People's Democratic Republic of Algeria Ministry of Higher Education and Scientific Research

University of 8 May 1945-Guelma-

Faculty of Mathematics, Computer Science and Science of Matter

Department of Computer Science

Master Thesis

Specialty : Computer Science

Option :

Computer System

Theme

Data Analysis and Processing for Recommendation System

present by :

M^r BOUGHAZI AKRAM

Jury Members:

N	**Full Name**	**Quality**
1	Dr MEHENNAOUI ZOHRA	Chairman
2	Pr KOUAHLA MOHAMED NADJIB	Supervisor
3	Dr BENDJEBAR SAFIA	Examiner

June 2023.

Appreciation

It is commonly stated that the journey is just as significant as the destination. The time we spent in college helped us comprehend the meaning of this sentence. This journey was not accomplished without obstacles and without posing numerous questions whose responses required many hours of effort.

First of all, I would like to thank God for giving me the strength, the knowledge, the ability and the opportunity to undertake this work, to persevere and to complete it satisfactorily.

*I would like to sincerely thank my Supervisor, Professor **KOUAHLA MOHAMED NADJIB,** for guiding me, advising me, and encouraging me throughout the completion of this work, and especially for his patience and understanding in allowing me to produce the results of a lengthy and challenging effort. Thank you for your trust and all you've done to bring me here.*

I also thank my parents for all the sacrifices they have made to enable me to continue my studies in the best possible conditions and to never cease to encourage me throughout these years.

Finally, I thank all the teachers of the Department of Computer Science of the University of Guelma 8 May 1945 for the quality of their training. Many thanks to the LabSTIC laboratory teams who contributed, each in their own way.

Dedication

I dedicate this work

To my dear parents

To my mother, who worked for my success through her love, her support, all the sacrifices made and her precious advice, for all her assistance and her presence in my life, receive through this work as modest as it is-it, the expression of my feelings and my eternal gratitude.

To my father, who can be proud and find here the result of long years of sacrifices and privations to help me progress in life. May God make this work bear fruit. Thank you for the noble values, education and ongoing support.

Not forgetting my dear friends ***ANIS, MALEK, AYMENE, ADEL, HAZEM, LOUFI, MOHAMED, SAYED, SANDRA*** *and all those who supported and encouraged me throughout the period of my work...*

AKRAM

Abstract

In this graduation thesis, we focused on exploring novel methods and approaches in data analysis and processing for recommendation systems. We aimed to address the challenges posed by diverse and heterogeneous data to contribute to the advancement of recommendation systems. By effectively analyzing and processing such data, we can unlock the true potential of recommendation systems, enabling users to make better decisions and discover new experiences.

To build our recommendation system, we established models for points of interest (POIs) and users. Our solution incorporated three key factors: sentiment analysis, user preferences, and ratings, culminating in the integration of the *lightGCN* model. Preprocessing and filtering of the data were performed to ensure data quality, followed by modelling the POIs based on their unique characteristics. The sentiment analysis factor played a crucial role in analyzing user reviews and predicting ratings. By employing sentiment analysis techniques, we accurately represented the user's opinion by aligning the sentiment expressed in textual reviews with the user's rating. The user preference factor enabled us to recommend the most suitable POIs based on individual preferences and interests. The rating factor, which examined the ratings users gave to visited POIs, facilitated tracking and updating user preferences. This allowed for dynamic adjustment and refinement of the user preference profile, ensuring recommendations aligned with their evolving interests. The culmination of these factors, along with the preprocessing, filtering, and modelling of the POIs, led to the integration of the *lightGCN* model. By combining similarity scores derived from the user preference profile, sentiment analysis score, and ratings score, the *lightGCN* model predicted the most suitable POIs for each user, enhancing the recommendation system's accuracy.

During the experimentation phase, we utilized Yelp datasets in a Jupyter environment to preprocess, filter, and model the POIs, incorporating sentiment analysis of reviews. The recommendation system, developed in the same environment, utilized the combined results of the three factors and the *lightGCN* model to provide improved POI recommendations for users.

Key Word: *Recommendation, Data Analysis and processing, Point Of interest (POI), preference, lightGCN, Sentiment analysis.*

Contents

List of Figures

List of Tables

General Introduction

This introduction presents the overall context of the work, the problem and objective that we have dealt with, and the general structure of the paper.

In the digital age, recommendation systems have become an indispensable tool for aiding users in making informed decisions. These systems make use of extensive datasets to provide personalized suggestions across different categories like movies, products, restaurants, and points of interest. However, effectively analyzing and processing this vast and diverse data presents a significant challenge. The data may vary in terms of formats, sources, and characteristics, making it difficult to extract meaningful insights and generate accurate recommendations. Addressing this challenge requires advanced techniques and algorithms to handle the complexity and heterogeneity of the data, ensuring the recommendations are relevant and valuable to the users.

This graduation thesis focuses on addressing the challenges associated with **data analysis and processing for recommendation system** to deal with heterogeneous data in recommendation systems and recommend the POIs that match the users' preferences. Our approach combines a hybrid methodology that incorporates sentiment analysis, data filtering, preprocessing stages, and graph neural network to enhance the accuracy and relevance of recommendations. By leveraging these advanced techniques, we aim to improve user satisfaction and engagement with the recommendation system.

One of the primary hurdles we tackle is the **continuous update of user preferences**. As user preferences evolve over time, it becomes crucial to adapt the recommendation system accordingly. Our proposed solution aims to dynamically update user preferences by incorporating users' feedback with users' preferences. By analyzing users' feedback, we can effectively track their evolving preferences and refine the recommendations accordingly.

Furthermore, ensuring that **users' feedback aligns with their real experiences with visited POIs** poses another challenge. It is vital to validate whether the recommendations provided by the system match users' expectations and whether they are satisfied with their chosen points of interest. To overcome this obstacle, we apply **sentiment analysis** techniques to evaluate users' feedback, enabling us to assess the match between recommendations and their actual experiences.

To optimize the recommendation process, we employ **data filtering and preprocessing** stages. These stages are essential for extracting relevant information from the vast and diverse datasets, thereby enhancing the accuracy and efficiency of the recommendation system. By filtering and preprocessing the data, we can **eliminate noise and irrelevant** information, ensuring that the recommendation algorithm operates on the most pertinent and meaningful data.

In addition, we utilize a **graph neural network** for recommendation purposes. Graphs provide a **flexible and intuitive** representation of the relationships between users, items, and their associated attributes. By leveraging graph structures, we can capture complex user-item relationships, uncover latent patterns, and generate more accurate recommendations.

Overall, we present a comprehensive approach to address the challenges of **data analysis and processing for recommendation system** of heterogeneous data in recommendation systems. By combining **sentiment analysis**, **data filtering**, **preprocessing** stages, and **graph neural network**, we aim to enhance the accuracy, relevance, and user satisfaction of recommendation systems. The subsequent sections delve into the details of our proposed methodology, highlighting the recommendation system and the implementation, evaluation, and the resulting benefits of our hybrid approach.

This work is divided into three different chapters, organized as follows:

Chapter 1: In the first chapter, we provide an overview of recommendation systems, delving into their definition and how they function. We explore the intricate relationships within recommender systems and discuss the different types, such as collaborative filtering, content-based filtering, and hybrid approaches. Additionally, we examine the process of acquiring data for recommendation systems and delve into the various methods and evaluation metrics used to measure their effectiveness. Moreover, we explore the integration of sentiment analysis into recommendation systems and review relevant works in the field.

Chapter 2: In the second chapter, we introduce our innovative approach for data analysis and processing in recommendation systems. We discuss the conception and modelling of our approach, highlighting the key components and techniques employed. This includes the incorporation of sentiment analysis to capture the emotional aspect of user preferences, as well as the integration of graph neural network for enhanced recommendation accuracy. We present the architecture and workflow of our approach, showcasing its potential to improve the recommendation process.

Chapter 3: The third chapter focuses on the experimental evaluation of our approach. We describe the Yelp datasets, the experimental setup, and the methodologies employed. We present the results obtained from applying our approach using Yelp dataset, comparing its performance with other state-of-the-art recommendation models and techniques on the same dataset. Furthermore, we provide insights and analysis of the experimental outcomes. To illustrate the practical application of our approach, we include a detailed scenario example demonstrating its effectiveness in generating personalized recommendations.

In the final section of this graduation thesis, we summarize the key findings and contributions of our research.

Part I

State of the art

Chapter 1

Recommendation system

1 Introduction

Recommendation systems play a crucial role in today's digital landscape, offering immense value and convenience to users across various domains. Their importance cannot be overstated, as they significantly enhance the intelligence and efficiency of platforms and services. Recommendation systems help people from many areas of life, not only tourists, by streamlining the selection process and providing relevant, customised recommendations.

By leveraging advanced algorithms and data analysis techniques, recommendation systems empower users to discover relevant and high-quality options amidst an overwhelming abundance of choices. Whether it's finding the ideal POI for a special occasion, these systems provide tailored recommendations based on factors such as past preferences, user behaviour, and contextual information.

This chapter focuses on these systems and examines in greater detail their fundamental concepts, their means, and their methods before presenting a number of works in various fields based on these systems.

2 Recommendation System RS

2.1 Definition

Consider them as instruments that assist users with information inundation, enhance customer relationship management, and provide recommendations (customized products and services) to users [1].

According to ADOMAVICIUS and TUZHILIN [2], recommendation systems predict ratings for unknown products for each user by utilizing ratings from other users, and then recommend the K top items with the highest predicted rating value.

2.2 How RS work: understanding the relations

Relationships provide recommendation systems with a thorough comprehension of consumers and their requirements. There are three major categories of relationships: product-user, product-product, and user-user.

2.2.1 Relationship user product

When particular consumers have a liking or preference for particular items they desire for, the user-product relation is established. For instance, a football player may have an eye for football-related merchandise, so social networks establish a user-to-player item relationship [3].

2.2.2 Relationship product-product

Product-to-product relations exist when two or more objects have similarities in their appearance or explanation. [3].

2.2.3 Relationship user-user

User-user relations exist when consumers share matching tastes for something in particular, such as having common acquaintances, similar tastes, or residing in the same city [3].

2.3 Recommendations types

There are a lot of recommendation types; we'll talk about three principal types below.

2.3.1 Content based

This strategy recommends products that are comparable to those favored by a particular user, Figure 1.1 represent illustration of how CB recommendation works. Content-based recommendation systems analyze the features and properties of items, such as text, metadata, or audiovisual content, to generate personalized recommendations for users based on their preferences and past interactions. [4]

There are two techniques used for making recommendations. The first is straightforward and employs measurements of similarity, like the Euclidean distance [5].

The second technique generates recommendations using techniques such as evaluative learning and machine learning [6]. the advantage and disadvantage of this technique are in table 1.1

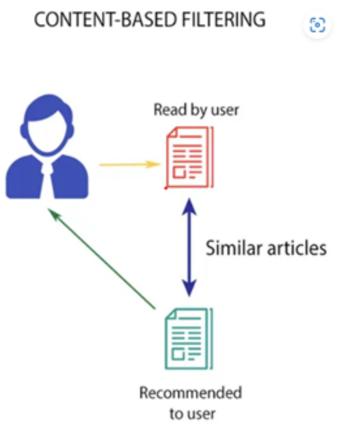


Figure 1.1: Content-based recommendation system [7]

Advantages	Disadvantages
It functions even when an item lacks feedback from customers.	Requires detailed information of all recommended content, that takes time.
	Because consumers have differing opinions about every item, it can be hard to set up massive item databases.

Table 1.1: Advantages and disadvantages of content-based RS techniques

2.3.2 Collaborative filtering

Collaborative filtering (CF) enables individuals to make decisions according to the ratings and preferences of others who match similar interests, Figure 1.2, such as nation, sex, and age, or who favor similar products. To recommend a novel item, CF-based RS utilize the interests of users with same tastes for specific products [8]. the advantage and disadvantage of this technique are in table 1.2

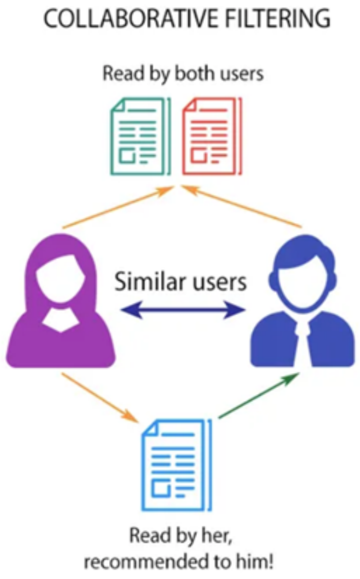


Figure 1.2: Collaborative filtering recommendation system [7]

Advantages	Disadvantages
It doesn't need any item's details.	We cannot endorse the product if there don't exist user evaluations (cold start problem).
	It is difficult to recommend items to new users, and people usually choose renowned items with many reviews.
	It gets difficult to recommend innovative items as fewer user have an opinion about them.

Table 1.2: Advantages and disadvantages of collaborative filtering RS techniques

2.3.3 RS hybrid

To improve efficiency and get around the problems with traditional recommendation methods, a hybrid recommendation technique that blends the best parts of multiple recommendation methods has been suggested [9], Figure 1.3. There are seven fundamental hybridization techniques of the combos used to create hybrids by recommendation systems: **Weighted** [10], **mixed** [11], **switching** [12], **characteristic combination, characteristics increase** [13, 14], **cascade** [15] and **meta-level** [16].

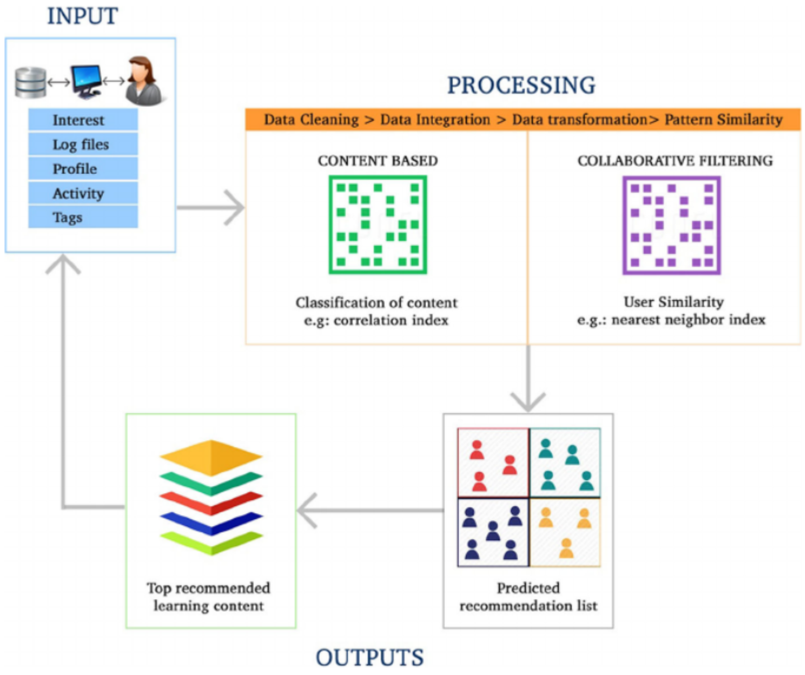


Figure 1.3: Hybrid filtering recommendation system [17]

3 Data Acquisition

Data acquisition is a crucial stage in the process of making recommendations. Data can be provided in various formats. There are three essential collection methods: explicit, implicit, and hybrid.

3.1 Explicitly

Typically, the system requires users to evaluate aspects via the GUI of the system for the purpose of building and enhancing its model. The sole downside of this strategy is that it demands effort from users, and users are not always willing to offer enough information. Despite the fact that specific comments demand more work from the user, they are thought to produce more trustworthy data.[18]

3.2 Implicitly

Monitoring the user's diverse actions, such as purchase history and past searches, the system automatically determines what the user wants. By deducing user preferences from their interaction with the system, implicit comments reduce the load on users. Although the technique requires no exertion on the part of the user, its accuracy is diminished. Moreover, it has been suggested that implicit preference statements may in fact remain more goal-oriented, as there is no bias that results from users responding in a socially desirable manner, as well as no self-image issues or the need to maintain the image for others.[18]

3.3 hybrid

The benefits of both implicit and explicit outputs can be mixed into a hybrid system to mitigate their weaknesses and create the most effective system possible.[18]

4 Recommendation System Methods

Recommendation systems are comprised of a set of tools and techniques that allow them to function and adapt to current circumstances in order to find a rapid and successful solution.

4.1 Similarity Methods

Analysing methods for quantifying degrees of similarity between users or products; this can aid in spotting shared traits and facilitating relevant recommendations.

4.1.1 The Euclidean Distance

With users serving as a focal point, this strategy has been effective. This metric is used to translate the Euclidean distance d between any two such users. The closer in similarity the users are, the less the distance value.[19]
To define out how similar two people are, we must first use the equation (1.2) to figure out how far apart they are, and then use the equation (1.1) to define out how similar they are.

$$\sum(U_a, U_b) = \frac{1}{1 + DIS(Ua, Ub)} \tag{1.1}$$

$$Dist(U_a, U_b) = \sum_{i=1}^{j} |R_u(a,i) - R_U(a,j)|^2 \tag{1.2}$$

4.1.2 cosine Similarity

Calculates similarity by quantifying the cosine angles generated among the two evaluation vectors provided by the user. Smaller angle values indicate greater similarity, and the opposite is true.[20]
The similarity in cosine is calculated [21] by the formula (1.3):

$$Sim(U, P) = \frac{\sum_{i=1}^{n} P_i C_i}{\sqrt{\sum_{i=1}^{n} P_i^2} \times \sqrt{\sum_{i=1}^{n} C_i^2}} \tag{1.3}$$

4.2 Classification

Exploring techniques for categorizing users or items into distinct classes or groups, enabling personalized recommendations based on shared characteristics or behavior patterns.

4.2.1 Naive Bayes NB

There is a group of classification algorithms called Naive Bayes classifiers that are grounded in the Bayesian theorem. Each algorithm in this family follows the same guiding principle, which is that classifications of any given pair of features should be made separately from any other pairs.[22]

4.2.2 Support Vector Machines SVM

The supervised machine learning approach known as support vector machines (SVM) is grounded on statistical learning theory and operates on the premise of Structural Risk Minimization as opposed to Empirical Risk Minimization. For data that falls into two categories, the SVM method locates the hyperplane with the biggest margin of separation, producing the largest possible gap between the hyperplane and the instances on either side of it.[23]

5 Evaluation

Various metrics can be used for evaluating the quality of a recommendation algorithm, and these metrics depend on the filtration strategy employed. we'll talk about the must widely used metrics to evaluate the recommender system,

- **PRECISION :** Or True Positive Accuracy this is the probability that a recommended item matches the user's inclinations and is determined as the proportion of pertinent recommendations to the overall number of recommendations.[24]

$$PRECISION = \frac{TruePositives}{TruePositives + Falsepositives} \tag{1.4}$$

- **RECALL :** Or true positive rate (also called sensitivity in psychology), It is determined as the proportion of pertinent recommended items to the overall quantity of pertinent items. This represents the probability that a pertinent item will be recommended.[24]

$$RECALL = \frac{FalseNegatives}{TruePositives + FalseNegatives} \tag{1.5}$$

- **MAP :** Mean average precision is a common search engine metric. It determines the accuracy of the recommendation set using the measurement corresponding to the pertinent item's position for each pertinent item. The mathematical mean of each of these accuracy levels is subsequently calculated. The overall mean average precision is determined by computing the mathematical mean of all users' average precisions.[24]

$$AP = \frac{\sum_{r=1}^{N}(P(r) \times rel(r))}{number\ of\ pertinent\ documents} \tag{1.6}$$

  In 1.6 formula, N represents the total number of documents in the ranking, $P(r)$ represents the precision at rank r, and $rel(r)$ is an indicator function that is 1 if the document at rank r is relevant and 0 otherwise.

$$MAP = \frac{\sum_{u=1}^{M} AP_u}{M} \tag{1.7}$$

  In 1.7 formula, M represents the total number of queries and $AP(u)$ is the AP score for the $u-th$ query.

- **NDCG :** Normalized Discounted Cumulative Gain it's similar to MAP metric, They both value placing highly pertinent documents at the top of suggested reading collections. However, the NDCG refines its assessment of recommended lists. It can utilize the fact that certain documents are more pertinent than others. Items with high relevance should come before those with medium relevance, which must come before those with no relevance.[25]

$$NDCG = \sum_{i=1}^{p} \frac{2^{R_i} - 1}{\log_2(i+1)} \tag{1.8}$$

In 1.8 formula, p represents the position of the item in the ranked list. Ri is the relevance score of the item at position i. The relevance scores are typically normalized to a certain scale (e.g., 0 to 1). $log2(i+1)$ is the logarithm (base 2) of the position i plus 1.

6 recommendation and sentiment analysis

Sentiment analysis, also known as opinion mining or sentiment analysis, is a subfield of automated language processing research.(TAL). It involves identifying positive or negative emotions, opinions, or assessments conveyed within an information unit. (i.e., paragraph, or document)[26]

6.1 Approaches to Measuring sentiment

There are essentially two methods for determining the sentiment of a text: the lexical method and statistical or machine learning methods.

6.1.1 The lexical approach

Based on the dictionary, lexicons are utilized. A predefined dictionary in which each term is assigned a polarity: predominantly positive, predominantly negative, and occasionally neutral. These are primarily verbs, adjectives, and adverbs, as well as prevalent nouns. This method has the advantage of ensuring the clarity of the classification criteria. To enhance results, adding, modifying, or removing terms from lexicons is simple and effective. However, the manual construction of such lexicons necessitates extensive effort with a relatively narrow scope. Terms must be specified for identification. In contrast, sentiment analysis lexicons are frequently restricted to single terms (termed uniterme or unigram)[26]

6.1.2 Approach to Machine Learning

Machine learning is a subfield of artificial intelligence that confers understanding through algorithms on a system. The goal is for computers to learn how to tackle specific tasks without being programmed by learning algorithms from data and making predictions based on that data.[27]

6.2 sentiment analysis method «VADER»

The Vader is a sentiment analysis instrument based on rules and specifically designed for social media. It was created by Hutto and colleagues in 2014; it accepts a chain and returns dictionaries of scores for each of the four categories. $Negative, Neutral, Positive, Composite.$[28] However, when a person experiences a positive emotion, it indicates that their need has been met. A negative emotion indicates that a person's need is not being met.[29]

Vader assigns a total score indicating whether a phrase is positive, negative, or neutral by summing the valence scores of every word in the lexicon, adjusting based on the rules, and then normalizing the score between -1 and +1.[30]

- $positive \geq 0,05$
- $neutral\ between\ -0,05\ and\ 0,05$
- $negative \leq -0,05$

And this method was chosen to be used in our application.

7 Related work

In this section, we review several key themes in the field of recommendation systems. Recommending points of interest (POIs) is an active research field, where algorithms aim to suggest relevant locations or venues to users based on their preferences and historical data. Deep learning has emerged as a powerful technique for recommendation systems, leveraging complex neural network architectures to extract meaningful patterns and make accurate predictions. Furthermore, we explore a recommender system based on the Yelp dataset, which provides valuable insights into user preferences and allows for personalized recommendations. Collaborative filtering, a widely used approach, involves analyzing user-item interactions to identify similarities and make recommendations based on the preferences of similar users. Content-based methods, on the other hand, focus on extracting features from items themselves, such as textual descriptions or metadata, to make recommendations based on item similarity. Lastly, hybrid approaches combine multiple techniques, such as collaborative filtering and content-based methods, to leverage the strengths of each approach and improve recommendation accuracy. By examining these different approaches, we gain a comprehensive understanding of the state-of-the-art techniques in recommendation systems. Through this exploration, we aim to contribute to the existing body of knowledge and shed light on the advancements made in the design and implementation of effective recommendation systems.

