelles dans les comportements des utilisateurs.

3.1.3.2. Modèles De Recommandation Basés Sur L'Autoencodeur

L'autoencodeur est un modèle qui tente de reconstruire les données d'entrée dans la couche de sortie. En général, la couche de goulot (couche intermédiaire) est utilisée comme représentation des caractéristiques dominantes des données d'entrée. Il existe deux façons générales d'appliquer l'autoencodeur à un système de recommandation[7] :

1. Utiliser l'autoencodeur pour apprendre des représentations de caractéristiques de dimensions inférieures dans la couche de goulot dans la couche de goulot

2. Remplir les infos manquantes de la matrice d'interaction directement dans la couche de reconstruction

L'autoencodeur est en effet une méthode puissante d'apprentissage de la représentation des caractéristiques, et son application dans les systèmes de recommandation est remarquable. Il permet d'apprendre des représentations de caractéristiques à partir des données des utilisateurs et des éléments, en capturant efficacement les structures sous-jacentes et les relations complexes entre eux. En utilisant les caractéristiques extraites par l'autoencodeur, les systèmes de recommandation peuvent mieux comprendre les préférences des utilisateurs et la nature des éléments, ce qui conduit à des recommandations plus précises et personnalisées.

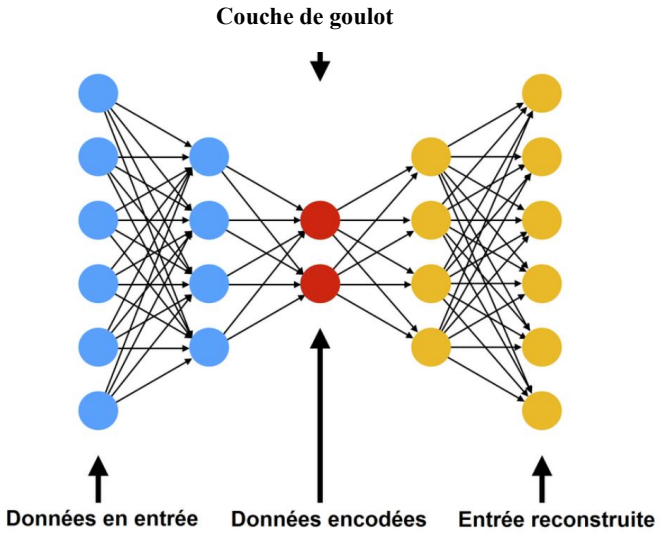

Figure 3. Exemple d'une architecture de base d'auto-encodeur qu'on pourra modifier selon l'application (Sublime, 2022) [16]

3.1.3.3. Système De Recommandation Hybride Basé Sur Les Graphes

Dans ce système, les relations entre utilisateurs et éléments sont représentées sous forme de graphes, où les nœuds représentent les utilisateurs et les éléments, et les arêtes représentent les interactions entre eux. Les algorithmes de traitement de graphe sont ensuite utilisés pour analyser ces données et générer des recommandations.

3.1.3.3.1. Caractéristiques Basées Sur Les Graphes

Page Rank : C'est un algorithme qui mesure l'influence transitive ou la connectivité des nœuds. Nous pouvons calculer le Page Rank en distribuant itérativement le rang d'un nœud (en fonction du degré) sur les voisins.

Centralité de Degré : La centralité de degré mesure le nombre de relations entrantes et sortantes d'un nœud. L'algorithme de centralité de degré peut être utilisé pour trouver la popularité des nœuds individuels. Les valeurs de centralité de degré sont normalisées en divisant par le degré maximal possible dans un graphe simple n-1, où n est le nombre de nœuds.

Centralité de Proximité : C'est une mesure de la proximité d'un noeud u par rapport à tous les autres noeuds du graphe. Un noeud avec un score de centralité de proximité élevé a des distances courtes par rapport à tous les autres noeuds. Dans la formule, la somme calcule la distance totale du plus court chemin du noeud u vers tous les autres noeuds du graphe. Nous divisons ensuite cette valeur par (n−1) (le nombre de noeuds excluant u lui-même) pour obtenir la distance moyenne. Enfin, nous prenons l'inverse de cette distance moyenne pour obtenir le score de centralité de proximité.

$$C_C(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(v,u))}$$

Figure 4. Formule de Centralité de Proximité

Où ; $C_C(u)$: Centralité de proximité du noeud u , n: Nombre de noeuds dans le graphe et d(v,u) : Distance du plus court chemin entre les noeuds v et u.

Centralité d'Intermédiarité : La centralité d'intermédiarité est un facteur que nous utilisons pour détecter la quantité d'influence qu'un nœud exerce sur le flux d'information dans un graphe. L'algorithme de centralité d'intermédiarité calcule le plus court chemin (pondéré) entre chaque paire de nœuds dans un graphe connecté. Chaque nœud reçoit un score, basé sur le nombre de ces plus courts chemins qui passent par le nœud. Les nœuds qui se trouvent le plus fréquemment sur ces plus courts chemins auront un score de centralité d'intermédiarité plus élevé.

$$C_B(u) = \frac{\sigma(s,t|u)}{\sum_{s,t \in V} \sigma(s,t)}$$

Figure 5. Formule de Centralité d'Intermédiarité

Où; $C_B(u)$ représente la centralité d'intermédiarité du noeud u, σ(s, t) représente le nombre de plus courts chemins entre les noeuds s et t, σ(s, t|u) représente le nombre de plus courts chemins entre les noeuds s et t qui passent par le noeud u et V est l'ensemble de tous les noeuds du réseau.

