Personalised Destination Suggestions via Data-Driven Recommendation Systems

Akshat Tokas
2022056

Dhruv Kantroo
2022167

***Abstract*— This paper investigates the development and application of a machine learning-based travel recommendation system.Furthermore, the selection of destinations is significantly impacted by the increasing availability of geographic data. User-based collaborative filtering was used in the development of the recommendation system covered in this report. To suggest the best places to visit, this model produces predictive values. Additionally, k-means clustering is employed to group similar travel destinations based on geographic features, enhancing the accuracy and relevance of the recommendations provided by the system.**

Keywords—collaborative filtering, travel recommendation, geographic data, k means clustering

I. Motivation

By providing tailored recommendations based on each traveler's individual needs and tastes, a travel recommendation system seeks to improve experiences for visitors. It simplifies travel planning, encourages exploration, maximizes resource allocation, and places a high priority on individual safety by employing data-driven algorithms. These days, most people plan their holidays using online services. Travel recommendation systems are thus made to sort through the enormous volume of data and find the best places for consumers to visit. In the end, it gives visitors the freedom to go out on rewarding adventures that align with their goals and passions.

II. Literature Review

A. Historical Developments of Recommendation Systems

During the 1990s, the research emphasis shifted towards prediction models for product ratings, leading to the inception of recommendation systems. These systems aimed to match consumers with suitable goods and services amidst the burgeoning data landscape.

B. Exploratory Data Analysis:

EDA is an approach used in analyzing datasets to summarize their main characteristics, often using graphical and statistical techniques.

C. Classification of Recommendation Systems:

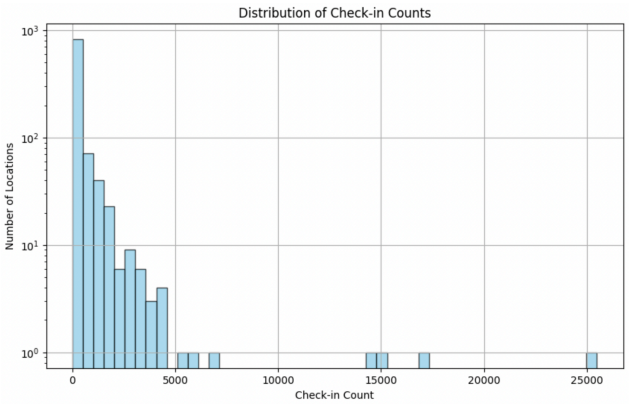
* Content-based Filtering: This approach links user characteristics with items likely to meet their preferences, leveraging the similarity between user profiles and item attributes.

* Collaborative Filtering: Grounded in the assumption that users with similar tastes and behaviours will prefer similar items, collaborative filtering predicts user preferences based on historical interactions.

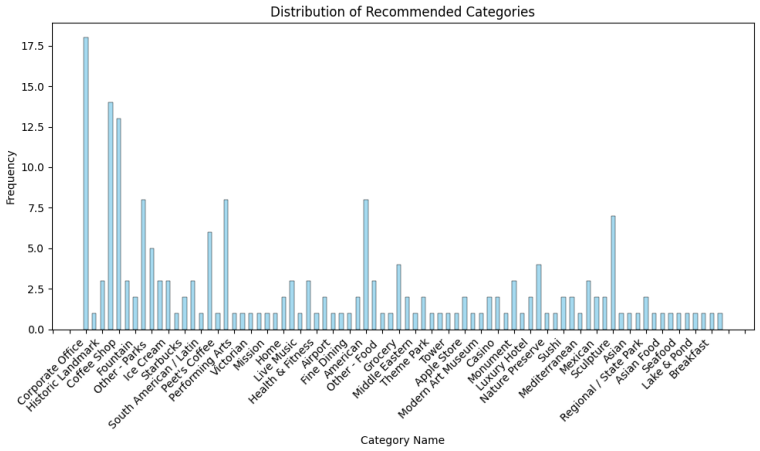
* Knowledge-based Systems: Particularly beneficial for recommending rare or specialized items, knowledge-based systems require extensive domain expertise and can be resource-intensive to implement.

* Hybrid Systems: Integrating the strengths of multiple techniques, hybrid systems aim to mitigate the limitations of individual approaches, offering more robust recommendations.

III. Datasets



The dataset we have used was gathered from social media Gowalla via towardsdatascience.com which consists of 36,001,959 check-ins by 407,533 users in 2,724,891 POIs. It includes user id, place id, date-time, geography, categorized destinations and check-in information. We have selected subsets of data points from the dataset to reduce the computing time of the data.



IV. Proposed Architecture

A. Collaborative Filtering using SVD

Collaborative Filtering is implemented using Matrix Factorisation involving Singular Value Decomposition (SVD). It decomposes the rating matrix into three matrices and then performs the following operation:

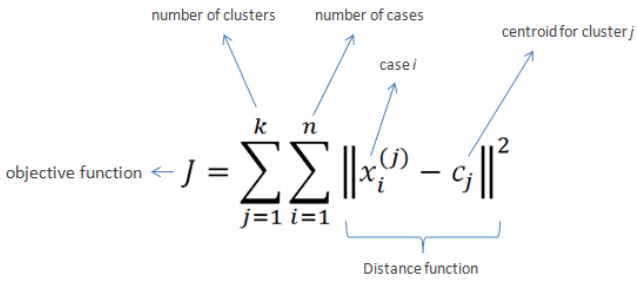
$$A_{(mxn)} = U * Sigma * V$$

A: User's rating matrix
U : User's feature matrix
Sigma: Singular matrix representing weights
V : POI feature matrix.

