PERSONALIZED TRAVEL POI RECOMMENDATIONS USING GEOGRAPHICAL & SOCIAL CHARACTERISTICS

Aravind Rajeev Nair, Naveen Sethuraman, Manikant Thatipalli, Prashanti M. Toram

ABSTRACT

This project aims to revolutionize travel recommendations by employing collaborative filtering (CF) methods including K-nearest neighbors (KNN), Singular Value Decomposition (SVD), and Non-Negative Matrix Factorization (NMF). Additionally, a specialized Geo-Social Location Recommender (GSLR) and a Two-Tower Neural Recommender are utilized. The dataset, sourced from Gowalla on Kaggle, contains crucial information such as user profiles, check-ins, locations, and friendships. The primary objective is to enhance travel decision-making, providing personalized recommendations. The technical approach involves meticulous data preprocessing, constructing CF features, and leveraging GSLR and Two Tower model for enhanced recommendations Model validation includes metrics such as RMSE, MAE, and precision and recall (Precision@K, Recall@K, F1@K) to assess accuracy. Anticipated challenges include sparse datasets and uncertainties in evaluation metrics, with contingency plans involving alternative weighting methods. Ultimately, the project aims to redefine travel decision-making, delivering personalized and memorable experiences.

1 INTRODUCTION

This ambitious project seeks to transform the travel industry by introducing an innovative recommendation system that tailors travel destinations to individual preferences. The overarching goal is to revolutionize the travel experience, augment customer satisfaction, and enhance decision-making within the tourism sector. The project employs advanced machine learning models, specifically collaborative filtering (CF) approaches such as K-nearest neighbors (KNN), Singular Value Decomposition (SVD), Non-Negative Matrix Factorization (NMF), an off-the-hood Geo-Social Location Recommender, and a Two-Tower Neural Recommender.

2 DATA COLLECTION AND PREPROCESSING

2.1 DATA SOURCE

The data for this project originates from Gowalla, a location-based social media app operational from 2007 to 2012. The datasets for analysis are sourced from Gowalla via Kaggle, encompassing essential information such as user profiles, check-in histories, locations, and user friendships. The data is comprised of four tables: user friendships, user features, location features, and user check-ins at various locations. These datasets provide the foundation for understanding user behaviors, preferences, and social interactions. To streamline the analysis, the last three tables were merged. It should also be noted that the user data is anonymized. By leveraging this data, the project aims to extract meaningful insights that contribute to the creation of a personalized travel recommendation system and generate location recommendations for users by leveraging collaborative filtering, geospatial location recommendation, and two-tower models.

2.2 PREPROCESSING

The data contained many users with ten or fewer location check-ins. Due to computational limitations and limited feasible accuracy in prediction, these users were removed from the dataset. Likewise, users with more than one hundred visits were also excluded.

Furthermore, the scope of predictions was narrowed down to locations within New York City. This would enable more accurate predictions and further reduce the computational requirements.

2.3 Exploratory Data Analysis (EDA)

After the preprocessing phase, the dataset was refined to approximately 142,000 data points, capturing the interactions between users and locations. This curated dataset encompasses a diverse range of information from around 21,000 distinct locations and involves interactions with a user base of 5,000 individuals.

2.3.1 Geographic Insights

A notable discovery from the EDA was the prevalence of visits to specific landmarks, indicating key points of interest. Notably, airports, Times Square, and Grand Central emerged as frequently visited locations within the dataset. This observation suggests that these landmarks hold substantial significance in the user check-in patterns and may be focal points for travel or social activities.

2.3.2 Popular Location Categories

The EDA unveiled insights into the distribution of locations and location categories based on user check-ins. The most popular categories included offices, airports, restaurants, and parks. The most popular individual locations included the airports, Times Square, Grand Central Terminal, and Madison Square Garden. Understanding these popular categories is crucial for tailoring recommendations, as they reflect the diverse interests and preferences of the user base. This insight allows for a more nuanced understanding of user behaviors within specific contexts.

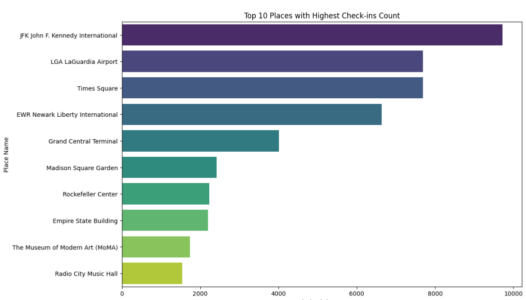


Figure 1: Top 10 visited locations

2.3.3 User Activity Metrics

Highly active users, characterized by those with over 10,000 check-ins, were identified as a distinct subset within the dataset. As per Figure 2 These users demonstrated a remarkable level of engagement, suggesting a deep involvement with the platform. Moreover, the concept of user-made groupings known as ”trips” emerged, highlighting users’ proclivity to organize and categorize their check-ins. Understanding these user-initiated groupings provides valuable context for interpreting the sequential nature of check-ins and potential patterns in user behavior.