La sommation Σ(s, t ∈ V) σ(s, t) est effectuée sur toutes les paires de noeuds s et t du réseau. Si s = t, alors σ(s, t) = 1, car il n'existe qu'un seul plus court chemin entre un noeud et lui-même (le chemin de longueur 0). Si v ∈ {s, t}, alors σ(s, t|u) = 0, car un plus court chemin ne peut pas passer par le même noeud deux fois.

Centralité de Charge : La centralité de charge d'un noeud est la fraction de tous les plus courts chemins qui passent par ce nœud.

Degré Moyen des Voisins : Retourne le degré moyen du voisinage de chaque nœud.

Dans notre projet, soit l'ensemble des n utilisateurs $U = \{u_1, ..., u_n\}$ et l'ensemble des m éléments, $I = \{i_1, ..., i_m\}$, toutes les paires utilisateur-élément peuvent être dénotées par une matrice n-par-m R = UxI, où l'entrée r_{ui} indique la valeur attribuée au retour d'information implicite de l'utilisateur u à l'élément i. Si r_{ui} a été observé (ou connu), il est représenté par une note associée à un intervalle spécifiques ; sinon, une note par défaut globale est nulle. Nous avons utilisé cette matrice pour trouver la similitude entre les préférences des utilisateurs. Après avoir généré le graphe de similarité qui représente les utilisateurs comme des nœuds et les relations comme des arêtes, nous extrayons les caractéristiques de ce graphe, $F = \{f_1, ..., f_g\}$, et les conservons dans la matrice n-par-g.

Nous recueillons certaines caractéristiques des utilisateurs à partir de l'ensemble de données, appelées informations secondaires, $F_u = \{f_{u1}, ..., f_{un}\}$, certaines informations secondaires des éléments $F_i = \{f_{i1}, ..., f_{in}\}$ et obtenons la matrice de caractéristiques combinée qui est n-par-g+s. Sans perte de généralité, nous avons classé toutes les caractéristiques (à la fois caractéristiques du graphe et informations secondaires) comme binaires, ce qui a élargi le vecteur de caractéristiques final pour chaque utilisateur.

3.2. CONCEPT CLES ET DEFIS

3.2.1. Démarrage à Froid

Il s'agit de la difficulté à laquelle les systèmes de recommandation sont confrontés lorsqu'ils doivent faire des recommandations à de nouveaux utilisateurs ou à des éléments pour lesquels il n'existe que peu ou pas de données historiques.

3.2.2. Rareté

La plupart des utilisateurs n'interagissent qu'avec une petite fraction de l'ensemble du catalogue des éléments, ce qui conduit à des matrices d'interaction utilisateur-élément clairsemées(rare), qui peuvent compromettre la qualité de la recommandation.

3.2.3. Évolutivité

Au fur et à mesure que le nombre d'utilisateurs et des éléments augmente, les systèmes de recommandation ont besoin d'algorithmes efficaces pour s'adapter et fournir des recommandations opportunes.

3.2.4. Diversité et Sérendipité

Les systèmes de recommandation visent non seulement à prédire les éléments qui plairont aux utilisateurs, mais aussi à les surprendre avec de nouvelles découvertes, en trouvant un équilibre entre la précision et la fourniture de recommandations nouvelles et diversifiées.

3.2.5. Confidentialité des Données Personnelles

Les systèmes de recommandation collectent souvent des données sensibles sur les préférences et les comportements des utilisateurs, ce qui pose des problèmes de confidentialité et nécessite la mise en œuvre de mécanismes de préservation des données personnelles.

3.2.5.1. KVKK

En Turquie, la loi KVKK est la principale norme à respecter lors de l'extraction de données, y compris les données publiques. Selon la loi KVKK, les données personnelles ne peuvent être collectées qu'avec le consentement explicite de la personne concernée. Les organisations doivent également informer les personnes concernées de la finalité du traitement des données et des tiers auxquels les données peuvent être transférées. Les organisations doivent également prendre des mesures pour protéger les données personnelles contre toute perte, vol, accès non autorisé, divulgation ou destruction.

Étant donné que nous ne collectons aucune donnée qui n'a pas été rendue publique dans le cadre du projet, nous agissons conformément à cette loi.

4. METHODOLOGIE ET MODELE

4.1. ARCHITECTURE

1. Nous construisons un graphe avec le nombre d'utilisateurs comme noeuds. Deux utilisateurs seront connectés en fonction de leurs similarités. L'arête connecte une paire d'utilisateurs ayant plus de alpha pourcentage d'éléments avec des évaluations similaires.

2. Un ensemble d'informations sera extrait du graphe de similarité pour chaque utilisateur. Par exemple, nous calculons le PageRank des noeuds, la centralité de degré, la centralité de proximité, la centralité de l'intermédiaire le plus court, la centralité de charge, et le degré moyen du voisinage de chaque noeud dans le graphe. En conséquence, cette matrice repose sur une magnitude de traitement des données différente en utilisant une approche collaborative basée sur les préférences.

3. Nous combinons des informations supplémentaires telles que le genre et l'âge avec des caractéristiques basées sur les graphes pour récupérer les films les plus pertinents pour les utilisateurs. Par conséquent, nous avons une matrice combinée à partir de différents types de caractéristiques, qui est ensuite utilisée comme entrée de l'étape Autoencodeur.

4. Nous appliquons l'autoencodeur pour extraire de nouvelles caractéristiques et réduire la dimension. Cela comprend la sélection d'un optimiseur approprié, l'utilisation d'une fonction de perte appropriée et d'une architecture de réseau neuronal, et la prévention du problème de surajustement.