7.1 Synthesis of research on the recommendation topic:

7.1.1 Point-of-Interest (POI) Recommendation

In the field of point-of-interest (POI) recommendation, a number of methods have been proposed to improve the precision and individualization of recommendations.
Carl Yang et al. introduce the PACE architecture, combining collaborative filtering and semi-supervised learning to recommend POIs of interest. It utilises a POI autoencoder and a user encoder, achieving promising results across various real-world check-in datasets [31]. Using geo-tagged photos with textual descriptions to extract user preference topics for comprehensive POI the city's recommendations [32] is a second technique that focuses on author-topic model-based collaborative filtering. Trust between users plays a significant role in another proposed method, which incorporates user covisiting relationships and network representation learning. This approach enhances user similarity and integrates geographic and temporal influences, improving the accuracy of POI recommendations [33]. Similarly, addressing the issue of overspecialization, a thesis proposes an algorithm leveraging check-in data to create user-location metrics and recommend new and interesting places.[34]. Urban POI-Mine (UPOI-Mine) presents a multistep approach that mines user preferences and check-in behaviours in order to recommend interesting urban POIs, which are restaurants. It entails extracting features from context-aware and environmental data and using a regression-tree model for prediction[35]. Another study by Mao Ye incorporates user preference, social influence, and geographical influence through user-based collaborative filtering, social ties analysis, and a power-law probabilistic model for spatial clustering in user check-in activities [36]. Lastly, using user check-in data from location-based social networks, this method predicts user-POI interactions[37].
These various approaches demonstrate the ongoing efforts to enhance the accuracy, personalisation, and diversity of POI recommendation systems, leveraging techniques such as collaborative filtering, content analysis, network representation learning, and geographical influences.

Articls	Technique	Database	Metrics	Result
[31]	PACE (Collaborative filtering and semi-supervised learning with a neural approach)	Real-world check-in datasets (Gowalla check-in dataset and the Yelp datase)	Pre@K, Rec@K, nDCG@K, MAP@K	8.5% improvements on Gowalla more than 6% improvements on Yelp
[32]	Author topic model-based collaborative filtering	Large collection of data	MAP and MAP@n	MAP= outperforms other methods MAP@n= when increases, the performance of all methods decreases
[33]	Network representation learning with trust-enhanced user similarity	Real-world LBSN datasets	Precision@K, Recal@Kl	best performance
[34]	Algorithm to recommend new interesting places	Check-in data from Gowalla	Precision, Recall, f-measure	impressive result
[35]	mining urban users' check-in behaviors. and propose UPOI-Mine ,regression-tree model	Dataset crawled from Gowalla	NDCG, MAE	better result
[36]	User-based collaborative filtering , social influence modeling, geographical influence modeling	real datasets (crawled the websites of Foursquare and Whrr)	Precision, Recall	better performance
[37]	Geography-aware inductive matrix completion approach	User check-in data from LBSNs	Area under ROC curve (AUC)	best performance

Table 1.3: comparison table of previous POI approach

7.1.2 recommender system based on yelp dataset

The Yelp dataset is a rich and diverse collection of user-generated content that provides valuable insights into various businesses and services. It encompasses a vast range of reviews, ratings, and other relevant information about local businesses, restaurants, shops, and more. With millions of reviews and ratings from a large and diverse user base [38].

In the field of Yelp dataset recommendation systems, several studies have been conducted to enhance the user experience and provide personalized recommendations. One approach, as described in the study by Sawant and Pai, utilized Singular Value Decomposition (SVD) to reduce the dimensionality of the feature space. They employed a hybrid cascade of K-nearest neighbors (KNN) algorithm, where the first KNN was applied on businesses and the second KNN on users to predict ratings for users on specific businesses. Additionally, they introduced a hybrid algorithm called Cascaded Clustered Multi-Step Weighted Bipartite Graph Projection, which aimed to leverage both strategies to improve the weighted bipartite graph projection algorithm. Another study [39], conducted by Chen et al. proposed a double-layered recommendation algorithm based on fast density clustering for Yelp social networks' dataset. The algorithm involved clustering users and items based on their features and constructing a double-layered bipartite network model. Recommendations were derived from the network structure and clustering results. The authors also incorporated regular updates to the recommender system using a time-weight function [40]. Ting and Ramaswamy focused on generating personalized business recommendations for Yelp users. Their approach involved predicting user ratings for businesses using various models. Principal Component Analysis (PCA) was applied to reduce complexity and eliminate random noise, while segmentation ensemble assigned different weights to each feature, leading to improved overall performance [41]. Le Xu and Xu aimed to develop a friend and business recommendation system using Yelp data. They employed K-means clustering and matrix factorization techniques to accomplish this goal [42]. In the realm of NLP analysis and recommendation systems for Yelp, Sun applied NLP techniques to user restaurant reviews. Preprocessing models such as unigram, bigram, trigram, and Latent Dirichlet Allocation (LDA) were used, along with text tokenization, normalization, and stop words. The study concluded with word vectorization using the Word2Vec method to capture word meanings. For recommendations, a location-based approach was adopted by applying K-means clustering to group restaurants in Las Vegas based on their locations [43].

These prior works contribute to the development of Yelp recommendation systems by exploring various techniques such as SVD, KNN, clustering, PCA, and NLP analysis. By leveraging these approaches, researchers have strived to enhance the accuracy and personalization of recommendations, ultimately improving the user experience.

article	Techniques And Methods used	Dataset	Metrics Evaluated	Result

[44]	KNN SVD,
	BIPARTITE GRAPH
		Yelp Dataset	RMSE,MAE	RMSE=1.092
				MAE=0.675

[45]	clustering , bipartite network		precision@K,recall@K,f1@K	Good result
[46]	PCA, segmentation ensemble		RMSE	Best result
[47]	K-Means , matrix factorization		RMSE	increase of performance

[48]	unigram, bigram trigram, LDA, word2vec, k-means		RMSE,MAE	RMSE=0.97
				MAE=0.75

Table 1.4: comparison table of previous recommender system based on yelp dataset

7.2 Synthesis of research on the type or technique of recommendation:

7.2.1 collaborative filtering

Several studies have been conducted on restaurant and hotel recommendation systems utilizing diverse methodologies. Jing Sun et al. a probabilistic factor analysis framework called RMSQ-MF was proposed to provide more accurate and efficient restaurant recommendations. The framework incorporated side information such as user profiles, restaurant characteristics, social factors, and mobility factors [49]. Another study by Alif Azhar Fakhri et al. focused on a user-based collaborative filtering approach for restaurant recommendations. The method involved building a user-item matrix, calculating the similarity between users, and using k-nearest neighbor prediction to recommend the top N items [50]. A restaurant recommendation system based on user ratings with collaborative filtering was explored in another study[51]. They used the Pearson correlation function to ascertain the degree of user similarity in restaurant recommendation systems based on previous ratings given by visitors or users. For hotel recommendations, a multi-criteria collaborative filtering approach was proposed by Qusai Y Shambour et al. This method aimed to provide accurate recommendations by considering crucial aspects such as price, location, amenities, and user ratings [52]. Additionally, a hotel recommendation system incorporating review and context information was investigated[53]. Techniques such as Google Spell Check, Stanford Core NLP toolkit, and TF-IDF were employed for pre-processing, feature extraction, and similarity calculation. Furthermore, research by KV Daya Sagar et al. focused on identifying contextual segments with a high impact on user overall ratings from different hotel classes and trip types. Collaborative filtering and regression techniques were used [54].

7.2.2 content based

Several studies have been conducted on restaurant and hotel recommendation systems. In [21], a restaurant recommender system is proposed for a mobile environment, utilizing user preference modelling and location information. In another study, Anant Gupta and Kuldeep Singh, personalized restaurant recommendations based on location and behavioral patterns are discussed [55]. Chung-Hua Chu and Se-Hsien Wu, introduces a restaurant recommendation system based on mobile context-aware services, taking into account user location, time, and preferences. The system utilizes contextual information, transforms it into vectors, and provides flexible recommendations [56]. For hotel recommendations, Cheryl Ayu Melyani, develops a hotel recommendation system using content-based filtering, employing TF-IDF weighting and cosine similarity. The dataset used in this study comprises hotel description data, and the accuracy of the system is evaluated with a reported value of 75% [57]. Additionally, Agung Muliawan et al. propose a hotel recommendation system using content-based filtering, K-Nearest Neighbour, and the Haversine formula [58]. Finally, Kristian Wahyudi et al. in this recommendation system a technique recommends items similar to those a given user has liked in the past by calculating the rating of hotel categories in a city [59].

7.2.3 hybrid

In the field of restaurant and hotel recommendation systems, several studies have explored the use of hybrid approaches to improve accuracy and address the limitations of individual techniques.

M Govindarajan proposes a hybrid classification method for restaurant review classification, combining different models and techniques for document pre-processing, feature extraction, and model training [60]. Another study by Realdo Dias et al. presents a hybrid framework for a restaurant recommendation, integrating collaborative filtering, content-based filtering, and knowledge-based filtering [61]. Additionally, Wei-Ta Chu and Ya-Lun Tsai, focuses on a hybrid recommendation system that incorporates visual information and text-based data to predict users' favorite restaurants [62]. Furthermore, a hybrid multi-criteria hotel recommender system is developed by [63], leveraging content data, user similarities, and implicit/explicit feedback within a layered architecture. Similarly, another work employs a hybrid approach using gradient boosting and neural networks for content-based and collaborative filtering to enhance hotel recommendations[64]. Lastly, a proposed hybrid-based recommendation method combines content-based and collaborative systems, utilizing clustering, Boolean data conversion, and association rule mining to improve accuracy and address data sparsity[65].

Methods	Articls	Technique	DataBase	Metrics
colaborative Based	[49]	Probabilistic factor analysis (RMSQ-MF)	Yelp.com (Manhattan district)	MAE, RMSE
	[50]	User-based collaborative filtering	zomato.com (Bandung), questionnaire data	MAE
	[51]	Pearson correlation function	Kaggle (restaurant and rating datasets)	RMSE
	[52]	Fusion-based Multi-Criteria Collaborative Filtering (FB-MCCF)	TripAdvisor MC dataset	MAE, RMSE
	[53]	context-aware personalized hotel	TripAdvisor.com	RMSE, MAE
	[54]	Regression	TripAdvisor.com	not specified
Content Based	[21]	BMCS, BWCS	Not specified	Not specified
	[55]	ML	Not specified	Not specified
	[56]	Context-aware services	Not specified	Not specified
	[57]	TF$_IDF, CosineSimilarity$	Nusatrip.com	Not specified
	[58]	KNN, Haversine Formula	hybrid data	Not specified
	[59]	similarity of two user	dataset from Kaggle	Precision, Recall, accuracy
Hybrid	[60]	bag-of-words and TF-IDF. (BFS), NB, SVM. GA	Yelp Dataset	Accuracy
	[61]	Semantic Web technology,Knowledge-based Filtering	Not specified	Not specified
	[62]	Deep Learning for Visual Feature Extraction, Content-based Filtering, Collaborative Filtering	social platform dedicated to restaurants	AUC
	[63]	Content-based Filtering, Collaborative Filtering, Multi-criteria Rating Approach, Layered Architecture	TripAdvisor	MAE, MSE, MRSE
	[64]	LightGBM,XGBoost	Not specified	Not specified
	[65]	Clustering, association mining algorithm	Public hotel dataset with user ratings	Sparsity, MAE

Table 1.5: comparison table of several recommendations approaches for hotels and restaurants.

7.3 Synthesis of research on the recommendation model:

7.3.1 Deep Learning And Recommendation System

In the field of personalised recommendation systems, several articles have made significant contributions.

Naumov et al. developed a deep learning recommendation model that combines matrix factorization (MF) and multilayer perceptron (MLP) techniques, improving accuracy through a specialised parallelization system [44]. Wang and Wang proposed a novel approach for content-based and hybrid music recommendation using deep learning, surpassing warm-start and cold-start stages without relying on collaborative filtering [45]. Zheng et al. introduced DRN, a deep reinforcement learning framework for news recommendation, utilizing a deep Q-network (DQN) architecture and an improved exploration method to enhance accuracy [46]. Van den Oord et al. addressed the semantic divide in music by training a deep convolutional neural network (CNN) for content-based music recommendation [47]. Zhang et al. integrated collaborative filtering with a deep neural network (DNN), increasing capacity and mitigating sparsity issues in recommendation systems [48].

These prior works cover diverse factors and methodologies, including deep learning models, reinforcement learning, offering valuable insights for the development of accurate and effective recommendation systems.

article	Techniques And Methods used	Dataset	Metrics Evaluated	Result
[44]	MF, MLP	random, synthetic and public data sets (The Criteo AI Labs Ad Kaggle and Terabyte data sets)	Accuracy	Accuracy = 0.79

[45]	DBN, PMF	Echo Nest Taste Profile Subset	RMSE, MAP	RMSE= 0.25
				MAP=0.013

[46]	DQN, DBGD	dataset collected from a commercial news recommendation application	Precision@k, nDCG, CTR	Precision@k=0.0149
				nDCG=0.0492
				CTR=0.0113

[47]	CNN, MSE, bag-of-word, WMF	The Million Song Dataset (MSD)	MAP, AUC	MAP=0.00672
				AUC=0.77192

[48]	QPR, DNN	MovieLens-100K, MovieLens1M, and Epinions.	RMSE, MAE	RMSE= 0.98
				MAE=0.69
				RMSE=0.93
				MAE=0.65
				RMSE=1.2
				MAE=0.17

Table 1.6: comparison table of previous Deep Learning And Recommendation System

7.3.2 recommendation systems based on graphs

Several studies have focused on improving recommender systems using graph convolutional networks (GCNs) and collaborative filtering techniques. The mechanism of how GCNs contribute to recommendation, particularly the core components such as neighbourhood aggregation, has been explored in various research.

Peng et al. investigate the efficacy of GCNs for recommendation by spectrally analysing them. They propose a new GCN learning algorithm that replaces neighbourhood aggregation with a Graph Denoising Encoder (GDE) to capture essential graph features. This method achieves comparable performance to indefinite-layer GCNs while dynamically adjusting gradients over negative samples [66]. Wu et al. propose a self-supervised graph learning (SGL) paradigm to improve recommendation accuracy and robustness against interaction noise on user-item graphs. This involves generating multiple views of nodes and maximizing agreement between views of the same node [67].

Collaborative filtering (CF) techniques have been widely employed for parameterizing users and items into latent representation spaces. GNN-based recommender systems, have demonstrated state-of-the-art performance. Xia et al. propose Hypergraph Contrastive Collaborative Filtering (HCCF) to tackle two challenges: the over-smoothing effect of deeper graph-based CF architectures, and the scarcity and skewed distribution of supervision signals. HCCF captures local and global collaborative relations using a hypergraph-enhanced cross-view contrastive learning architecture, and integrates hypergraph structure encoding with self-supervised learning to reinforce the quality of recommender systems. Experimental results demonstrate the superiority of the proposed model compared to existing methods [68]. The issue of popularity bias in recommender systems is addressed by Wei et al. from a cause-effect perspective. Wei et al. propose a causal graph-based solution to tackle popularity bias in recommender models, using multi-task learning and counterfactual inference to eliminate the effect of item popularity. This approach can be easily implemented in existing models, and experimental results on real-world datasets demonstrate its effectiveness [69]. Hu et al. propose a framework called Markov Graph Diffusion Collaborative Filtering (MGDCF) to investigate GNN-based CF from the perspective of Markov processes for distance learning. They introduce a novel GNN encoder called Markov Graph Diffusion Network (MGDN) which learns vertex representations by considering two types of distances through a Markov process. To optimize MGDCF, they propose the InfoBPR loss function, an extension of the commonly used BPR loss that leverages multiple negative samples for improved performance. Experiments are conducted to provide a detailed analysis of MGDCF's effectiveness [70].

Articls	Technique - Methods	Dataset	Metrics Evaluated	Result
[66]	Spectral Features', Analyse and encode	Pinterest, CiteULike-a, MovieLens, Gowalla	nDCG@k, RECALL@k	outperforms all results
[67]	self-supervised learning. Contrastive learning, GCN combination	Yelp2018, Amazon-Book, Alibaba-iFashion		
[68]	Graph-based message passing module, Hypergraph neural network with global dependency structure learning, Hypergraph contrastive learning architecture	Yelp, Movielens, Amazon-Book		
[70]	(MACR) framework	Gowalla, Yelp2018, Amazon-Book		competitive or superior performance
[69]	model-agnostic counterfactual reasoning framework	Yeep, Gowalla, ML10M, GLOBO, Adressa	HR@k, RECALL@k, NDCG@k	Performance improvement

Table 1.7: comparison table of previous recommender based on graphs

8 Conclusion

In this chapter, we have explored the landscape of related work in the domain of recommendation systems. Our analysis has led us to propose a novel approach that integrates data analysis techniques and graphs to offer personalized recommendations for Point-of-Interest (POI) selection 5. Unlike traditional models that rely on complex techniques such as matrix factorization or deep neural networks 7.3.1, 7.2.1, 7.2.2, 7.2.3, we have employed the *LightGCN* [71] model, which simplifies the recommendation process by leveraging the graph structure of user-item interactions. By utilizing the Yelp dataset 3, we have trained the *LightGCN* [71] model and incorporated sentiment analysis using VADER 6.2 to analyze user feedback and enhance the recommendation accuracy 7.2.1. Furthermore, we have addressed the dynamic nature of user preferences by introducing a method to continuously track and analyze user behaviors 7.1.2, ensuring that the recommendations align with their evolving needs.

Building upon this research, the next chapter will delve deeper into the methodology used for incorporating graph-based techniques and sentiment analysis into the recommendation process. We will provide a detailed explanation of the *LightGCN* [71] model and its advantages in capturing user-item interactions. Additionally, we will discuss the implementation of sentiment analysis using VADER and its role in understanding user feedback. Through this exploration, we aim to demonstrate the effectiveness of our proposed approach and highlight its potential for improving recommendation systems.

Part II

Conception And Implimentation

Chapter 2

Conception of our Approach

1 Introduction

This chapter is devoted to the modeling of the proposed approach, which covers several of the important phases of the right layout of a POI recommendation system. We will examine how this technique produces effective results, as well as the various modules upon which it relies to achieve our objective.

2 Functionality and Objective

The objective of this project is to analyze and precessing of the data to create an efficient recommender system of point of interest (POI) for users, this recommender is based on user preferences and feedback However, this technique is a means of personalization and a very powerful tool.

Our approach describes a situation in which a user in need of food, sleep, or fun receives suggestions that include a set of restaurants or nightclubs, depending on his point of interest, that meet their needs and are useful and effective. The treatment with the user begins with a request from the suggested preference list or user feedback and ends with the list of ranked POI.

3 General Architecture

The recommendation system provides the possibility of calculating and filtering the data to obtain high-quality results for the user.

This section presents the general scheme of our POI recommendation system. Indeed, the recommended technique is focused on the feedback and preferences of users in order to address the low choice of POI for users.

In order to improve users' lives, we have filled this system with multiple user profiles and POI features, as well as their feedback. The diagram 2.1 below represents the overall architecture of the system, as well as the models that make up it and how it works.

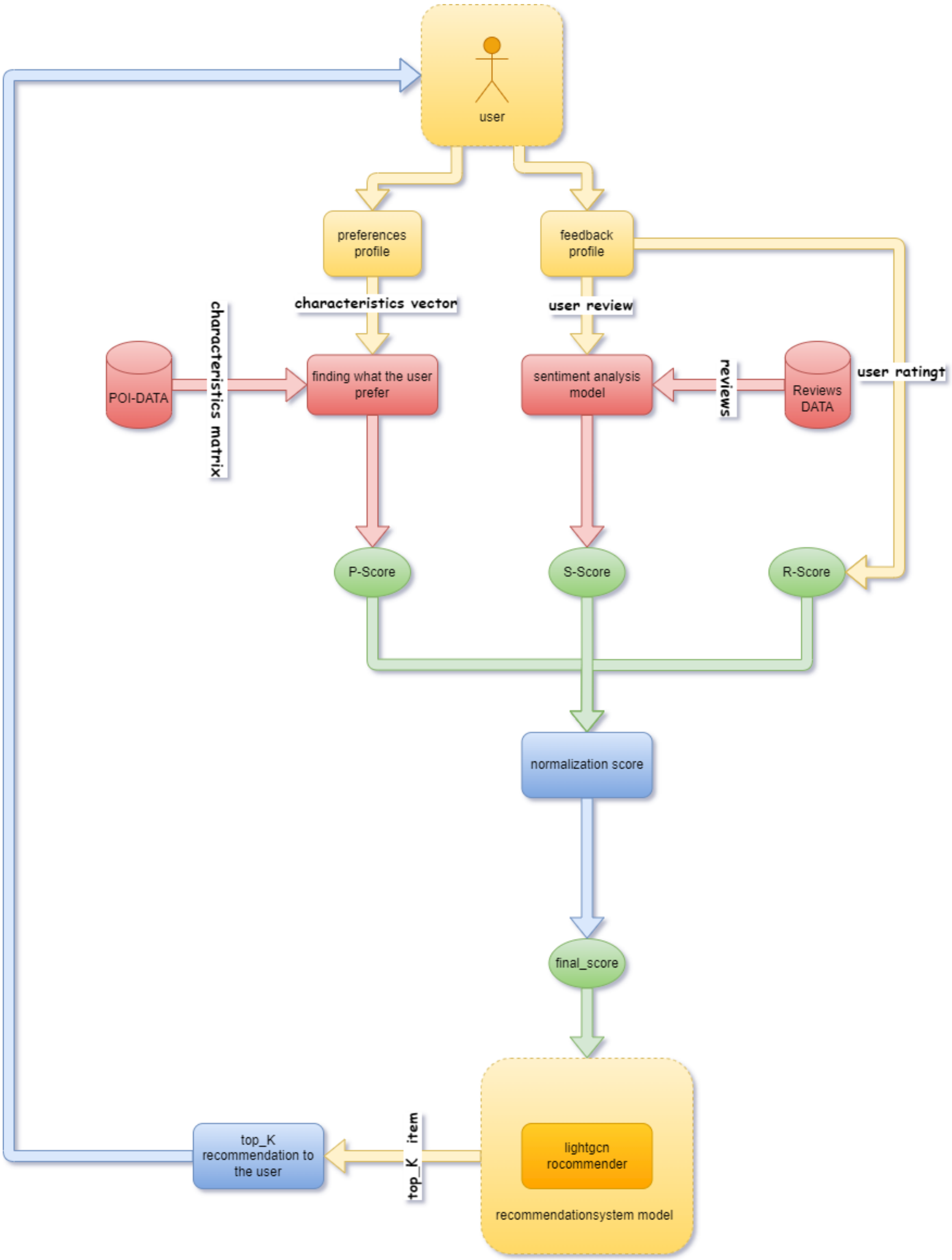


Figure 2.1: General Architecture of the system

Our approach is divided into several components, including:

- **User:** Incorporating diverse demographic and preference data, this component plays a crucial role in delivering personalized recommendations. With the aim of providing individualized suggestions, it diligently maintains user profiles, constantly updating them with the latest information.
- **POI:** determine what characteristics of the POI are inputted into the system and how to filter and recommend based on those characteristics.
- **Sentiment model:** it enables users to share their thoughts about a specific POI in order

to assign it a score between 1 and 5 for use in the recommendation phase.

- **Recommendation model:** Recommendation Model: Serving as the backbone of the system, this module leverages user and point-of-interest data to deliver personalized recommendations, empowering users to achieve optimal outcomes. Section 7 delves into the intricate workings of this module, providing a comprehensive understanding of its internal mechanisms.

In the following sections, we will have a closer look at these units and learn everything there is to know about them.

4 User modelling

The implementation of the POI recommendation method is largely based on user modelling. It consists of describing the information preferences of users using the profile form. User profile modelling is a two-step process [72]. The first step involves data collection strategies for users, and the second step is to formulate and develop the user profile, which can be done in different ways to arrange the acquired data in the desired representative structure.

4.1 Construction of profiles

A good management of the best recommendations allows the results to be adjusted according to the requirements of the users by correctly creating their profiles from their information data. For this reason, there are three basic types of profile representation [73]:

- **Assembly representation:** Typically, the profile is represented as a vector of weighted terms, or vector classes.
- **semantic representation:** The description of the profile highlights the semantic connections between the data it includes. On that representation, probabilistic concepts and semantic networks are generally employed.
- **multidimensional representation:** The profile is organised into several dimensions that are presented in various ways. [73]

The notion of a profile is frequently associated with preferences and context, along with true facts about a person [74]:

1. A user profile is a compilation of information about a particular person.
2. A context is a compilation of information that describes the context of user-system interaction.
3. A preference is a concept or expression that enables us to classify the significance of profile or context-related facts.