Once decomposed, the original matrix is reconstructed by multiplying these matrices, capturing the interaction between users and items. After obtaining the factorized matrices, predictions for missing or unobserved ratings can be made by taking dot products of user and item latent vectors.
For a given user, recommendations are generated by selecting items with the highest predicted ratings that the user has not already rated.
The reason for using Matrix factorization is that we can capture latent factors or features that represent user preferences and item characteristics.

B. K-Means Clustering

The K-means clustering algorithm is applied to group the places (POIs) into clusters based on their latitude and longitude coordinates. Each place is assigned to the cluster with the nearest centroid. User profiling is then used to find the cluster of the user's profile. Finally, places within the user's cluster are recommended, and their categories are visualized through a table and a histogram to understand the distribution of recommended categories.



$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

It aims to find the centroids that minimizes the objective function J.

C. Cosine Similarity and Count Vectorization

Here, the algorithm calculates the cosine similarity between points of interest (POIs) based on certain features extracted from two subsets of data. Specific features related to each POI are concatenated into a single string. Using the Count Vectorize method, this concatenated string is transformed into a matrix of token counts.

$$similarity(A, B) = cos(\theta) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sum_{i=1}^{n} B_i^2}.$$

Cosine similarity is then computed between each pair of POIs using the cosine_similarity function. The script identifies the most similar POIs to a target POI by sorting the similarity scores. Then the names and locations of the top POIs to the target POI are printed.

Count vectorizer with cosine similarity method is used as it efficiently converts text data into a numerical format suitable for computation, allowing for fast processing of large datasets and it is scalable and can handle large datasets with millions of documents (POIs) and thousands of features.

D. Exploratory Data Analysis

Calculate summary statistics such as mean, median, minimum, and maximum of the check-in counts and plot various tables & graphs to understand the distribution of popularity across different places.
We can establish a direct proportionality between the number of check-ins and the popularity of destinations using this method leading to a static model.

V. Experimental settings:

The dataset was originally taken from Gowalla through towardsdatascience.com but in order to create functioning models we have filtered the datasets.

The models make use of csv files containing various columns including latitude, longitude, place id, user id, no. of checkins and date-time.

Since the dataset was huge so in order to reduce the computing time of the data:

We use random samples from the dataset.
Check-ins: 1,000,000
Users: 5000
POIs: 3000

VI. Results

A. Collaborative Filtering using SVD

Outputs recommended POIs and their predicted ratings. Recommendations are personalized to each user, leveraging latent factors to predict ratings for items not yet rated by the user. Offers recommendations based on historical user-item interactions.
Algorithm involves taking into account more user-place interactions therefore results are more relevant to the user.

B. K-Means Clustering

Outputs some recommendations based on similarity in feature space.
Recommendations are based on clustering similar items together. It tends to recommend items that share common features or attributes, irrespective of individual user preferences.
The algorithm is less user centric therefore results are comparatively less relevant.

C. Cosine Similarity and Count Vectorization

Outputs a select number of similar recommended POIs and their descriptions, based on a given POI. It does not take into account any user preferences or user features and is purely based on similarity of place features.
The algorithm is less user centric, therefore results are comparatively less relevant.

VII. Comparing the models

CF using SVD excels in capturing latent patterns in user-item interactions, handling sparse and high-dimensional data efficiently. It relies heavily on existing user-item interaction data, potentially leading to biased recommendations in scenarios with limited diversity in user preferences.

K-means clustering groups similar items based on features, offering simplicity and interpretability. It's effective for segmentation tasks and can reveal patterns in data. However, its performance relies on choosing the right number of clusters and quality of features.

Cosine similarity is robust to sparse data and diverse feature types, suitable for scenarios where item characteristics are vital. However, it may overlook subtle item relationships and struggle with capturing contextual relevance in recommendations.

The most suitable model depends on the specific characteristics of the dataset and the requirements of the recommendation task.
SVD tends to perform well when user-item interactions are abundant and diverse.
K-means clustering is effective for uncovering item groupings and segmenting users based on preferences.
Cosine similarity excels in scenarios where item features play a significant role in recommendation relevance.
According to our model and dataset, Clustering excels in providing the most accurate predictions but requires geographic data as input from user.

VIII. Contributions

Akshat Tokas: K-Means Clustering, Cosine Similarity, Exploratory Data Analysis
Dhruv Kantroo: Collaborative Filtering, Cosine Similarity, Dataset Filtering

References / Acknowledgements

https://en.wikipedia.org/wiki/Singular_value_decomposition
https://towardsdatascience.com/where-to-travel-next-a-guide-to-building-a-recommender-system-for-pois-5116adde6db
https://memgraph.com/blog/cosine-similarity-python-scikit-learn
https://www.saedsayad.com/clustering_kmeans.htm
https://realpython.com/build-recommendation-engine-collaborative-filtering
Inspired and help taken from Analysis and Prediction of Movie Recommendations based on User's Personal Preference provided by Sir.