5. Nous utilisons les nouvelles caractéristiques encodées par l'Autoencodeur pour le regroupement des utilisateurs, en utilisant l'algorithme K-means pour créer un petit nombre de groupes de pairs. Cela comprend la recherche d'un nombre approprié de clusters pour chaque ensemble de données.

6. Nous assignons les nouveaux utilisateurs à des clusters en fonction des caractéristiques encodées et calculons la nouvelle notation des éléments en fonction de leur similarité avec d'autres éléments.

7. Nous calculons les taux estimés de tous les éléments pour chaque utilisateur en fonction de la notation moyenne de son cluster.

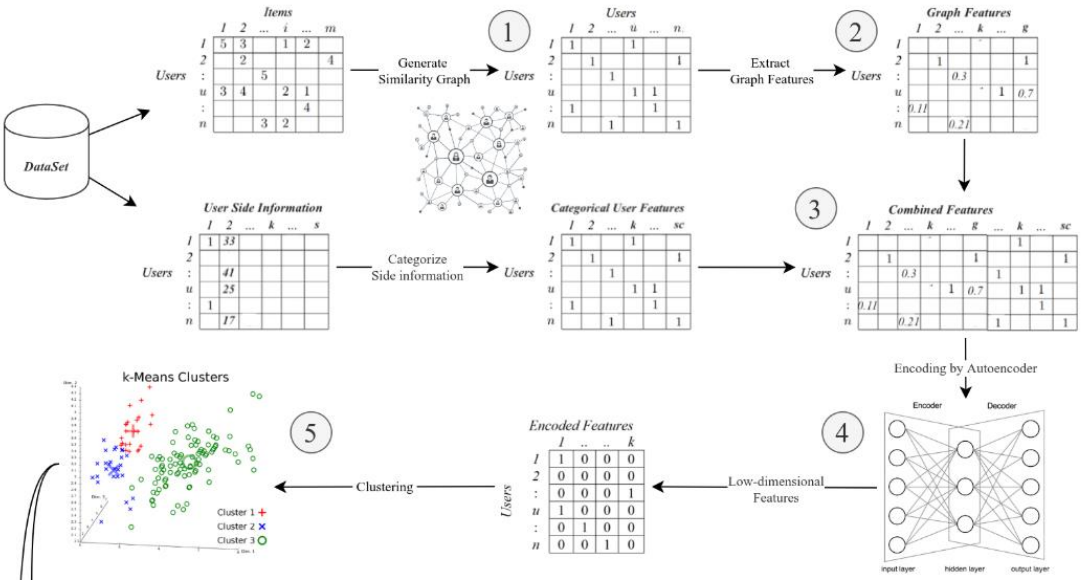

Figure 6. La méthode code les caractéristiques combinées avec un autoencodeur et crée le modèle en regroupant les utilisateurs à l'aide des caractéristiques codées [6]

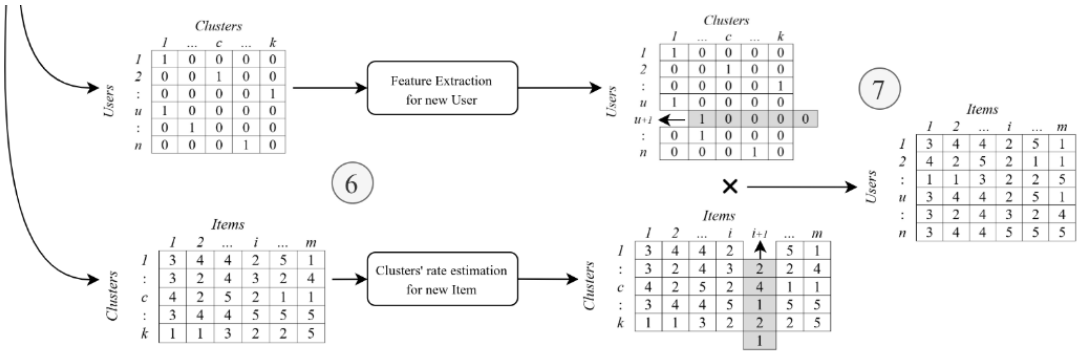

Figure 7. Un modèle de classement basé sur les préférences est utilisé pour récupérer le classement prédit des films pour l'utilisateur cible [6]

4.2. ALGORITHME PROPOSE

Entrée :

- U : Ensemble d'utilisateurs
- I : Ensemble des éléments
- R : Matrice de notation utilisateur-élément
- F_u : Caractéristiques supplémentaires de l'utilisateur

Sortie :

- R' : Matrice d'évaluation estimée pour l'utilisateur-élément

Étapes :

1. **Définir alpha**
   - Alpha correspond au pourcentage d'élément avec des évaluations similaires entre deux utilisateurs.
2. **Calculer la similarité agrégée entre les utilisateurs**
   - Calculer la similarité agrégée entre les utilisateurs en fonction du pourcentage d'élément que deux utilisateurs ont évalués de manière similaire.
3. **Construire le graphe de similarité**
   - Représenter les utilisateurs comme des noeuds dans le graphe de similarité.
4. **Extraire les caractéristiques basées sur le graphe pour les utilisateurs**
   - Extraire des caractéristiques du graphe de similarité qui représentent les utilisateurs (noeuds).
5. **Prétraiter et catégoriser les informations secondaires des utilisateurs**
   - Prétraiter et catégoriser les informations supplémentaires sur les utilisateurs (détails démographiques).
6. **Combiner les caractéristiques**
   - Combiner les caractéristiques extraites du graphe et les informations secondaires de l'utilisateur dans un seul vecteur de caractéristiques.
   - Appliquer l'autoencodeur sur le vecteur de caractéristiques combiné et entraîner le modèle avec les meilleurs paramètres.
7. **Encoder les caractéristiques**
   - Encoder le vecteur de caractéristiques combiné à l'aide de l'autoencodeur pour extraire un vecteur de caractéristiques de dimension inférieure.
8. **Trouver les clusters optimaux**
   - Trouver le nombre optimal de clusters pour regrouper les utilisateurs en fonction du vecteur de caractéristiques de dimension inférieure.
9. **Clustering d'utilisateurs**
   - Regrouper les utilisateurs en fonction du vecteur de caractéristiques de dimension inférieure.
10. **Générer la matrice utilisateur-cluster**
    - Générer une matrice indiquant l'appartenance de chaque utilisateur à un cluster.
11. **Estimer les notes des clusters pour les articles**