Indeed, the user's preferences are a crucial part of his description, and they can alter depending on the situation in which he finds himself. The objective of the construction is to produce a generic model that can be used to characterise a user and their preferences via a set of features or categories that can be modified, enhanced, and instantiated as necessary.

We analysed the profile in multiple dimensions that represent all three elements for feedback data, context, and preferences, which are the primary components in our system's user profile, and utilised the most suitable multidimensional representation to convey information about users.

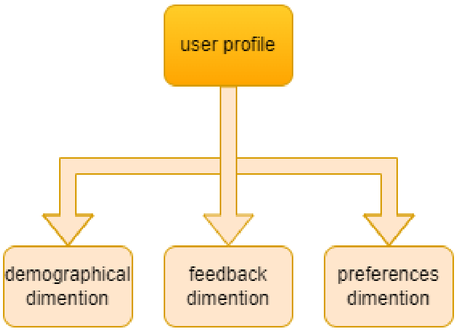


Figure 2.2: user profile dimension

4.1.1 Demographical profile

This dimension represents the element of the user's profile that is established at the time of their initial login by filling out a form that obtains the user's information. It is composed of two types of data: the primary type of which is the user's identifier, and the second type of which includes details that include their name, age, gender, etc. This information belongs to a group of personal data.

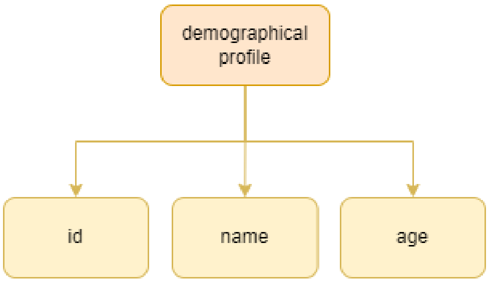


Figure 2.3: demographical profile

4.1.2 Feedback profile

This dimension represents the viewpoint section of the profile, which contains all the necessary information to define the user's perspective. Primarily, this is the spatial element required to convey the user's emotion.

The rating and the reviews are two features that represent user interaction and perspective on one of the POIs. The rating represents the quick and direct manner in which a user left his opinion concerning the point of interest (POI) he went to, while a review represents the extensive significance of the feedback users left.

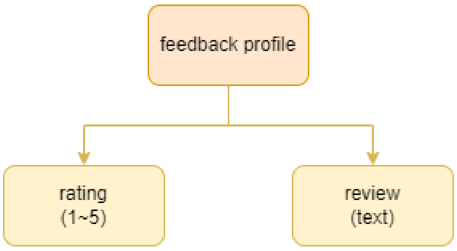


Figure 2.4: feedback profile

4.1.3 Preferences profile

The preference profile can be based on POI criteria that are reviewed at the time of entry into the system. To create a preference profile, the user must correctly specify their preferences. Each element offers a variety of options, such as price ranges, as well as many types of payment, like AcceptCreditCard, Bitcoin, smoking outdoor or yes etc.

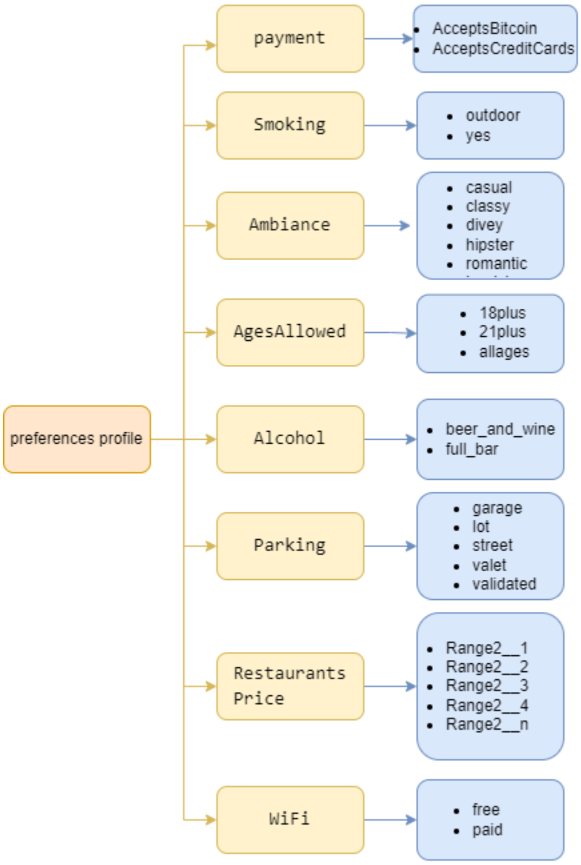


Figure 2.5: preferences profile

5 POI modelling

The POI entity represents the fundamental concept of providing a list of ideas, with each POI illustrating a set of categories (business) that have the same characteristics and which the user may customise to meet his needs and preferences. We examined the requirements or advantages of a complete POI because of the increasing rivalry in the POI business. We were particularly interested in the aspects of the point of interest that impact the customer's impression. In Figure 2.11 we have therefore compiled a list of the most common characteristics

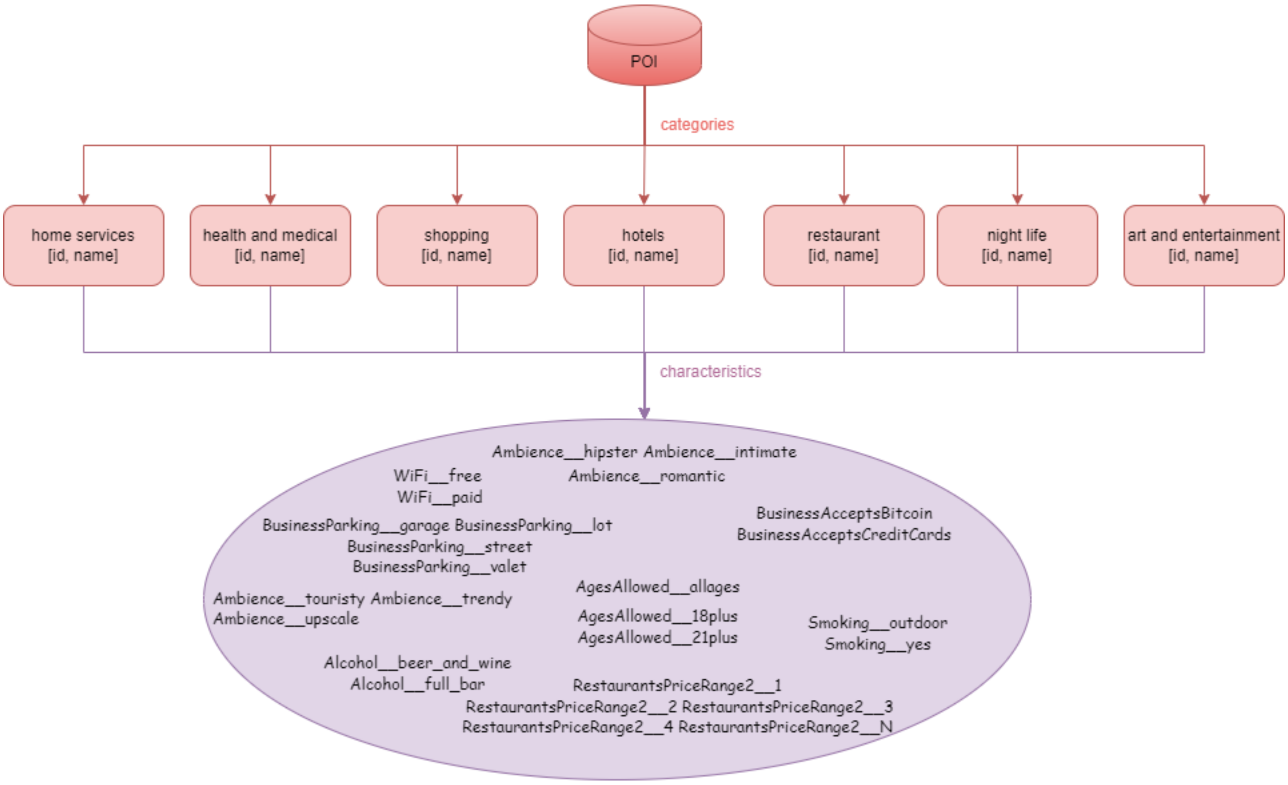


Figure 2.6: POI characteristic

5.1 Characteristics

In the real world, each POI has numerous characteristics; therefore, we are able to deal with those that we discovered in the dataset. Each POI category (business), including hotels, restaurants, and nightlife, might or might not have similar characteristics, like wifi___free or smoking___outdoor. For more comprehension, have a look at figure 2.7, For the same category, the businesses whose ID = 'fdsafjhksafoijsfa_ia' have wifi_free, and the businesses whose ID = 'kjkjkfdshhhsfsaf' have wifi_free and businessacceptBitcoin. Figure 2.8 explains this.

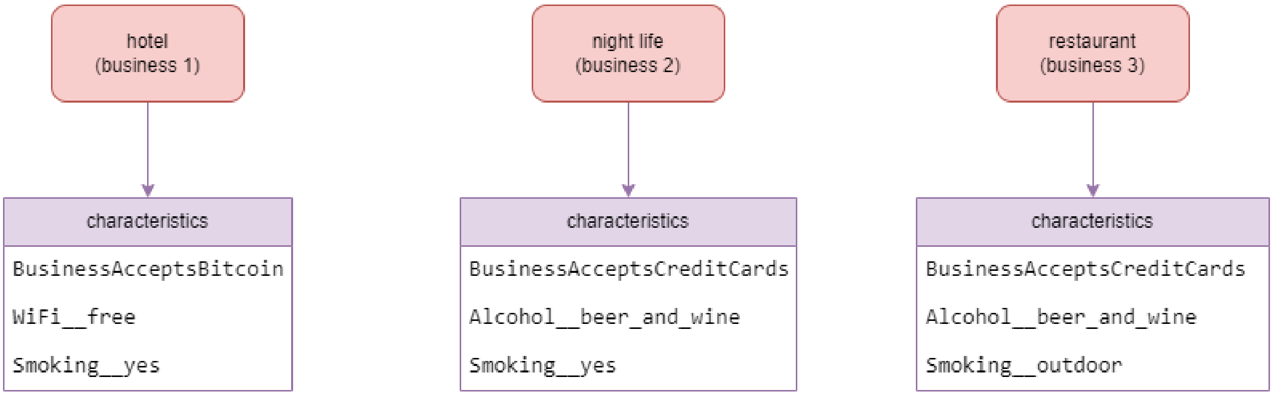


Figure 2.7: POI characteristic comparison

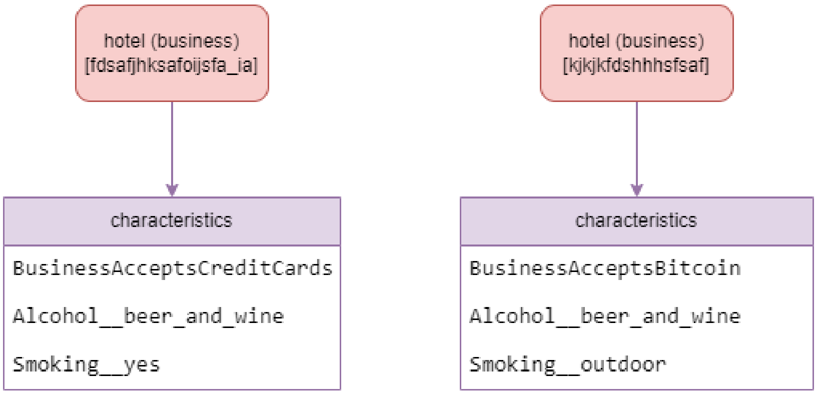


Figure 2.8: Business characteristic comparison

6 Sentiment analysis model

It's strongly associated with (or may be regarded as an element of) the fields of computational linguistics and natural language processing (NLP) as a scientific discipline [75]. To differentiate and classify a text review conveying the user's positive, negative, or neutral behaviour, or to transform it into a score between 1 and 5 for an item, event [76].
Once our data is provided, a two-step sentiment analysis process is performed, including preprocessing, text classification (positive, negative, or neutral), The structure of the process is shown in figure 2.9, After the stage of text preprocessing is complete, the processed text will be delivered to VADER to obtain a sentiment score. We picked VADER since it offers numerous advantages over standard sentiment analysis techniques, including [77] :

- It performs exceptionally well with social media text, but can be readily generalised to other domains.
- It is constructed from a generalizable, valence-based, human-edited, highest-quality sentiment lexicon and requires no training data.

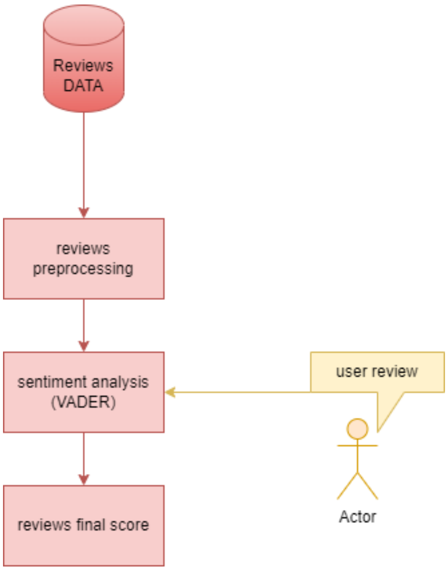


Figure 2.9: Sentiment analysis model

6.1 Review preprocessing

During the preparation stage, it is essential to ensure that the reviews data is clean and free from any unwanted elements. This can be achieved through the use of various cleaning methods.

1. **convert text to lowerCase:** Lowercasing text is a crucial step in text preprocessing that serves multiple purposes. It simplifies the text and facilitates its processing, as different variations of the same word are treated as identical. Additionally, some natural language processing models and algorithms perform better when the text is in lowercase. In review preprocessing, lowercasing text helps standardize it and makes it easier to compare different reviews. It also diminishes the impact of capitalization on sentiment analysis, allowing the algorithm to focus more on the words and their meanings.

2. **remove punctuation double-spacing and numbers:** This process is essential to improve the quality and readability of text data. It involves removing punctuation, double-spacing, and numbers. Punctuation marks, such as commas, periods, and question marks, provide structure to sentences, but are often irrelevant when analyzing the meaning of text. Double-spacing occurs when there are unnecessary gaps between words or lines, which can negatively impact subsequent text analysis tasks. Numbers, especially when analyzing textual data, can be considered noise or irrelevant information. Removing numbers allows us to concentrate on the textual context itself, enabling more accurate analysis and extraction of meaningful patterns. [78].

3. **tokenization:** is the process of breaking down a text or a sequence of characters into smaller units, called tokens. These tokens can be words, sentences, or even smaller subword units, depending on the level of granularity required for analysis.The goal of tokenization is to segment the text into meaningful units that can be easily processed and analyzed. By dividing the text into tokens, we can perform various operations such as counting words, analyzing word frequencies, identifying sentence boundaries, and applying further linguistic or statistical analysis [78].

4. **remove stop words:** Stop words are frequently used words that do not contribute much meaning to the text, such as “the”, “and”, “a”, “in”, etc. Removing these words can reduce the dimensionality of the data and enhance the efficiency of subsequent text analysis tasks like sentiment analysis or topic modeling. Stop words generally do not aid in determining the sentiment or topics of the text, so removing them can improve the accuracy and relevance of the analysis results [78].

5. **Stemming:** In review preprocessing, stemming is techniques used to simplify words to their base form, ultimately reducing the complexity of vocabulary and improving the efficiency of analysis. Stemming involves stripping suffixes from words to obtain their root form or stem, such as "jump" from "jumping." This technique groups together words with similar meanings and helps to reduce the number of unique words in a text [79].

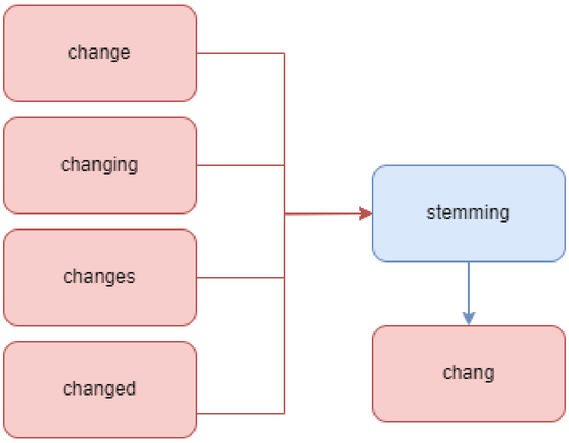


Figure 2.10: Stemming process [80]

6. **lemmatizing:** Lemmatization reduces words to their dictionary form or lemma, such as "chang" for "change, changes, changing." This ensures that different inflected forms of a word are treated as the same word, improving the accuracy of the analysis [81].

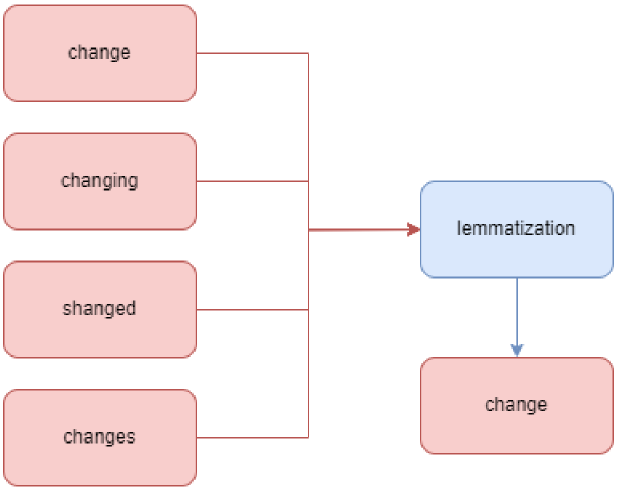


Figure 2.11: Lemmatization process [80]

7. **Joining the tokens back into a single string:** The reason for joining the tokens back into a single string in the review preprocessing is that subsequent analysis typically requires the input text to be in the form of a single string, rather than a list of individual tokens. This step guarantees that the preprocessed review is in the correct format to be utilized in further analysis, such as sentiment analysis or topic modeling.

6.2 Sentiment analysis

After the review (text) finishes the stage of preprocessing, it is going to be delivered through our sentiment analysis model (VADER6.2), which will analyse it and provide a score for each review (text), for example, “The cuisine is very good! But the service is dreadful” after preprocessing; it becomes “food really good service dread”. This text will be handled by our sentiment analysis model to provide us with a sentiment score for this text, Vader return if the text is positive, negative, or neutral, and finally the compound, which is the sum of the lexicon rating. The figure 2.12 below illustrates an example of this:

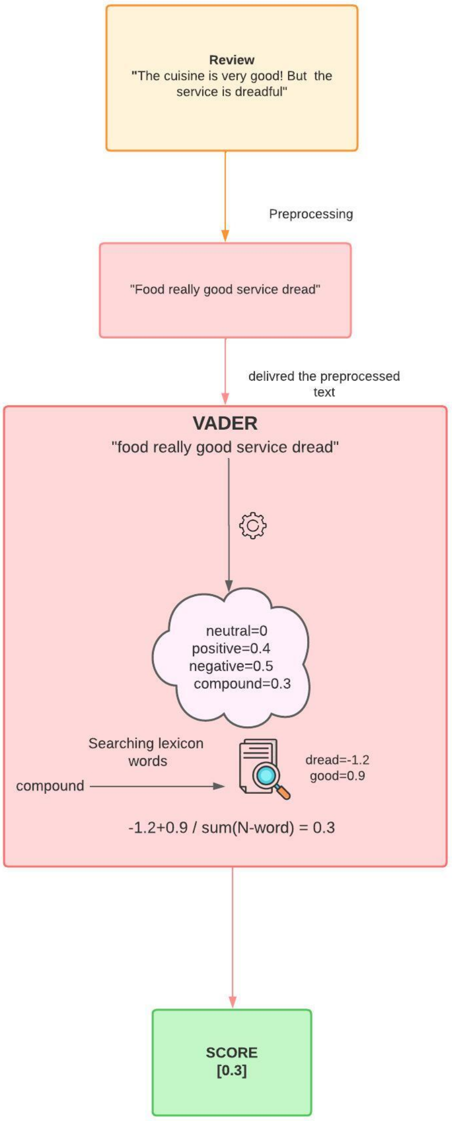

Review
"The cuisine is very good! But the service is dreadful"
Preprocessing
"Food really good service dread"
delivred the preprocessed text
VADER
"food really good service dread"
neutral=0
positive=0.4
negative=0.5
compound=0.3
Searching lexicon words
compound
dread=-1.2
good=0.9
-1.2+0.9 / sum(N-word) = 0.3
SCORE
[0.3]

Figure 2.12: VADER example

When the score of sentiment analysis model is provided, we normalize the score to be between 0 and 5 the form used to normalize this score is the min-max equation.

$$x_{scaled} = a + \frac{(x - x_{min})(b - a)}{x_{max} - x_{min}} \tag{2.1}$$

- Where a and b are the minimum and the maximum values of the new range, respectively.
- x_{min} and x_{max} are the minimum and the maximum value of the original data, respectively.
- x is the original data value

7 Recommender model

The fundamental component of our system is the recommendation unit, which generates the outcome of the recommendation. We showcase how we used collaborative and content-based filtering techniques to create a highly accurate model based on the section's 7 ideas and research in chapter 1. In order to present a greater diversity of ideas, we prefer the mixed filtering method in our filtering strategy. This hybrid referral system uses both collaborative and content-based filtering techniques.As was already established in Section 2.3.3 of Chapter 1, the idea is straightforward; for combining both techniques, a hybrid approach is used.
However, we must first establish how users and IOPs will be represented in the system and how they interact with each other before we can begin developing the recommendation engine.

7.1 Representation of the users and POI

Vectors with n dimensions that have an attribute for each element represent users and POIs. A single element within the characteristic vector expresses every Boolean characteristic.

7.1.1 Point of interest POI

Each POI is represented by a vector C of n characteristics indicating its quality $n = 94$, Each of the characteristics c in the victor C is a boolean with 1 or 0 value which mean the existence of this characteristic in the POI or not if $ci = 1$ this mean the characteristic exist in the POI and the vice versa.

$$C = [c_1, c_2, ..., c_n] \tag{2.2}$$

As illustrated in Figure 2.13, a matrix can be created by combining the current features and POI. In the matrix, POI are represented by rows, and characteristics are represented by columns. $(pi, c_j) = 1$ if the $6th$ POI contains the same characteristic. It is crucial to observe that each POI may offer one or more characteristics.

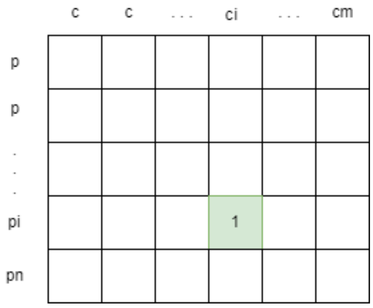


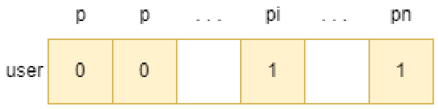
Figure 2.13: POI characteristic matrix representation

7.1.2 User

A user's vector P includes details about their preferences p of range $n = 94$, as well as two sets of 0 and 1 values demonstrating their preference for particular characteristics if $pi = 1$ this mean the user prefer this characteristic.

$$P = [p_1, p_2, ..., p_n] \tag{2.3}$$

The characteristics of the POI were explored in the previous section, These characteristics can be used to represent the vector of a user's preferences because they can demonstrate the user's preferences. When a user first uses the suggested system, they are invited to select the initial value of their preference vector according to the distinctive characteristics of the POI; when the user did not select it a value of 0 denotes an unfavourable opinion towards the characteristic, while if the user selects it a value of 1 denotes a favourable opinion, Fig 2.14 shows the user preferences vector.