- Créer une matrice qui représente les notes moyennes des éléments dans chaque cluster.

12. **Estimer les notes des utilisateurs**
    - Pour chaque élément, estimer la note d'un utilisateur en fonction de:
        - Si l'utilisateur a déjà évalué l'élément, utiliser sa note.
        - Sinon, si des éléments similaires ont été évalués par des utilisateurs dans le même cluster, utiliser la note moyenne de ces éléments similaires.
        - Sinon, utiliser la note moyenne de tous les éléments du cluster.
13. **Générer la liste de recommandations**
    - Générer une liste d'éléments recommandés pour l'utilisateur cible en fonction des notes estimées.

4.3. TECHNOLOGIES A UTILISER

Python : Langage de programmation populaire utilisé pour le développement de divers projets, y compris l'analyse de données et l'apprentissage automatique.

Jupyter : Un environnement de développement interactif largement utilisé dans le domaine de la science des données et de l'apprentissage automatique. Il permet d'écrire et d'exécuter du code dans des cellules séparées, ce qui facilite l'exploration des données, la visualisation et le prototypage de modèles.

Pandas : Une bibliothèque Python puissante et flexible utilisée pour l'analyse et la manipulation des données. Elle offre des structures de données efficaces pour travailler avec des données tabulaires et des séries temporelles, ce qui en fait un outil essentiel pour le prétraitement des données dans les projets d'apprentissage automatique.

TensorFlow : Un cadre d'apprentissage automatique développé par Google, largement utilisé pour la création de modèles d'apprentissage en profondeur. TensorFlow offre une flexibilité et une extensibilité considérables, ainsi qu'un support pour l'entraînement et le déploiement de modèles sur une variété de plates-formes.

Keras : Une interface de programmation d'applications (API) haut niveau écrite en Python, conçue pour être simple et intuitive, tout en permettant une flexibilité et une puissance d'expression élevées. Keras est souvent utilisé en conjonction avec TensorFlow pour simplifier le processus de création et de formation des réseaux de neurones.

Scikit-learn : Une bibliothèque d'apprentissage automatique open-source qui offre des outils simples et efficaces pour l'analyse des données et l'apprentissage statistique en Python. Scikit-learn fournit une large gamme d'algorithmes d'apprentissage supervisé et non supervisé, ainsi que des outils pour l'évaluation des modèles et la validation croisée.

NumPy : Une bibliothèque Python essentielle pour le calcul scientifique, offrant des structures de données puissantes et des fonctions pour travailler avec des tableaux multidimensionnels. NumPy est largement utilisé pour le traitement et la manipulation efficaces des données numériques dans le domaine de la science des données et de l'apprentissage automatique.

4.4. METHODES A UTILISER POUR LE GROUPEMENT DES UTILISATEURS

Dans la méthode proposée, nous utilisons l'algorithme K-Mean pour regrouper les utilisateurs sur la base des caractéristiques extraites par l'autoencodeur. L'une des questions importantes dans l'utilisation de ces algorithmes est de trouver le nombre adéquat de clusters en fonction des facteurs de performance. Nous utilisons deux méthodes pour choisir le nombre de clusters : la méthode d'Elbow et l'algorithme de la silhouette moyenne.

4.4.1. Algorithm de K-Means

L'algorithme K-means représente une méthode itérative de regroupement des données. En se basant sur une métrique de distance et un nombre prédéfini de catégories K dans l'ensemble de données, il calcule la moyenne de la distance, ce qui donne le centroïde initial, chaque classe étant décrite par le centroïde. Pour un ensemble de données X donné avec n échantillons et K catégories, la distance euclidienne est utilisée comme mesure de similarité. L'objectif du regroupement est de minimiser la somme des carrés des écarts, cherchant ainsi à réduire les disparités entre les différentes catégories. [17]

$$d = \sum_{k=1}^{K} \sum_{i=1}^{n} \left\| (x_i - u_k) \right\|^2$$

Figure 8. Formule d'Algorithme de K-Means [17]

Où ; k représente K centres de clusters, u_k représente le k-ième centre et x_i représente le i-ème point de l'ensemble de données.