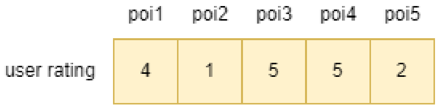


	p	p	. . .	pi	. . .	pn
user	0	0		1		1

Figure 2.14: User preferences vector representation

The user also have another important dimension the feedback which contain the rating as we explained in section 4.1 and in section 4.1.2, this play a big role when it comes to know what the user prefers this can be represented in a vector W and each element in this vector represent user rating (0 to 5) for a particular POI he visited this vector size $n = number\ of\ visited\ POI$, Fig 2.15 shows the representation of this vector.

$$W = [r_1, r_2, ..., r_n] \tag{2.4}$$



	poi1	poi2	poi3	poi4	poi5
user rating	4	1	5	5	2

Figure 2.15: User rating vector

Zeng et al [21], they add the restaurant's features to a user when he/she, visited a restaurant corresponding to he/she, preference features and normalizing them, Fig 2.16 represent the user preference model of Zeng et al.

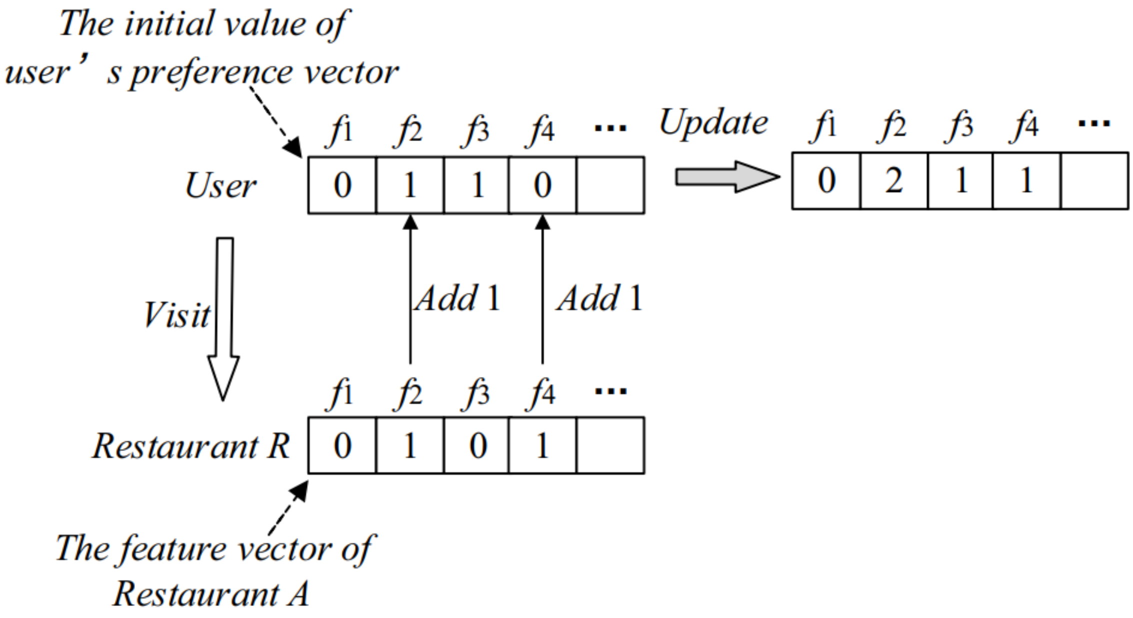


Figure 2.16: user preferences model of Zeng et al[21]

This may not be accurate, as a user's visit to a restaurant does not always mean that the user likes that restaurant; therefore, adding directly the features of the restaurant the user visited is not precise; even if a user visits the same restaurant twice, this does not necessarily indicate that the user likes that restaurant when viewed from a semantic perspective.
Inspired by what [21] did to represent the user preference model and trying to make it more accurate, we used the user's rating for the POI they visited as a weight of preference which mean every POI user has visited will take from the feedback the rating and multiplicated it by all the characteristics of the visited POI and then normalized it by divided it with the sum of the rating and if the result is grater than or equal to 0.5 put 1 else put 0 this is the round process of the result, equation 2.5 represent this process, Figure 2.17 shows the process between user and one of visited POI and Figure 2.18 shows the user preference model.

$$\sum_{j=0}^{m} P[j] = round(\sum_{i=0}^{n}(W_i \cdot C_{ij})\frac{1}{\sum_{j=0}^{n} W_j}) \tag{2.5}$$

The vector P will contain 1 and 0 values, representing the user's preference. For example $P = [1,0,1,1]$, after multiplying with the wights with POI matrix the vector P it will be $P = [5,0,3,2]$ After normalisation, these results will be $P = [0.8,0,0.4,0.1]$, and after rounding, the end result will be $P = [1,0,0,0]$.

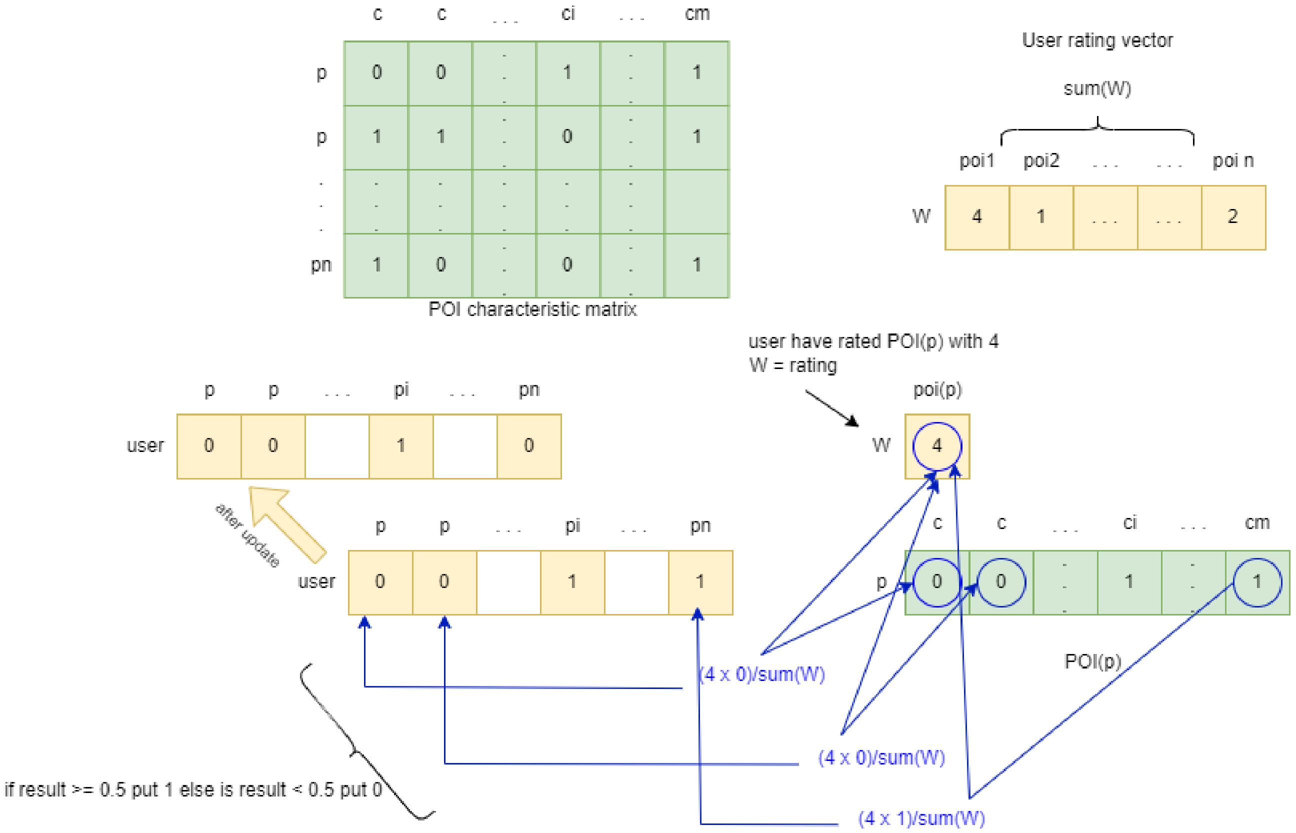


Figure 2.17: User model with one of visited POI example

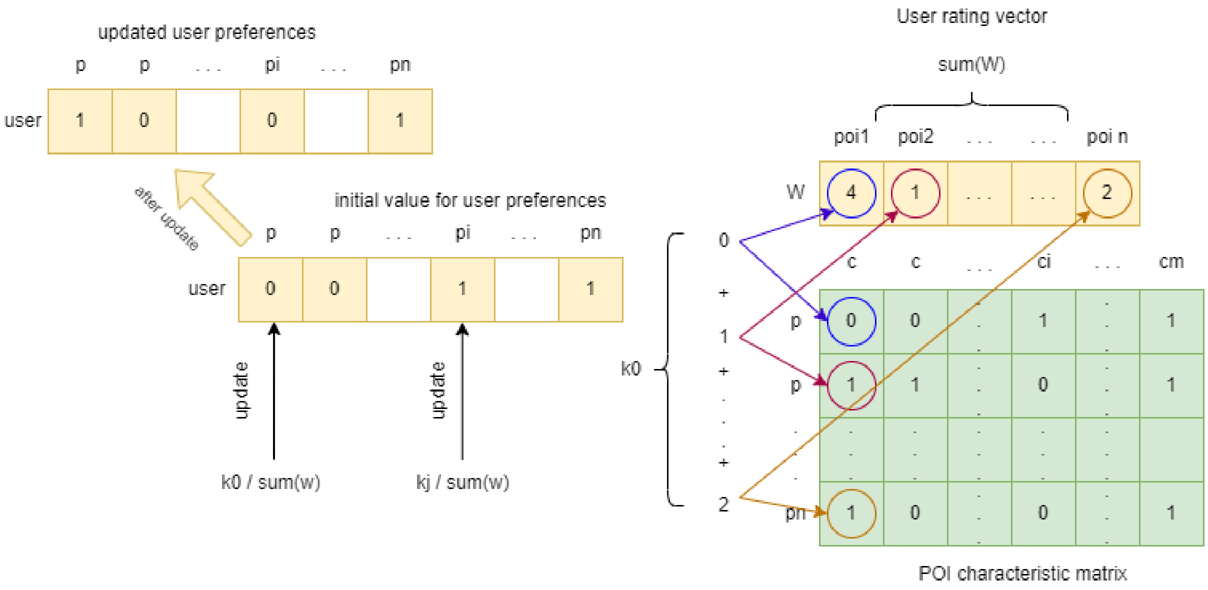


Figure 2.18: User preferences model representation

7.2 Recommendation factors

The proposed system of recommendations employs statistical techniques and experimental data analysis, and takes three factors into account.

- Score of sentiment analysis model.

- Score of content based, which determine the degree of similarity between the user and the POI.
- Score of the user feedback, the rating which determine the user visited POI and the degree of how much the user enjoin the visited POI.

This section discusses the whole POI recommendation strategy.

7.2.1 Sentiment analysis - reviews score

Each POI have a various feedback from the visitors and users in the real word, because they allow interaction between users and POI. Therefore, in contextual systems, whose main purpose is to recommend POI appropriately, POI reviews score are essential.
We used the sentiment analysis method as mentioned in section 6 when the user leaves a feedback review of the visited POI this review will be passed in the sentiment analysis method to calculate the sentiment score as mentioned in Fig 2.9 This method allows us to determine if the rating a user has left for a visited POI is accurate, because what the user written about his opinion is more accurate than a rating what if a user by accident put three stars rating for a visited POI, and he really enjoined it but when the user left a review he wrote his opinion about the visited POI it will be more accurate than just a simple rating end this is the power of the sentiment analysis in our approach, see the Fig 2.19. The Sentiment Analysis Step Score is required for the development of the recommendation algorithm and the extraction of a strong recommendation result.

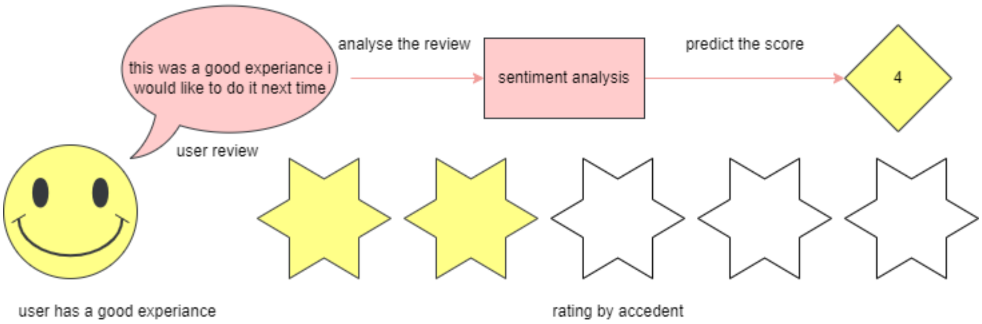

this was a good experiance i would like to do it next time
analyse the review
sentiment analysis
predict the score
4
user review
user has a good experiance
rating by accedent

Figure 2.19: Sentiment analysis example

7.2.2 Content based - similarity score

The matched recommendation ($item/user$) is determined by the content that identifies the POI that are most similar to the user's profile. For instance, a user who has rated multiple POI who accept credit cards positively will have a profile that resembles to accept credit cards. To develop a technique for combining a user U, whose vector of preferences $Pi = [p1, p2, ..., pn]$, and a POI P, whose vector of characteristics $Ci = [c1, c2, ..., cn]$, in order to receive POI recommendations from users, we must calculate the similarity to determine the degree of similarity between both vectors to ensure they can offer POI based on the user's preferences.
To determine which POI the user prefers, we must discover a way to calculate the user's and the POI similarity. In Section 4.1 of Chapter 1, we discussed many similarity models that could

be used to determine this similarity and chose to use a cosine similarity metric for this system, 2.6.

$$Sim(U,P) = \frac{\sum_{i=1}^{n} P_i C_i}{\sqrt{\sum_{i=1}^{n} P_i^2} \times \sqrt{\sum_{i=1}^{n} C_i^2}} \tag{2.6}$$

The similarity function always returns a value between 0 and 1. If the value is close to 1, then the POI and the user are very similar; otherwise, they are not similar. This value will be normalized to be between 0 and 5 by the equation min-max 2.1.

7.3 Recommender Algorithm

In this section, we define the concepts that permit us to filter using two algorithms:

- The first method is an equation of normalization score, which calculates the final score by using content-based filtering, which returns a similarity score between the user and the POI, and the sentiment analysis model, which returns a sentiment score, and finally the rating.
- The second is a deep learning model, *LightGCN* designed for collaborative filtering tasks.

7.3.1 Final score

When a user is connected to the system, a final score is generated for each user based on three entries.

- the review that belongs to the feedback of the user who visited POI.
- the rating which also belong to the feedback of the user to visited POI
- the vector of preferences of a user and the vector of characteristics of a POI.

In order to compute the final score, the equation makes use of the sentiment analysis result, which takes a review from the user and returns the sentiment score, the similarity score between the user and the POI, and finally the review given by the user to the POI. the final score computed by the equation 2.7.

$$F\_Score = \alpha P\_Score + \beta S\_Score + \lambda R\_Score \tag{2.7}$$

In equation 2.7 the alpha beta lambda are wights, each one started with a value of 0.333, this value can be considered as a normalization value beside of wight to keep the final score in range of 0 to 5, the total sum of them is equal to one.
The final score of each POI will be utilised in the subsequent stage, which discusses the collaborative filtering technique.

7.3.2 Collaborative filtering - *lightGCN*

Collaborative filtering is a popular technique used in recommendation systems to provide personalized recommendations to users. It is based on the idea that people with similar preferences or behaviours in the past are likely to have similar preferences in the future. Collaborative Filtering utilises the collective wisdom of users to make recommendations by identifying patterns or similarities among their interactions with items.
In our approach, we decided to use *lightGCN* rather than traditional collaborative filtering methods because, as is known, deep learning performs well when it comes to dealing with big data.

7.3.2.1 lightGCN *LightGCN* is a state-of-the-art model in collaborative filtering, specifically in the field of graph-based recommendation systems. It is designed to address the limitations of traditional Graph Neural Networks (GNNs) by simplifying the model architecture and improving scalability and efficiency [71] Figure 2.23 illustrate the architecture of *lightGCN*.
LightGCN leverages the collaborative filtering principle and represents the user-item interaction data as a bipartite graph. It utilizes a weighted sum aggregator2.8 to learn user and item embeddings directly from the user-item graph structure. The key idea is to propagate the user and item embeddings by aggregating the embeddings of their neighboring nodes (users/items) in the graph, e.g. in Figure 2.20 2.21. At layer combination, instead of taking the embedding of the final layer, *LightGCN* computes a weighted sum 2.9 of the embeddings at different layers example in figure 2.22, Finally, *LightGCN* predicts based on the inner product 2.10 of the final user and item (POI) embeddings[71].

$$\begin{aligned} e_u^{(k+1)} &= \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|Ni|}} e_i^k \\ e_i^{(k+1)} &= \sum_{i \in N_i} \frac{1}{\sqrt{|N_i|}\sqrt{|Nu|}} e_u^k \end{aligned} \tag{2.8}$$

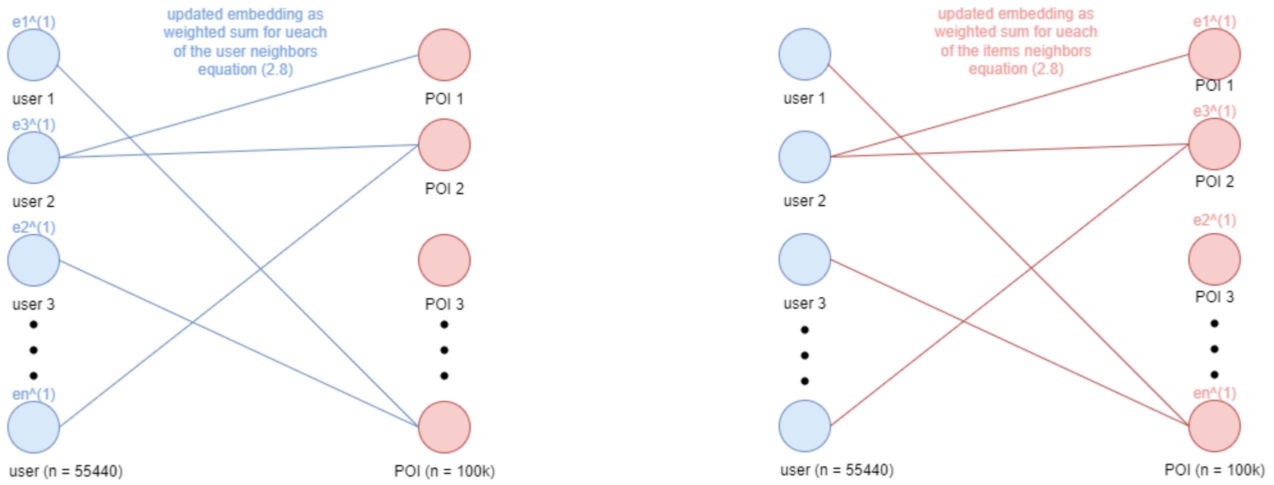
where $e_u^{(k)}$ and $e_i^{(k)}$ are the user and item (POI) node embeddings at the k-th layer. $|N_u|$ and $|N_i|$ are the user and item nodes' number of neighbors [71].

$$\begin{aligned} e_u &= \sum_{k=0}^{k} \alpha_k e_u^{(k)} \\ e_i &= \sum_{k=0}^{k} \alpha_k e_i^{(k)} \end{aligned} \tag{2.9}$$

with $\alpha_k \geq 0$. Here, alpha values can either be learned as network parameters, or set as empirical hyperparameters. It has been found that $\alpha = 1/(K+1)$ works well, α_k haven't a special component to optimize to avoid complicating the *lightGCN* [71].

$$\hat{y}_{ui} = e_u^T e_i \tag{2.10}$$

This inner product measures the similarity between the user and POI, therefore allowing us to understand how likely it is for the user to like the POI.



(a) aggregation users process example

(b) aggregation item process example

Figure 2.20: (user/item)aggregation process example

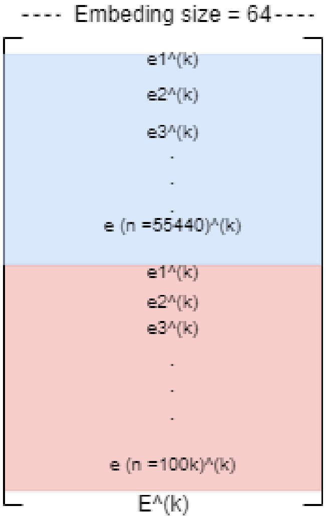


Figure 2.21: example of the embedding result

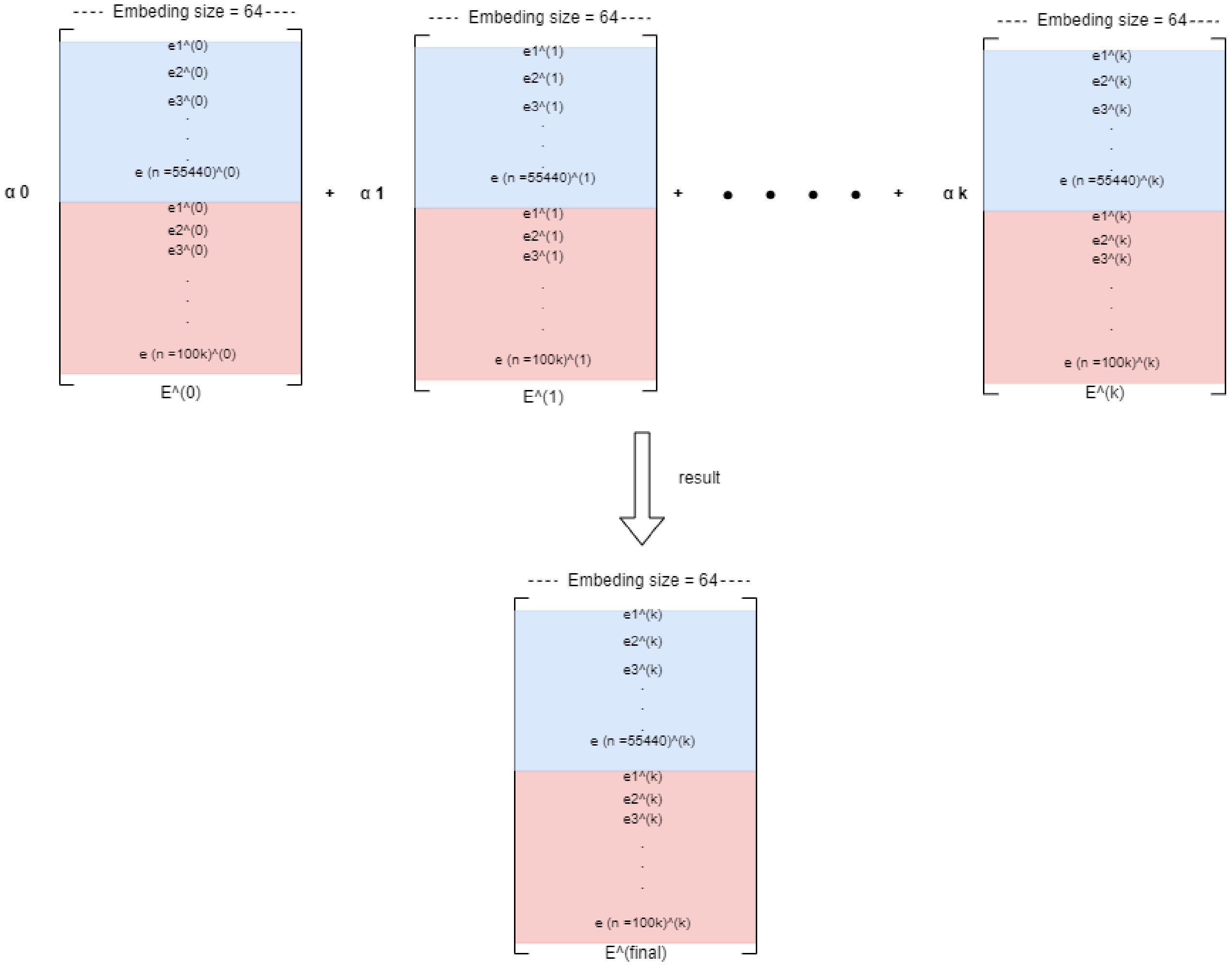


Figure 2.22: example of layer combination

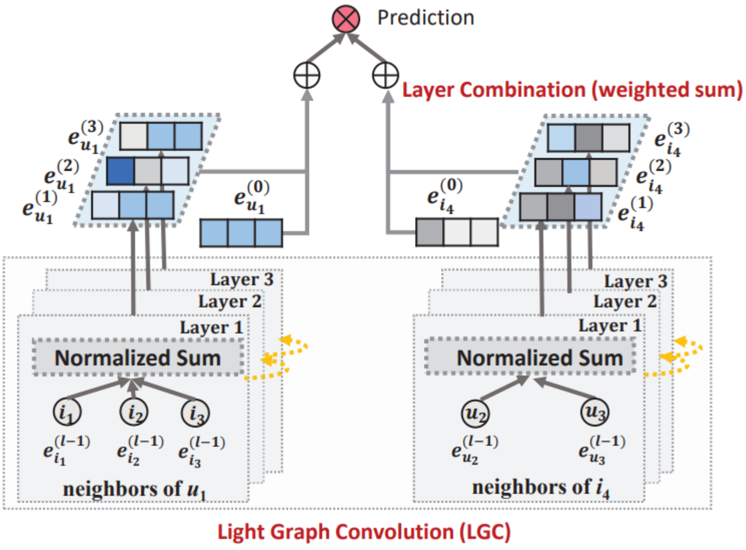


Figure 2.23: Illustration of *lightGCN* architecture [71]

7.3.3 Recommendation results

The recommendation system is an effective tool for providing personalised and beneficial information to users. This system employs a user preference model based on the characteristics of the visited POI, as well as feedback information about the visited POI, to generate dynamically recommended results. We used two methods of filtering to identify successful outcomes for the target user requesting POI recommendations: content-based filtering, which uses similarity as the primary factor for calculating the final score, and collaborative filtering, which uses the $lightGCN$ model to provide a more precise final recommendation result.
Figure 2.24 represent the final step of recommendation system and Table 2.1 presents a simple example of what will be displayed to the target user and the final result of a Top_k recommendation list for evaluating the functionality of the recommendation system proposed in this section, POI 1 POI 2 ..., are the list of the final recommendation POI, which are the most similar to user and are the best final results.