4.4.2. Méthode d'Elbow

C'est une technique utilisée pour déterminer le nombre optimal de clusters dans un algorithme de regroupement. Elle consiste à tracer la somme des carrés des distances intra-cluster en fonction du nombre de clusters, puis à observer le point où cette somme commence à diminuer de manière significativement plus lente. Ce point est appelé Elbow. L'idée est de choisir le nombre de clusters qui se situe juste avant d'Elbow. Cela représente un compromis optimal entre la réduction de l'erreur de regroupement et la complexité accrue du modèle. [17]

4.4.3. Méthode De La Silhouette Moyenne

La méthode de la silhouette a été proposée pour la première fois par Peter J. Rousseeuw. Elle combine les deux facteurs de cohésion et de résolution. La cohésion représente la similarité entre l'objet et le cluster auquel il appartient, tandis que la résolution mesure la similarité de l'objet par rapport aux autres clusters. Cette comparaison est réalisée à l'aide de la valeur de la silhouette, qui est dans la range de -1 à 1. Une valeur de silhouette proche de 1 indique qu'il existe une relation étroite entre l'objet et le cluster. Si un cluster de données dans un modèle est généré avec une valeur de silhouette relativement élevée, le modèle est considéré comme approprié et acceptable. [17]

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$$
$$= \begin{cases} 1-\frac{a(i)}{b(i)}, & a(i) < b(i) \\ 0, & a(i) = b(i) \\ \frac{b(i)}{a(i)}-1, & a(i) > b(i) \end{cases}$$

Figure 9. Formule de Calcul de la Coefficient Silhouette [17]

Méthode de calcul :

1. Similarité intra-classe :

Calculer la distance moyenne a(i) de l'échantillon i par rapport aux autres échantillons du même cluster. Plus a(i) est petit, plus l'échantillon i doit être regroupé dans ce cluster. a(i) est appelé la dissimilarité intra-classe de l'échantillon i. La moyenne des a(i) pour tous les échantillons du cluster c est appelée la dissimilarité du cluster c.

2. Similarité inter-classe :

Calculer la distance moyenne b(i) de tous les échantillons par rapport à l'échantillon i dans l'autre cluster, nommé cluster c(i). Cette distance est appelée la dissimilarité entre l'échantillon i et le cluster c(i). Elle est définie comme la dissimilarité inter-classe de l'échantillon i : $b(i) = \min \{b_{i1}, b_{i2}, ..., b_{ik}\}$. Plus b(i) est grand, moins l'échantillon i appartient aux autres clusters.

3. Coefficients de contour :

Les coefficients de contour de l'échantillon i sont définis en fonction de la dissimilarité intra-classe a(i) de l'échantillon i et de la dissimilarité inter-classe b(i).

5. APPLICATION DU MODELE PROPOSE

5.1. DATASET

Nous utilisons l'ensemble de données MovieLens 1M dans notre système de recommandation pour mettre en œuvre le modèle. Il comprend 3 fichier nommé ratings.dat, users.dat et movies.dat.

Le fichier « ratings.dat » contient toutes les évaluations et est structuré comme suit : UserID::MovieID::Rating::Timestamp. Les identifiants d'utilisateur (UserID) varient de 1 à 6040, tandis que les identifiants de film (MovieID) varient de 1 à 3952. Les évaluations sont faites sur une échelle de 5 étoiles (notation entière uniquement), et le champ Timestamp est représenté en secondes depuis l'époque (epoch). Chaque utilisateur a au moins 20 évaluations.

Les informations sur les utilisateurs sont stockées dans le fichier « users.dat » et suivent le format UserID::Gender::Age::Occupation::Zip-code. Les données démographiques sont fournies volontairement par les utilisateurs et ne sont pas vérifiées pour leur précision. Seuls les utilisateurs ayant fourni des informations démographiques sont inclus dans cet ensemble de données. Ainsi qu'on l'a vu, la méthode proposée utilise les données démographiques comme informations supplémentaires sur l'utilisateur pour résoudre le problème du démarrage à froid des nouveaux utilisateurs. Le genre est indiqué par "M" pour masculin et "F" pour féminin. L'âge est choisi parmi plusieurs catégories, allant de "Moins de 18 ans" à "56+". L'occupation est également spécifiée par un code numérique correspondant à différents métiers, comme "academic/educator" (1), "artist" (2), ou "retired" (13).

Les informations sur les films sont stockées dans le fichier « movies.dat » et suivent le format MovieID::Title::Genres. Les titres sont identiques à ceux fournis par l'IMDB (incluant l'année de sortie), tandis que les genres sont séparés par des barres verticales et peuvent inclure des catégories telles que "Action", "Comédie", "Horreur", etc. Certains identifiants de film peuvent ne pas correspondre à un film en raison de doublons accidentels ou d'entrées de test. Les informations sur les films sont généralement saisies manuellement, ce qui peut entraîner des erreurs et des incohérences.

5.2. CARACTERISTIQUES UTILISEES POUR L'UTILISATEUR

Dans la méthode proposée, nous utilisons deux types de caractéristiques : des informations supplémentaires (données démographiques des utilisateurs) et des caractéristiques extraites du graphe de similarité entre les utilisateurs. Nous avons transformé les données démographiques en format catégorique, concaténé les deux types de caractéristiques, puis constitué l'ensemble de caractéristiques brut avant la réduction de dimension avec un autoencodeur. Comme toutes les caractéristiques démographiques sont transformées en format catégorique, le vecteur de caractéristiques démographiques est encodé de manière one-hot et présente un niveau de clairsemé spécifique pour chaque ensemble de données.

Le seul paramètre que nous utilisons pour générer le graphe est alpha, la valeur d'un seuil pour relier deux utilisateurs ayant au moins plusieurs films identiques dans leurs évaluations. Ce seuil est représenté sous la forme d'un pourcentage du nombre total de films dans l'ensemble de données. Nous avons donc une exploration d'un graphe très clairsemé pour approcher un graphe à maille complète.

Comme nous avons déclaré que la valeur des caractéristiques basées sur les graphes est liée à la taille du graphe de similarité, et que la taille du graphe est directement liée au facteur alpha. La figure 10 montre que la rareté (sparsity) de l'ensemble de caractéristiques augmente lorsque la valeur du facteur alpha augmente.

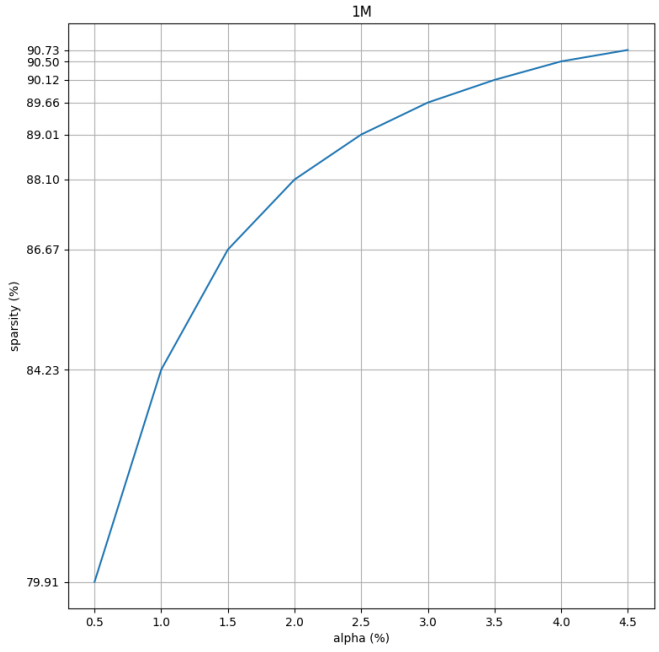

Figure 10. Graphique de Relation Entre Rareté et Facteur Alpha

5.3. ETAPES APPLIQUE

Dans la phase de préparation des données, les trois ensembles de données provenant de données MovieLens 1M ont été lus dans un environnement Jupyter Notebook à l'aide de la bibliothèque pandas, et ont été prétraités pour être utilisés dans l'étape suivante. Les données sur les évaluations des films par les utilisateurs ont été extraites de l'ensemble de données "ratings.dat". Les données démographiques des utilisateurs, telles que l'âge, le sexe et la profession, ont été extraites de l'ensemble de données "users.dat". À partir de ces informations, les données sur le sexe et la profession ont été converties en données catégorielles en utilisant la technique de codage one-hot. De plus, les données sur l'âge ont été regroupées en intervalles d'âge spécifiques et étiquetées pour être converties en données catégorielles. Dans la deuxième étape, il est calculé les paires de 'UID' (UserID) qui ont évalué les mêmes films en regroupant les données par 'MID' (MovieID) et 'rate' (évaluation). Cela permet de mesurer la similarité entre les utilisateurs en fonction de leurs évaluations partagées, ce qui est essentiel pour générer des recommandations pertinentes. Pour construire le graphe, le coefficient alpha a été défini à 0,005 en tenant compte de la rareté des données et du fait que les utilisateurs évaluent un nombre limité de films par rapport à l'ensemble de films disponibles. Il a également été observé que pour accorder plus de poids aux utilisateurs ayant des préférences très similaires, une valeur plus petite du coefficient alpha était nécessaire. Les paires de 'UID' sont transformées en arêtes dans un graphe, où chaque noeud représente un utilisateur et chaque arête représente une similarité entre deux utilisateurs. Cette représentation graphique facilite l'analyse des relations entre les utilisateurs et la génération de recommandations basées sur ces relations. Puis, les mesures de centralité telles que le PageRank, la centralité degré, la centralité de proximité, centralité d'intermédiarité, centralité de charge et degré moyen des voisins sont calculées pour chaque utilisateur dans le graphe de similarité. Ces mesures permettent de quantifier l'importance de chaque utilisateur dans le réseau de recommandation, ce qui est utile pour identifier les utilisateurs influents et générer des recommandations personnalisées. Enfin, nous enregistrons le nouvel ensemble de données transformé que nous obtenons en combinant les informations obtenues par le graph et les informations supplémentaires que nous convertissons en données catégorielles, dans un format de fichier pickle.

Dans l'étape suivante, nous divisons nos données en ensembles de test et d'entraînement afin d'appliquer un autoencodeur, qui est un algorithme d'apprentissage automatique, pour extraire de nouvelles caractéristiques et réduire la dimension de l'ensemble de donnée. Le processus commence par la définition d'un modèle autoencodeur (AE) à l'aide d'une fonction nommée "create_ae". Ce modèle est construit avec des couches d'entrée, des couches cachées, une couche de codage et des couches de décodage. Nous utilisons les paramètres suivants : input_dim = 35, hidden_dim = 16, encoding_dim = 4 et noise_factor = 0.5. Cela signifie que chaque exemple de données d'entrée comportait 35 caractéristiques, avec un réseau de neurones cachés composé de 16 neurones. Les données sont compressées dans un espace de représentation latent de dimension 4 avant d'être reconstruites. De plus, un facteur de bruit gaussien de 0,5 est ajouté aux données d'entrée pour aider à la régularisation et à la généralisation du modèle. Une fois l'architecture définie, le modèle est compilé pour l'entraînement. Ensuite, le modèle est

entraîné en utilisant une validation croisée (cross-validation) à 10 plis. Les données sont divisées en ensembles d'entraînement et de validation, et le modèle est entraîné sur chaque pli d'entraînement, avec une validation sur un sous-ensemble de validation. Les courbes de perte de validation sont tracées pour chaque pli, permettant de sélectionner le modèle avec la meilleure performance de validation. Après l'entraînement sur tous les plis, le modèle est évalué sur les données de test pour chaque pli afin de calculer la perte de reconstruction. Enfin, le meilleur modèle est sélectionné en fonction de la perte de reconstruction la plus faible moyenne sur tous les plis, et ce modèle sélectionné est imprimé avec son résumé.