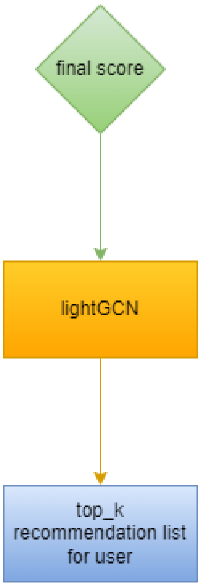


Figure 2.24: final stem of recommendation system

POI	**rating**
POI 1	7
POI 2	5
POI 3	2
POI 4	1

Table 2.1: Example of $lightGCN$ recommendation of top_k POI using the final score results

7.3.4 Recommendation Algorithm pseudocode

This section provides a summary of our recommendation system through a pseudocode that examines two states: when a user goes through the system for the first time and when it's currently present in the system. This code provides all the methods necessary for the calculation of the above-mentioned suggestion and explains completely the steps of this phase, where he receives as input the list of POI L_POI and the list of Lu users who currently exist in the system in order to calculate the final score of each POI and apply collaborative filtering with the *lightGCN* model to these final scores.

Algorithme 1 General Algorithm of the Recommendation system

```
Data: $Lu(list\ of\ users)$, $LPOI(list\ of\ POI)$, $Urv(user\ review)$
Result: $Top\_K(list\ of\ top\ k\ POI\ recommended)$
$\alpha, \beta, \lambda = 0.33$
for $P \in LPOI$ do
    if $U \notin Lu$ then
                                                              ▷ new user
        $P\_score(U, P)$                                      ▷ content based
        $F\_score(U, P) = P\_score(U, P))$                    ▷ final score
        $Top\_k \leftarrow lightGCN(F\_score(U, P)$            ▷ collaborative filtering
        $ReturnTop\_k$                                        ▷ recommendation list
    else
                                                              ▷ if user exist in the system
        $U = update\ U$                                       ▷ update the user
        $P\_score(U, P)$                                      ▷ content based
        $S\_score(U, P)$                                      ▷ sentiment analysis
        $R\_score(U, P)$                                      ▷ rating score
        $F\_score(U, P) = \alpha P\_score(U, P) + \beta S\_score(U, P) + \lambda R\_score(U, P)$   ▷ final score
        $Top\_k \leftarrow lightGCN(U, F\_score(U, P))$       ▷ collaborative filtering
        $ReturnTop\_k$                                        ▷ recommendation list

        if $U\ left\ a\ review\ feedback\ for\ P$ then
            $Urv \leftarrow PreProcessing(Urv)$               ▷ preprocessing
            $S\_score(U, P) \leftarrow VADER(U, P, Urv)$      ▷ sentiment analysis
            $F\_score(U, P) = \alpha P\_score(U, P) + \beta S\_score(U, P) + \lambda R\_score(U, P)$   ▷ final score
            $Top\_k \leftarrow lightGCN(F\_score(U, P))$      ▷ collaborative filtering
            $ReturnTop\_k$                                    ▷ recommendation list
        end
    end
end
```

8 Conclusion

In this chapter, we describe the stages of development of a POI recommender system. These stages include modelling user and POI profiles based on contextual and feedback data, as well as the sentiment analysis process that allows evaluating the user's opinion and converting it into a real note.

We were able to explain the primary phase of the proposition and recommendation based on the two filtering techniques and the overall system performance, utilising the user and POI modelling phases and the sentiment analysis phase, as well as explaining their relationships and processes.

Following this chapter on conception and modelling, we will discuss the experiments and the system's implementation, as well as the attained results.

Part III

Experiments and Results

Chapter 3

Experiments and results of proposed approach

1 Introduction

After completing the design and formalisation phases of our approach. In this section, we present a comprehensive overview of the experiments conducted to evaluate the performance and effectiveness of our proposed approach. The experiments were designed to assess various aspects of our system, including the development tools used, the dataset employed, data preprocessing and filtering techniques, sentiment analysis experiments, recommender system experiments, and a scenario example. Each of these components contributes to a thorough understanding of the capabilities and potential applications of our system.

2 Presentation of development tools

Over the development and setup phases of our system, we apply a variety of tools that assist us in creating an appropriate approach. In the following sections, we will examine these instruments in depth.

2.1 Equipment

The implementation was conducted on a PC that had an I5 processor, 16 GB of RAM, along with Windows 11; however, the system is functional on any computer with online interactive programming environments such as jupyter and Google collaboratory.

2.2 Work environment

- **Jupyter:** Free software, open standards, and web services for all programming languages' collaborative computation. The Jupyter Notebook is the first web application designed to create and share computational documents. It provides a straightforward, streamlined, document-focused experience [82].

2.3 Programming language

- **Python:** Python is an object-oriented, high-level programming language that is interpretable and has flexible semantics. Its built-in data structures, dynamic encoding, and binding make it a desirable language for fast application development and scripting. Its straightforward, easy-to-learn syntax emphasises clarity and saves on maintenance expenses. It offers modules and packages, thereby promoting programme modularity and code reuse. The Python interpreter and common library are accessible in source or binary form and may be distributed without restriction [83].

2.4 Library

- **Numpy:** NumPy is a widely used Python library used primarily for quantitative and scientific computations. It offers numerous tools and functions that can be beneficial for data science applications. An essential step in a data science training endeavour is familiarising oneself with NumPy [84].

- **Pandas:** Pandas is an open-source library designed for handling tabular or labelled data in a straightforward and easy manner. It offers numerous data structures as well as methods for handling numerical and time-series data. This library is a NumPy extension. Pandas is quick and offers superior efficiency and effectiveness for its consumers [85].

- **Matplotlib:** Matplotlib is a plotting library employed in the Python programming language to serve two-dimensional visuals. It is compatible with Python programmes, the shell, web-based application servers, and graphical user interface toolkits [86].

- **Seaborn:** Seaborn is a Matplotlib-based Python data visualisation library. It provides a sophisticated interface for creating visually appealing and informative statistical graphics. It provides a significantly more appealing interface compared to Matplotlib. While Matplotlib is simple to use and has its benefits, it is not without its drawbacks [87].

- **NLTK:** Is a collection of Python-written libraries and programmes for symbolic and statistical natural language processing (NLP) for English [88].

- **Gensim:** Gensim is a Python library for open-source natural language processing (NLP) that facilitates topic modelling. It provides many features and algorithms for preprocessing textual data prior to analysis, such as stopword elimination, lemmatization, case normalisation, and frequent word extraction [89].

- **Sklearn:** Scikit-learn is a Python library for machine learning algorithm development. It also includes data preprocessing stages, data resampling strategies, evaluation parameters, and search interfaces for adjusting and optimising algorithm performance, which are all essential components of the machine learning pipeline [90].

- **Tensorflow:** Transformers revolves around pre-trained transformer models. These transformer models appear in various forms, dimensions, and designs, and each has its own method for accepting input data: tokenization. A configuration class, a tokenizer class, and a model class form the foundation of the library [91].

- **Torch:** It is a Python-based module that replaces Numpy and offers flexibility as a deep learning (DL) development platform [92].

3 Dataset

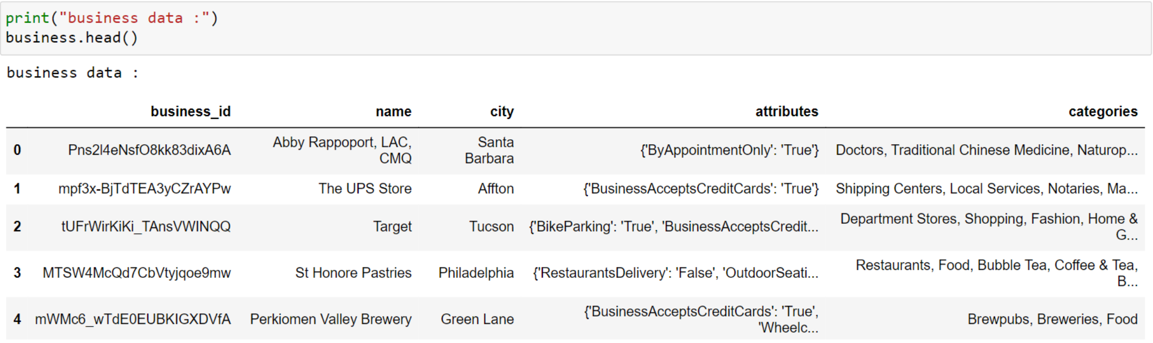
The Yelp dataset [93], one of the largest and most comprehensive datasets in the field, was utilized as the primary source of data for this study. With over 6 *million reviews* and associated metadata, encompassing more than $100,000$ *businesses* and over 1 *million* user, the Yelp dataset provides a rich and extensive collection of customer opinions and interactions with businesses. after the preprocessing and filtering, the data we collect about 55440 *user* and 2193563 *review* and 120430 *business* and this data from the yelp data set is used in our work.

3.1 Data preprocessing and filtering

Since we have a heterogeneous data-set, we need to filter and preprocessing it to clear it from duplicate and missing value and make it clear to use.
In this stage, we'll take a look of Yelp data-set, and we'll go into different process to clean the data.
First, we need to understand the content of the data-set, here we have data contains business data and reviews data and user data each user have a related business and related reviews for each business the user have interaction with, let's start by the business data next we look at the reviews data and finally the users' data.

3.1.1 business data

This data shape is (150346, 5) it contains over $100k$ business (rows) and 5 columns $['business\_id'$ $,' name' ,' city' ,' attributes' ,' categories']$ (we select only the columns we need for this work) the Figure 3.1 shows what the business data contains.



```
print("business data :")
business.head()
```

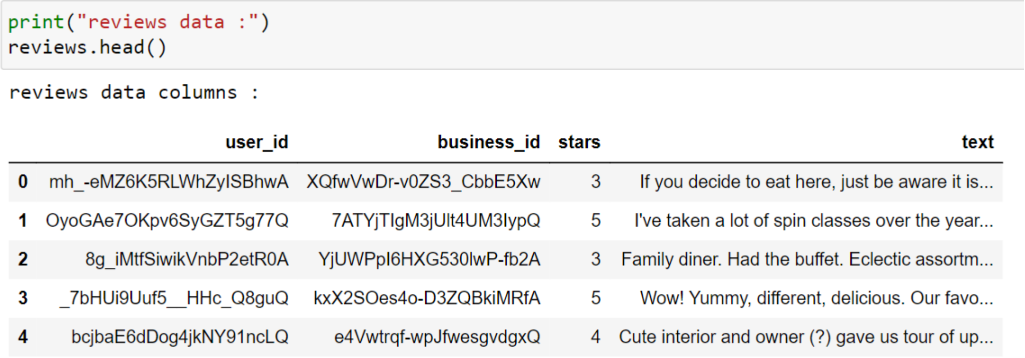
```
business data :
```

	business_id	name	city	attributes	categories
0	Pns2l4eNsfO8kk83dixA6A	Abby Rappoport, LAC, CMQ	Santa Barbara	{'ByAppointmentOnly': 'True'}	Doctors, Traditional Chinese Medicine, Naturop...
1	mpf3x-BjTdTEA3yCZrAYPw	The UPS Store	Affton	{'BusinessAcceptsCreditCards': 'True'}	Shipping Centers, Local Services, Notaries, Ma...
2	tUFrWirKiKi_TAnsVWINQQ	Target	Tucson	{'BikeParking': 'True', 'BusinessAcceptsCredit...	Department Stores, Shopping, Fashion, Home & G...
3	MTSW4McQd7CbVtyjqoe9mw	St Honore Pastries	Philadelphia	{'RestaurantsDelivery': 'False', 'OutdoorSeati...	Restaurants, Food, Bubble Tea, Coffee & Tea, B...
4	mWMc6_wTdE0EUBKIGXDVfA	Perkiomen Valley Brewery	Green Lane	{'BusinessAcceptsCreditCards': 'True', 'Wheelc...	Brewpubs, Breweries, Food

Figure 3.1: screenshots of the business data

3.1.2 reviews data

This data shape is (6990280, 4) it contains over 6 *millions* review (rows) and 4 columns [$'user\_id', 'business\_id', 'stars', 'text'$] (the selected columns we need for this work) the Figure 3.2 shows what the reviews data contains.



```
print("reviews data :")
reviews.head()
```

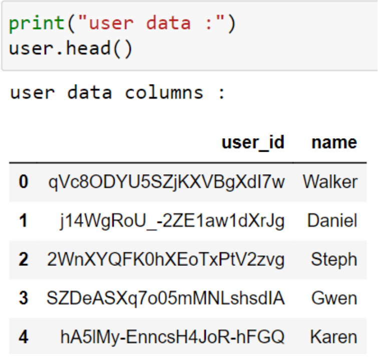
```
reviews data columns :
```

	user_id	business_id	stars	text
0	mh_-eMZ6K5RLWhZyISBhwA	XQfwVwDr-v0ZS3_CbbE5Xw	3	If you decide to eat here, just be aware it is...
1	OyoGAe7OKpv6SyGZT5g77Q	7ATYjTIgM3jUlt4UM3IypQ	5	I've taken a lot of spin classes over the year...
2	8g_iMtfSiwikVnbP2etR0A	YjUWPpI6HXG530lwP-fb2A	3	Family diner. Had the buffet. Eclectic assortm...
3	_7bHUi9Uuf5__HHc_Q8guQ	kxX2SOes4o-D3ZQBkiMRfA	5	Wow! Yummy, different, delicious. Our favo...
4	bcjbaE6dDog4jkNY91ncLQ	e4Vwtrqf-wpJfwesgvdgxQ	4	Cute interior and owner (?) gave us tour of up...

Figure 3.2: screenshots of the reviews data

3.1.3 users' data

This data shape is (1987897, 2) it contains over 1 *millions* user and 2 columns [$'user\_id', 'name'$] (selected columns we need for this work) the Figure 3.3 shows what the reviews data contains.



```
print("user data :")
user.head()
```

```
user data columns :
```

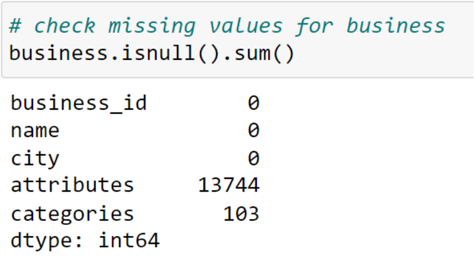
	user_id	name
0	qVc8ODYU5SZjKXVBgXdI7w	Walker
1	j14WgRoU_-2ZE1aw1dXrJg	Daniel
2	2WnXYQFK0hXEoTxPtV2zvg	Steph
3	SZDeASXq7o05mMNLshsdIA	Gwen
4	hA5lMy-EnncsH4JoR-hFGQ	Karen

Figure 3.3: screenshots of users' data

3.1.4 Data cleaning

To make sure that our data is clean of messing values and duplicate values and to prepare our data, we start with the business data, and next we check for reviews data and finally the users' data:

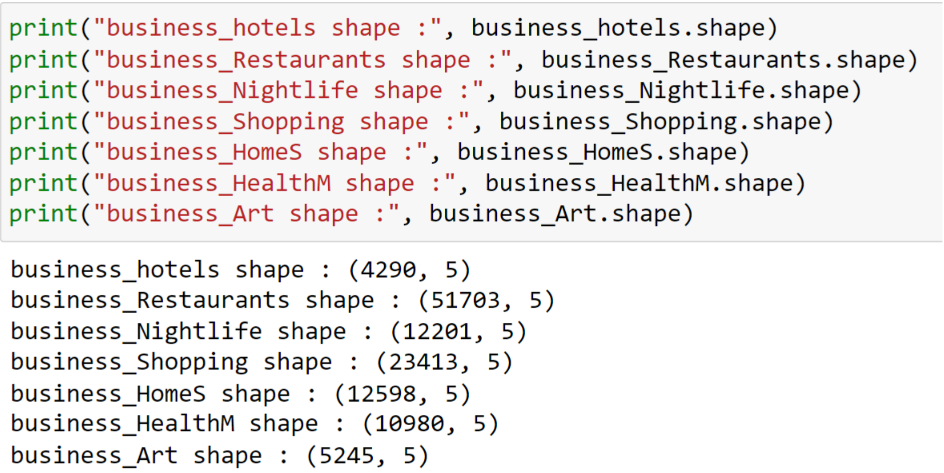
- **business:** in our business data after checking for duplicate values we found that there is no duplicate value after that we checked for missing value we found there is 13744 messing value in attribute and 103 missing value in the categories Figure 3.4 and for that, we dropped the rows that contain the missing values because we don't need them. The data after this process became 136601 rows, which decreased the data in this scenario by approximately 9.14%. after that, we need to select the POI from the cleaned business data and for that we filtered the data using the attribute by *Hotels&Travel*, *Restaurants*,*Nightlife*,*Shopping*,*HomeServices*,*Health&Medical*,*Arts&Entertainment* to get these POI Figure 3.5 shows the POI we've selected. After that, we concatenated them in POI-data, and now we have 120430 rows in the POI-data, which decreased by approximately 11.86% from the cleaned data and by approximately 19.90% from the original data.



```
# check missing values for business
business.isnull().sum()
```

```
business_id         0
name                0
city                0
attributes      13744
categories        103
dtype: int64
```

Figure 3.4: screenshot of business checking messing data

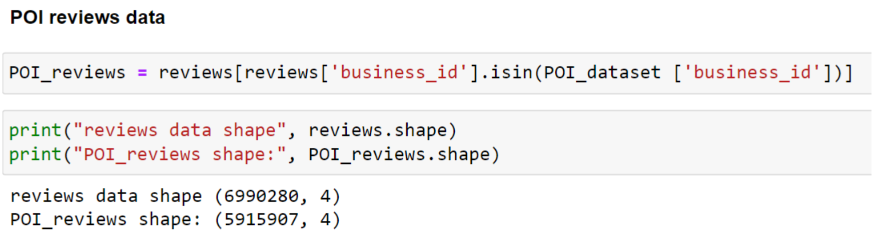


```
print("business_hotels shape :", business_hotels.shape)
print("business_Restaurants shape :", business_Restaurants.shape)
print("business_Nightlife shape :", business_Nightlife.shape)
print("business_Shopping shape :", business_Shopping.shape)
print("business_HomeS shape :", business_HomeS.shape)
print("business_HealthM shape :", business_HealthM.shape)
print("business_Art shape :", business_Art.shape)
```

```
business_hotels shape : (4290, 5)
business_Restaurants shape : (51703, 5)
business_Nightlife shape : (12201, 5)
business_Shopping shape : (23413, 5)
business_HomeS shape : (12598, 5)
business_HealthM shape : (10980, 5)
business_Art shape : (5245, 5)
```

Figure 3.5: screenshots of the POI selected

- **reviews:** first, we need to filter the reviews data to get only the reviews belongs to the POI-data Figure 3.6 shows this process and the new reviews' data shape, we see that the data has decreased by approximately 15.36%. In terms of reducing the size of the data and make it much useful we'll filter the POI-reviews data and get only the users who appear at least 10 times in the data these mean we have at minimum users who reviewed at least 10 time deferent business (POI) and maximum 50 times deferent business (POI) this also helps to make the recommendation more valuable and hopefully make the model more precession. after this process we get 2193563 reviews and 55440 user, which decreased by approximately 62.86% from POI-reviews data and by approximately 68.60% from reviews data,



POI reviews data

```
POI_reviews = reviews[reviews['business_id'].isin(POI_dataset ['business_id'])]
```

```
print("reviews data shape", reviews.shape)
print("POI_reviews shape:", POI_reviews.shape)
```

```
reviews data shape (6990280, 4)
POI_reviews shape: (5915907, 4)
```

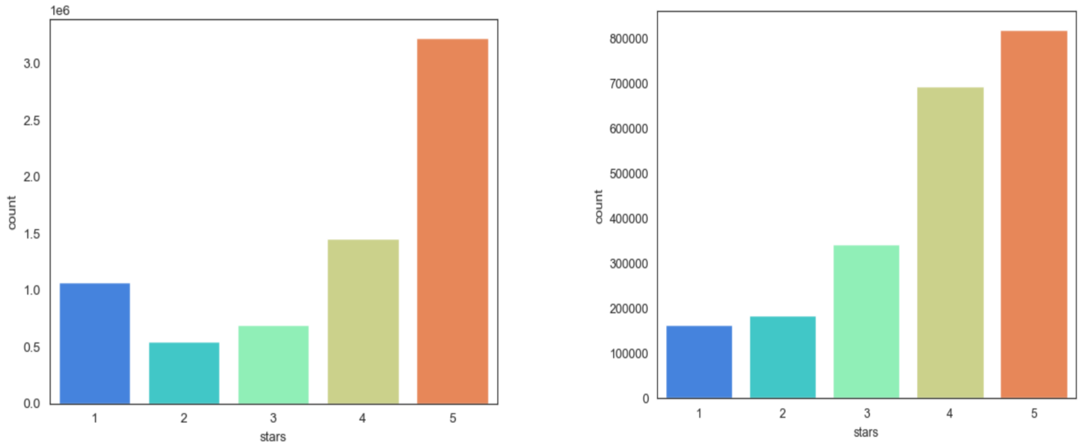
Figure 3.6: screenshots of POI-reviews data

after we filtered the reviews' data, we checked whether it has duplicated business (POI) in the POI-reviews data to make sure we didn't lose the semantic of the data and after checked it we found there is 2097751 duplicated business (POI) which mean approximately 95.56% of the data which validates our data. next we look for missing value, and we found that the data is clean from that.

- **users':** for users' data, we just filtered it by getting only users presented in POI-reviews data.

3.1.5 Discussing the data

After cleaning and preprocessing the data to meet our requirements, we compared the counts and distributions of ratings between the original dataset and the preprocessed and filtered dataset. In the original data, ratings 5 and 4 had the highest counts, indicating a concentration of positive ratings. However, after preprocessing and filtering, the counts shifted, with rating 5 remaining the highest but with a reduced count. The distribution became more balanced, spreading the ratings more evenly across the range. This analysis highlights the impact of data preprocessing on the distribution and relative frequencies of ratings, ultimately shaping the dataset to better suit our needs. Figure 3.7 visually presents the contrast between rating in the original review data and the rating in processed POI-reviews data.