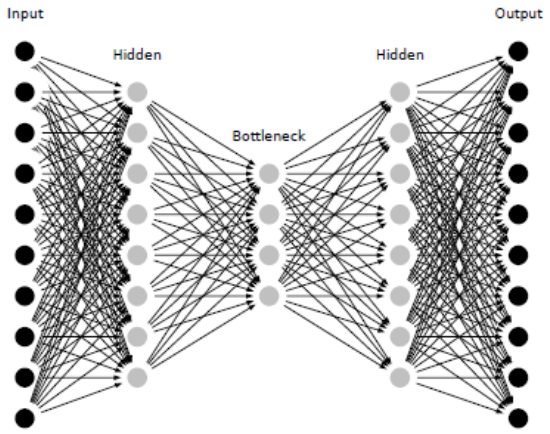

Figure 11. Structure d'Autoencodeur Utilisé [6]

Dans l'étape suivante, nous encodons les nouvelles données que nous avons enregistrées au format pickle en utilisant le meilleur modèle d'encodeur sélectionné. Pour regrouper les utilisateurs en utilisant l'algorithme K-means, il est nécessaire de trouver le nombre optimal de groupes. Pour ce faire, nous utilisons d'abord la méthode de la silhouette moyenne, puis la méthode d'elbow pour rechercher le nombre recommandé de groupes le plus approprié. Comme les Figures 12 et 13 montres, puisque le nombre de groupes recommandés est de 12 pour les deux méthodes, nous acceptons que 12 soit notre nombre optimal de groupes. Nous entraînons l'algorithme K-means avec le nombre optimal de clusters et enregistré le modèle K-means obtenu. Ensuite, il est enregistré les utilisateurs de chaque cluster, comme illustré dans la figure 16, pour une utilisation ultérieure dans l'analyse et la recommandation. Puis, Nous créons une matrice de similarité pour tous les utilisateurs et toutes les groupes sous la forme ["UID", "cluster"]. Nous enregistrons la matrice dans un format de fichier pickle nommé « all_users_cluster.pkl ». Nous faisons de même une matrice de similarité pour des éléments et des groupes et l'enregistrons. Enfin, nous effectuons les prédictions pour le nouvel utilisateur et nous évaluerons les performances du modèle.

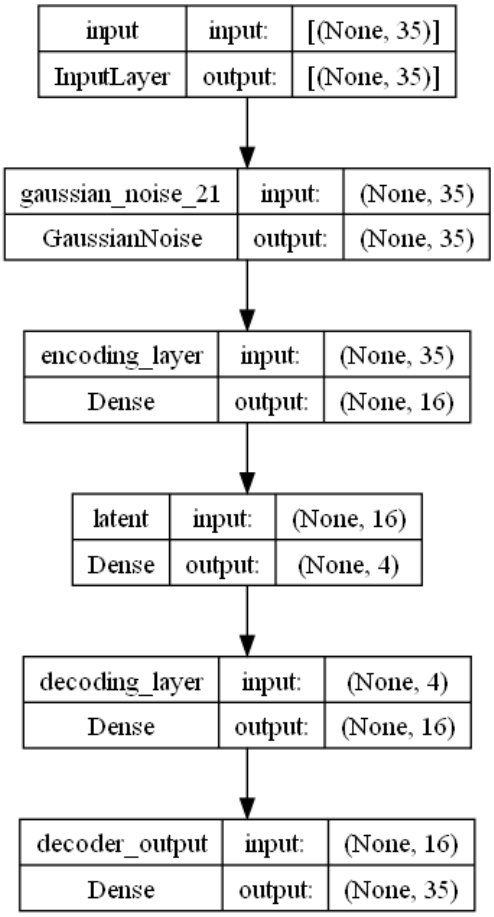

Figure 12. Architecture d'Autoencodeur Appliqué

```
Model: "autoencoder"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input (InputLayer)          [(None, 35)]              0

 gaussian_noise_19 (Gaussian  (None, 35)               0
 Noise)

 encoding_layer (Dense)      (None, 16)                576

 latent (Dense)              (None, 4)                 68

 decoding_layer (Dense)      (None, 16)                80

 decoder_output (Dense)      (None, 35)                595

=================================================================
Total params: 1,319
Trainable params: 1,319
Non-trainable params: 0
_________________________________________________________________
```

Figure 13. Résumé du Modèle Autoencodeur

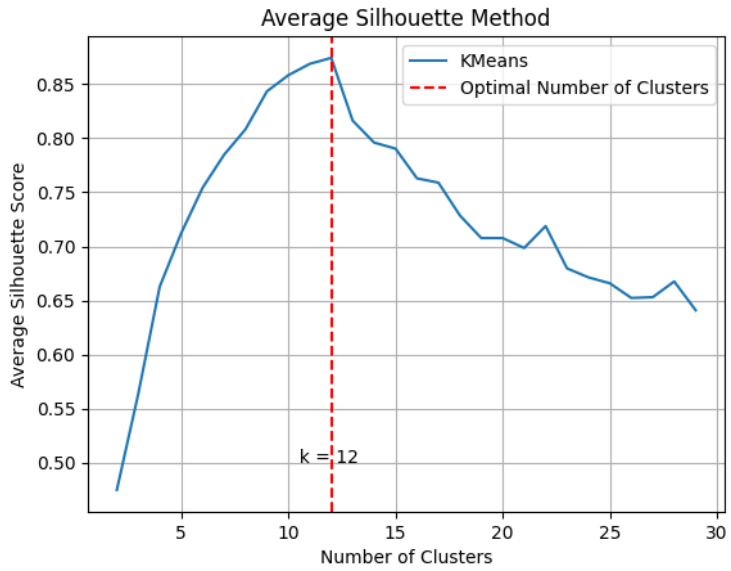

Figure 14. Le Nombre Optimal Proposé par La Méthode De La Silhouette Moyenne

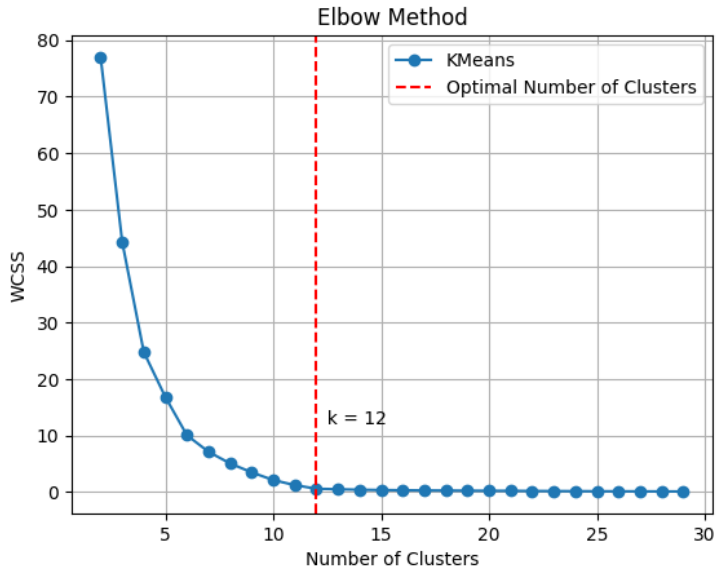

Figure 15. Le Nombre Optimal Proposé par La Méthode d'Elbow

0	1130
1	163
2	338
3	1397
4	437
5	499
6	371
7	855
8	141
9	515
10	135
11	59

Figure 16. Nombre d'Utilisateurs Dans Chaque Groupe

RESULTATS

Pour le moment nous calculons la RMSE (Root Mean Square Error) de 0,95 est une mesure de l'erreur moyenne entre les notes réelles et les notes prédites de toutes les évaluations. Un RMSE de 0,95 signifie que, en moyenne, les prédictions du modèle sont écartées de 0,95 unité de la vérité.

Dans ce cas, nous devons mesurer les performances de notre modèle à l'aide d'autres mesures de performances et apporter les améliorations nécessaires.

BIBLIOGRAPHIE

[1] Burke, R., Felfernig, A., & Göker, M. H. (2011). Recommender Systems: An Overview. AI Magazine, 32(3), 13-18. https://doi.org/10.1609/aimag.v32i3.2361

[2] Salah, A., Rogovschi, N., Nadif, M., 2016. A dynamic collaborative filtering system via a weighted clustering approach. Neurocomputing 175, 206–215

[3] Koren, Y., Bell, R., 2015. Advances in collaborative filtering, in: Recommender Systems Handbook. Springer, pp. 77–118

[4] Lops, P., Gemmis, M., Semeraro, G., 2011. Content-based recommender systems: State of the art and trends, in: Recommender systems handbook. Springer, pp. 73–105

[5] J. Lund and Y. -K. Ng, "Movie Recommendations Using the Deep Learning Approach," *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, Salt Lake City, UT, USA, 2018, pp. 47-54, https://doi.org/10.1109/IRI.2018.00015

[6] Darban, Z. Z., & Valipour, M. H. (2022). GHRS: Graph-based hybrid recommendation system with application to movie recommendation. Expert Systems with Applications, 200, 116850. https://doi.org/10.48550/arXiv.2111.11293

[7] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, Jae Kyeong Kim, A literature review and classification of recommender systems research, Expert Systems with Applications, Volume 39, Issue 11, 2012, Pages 10059-10072, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2012.02.038

[8] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper, Proceedings of the ACM SIGIR'99 Workshop on Recommender Systems

[9] Blanco-Fernandez, Y., Pazos-arias, J. J., Gil-Solla, A., Ramos-Cabrer, M., & Lopez-

Nores, M. (2008). Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems. IEEE Transactions on Consumer Electronics, 54, 727–735.

[10] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Comput. Surv. 52, 1, Article 5 (January 2020), 38 pages. https://doi.org/10.1145/3285029

[11] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, Zuoyin Tang, Collaborative filtering and deep learning based recommendation system for cold start items, Expert Systems with Applications, Volume 69, 2017, Pages 29-39, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2016.09.040

[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In Proceedings of the 26th International Conference

on World Wide Web (WWW '17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. https://doi.org/10.1145/3038912.3052569

[13] Cintia Ganesha Putri D, Leu J-S, Seda P. Design of an Unsupervised Machine Learning-Based Movie Recommender System. Symmetry. 2020; 12(2):185. https://doi.org/10.3390/sym12020185

[14] An Efficient Deep Learning Approach for Collaborative Filtering Recommender System, Procedia Computer Science, Volume 171, 2020, Pages 829-836, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.04.090

[15] Ruiping Yin, Kan Li, Guangquan Zhang, Jie Lu, A deeper graph neural network for recommender systems, Knowledge-Based Systems, Volume 185, 2019, 105020, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2019.105020

[16] Sublime, Jeremie. (2022). L'apprentissage non-supervisé et ses contradictions. Bulletin 1024. 145-156. 10.48556/SIF.1024.19.145.

[17] Yuan C, Yang H. Research on K-Value Selection Method of K-Means Clustering Algorithm. J. 2019; 2(2):226-235. https://doi.org/10.3390/j2020016