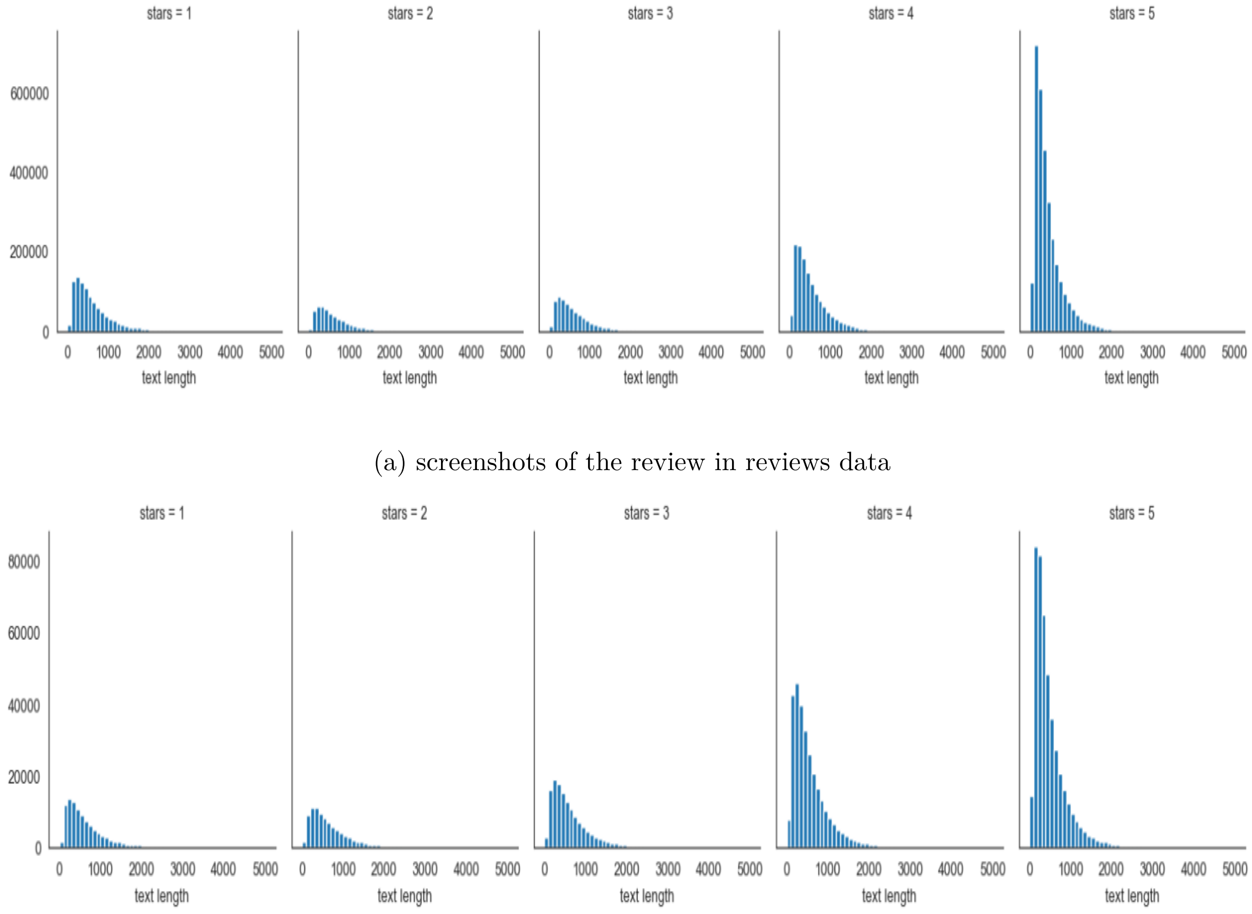


(a) screenshots of the rating in reviews data

(b) screenshots of the rating in POI-reviews data

Figure 3.7: screenshots shows the deference between rating in original data and preprocessed data

the same thing can see in the user review (text) in the original reviews data and the POI-reviews data which shows more balanced Figure 3.8 visually presents the contrast between review (text) in the original review data and the review (text) in processed POI-reviews data.



(a) screenshots of the review in reviews data

(b) screenshots of the review in POI-reviews data

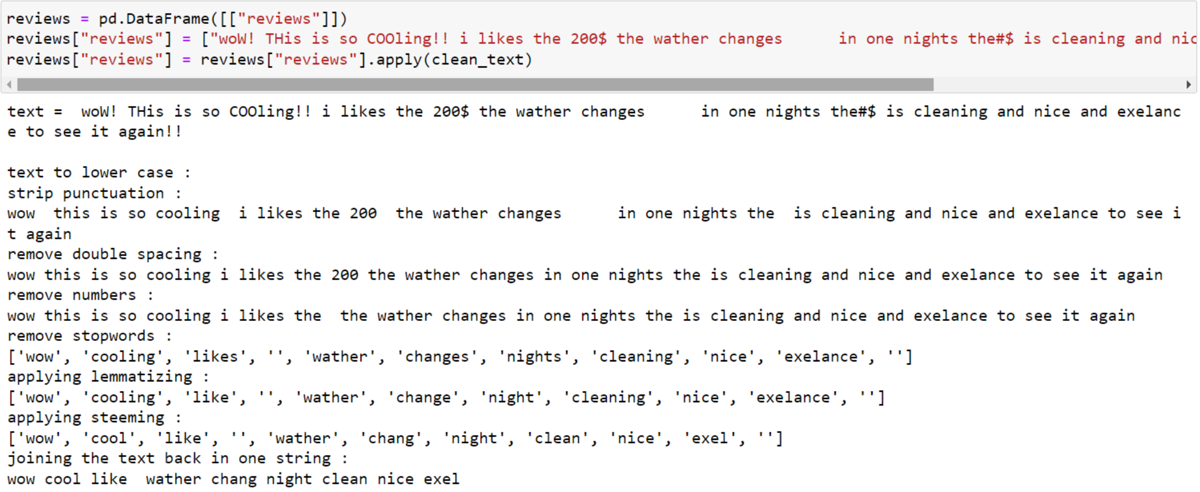
Figure 3.8: screenshots shows the difference between review in original data and preprocessed data

4 Sentiment analysis experiments

In this section, we're going to show the sentiment analysis process and experiments, and for this process, we used the POI-reviews data, which contains the users' reviews (text) that we're going to analyse.
First, we need to preprocess the data and then pass it to the sentiment analysis model.

4.1 Preprocessing the data

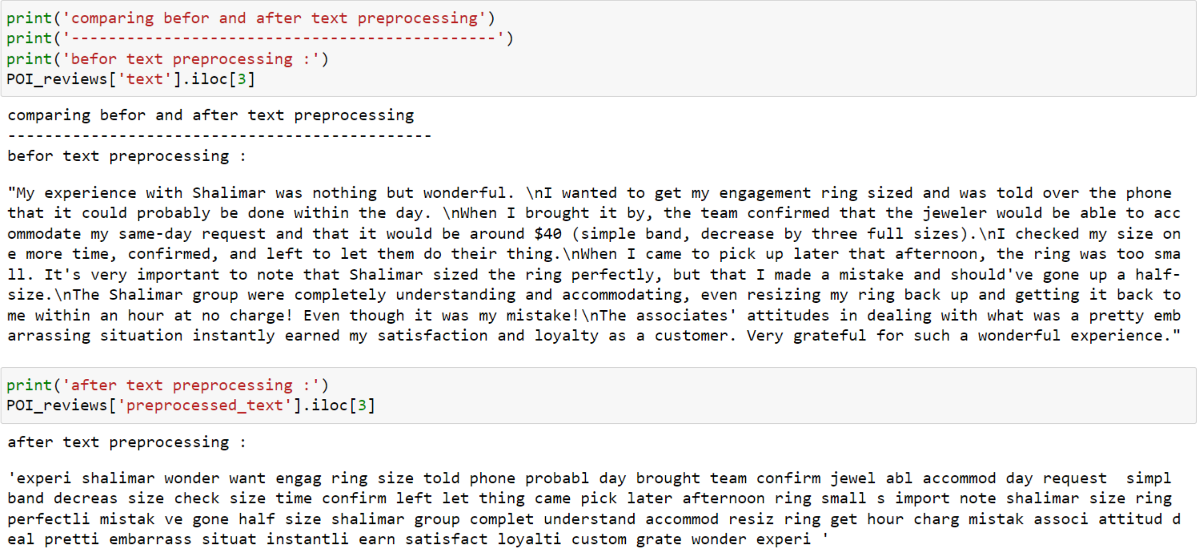
As we discussed in Section 6, the data will pass into the deference preprocessing stage, Figure 3.9 shows an example of the deferent process with its results and Figure 3.10 shows an example of a user review before and after the preprocessing.



```
reviews = pd.DataFrame([["reviews"]])
reviews["reviews"] = ["woW! THis is so COOling!! i likes the 200$ the wather changes      in one nights the#$ is cleaning and nic
reviews["reviews"] = reviews["reviews"].apply(clean_text)
```

```
text =  woW! THis is so COOling!! i likes the 200$ the wather changes      in one nights the#$ is cleaning and nice and exelanc
e to see it again!!

text to lower case :
strip punctuation :
wow  this is so cooling  i likes the 200  the wather changes      in one nights the  is cleaning and nice and exelance to see i
t again
remove double spacing :
wow this is so cooling i likes the 200 the wather changes in one nights the is cleaning and nice and exelance to see it again
remove numbers :
wow this is so cooling i likes the  the wather changes in one nights the is cleaning and nice and exelance to see it again
remove stopwords :
['wow', 'cooling', 'likes', '', 'wather', 'changes', 'nights', 'cleaning', 'nice', 'exelance', '']
applying lemmatizing :
['wow', 'cooling', 'like', '', 'wather', 'change', 'night', 'cleaning', 'nice', 'exelance', '']
applying steeming :
['wow', 'cool', 'like', '', 'wather', 'chang', 'night', 'clean', 'nice', 'exel', '']
joining the text back in one string :
wow cool like  wather chang night clean nice exel
```

Figure 3.9: screenshots of review preprocessing example



```
print('comparing befor and after text preprocessing')
print('---------------------------------------------')
print('befor text preprocessing :')
POI_reviews['text'].iloc[3]
```

```
comparing befor and after text preprocessing
---------------------------------------------
befor text preprocessing :

"My experience with Shalimar was nothing but wonderful. \nI wanted to get my engagement ring sized and was told over the phone
that it could probably be done within the day. \nWhen I brought it by, the team confirmed that the jeweler would be able to acc
ommodate my same-day request and that it would be around $40 (simple band, decrease by three full sizes).\nI checked my size on
e more time, confirmed, and left to let them do their thing.\nWhen I came to pick up later that afternoon, the ring was too sma
ll. It's very important to note that Shalimar sized the ring perfectly, but that I made a mistake and should've gone up a half-
size.\nThe Shalimar group were completely understanding and accommodating, even resizing my ring back up and getting it back to
me within an hour at no charge! Even though it was my mistake!\nThe associates' attitudes in dealing with what was a pretty emb
arrassing situation instantly earned my satisfaction and loyalty as a customer. Very grateful for such a wonderful experience."
```

```
print('after text preprocessing :')
POI_reviews['preprocessed_text'].iloc[3]
```

```
after text preprocessing :

'experi shalimar wonder want engag ring size told phone probabl day brought team confirm jewel abl accommod day request  simpl
band decreas size check size time confirm left let thing came pick later afternoon ring small s import note shalimar size ring
perfectli mistak ve gone half size shalimar group complet understand accommod resiz ring get hour charg mistak associ attitud d
eal pretti embarrass situat instantli earn satisfact loyalti custom grate wonder experi '
```

Figure 3.10: screenshots of review preprocessing example

4.2 Sentiment analysis

After preprocessing the data, we passed it into the sentiment analysis model to analyse it and give us the sentiment score. Figure 3.11 shows us statistics of the sentiment analysis result.

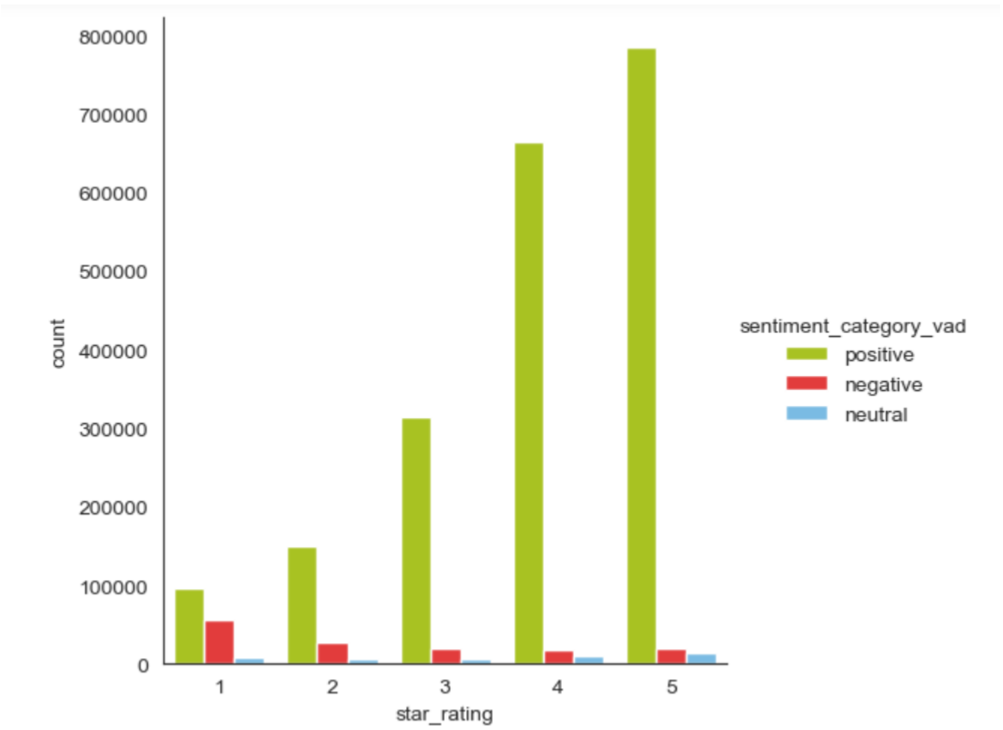
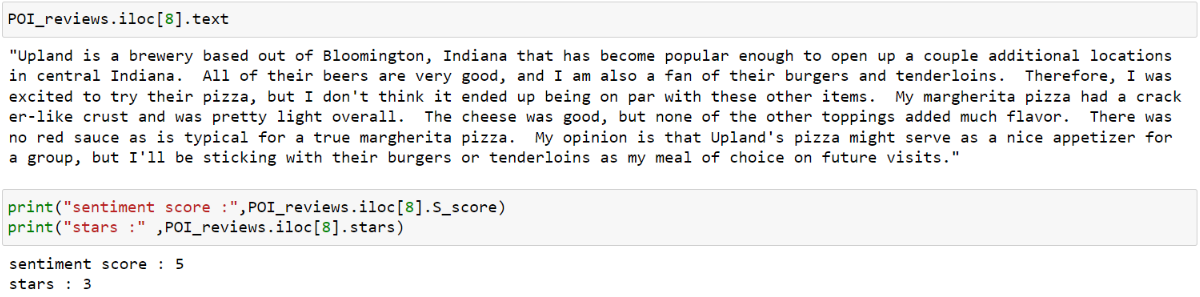


Figure 3.11: screenshots of sentiment analysis static result

These results show us that the reviews that have a rating of 1 contain some positive results, while other reviews that have a rating of 5 contain some negative results, and this shows us that the user may enjoy some POI but give it a low rating, and vice versa. Figure 3.12 shows an example of sentiment analysis for one of the users' reviews.



```
POI_reviews.iloc[8].text
```

```
"Upland is a brewery based out of Bloomington, Indiana that has become popular enough to open up a couple additional locations
in central Indiana.  All of their beers are very good, and I am also a fan of their burgers and tenderloins.  Therefore, I was
excited to try their pizza, but I don't think it ended up being on par with these other items.  My margherita pizza had a crack
er-like crust and was pretty light overall.  The cheese was good, but none of the other toppings added much flavor.  There was
no red sauce as is typical for a true margherita pizza.  My opinion is that Upland's pizza might serve as a nice appetizer for
a group, but I'll be sticking with their burgers or tenderloins as my meal of choice on future visits."
```

```
print("sentiment score :",POI_reviews.iloc[8].S_score)
print("stars :" ,POI_reviews.iloc[8].stars)
```

```
sentiment score : 5
stars : 3
```

Figure 3.12: screenshots show example of sentiment analysis result

In Figure 3.12 the user in his review looks like he enjoyed the visited POI, but he gave it a normal rating, and the result of the sentiment analysis shows that the user may give a higher rating for this POI.

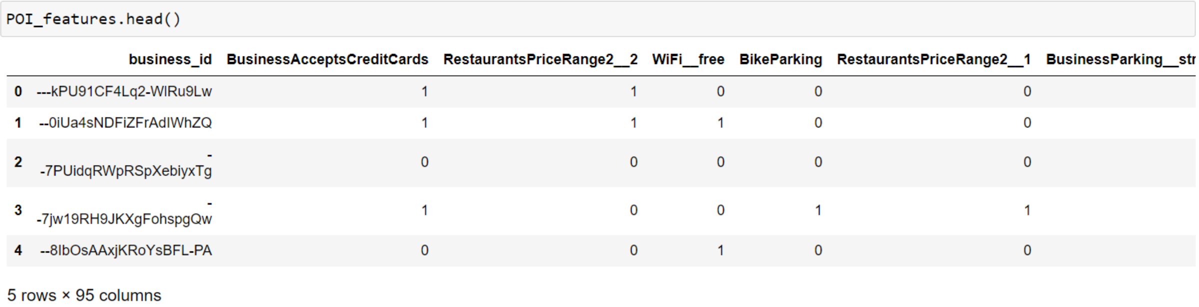
5 Recommendation system experiments

The recommendation system is based on two methods: content-based and collaborative filtering. We're going to start with content-based and then move on to the last stage, collaborative filtering.

5.1 content based

In this method, we're going to pass through different stages, starting with preparing the data and then computing the similarity between the user and the POI, as well as updating the users' preference vector:

- **preparing the data:** Because we have heterogeneous data, we need to preprocess and binarize it. First, we loop into the POI-data to extract the characteristics; after that, we check for each POI in the data and see if it contains these characteristics or not to create the POI characteristics matrix, Figure 3.13.



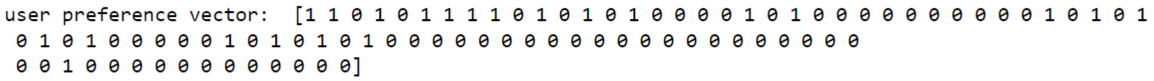
```
POI_features.head()
```

	business_id	BusinessAcceptsCreditCards	RestaurantsPriceRange2__2	WiFi__free	BikeParking	RestaurantsPriceRange2__1	BusinessParking__str
0	---kPU91CF4Lq2-WlRu9Lw	1	1	0	0	0	
1	--0iUa4sNDFiZFrAdIWhZQ	1	1	1	0	0	
2	- -7PUidqRWpRSpXebiyxTg	0	0	0	0	0	
3	- -7jw19RH9JKXgFohspgQw	1	0	0	1	1	
4	--8lbOsAAxjKRoYsBFL-PA	0	0	1	0	0	

5 rows × 95 columns

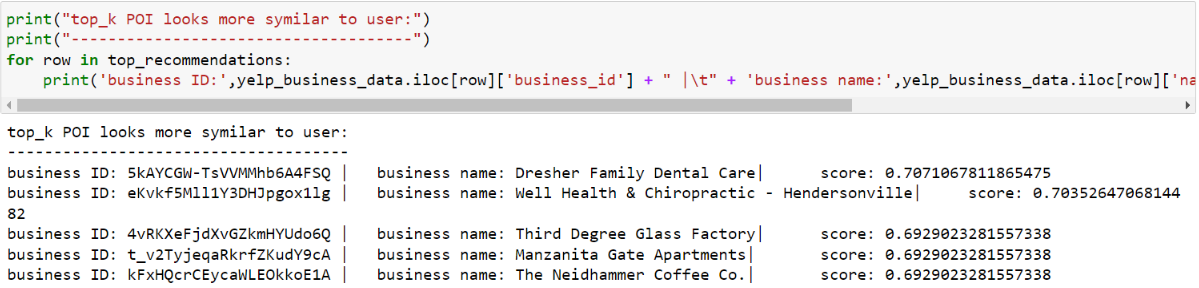
Figure 3.13: screenshots of POI characteristics data

- **computing the similarity:** After we prepare the data and after the user selects his preference, Figure 3.14 the system'll calculate the similarity between the user and all the POI in the data and provide it with the top_K POI that looks more similar to the user's preference, Figure 3.15.



```
user preference vector:  [1 1 0 1 0 1 1 1 1 0 1 0 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1
 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]
```

Figure 3.14: screenshots of user preference vector

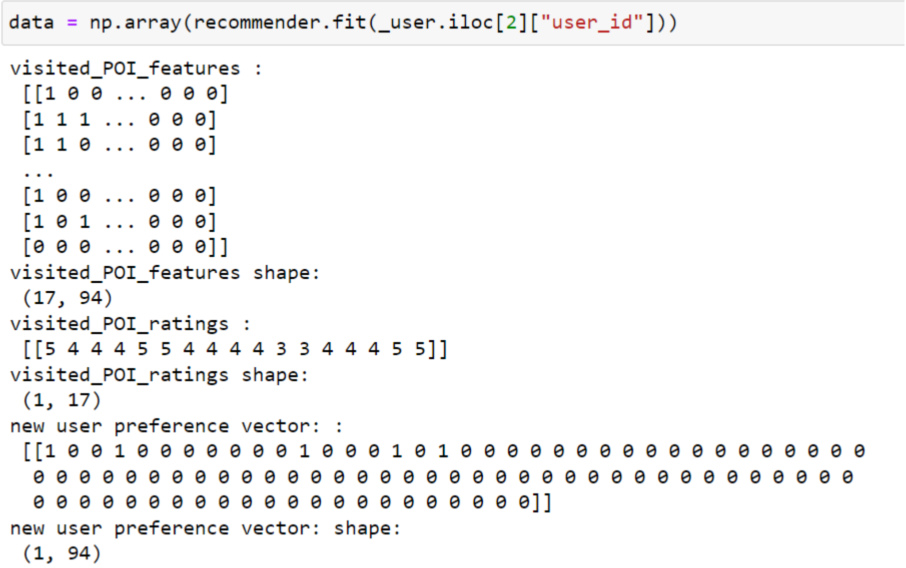


```
print("top_k POI looks more symilar to user:")
print("------------------------------------")
for row in top_recommendations:
    print('business ID:',yelp_business_data.iloc[row]['business_id'] + " |\t" + 'business name:',yelp_business_data.iloc[row]['na
```

```
top_k POI looks more symilar to user:
------------------------------------
business ID: 5kAYCGW-TsVVMMhb6A4FSQ |   business name: Dresher Family Dental Care|      score: 0.7071067811865475
business ID: eKvkf5Ml1Y3DHJpgox1lg |   business name: Well Health & Chiropractic - Hendersonville|      score: 0.70352647068144
82
business ID: 4vRKXeFjdXvGZkmHYUdo6Q |   business name: Third Degree Glass Factory|      score: 0.6929023281557338
business ID: t_v2TyjeqaRkrfZKudY9cA |   business name: Manzanita Gate Apartments|      score: 0.6929023281557338
business ID: kFxHQcrCEycaWLEOkkoE1A |   business name: The Neidhammer Coffee Co.|      score: 0.6929023281557338
```

Figure 3.15: screenshots of similarity result between user and POI

- **updating users' preference vector:** We know that the users' preferences change over time, and to handle the users' changes, we used a technique to deal with it: we keep tracking the visited POI and using them to update our users' preferences section 7.1.2. Figure 3.16 shows an example of a user preference update.



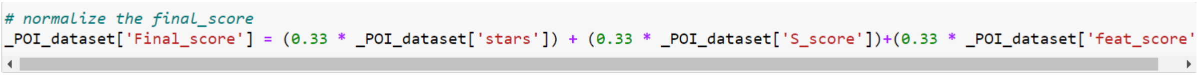
```
data = np.array(recommender.fit(_user.iloc[2]["user_id"]))
```

```
visited_POI_features :
 [[1 0 0 ... 0 0 0]
 [1 1 1 ... 0 0 0]
 [1 1 0 ... 0 0 0]
 ...
 [1 0 0 ... 0 0 0]
 [1 0 1 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
visited_POI_features shape:
 (17, 94)
visited_POI_ratings :
 [[5 4 4 4 5 5 4 4 4 4 3 3 4 4 4 5 5]]
visited_POI_ratings shape:
 (1, 17)
new user preference vector: :
 [[1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
new user preference vector: shape:
 (1, 94)
```

Figure 3.16: screenshots of user preference vector update example

5.2 final score

After computing the different scores, we need to normalise them and prepare them for the final stage; the process is described in Section 2.7. Figure 3.17 shows this process for our data.

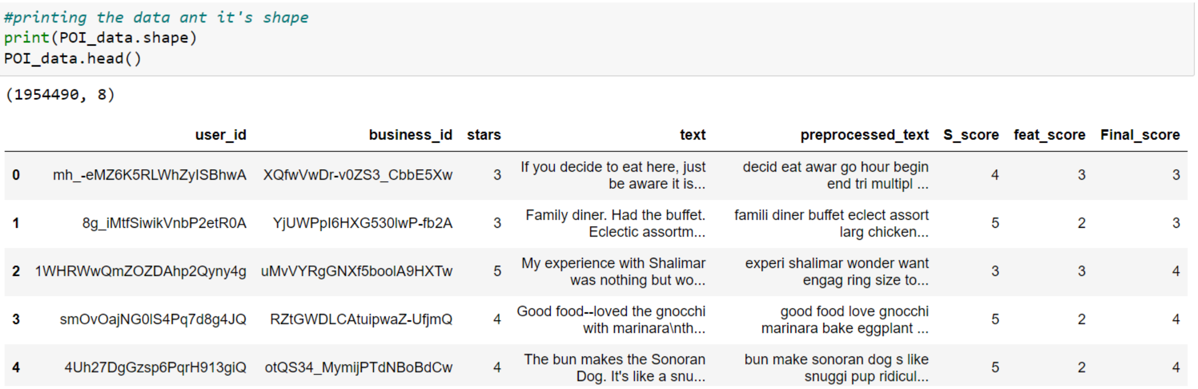


```
# normalize the final_score
_POI_dataset['Final_score'] = (0.33 * _POI_dataset['stars']) + (0.33 * _POI_dataset['S_score'])+(0.33 * _POI_dataset['feat_score'
```

Figure 3.17: screenshots of the normalisation process

5.3 collaborative filtering - lightGCN

This is the final stage of providing the recommendation to the user, after passing through the previous phases and preparing the data. Figure 3.18. this data used to train the *lightGcn* model, after the model trained it'll be able to provide users' by the recommendation of POI that match their needs Figure 3.19 shows an example of the final recommendation.

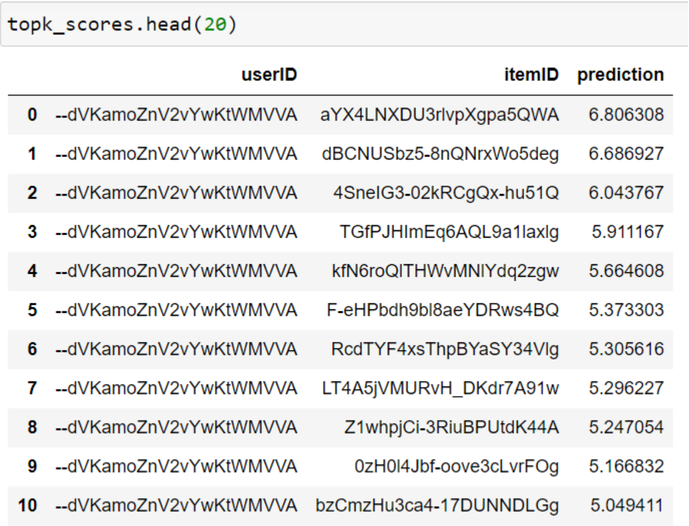


```
#printing the data ant it's shape
print(POI_data.shape)
POI_data.head()
```

```
(1954490, 8)
```

	user_id	business_id	stars	text	preprocessed_text	S_score	feat_score	Final_score
0	mh_-eMZ6K5RLWhZylSBhwA	XQfwVwDr-v0ZS3_CbbE5Xw	3	If you decide to eat here, just be aware it is...	decid eat awar go hour begin end tri multipl ...	4	3	3
1	8g_iMtfSiwikVnbP2etR0A	YjUWPpI6HXG530lwP-fb2A	3	Family diner. Had the buffet. Eclectic assortm...	famili diner buffet eclect assort larg chicken...	5	2	3
2	1WHRWwQmZOZDAhp2Qyny4g	uMvVYRgGNXf5boolA9HXTw	5	My experience with Shalimar was nothing but wo...	experi shalimar wonder want engag ring size to...	3	3	4
3	smOvOajNG0lS4Pq7d8g4JQ	RZtGWDLCAtuipwaZ-UfjmQ	4	Good food--loved the gnocchi with marinara\nth...	good food love gnocchi marinara bake eggplant ...	5	2	4
4	4Uh27DgGzsp6PqrH913giQ	otQS34_MymijPTdNBoBdCw	4	The bun makes the Sonoran Dog. It's like a snu...	bun make sonoran dog s like snuggi pup ridicul...	5	2	4

Figure 3.18: screenshots of the final data



```
topk_scores.head(20)
```

```
                        userID                  itemID  prediction
 0  --dVKamoZnV2vYwKtWMVVA  aYX4LNXDU3rlvpXgpa5QWA    6.806308
 1  --dVKamoZnV2vYwKtWMVVA  dBCNUSbz5-8nQNrxWo5deg    6.686927
 2  --dVKamoZnV2vYwKtWMVVA   4SnelG3-02kRCgQx-hu51Q    6.043767
 3  --dVKamoZnV2vYwKtWMVVA   TGfPJHImEq6AQL9a1laxlg    5.911167
 4  --dVKamoZnV2vYwKtWMVVA   kfN6roQlTHWvMNlYdq2zgw    5.664608
 5  --dVKamoZnV2vYwKtWMVVA  F-eHPbdh9bl8aeYDRws4BQ    5.373303
 6  --dVKamoZnV2vYwKtWMVVA   RcdTYF4xsThpBYaSY34Vlg    5.305616
 7  --dVKamoZnV2vYwKtWMVVA  LT4A5jVMURvH_DKdr7A91w    5.296227
 8  --dVKamoZnV2vYwKtWMVVA    Z1whpjCi-3RiuBPUtdK44A    5.247054
 9  --dVKamoZnV2vYwKtWMVVA     0zH0l4Jbf-oove3cLvrFOg    5.166832
10  --dVKamoZnV2vYwKtWMVVA  bzCmzHu3ca4-17DUNNDLGg    5.049411
```

Figure 3.19: screenshots of $lightGCN$ recommendation list for specific user from the dataset

Figure 3.19 represents the recommendation result for a specific user, the $userID$ is the ID of the user, the $itemID$ is the ID of the different recommended POI and the prediction is the score of each POI recommended for the user.

6 Evaluation and discussion of final results

In this part, we'll discuss the evaluation result and the parameter used to train $lightGCN$ and comparing it with other different recommendation models.

6.1 experiments setting

To train the $lightGCN$ and get a better result, we've tried different settings to train the model. Table 3.1 shows the different results with different parameter sets. To reduce the experiment workload and keep the comparison fair with the SVD and $BiVAE$ model, we picked the same shared final parameters used to train, $lightGCN$ which are $batch\ size = 1024$, $learning\ rate = 0.001$, $epoch = 50$ and we split the data into 80% train and 20% test for the three mode. Table 3.2 shows the different results of the three models, and Figure 3.20 illustrates the different results of $Recall@k$, $NDGC$, MAP metrics for the different models.

dataset	Preprocessed Yelp dataset (100k)					
# layer	batch size	learning rate	embed size	RECALL@10	NDGC	MAP
1 layer	**1024**	**0.001**	**64**	0.069183	0.040142	0.029634
2 layer				0.087943	0.056420	0.045199
3 layer				**0.257155**	**0.115672**	**0.068016**
4 layer				0.133932	0.077119	0.057520
1 layer	1024	0.001	128	0.085158	0.048261	0.035996
2 layer				0.130612	0.081289	0.063716
3 layer				0.131733	0.076968	0.057892
4 layer				0.151325	0.091124	0.069659
1 layer	1024	0.01	64	0.075554	0.040269	0.028181
2 layer				0.094026	0.053665	0.039709
3 layer				0.077488	0.042198	0.030396
4 layer				0.089592	0.050134	0.036770
1 layer	128	0.01	64	0.038945	0.018599	0.011725
2 layer				0.031838	0.031838	0.031838
3 layer				0.074302	0.024439	0.024439
4 layer				0.072589	0.037764	0.026074

Table 3.1: Different result of the different parameter used to train lightGCN

The table 3.1 showcases the performance comparison of different configurations of the *LightGCN* model on the preprocessed Yelp dataset (100k). The results highlight the impact of various parameters on the model's performance metrics. Notably, increasing the number of layers in *LightGCN* leads to improvements in RECALL@10, NDGC, and MAP values, indicating enhanced recommendation accuracy. Specifically, the configuration with three layers, a batch size of 1024, a learning rate of 0.001, and an embedding size of 64 demonstrates the highest performance across the metrics. These findings suggest that deeper *LightGCN* models, with appropriate parameter settings, have the potential to capture intricate patterns and boost recommendation quality.

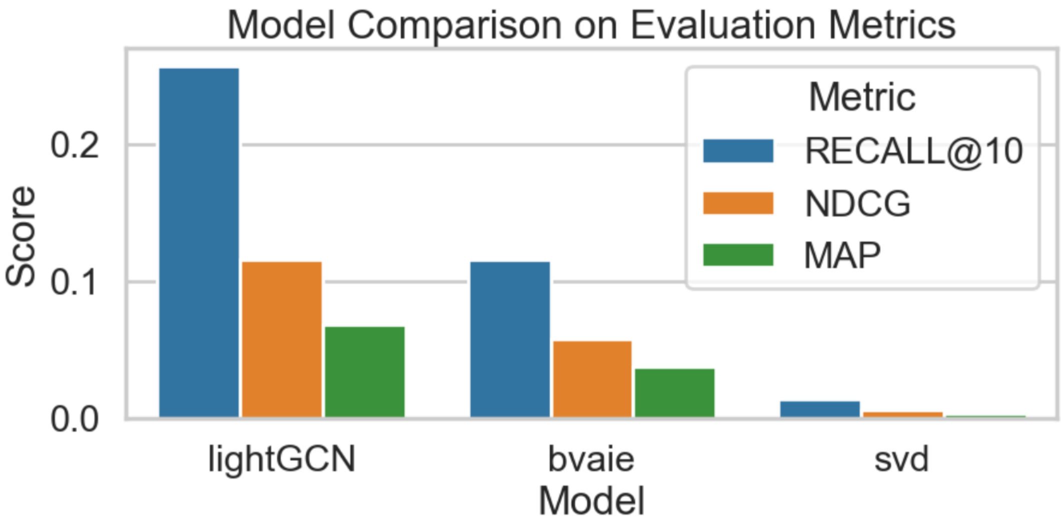


Figure 3.20: screenshots of the evaluation results on different metrics

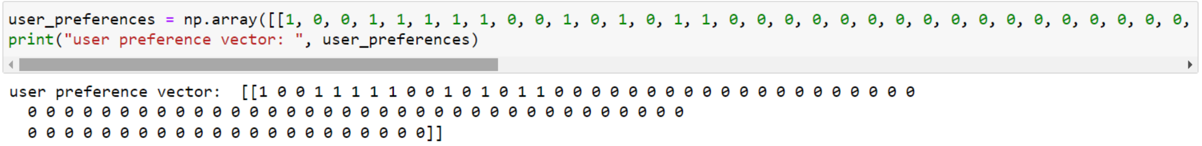
dataset	Preprocessed Yelp dataset (100k)		
models	RECALL@10	NDGC	MAP
lightGCN	**0.257155**	**0.115672**	**0.068016**
bvaie	0.115680	0.057732	0.037779
svd	0.014165	0.006153	0.003199

Table 3.2: Metrics result of each model on the same configuration and dataset

The table 3.2 and the figure 3.20 presents a comparison of different models on the preprocessed Yelp dataset (100k) based on three evaluation metrics: RECALL@10, NDCG, and MAP. Among the models, *lightGCN* demonstrates the highest performance across all three metrics. It achieves a RECALL@10 score of 0.257155, indicating that it successfully captures 25.7% of the relevant items within the top 10 recommendations. Furthermore, *lightGCN* obtains a notable NDGC (Normalized Discounted Cumulative Gain) score of 0.115672, suggesting that it effectively ranks the recommended items based on their relevance. Additionally, the Mean average precision (MAP) for *lightGCN* is 0.068016, reflecting its ability to provide accurate and relevant recommendations on average.
In comparison, the *bvaie* model shows lower performance in terms of all three metrics. It achieves a RECALL@10 score of 0.115680, indicating that it captures only 11.6% of the relevant items within the top 10 recommendations. The NDCG score for *bvaie* is 0.057732, suggesting that its ranking of recommended items based on relevance is relatively weaker. Similarly, the MAP score for *bvaie* is 0.037779, indicating a lower accuracy and relevance of its recommendations compared to *lightGCN*.
Lastly, the *svd* model performs the weakest among the three models, with significantly lower scores across all metrics. It achieves a RECALL@10 score of 0.014165, implying a relatively low ability to capture relevant items within the top 10 recommendations. The NDCG score for *svd* is 0.006153, indicating poor ranking of the recommended items based on relevance. Similarly, the MAP score for *svd* is 0.00319, highlighting the limited accuracy and relevance of its recommendations compared to the other models.
In summary, the evaluation results clearly show that *lightGCN* outperforms both *bvaie* and *svd* models on the preprocessed Yelp dataset (100k) based on the metrics of RECALL@10, NDCG, and MAP. It consistently demonstrates higher recall, better ranking of recommended items, and higher accuracy in providing relevant recommendations.

7 scenario example

In this section, we'll design a scenario for the system's use. First, when the user starts in the system, he'll pick some POI Figure 3.21 shows the initial preferences vector of the user.

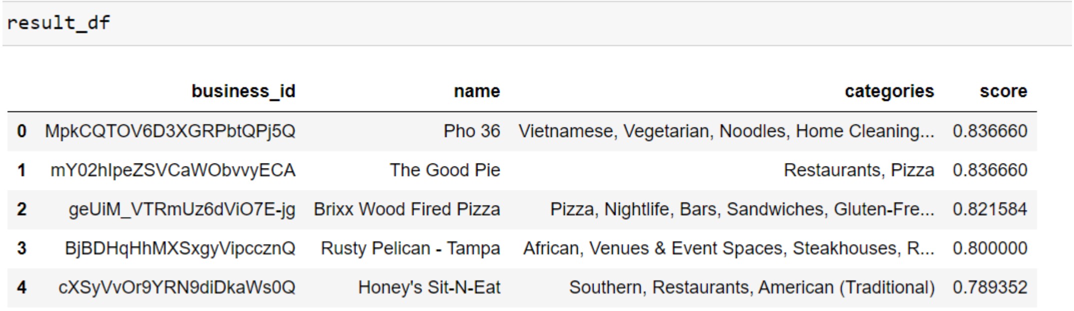


```
user_preferences = np.array([[1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
print("user preference vector: ", user_preferences)
```

```
user preference vector:  [[1 0 0 1 1 1 1 1 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
```

Figure 3.21: screenshots of user preference vector example

after that the system'll compute the similarity between the user and POI in the system figure 3.22 shows the result of the content based (similarity score)

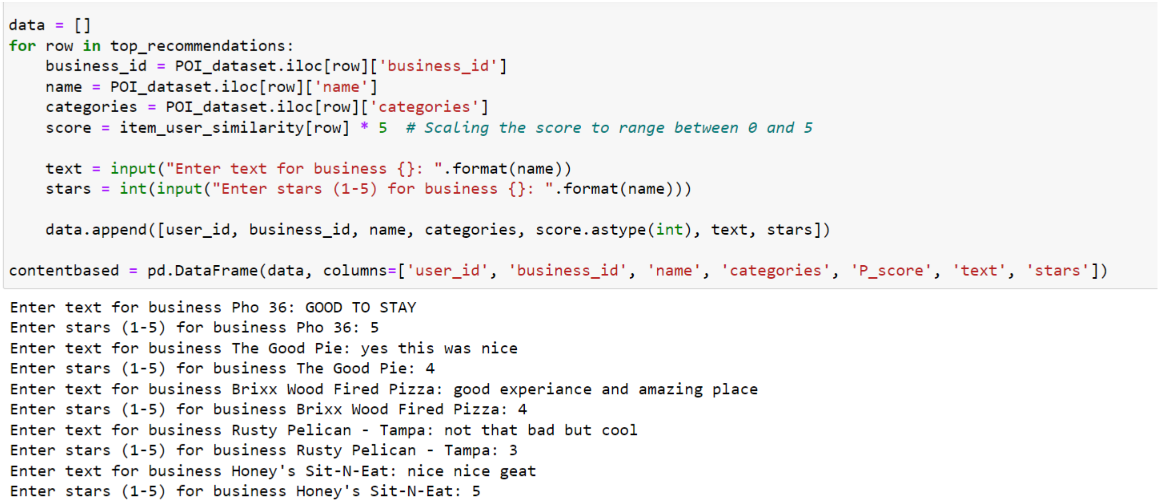


```
result_df
```

	business_id	name	categories	score
0	MpkCQTOV6D3XGRPbtQPj5Q	Pho 36	Vietnamese, Vegetarian, Noodles, Home Cleaning...	0.836660
1	mY02hlpeZSVCaWObvvyECA	The Good Pie	Restaurants, Pizza	0.836660
2	geUiM_VTRmUz6dViO7E-jg	Brixx Wood Fired Pizza	Pizza, Nightlife, Bars, Sandwiches, Gluten-Fre...	0.821584
3	BjBDHqHhMXSxgyVipccznQ	Rusty Pelican - Tampa	African, Venues & Event Spaces, Steakhouses, R...	0.800000
4	cXSyVvOr9YRN9diDkaWs0Q	Honey's Sit-N-Eat	Southern, Restaurants, American (Traditional)	0.789352

Figure 3.22: screenshots of user POI similarity example

After that, let's assume that the user went to these POI. The user'll leave his feedback about these visited POI figure 3.23 shows the user feedback,

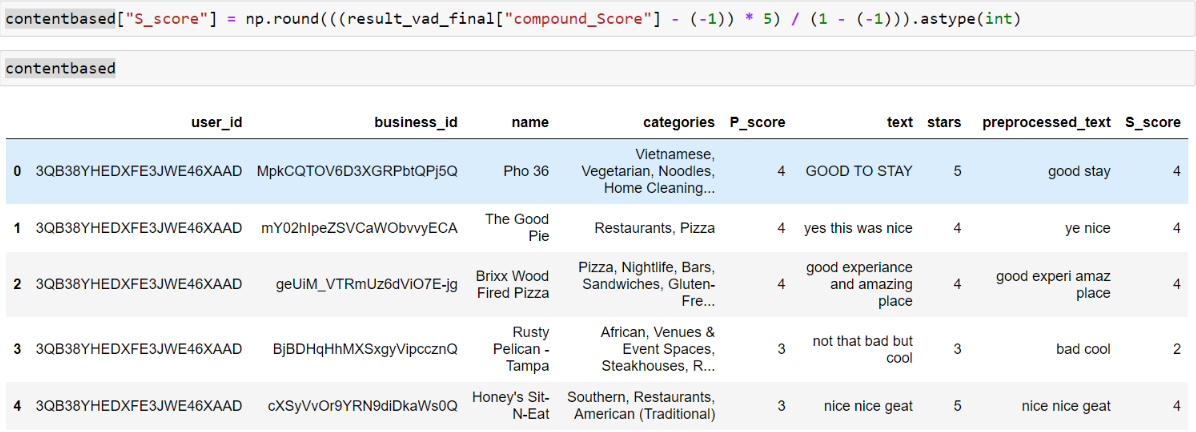


```
data = []
for row in top_recommendations:
    business_id = POI_dataset.iloc[row]['business_id']
    name = POI_dataset.iloc[row]['name']
    categories = POI_dataset.iloc[row]['categories']
    score = item_user_similarity[row] * 5  # Scaling the score to range between 0 and 5

    text = input("Enter text for business {}: ".format(name))
    stars = int(input("Enter stars (1-5) for business {}: ".format(name)))

    data.append([user_id, business_id, name, categories, score.astype(int), text, stars])

contentbased = pd.DataFrame(data, columns=['user_id', 'business_id', 'name', 'categories', 'P_score', 'text', 'stars'])
```

```
Enter text for business Pho 36: GOOD TO STAY
Enter stars (1-5) for business Pho 36: 5
Enter text for business The Good Pie: yes this was nice
Enter stars (1-5) for business The Good Pie: 4
Enter text for business Brixx Wood Fired Pizza: good experiance and amazing place
Enter stars (1-5) for business Brixx Wood Fired Pizza: 4
Enter text for business Rusty Pelican - Tampa: not that bad but cool
Enter stars (1-5) for business Rusty Pelican - Tampa: 3
Enter text for business Honey's Sit-N-Eat: nice nice geat
Enter stars (1-5) for business Honey's Sit-N-Eat: 5
```

Figure 3.23: screenshots of user feedback example

next, the system'll analyse user sentiment by analysing the user review figure 3.24.



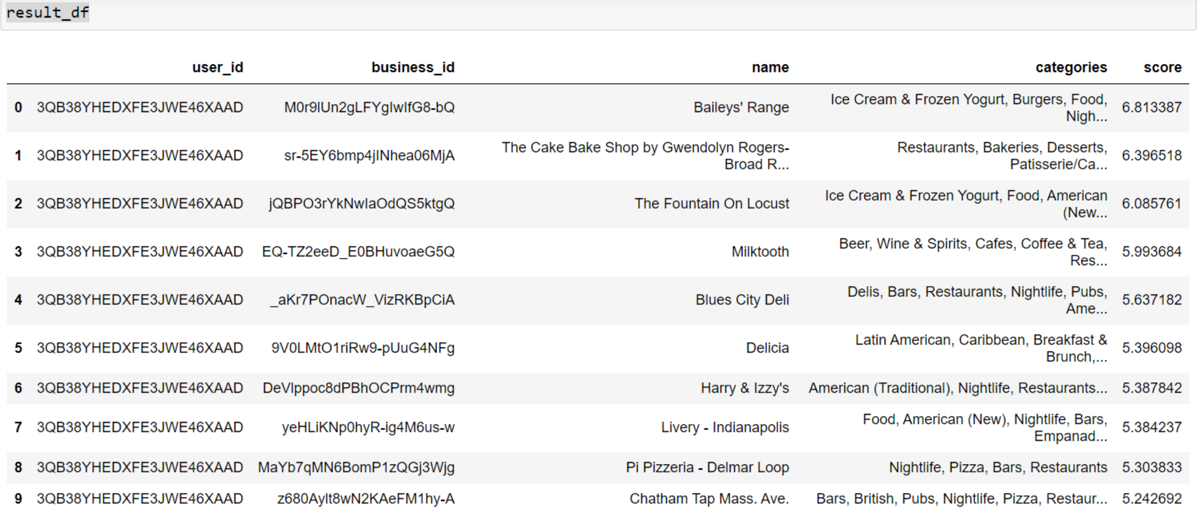
```
contentbased["S_score"] = np.round(((result_vad_final["compound_Score"] - (-1)) * 5) / (1 - (-1))).astype(int)
```

```
contentbased
```

	user_id	business_id	name	categories	P_score	text	stars	preprocessed_text	S_score
0	3QB38YHEDXFE3JWE46XAAD	MpkCQTOV6D3XGRPbtQPj5Q	Pho 36	Vietnamese, Vegetarian, Noodles, Home Cleaning...	4	GOOD TO STAY	5	good stay	4
1	3QB38YHEDXFE3JWE46XAAD	mY02hIpeZSVCaWObvvyECA	The Good Pie	Restaurants, Pizza	4	yes this was nice	4	ye nice	4
2	3QB38YHEDXFE3JWE46XAAD	geUiM_VTRmUz6dViO7E-jg	Brixx Wood Fired Pizza	Pizza, Nightlife, Bars, Sandwiches, Gluten-Fre...	4	good experiance and amazing place	4	good experi amaz place	4
3	3QB38YHEDXFE3JWE46XAAD	BjBDHqHhMXSxgyVipccznQ	Rusty Pelican - Tampa	African, Venues & Event Spaces, Steakhouses, R...	3	not that bad but cool	3	bad cool	2
4	3QB38YHEDXFE3JWE46XAAD	cXSyVvOr9YRN9diDkaWs0Q	Honey's Sit-N-Eat	Southern, Restaurants, American (Traditional)	3	nice nice geat	5	nice nice geat	4

Figure 3.24: screenshots of sentiment analysis of user feedback example

after that, the system'll compute the final score; and the *lightGCN* will provide the user with the recommendation list; figure 3.25 displays the recommendation list for the user; the system will track the user visited POI,

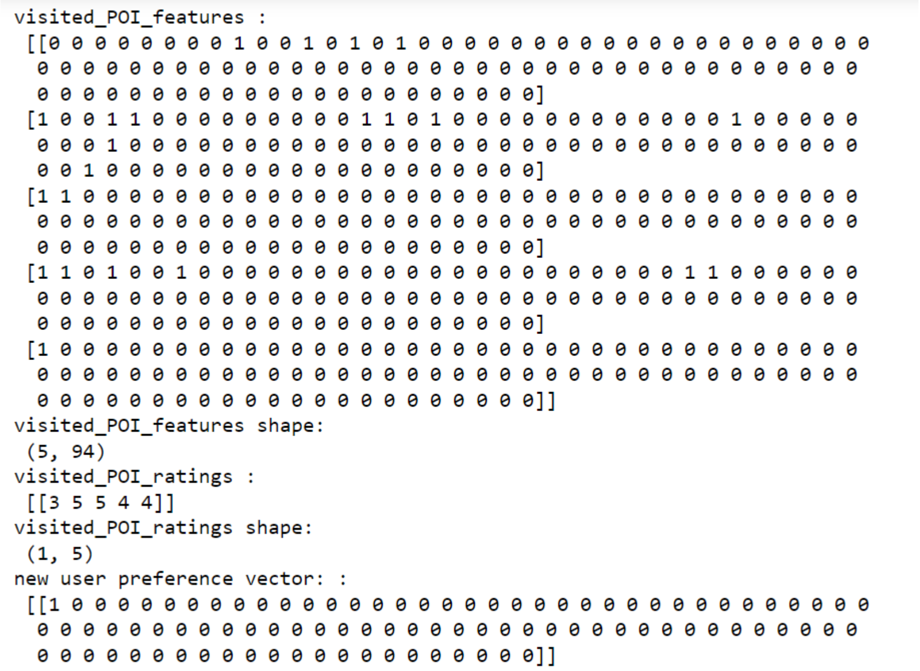


```
result_df
```

	user_id	business_id	name	categories	score
0	3QB38YHEDXFE3JWE46XAAD	M0r9lUn2gLFYglwlfG8-bQ	Baileys' Range	Ice Cream & Frozen Yogurt, Burgers, Food, Nigh...	6.813387
1	3QB38YHEDXFE3JWE46XAAD	sr-5EY6bmp4jlNhea06MjA	The Cake Bake Shop by Gwendolyn Rogers-Broad R...	Restaurants, Bakeries, Desserts, Patisserie/Ca...	6.396518
2	3QB38YHEDXFE3JWE46XAAD	jQBPO3rYkNwlaOdQS5ktgQ	The Fountain On Locust	Ice Cream & Frozen Yogurt, Food, American (New...	6.085761
3	3QB38YHEDXFE3JWE46XAAD	EQ-TZ2eeD_E0BHuvoaeG5Q	Milktooth	Beer, Wine & Spirits, Cafes, Coffee & Tea, Res...	5.993684
4	3QB38YHEDXFE3JWE46XAAD	_aKr7POnacW_VizRKBpCiA	Blues City Deli	Delis, Bars, Restaurants, Nightlife, Pubs, Ame...	5.637182
5	3QB38YHEDXFE3JWE46XAAD	9V0LMtO1riRw9-pUuG4NFg	Delicia	Latin American, Caribbean, Breakfast & Brunch,...	5.396098
6	3QB38YHEDXFE3JWE46XAAD	DeVlppoc8dPBhOCPrm4wmg	Harry & Izzy's	American (Traditional), Nightlife, Restaurants...	5.387842
7	3QB38YHEDXFE3JWE46XAAD	yeHLiKNp0hyR-ig4M6us-w	Livery - Indianapolis	Food, American (New), Nightlife, Bars, Empanad...	5.384237
8	3QB38YHEDXFE3JWE46XAAD	MaYb7qMN6BomP1zQGj3Wjg	Pi Pizzeria - Delmar Loop	Nightlife, Pizza, Bars, Restaurants	5.303833
9	3QB38YHEDXFE3JWE46XAAD	z680Aylt8wN2KAeFM1hy-A	Chatham Tap Mass. Ave.	Bars, British, Pubs, Nightlife, Pizza, Restaur...	5.242692

Figure 3.25: screenshots of recommendation list example

and it'll update the user preferences; figure 3.26 shows the new user preferences vector.



```
visited_POI_features :
 [[0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
  0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
visited_POI_features shape:
 (5, 94)
visited_POI_ratings :
 [[3 5 5 4 4]]
visited_POI_ratings shape:
 (1, 5)
new user preference vector: :
 [[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
```

Figure 3.26: screenshots of updating user preferences, example

8 Conclusion

In this chapter, we describe the phases of development of our POI recommendation system based on sentiment analysis and hybrid filtering, as well as the tools and results obtained during the testing of the two sentiment analysis and recommendation modules. We can say that the results meet the fundamental requirements of a recommendation system for users' POI, as they are largely outstanding and satisfactory.

General Conclusion

In this graduation project, we focused on the area of Data Analysis and Processing for Recommendation Systems and proposed an approach to recommend different POIs.

Recommendation systems are a good way to give people individual and valuable information, so they can make better decisions in their daily lives. Therefore, we have introduced our new recommendation approach in this brief to improve these decisions, which provide suggestions from contextual point of interest. Our recommendation system is a hybrid and contextual system that adapts to the user's type preferences and update them and take uses of the characteristics of the visited point of interest, and also produces dynamic suggestion results based on previous methods.

The results of the case study, conducted using the Yelp dataset, demonstrated that the proposed POI recommendation system can effectively utilise user preferences, feedback information, and other relevant criteria, such as sentiment analysis, to provide different users with customised and relevant POI recommendations.

our work contributes to the advancement of recommendation systems, enabling more accurate and personalized recommendations for users in various domains.

By leveraging sentiment analysis, data filtering, and preprocessing techniques, along with graph neural network model, our approach empowers recommendation systems to provide valuable insights and recommendations tailored to individual users' preferences, while addressing the challenges associated with heterogeneous data.

From a future research perspective, we anticipate the following enhancements to our work:

1. Utilisation of social media data in the system, to obtain information regarding previous visits and user interactions on POI context.

2. Add geolocation in real time option to the system which help to provide POI the closest to the users'

3. Add the close friends feature if the user's friends are nearby, their position is indicated on a map, and recommendations for the same POI are provided.

Bibliography

[1] Guocan Yu, Kecheng Jie, and Feihe Huang. Supramolecular amphiphiles based on host–guest molecular recognition motifs. *Chemical reviews*, 115(15):7240–7303, 2015.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[3] Badreesh Shetty. An in-depth guide to how recommender systems work. `https://builtin.com/data-science/recommendersystems`, March 2023. Online; accessed 10 May 2023.

[4] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.

[5] Per-Erik Danielsson. Euclidean distance mapping. *Computer Graphics and image processing*, 14(3):227–248, 1980.

[6] Xian-Da Zhang. A matrix algebra approach to artificial intelligence. 2020.

[7] Marwa Hussien Mohamed, Mohamed Helmy Khafagy, and Mohamed Hasan Ibrahim. Recommender systems challenges and solutions survey. In *2019 international conference on innovative trends in computer engineering (ITCE)*, pages 149–155. IEEE, 2019.

[8] Yassine Afoudi, Mohamed Lazaar, and Mohamed Al Achhab. Collaborative filtering recommender system. In *Advanced Intelligent Systems for Sustainable Development (AI2SD'2018) Volume 5: Advanced Intelligent Systems for Computing Sciences*, pages 332–345. Springer, 2019.

[9] Robin Burke. Hybrid web recommender systems. *The adaptive web: methods and strategies of web personalization*, pages 377–408, 2007.

[10] Bamshad Mobasher, Xin Jin, and Yanzan Zhou. Semantically enhanced collaborative filtering on the web. In *Web Mining: From Web to Semantic Web: First European Web Mining Forum, EWMF 2003, Cavtat-Dubrovnik, Croatia, September 22, 2003, Invited and Selected Revised Papers*, pages 57–76. Springer, 2004.

[11] Barry Smyth and Paul Cotter. A personalised tv listings service for the digital tv age. *Knowledge-Based Systems*, 13(2-3):53–59, 2000.

[12] Daniel Billsus and Michael J Pazzani. User modeling for adaptive news access. *User modeling and user-adapted interaction*, 10:147–180, 2000.

[13] DERRY O' SULLIVAN, Barry Smyth, and DAVID WILSON. Preserving recommender accuracy and diversity in sparse datasets. *International Journal on Artificial Intelligence Tools*, 13(01):219–235, 2004.

[14] David C Wilson, Barry Smyth, and Derry O' Sullivan. Sparsity reduction in collaborative recommendation: A case-based approach. *International journal of pattern recognition and artificial intelligence*, 17(05):863–884, 2003.

[15] Michael J Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review*, 13:393–408, 1999.

[16] Alejandro Bellogín, Iván Cantador, Fernando Díez, Pablo Castells, and Enrique Chavarriaga. An empirical comparison of social, collaborative filtering, and hybrid recommenders. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):1–29, 2013.

[17] Shristi Shakya Khanal, PWC Prasad, Abeer Alsadoon, and Angelika Maag. A systematic review: machine learning based recommendation systems for e-learning. *Education and Information Technologies*, 25:2635–2664, 2020.

[18] Folasade Olubusola Isinkaye, Yetunde O Folajimi, and Bolande Adefowoke Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal*, 16(3):261–273, 2015.

[19] Taner Arsan, Efecan Köksal, and Zeki Bozkuş. Comparison of collaborative filtering.

[20] Gourav Jain, Tripti Mahara, and Kuldeep Narayan Tripathi. A survey of similarity measures for collaborative filtering-based recommender system. In *Soft Computing: Theories and Applications: Proceedings of SoCTA 2018*, pages 343–352. Springer, 2020.

[21] Jun Zeng, Feng Li, Haiyang Liu, Junhao Wen, and Sachio Hirokawa. A restaurant recommender system based on user preference and location in mobile environment. In *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 55–60. IEEE, 2016.

[22] Korab Rrmoku, Besnik Selimi, and Lule Ahmedi. Application of trust in recommender systems—utilizing naive bayes classifier. *Computation*, 10(1):6, 2022.

[23] Jin An Xu and Kenji Araki. A svm-based personal recommendation system for tv programs. In *2006 12th International Multi-Media Modelling Conference*, pages 4–pp. IEEE, 2006.

[24] Gunnar Schröder, Maik Thiele, and Wolfgang Lehner. Setting goals and choosing metrics for recommender system evaluations. In *UCERSTI2 workshop at the 5th ACM conference on recommender systems, Chicago, USA*, volume 23, page 53, 2011.

[25] Moussa Taifi PhD. "mrr vs map vs ndcg: Rank-aware evaluation metrics and when to use them". `https://medium.com/swlh/rank-aware-recsys-evaluation-metrics-5191bba16832`, 2019. Online; accessed 26 May 2023.

[26] Marilyne Latour. Analyse de sentiments dans les textes économiques: un exemple d'application chez reportlinker. In *CORIA*, 2021.

[27] Larbes Abdelkrim, Amrani Okba, and Khantoul Bilel. Une approche deep learning pour l'analyse des sentiments. 2021.

[28] Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225, 2014.

[29] Karsten Tymann, Matthias Lutz, Patrick Palsbröker, and Carsten Gips. Gervader-a german adaptation of the vader sentiment analysis tool for social media texts. In *LWDA*, pages 178–189, 2019.

[30] NESRINE GUERRI. Findtome: Un système de recommandation collaboratif de lieux. 2022.

[31] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1245–1254, 2017.

[32] Shuhui Jiang, Xueming Qian, Jialie Shen, Yun Fu, and Tao Mei. Author topic model-based collaborative filtering for personalized poi recommendations. *IEEE transactions on multimedia*, 17(6):907–918, 2015.

[33] Wei Wang, Junyang Chen, Jinzhong Wang, Junxin Chen, Jinquan Liu, and Zhiguo Gong. Trust-enhanced collaborative filtering for personalized point of interests recommendation. *IEEE Transactions on Industrial Informatics*, 16(9):6124–6132, 2019.

[34] Sadok Ben Yahia and Imen Ben Sassi. Poi based serendipitous recommender algorithm.

[35] Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wen-Ning Kuo, and Vincent S Tseng. Urban point-of-interest recommendation by mining user check-in behaviors. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, pages 63–70, 2012.

[36] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 325–334, 2011.

[37] Wei Wang, Junyang Chen, Jinzhong Wang, Junxin Chen, and Zhiguo Gong. Geography-aware inductive matrix completion for personalized point-of-interest recommendation in smart cities. *IEEE Internet of Things Journal*, 7(5):4361–4370, 2019.

[38] Nabiha Asghar. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*, 2016.

[39] Sumedh Sawant and Gina Pai. Yelp food recommendation system, 2013.

[40] Jinyin Chen, Xiang Lin, Yangyang Wu, Yixian Chen, Haibin Zheng, Mengmeng Su, Shanqing Yu, and Zhongyuan Ruan. Double layered recommendation algorithm based on fast density clustering: Case study on yelp social networks dataset. In *2017 International Workshop on Complex Systems and Networks (IWCSN)*, pages 242–252. IEEE, 2017.

[41] Jason Ting and Swaroop Indra Ramaswamy. Yelp recommendation system. 2013.

[42] Jia Le Xu and Yingran Xu. Recommendation system using yelp data, 2021.

[43] Jiancong Sun. *NLP Analysis and Recommendation System for Yelp*. University of California, Los Angeles, 2020.

[44] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. Deep learning recommendation model for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091*, 2019.

[45] Xinxi Wang and Ye Wang. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 627–636, 2014.

[46] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 world wide web conference*, pages 167–176, 2018.

[47] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. *Advances in neural information processing systems*, 26, 2013.

[48] Libo Zhang, Tiejian Luo, Fei Zhang, and Yanjun Wu. A recommendation model based on deep neural network. *IEEE Access*, 6:9454–9463, 2018.

[49] Jing Sun, Yun Xiong, Yangyong Zhu, Junming Liu, Chu Guan, and Hui Xiong. Multi-source information fusion for personalized restaurant recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 983–986, 2015.

[50] Alif Azhar Fakhri, ZKA Baizal, and Erwin Budi Setiawan. Restaurant recommender system using user-based collaborative filtering approach: a case study at bandung raya region. In *Journal of Physics: Conference Series*, volume 1192, page 012023. IOP Publishing, 2019.

[51] Achmad Arif Munaji and Andi Wahju Rahardjo Emanuel. Restaurant recommendation system based on user ratings with collaborative filtering. In *IOP Conference Series: Materials Science and Engineering*, volume 1077, page 012026. IOP Publishing, 2021.

[52] Qusai Y Shambour, Ahmad Adel Abu-Shareha, and Mosleh M Abualhaj. A hotel recommender system based on multi-criteria collaborative filtering. *Information Technology and Control*, 51(2):390–402, 2022.

[53] Ya-Han Hu, Pei-Ju Lee, Kuanchin Chen, J Michael Tarn, and Duyen-Vi Dang. Hotel recommendation system based on review and context information: a collaborative filtering appro. In *PACIS*, page 221, 2016.

[54] KV Daya Sagar, PSG Arunasri, Sridevi Sakamuri, J Kavitha, and DBK Kamesh. Collaborative filtering and regression techniques based location travel recommender system based on social media reviews data due to the covid-19 pandemic. In *IOP Conference Series: Materials Science and Engineering*, volume 981, page 022009. IOP Publishing, 2020.

[55] Anant Gupta and Kuldeep Singh. Location based personalized restaurant recommendation system for mobile environments. In *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 507–511. IEEE, 2013.

[56] Chung-Hua Chu and Se-Hsien Wu. A chinese restaurant recommendation system based on mobile context-aware services. In *2013 IEEE 14th International Conference on Mobile Data Management*, volume 2, pages 116–118. IEEE, 2013.

[57] Cheryl Ayu Melyani. Hotel recommendation system with content-based filtering approach (case study: Hotel in yogyakarta on nusatrip website). *J Statistika: Jurnal Ilmiah Teori dan Aplikasi Statistika*, 15(1), 2022.

[58] Agung Muliawan, Tessy Badriyah, and Iwan Syarif. Membangun sistem rekomendasi hotel dengan content based filtering menggunakan k-nearest neighbor dan haversine formula. *Technomedia Journal*, 7(2 October):231–247, 2022.

[59] Kristian Wahyudi, Johanes Latupapua, Ritchie Chandra, and Abba Suganda Girsang. Hotel content-based recommendation system. In *Journal of Physics: Conference Series*, volume 1485, page 012017. IOP Publishing, 2020.

[60] M Govindarajan. Sentiment analysis of restaurant reviews using hybrid classification method. *International Journal of Soft Computing and Artificial Intelligence*, 2(1):17–23, 2014.

[61] Realdo Dias, SC Ng, and Norriza Hussin. A hybrid framework for restaurant recommender system. *International Journal of Scientific Engineering and Technology*, 5(12):546–548, 2016.

[62] Wei-Ta Chu and Ya-Lun Tsai. A hybrid recommendation system considering visual information for predicting favorite restaurants. *World Wide Web*, 20:1313–1331, 2017.

[63] Ashkan Ebadi and Adam Krzyzak. A hybrid multi-criteria hotel recommender system using explicit and implicit feedbacks. *International Journal of Computer and Information Engineering*, 10(8):1450–1458, 2016.

[64] Huang Kung-Hsiang, Fu Yi-Fu, Lee Yi-Ting, Lee Tzong-Hann, Chan Yao-Chun, Lee Yi-Hui, and Lin Shou-De. A-ha: A hybrid approach for hotel recommendation. In *Proceedings of the Workshop on ACM Recommender Systems Challenge*, pages 1–5, 2019.

[65] S Pandya, J Shah, N Joshi, H Ghayvat, SC Mukhopadhyay, and MH Yap. A novel hybrid based recommendation system based on clustering and association mining. In *2016 10th international conference on sensing technology (ICST)*, pages 1–6. IEEE, 2016.

[66] Shaowen Peng, Kazunari Sugiyama, and Tsunenori Mine. Less is more: Reweighting important spectral graph features for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1273–1282, 2022.

[67] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 726–735, 2021.

[68] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pages 70–79, 2022.

[69] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1791–1800, 2021.

[70] Jun Hu, Shengsheng Qian, Quan Fang, and Changsheng Xu. Mgdcf: Distance learning via markov graph diffusion for neural collaborative filtering. *arXiv preprint arXiv:2204.02338*, 2022.

[71] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

[72] Nesrine Zemirli. *Modèle d'accès personnalisé à l'information basé sur les Diagrammes d'Influence intégrant un profil utilisateur évolutif.* PhD thesis, Université Paul Sabatier-Toulouse III, 2008.

[73] Lynda Tamine and Wahiba Bahsoun. Définition d'un profil multidimensionnel de l'utilisateur: vers une technique basée sur l'interaction entre dimensions. In *Actes de la*

Conférence francophone en Recherche d'Information et Applications (CORIA 2006), pages 225–236, 2006.

[74] Dimitre Kostadinov. *Personnalisation de l'information: une approche de gestion de profils et de reformulation de requêtes*. PhD thesis, Université de Versailles-Saint Quentin en Yvelines, 2007.

[75] Yelena Mejova. Sentiment analysis: An overview. *University of Iowa, Computer Science Department*, 2009.

[76] Saqib Alam and Nianmin Yao. The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis. *Computational and Mathematical Organization Theory*, 25:319–335, 2019.

[77] Manish Todi. Sentiment analysis using the vader library. https://www.linkedin.com/pulse/sentiment-analysis-using-vader-library-manish-todi?fbclid=IwAR1Llgod-_irE5yNlK2_WbNs96uXYNHVXuc3ljAX7xutd_2lB297DboQ7Ik, 2019. Online; accessed 27 May 2023.

[78] Subbu Kannan, Vairaprakash Gurusamy, S Vijayarani, J Ilamathi, Ms Nithya, S Kannan, and V Gurusamy. Preprocessing techniques for text mining. *International Journal of Computer Science & Communication Networks*, 5(1):7–16, 2014.

[79] S Vijayarani, Ms J Ilamathi, Ms Nithya, et al. Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, 5(1):7–16, 2015.

[80] Divya Khyani, BS Siddhartha, NM Niveditha, and BM Divya. An interpretation of lemmatization and stemming in natural language processing. *Journal of University of Shanghai for Science and Technology*, 22(10):350–357, 2021.

[81] Louis Hickman, Stuti Thapa, Louis Tay, Mengyang Cao, and Padmini Srinivasan. Text preprocessing for text mining in organizational research: Review and recommendations. *Organizational Research Methods*, 25(1):114–146, 2022.

[82] Jupyter. https://jupyter.org/, 2023. Online; accessed 10 June 2023.

[83] @ThePSF. Python. https://www.python.org/doc/essays/blurb/. Online; accessed 10 June 2023.

[84] Numpy : the most used python library in data science. https://datascientest.com/en/numpy-the-python-library-in-data-science, 2023. Online; accessed 10 June 2023.

[85] Shubhang Agrawal. Introduction to pandas. https://medium.com/analytics-vidhya/introduction-to-pandas-90b75a5c2278, 2020. Online; accessed 10 June 2023.

[86] Aayushi Johari. Python matplotlib guide - learn matplotlib library with examples. https://medium.com/edureka/python-matplotlib-tutorial-15d148a7bfee, 2017. Online; accessed 10 June 2023.

[87] Vibhav Sharma. Starting with matplotlib and seaborn ! https://medium.datadriveninvestor.com/starting-with-matplotlib-and-seaborn-cba16c7beabf, 2021. Online; accessed 10 June 2023.

[88] Pema Grg. Getting started with nlp using nltk. https://becominghuman.ai/nlp-for-beginners-using-nltk-f58ec22005cd, 2018. Online; accessed 10 June 2023.

[89] Gensim: The python library for topic modelling. https://datascientest.com/gensim-tout-savoir#:~:text=Gensim%20est%20une%20biblioth%C3%A8que%20Open,est%20la%20mod%C3%A9lisation%20de%20sujet., 2023. Online; accessed 10 June 2023.

[90] Ekaba Bisong. Building machine learning and deep learning models on google cloud platform: A comprehensive guide for beginners. https://www.researchgate.net/publication/336113323_Building_Machine_Learning_and_Deep_Learning_Models_on_Google_Cloud_Platform_A_Comprehensive_Guide_for_Beginners, 2019. Online; accessed 10 June 2023.

[91] Lysandre Debut. Hugging face: State-of-the-art natural language processing in ten lines of tensorflow 2.0. https://medium.com/tensorflow/using-tensorflow-2-for-state-of-the-art-natural-language-processing-102445cda54a, 2019. Online; accessed 10 June 2023.

[92] Sanyam Bhutani. Pytorch basics in 4 minutes. https://medium.com/hackernoon/pytorch-basics-9c1c627cd0d2, 2018. Online; accessed 10 June 2023.

[93] Yelp dataset. https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset, 2023. Online; accessed 10 June 2023.

.