2.3.4 T-SNE Clustering Analysis

An attempt to discern patterns based on friend count using t-SNE clustering as per Figure 3 did not yield significant observations. This outcome suggests that, at least based on the available features, the user-friendship dynamic did not exhibit clear clustering patterns. While friend count may not be a predominant factor influencing check-in behaviors, this insight guides the understanding of social dynamics within the user base.

In summary, the EDA phase uncovered valuable insights into user behavior, preferences, and geographic trends within the dataset. These findings lay the foundation for subsequent modeling

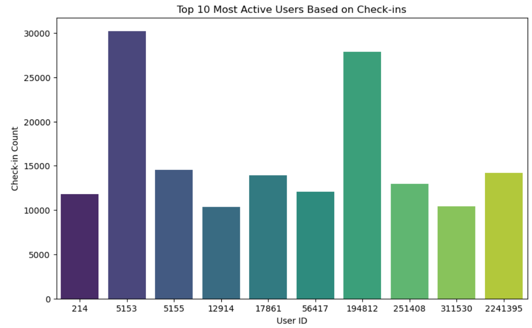


Figure 2: Top 10 active users (check-in count)

approaches, providing a nuanced understanding of the diverse interactions between users and locations. The identified patterns and characteristics contribute to the formulation of targeted and effective recommendation strategies, enhancing the overall success of the recommendation system.

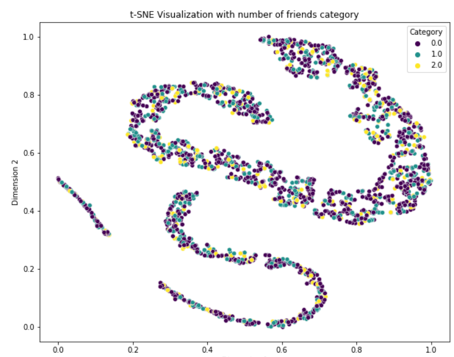


Figure 3: T-SNE Clustering

3 Model Description and Training

3.1 Collaborative Filtering

We first took a standard approach using collaborative filtering. This method entails creating a matrix enumerating the count of check-ins of each user at each location. In lieu of a check-in count, we calculated a rating value to keep values bounded from zero to ten. This was calculated as $10 \times \tanh\left(\frac{x-\text{locmin}}{\text{locmax}-\text{locmin}}\right)$, using the hyperbolic tangent to emphasize differences among lower check-in counts, as check-in counts by user to any given location are heavily left-skewed.

As most users would not have visited most locations, this matrix is expectedly sparse. As such, it may be used to predict unfilled values. Collaborative filtering allows for three approaches in making these predictions:

1. User-User filtering, where the users are taken as entries and the similarity of user behavior is used to make recommendations.
2. Item-Item filtering, considers the similarity of items to items with which the user has interacted to make predictions.
3. User-Item filtering, which predicts items for a user based on their past interactions and preferences.

For the purposes of this analysis, we chose to use user-user collaborative filtering.

3.1.1 Model 1: K-Nearest Neighbors

In order to make user rating predictions from the collaborative filtering matrix, we employed three different models, the first of which is a K-Nearest Neighbor Model. The model takes a user for whom we wish to predict ratings and identifies the k most similar users (as measured by cosine distance). Their ratings for each location are then used to predict the selected user's ratings by taking a weighted average.

For the KNN model, hyperparameters were iteratively tested during the training process to optimize their performance. We employed Grid search with CV of 5 folds to evaluate the performance of different hyperparameter combinations through measures such as RMSE and MSE more reliably and to mitigate the risk of over-fitting to a specific train-test split. This involved experimenting with different values for parameters such as the number of neighbors (K) in KNN, as this K gives the list of similar users for the user-user similarity approach. We tried with K values ranging from 5 to 50 and explored different ways of measuring similarity metrics using cosine and Pearson based methods. For weightage methods, we explored 'uniform' and 'distance' based types to assign weightage to each neighbor.

```
param_grid = {'n_neighbors': [5, 10, 20,30,40,50],
              'metric': ['cosine', 'euclidean'],
              'weights': ['uniform', 'distance']}
```

3.1.2 Model 2: Singular Value Decomposition (SVD)

Singular Value Decomposition is a matrix factorization technique that decomposes the user-item interaction matrix into three matrices: U (user-feature matrix), Σ (diagonal matrix of singular values), and V (item-feature matrix). The product of these matrices approximates the original matrix. In the context of collaborative filtering, SVD is applied to predict missing values in the user-item matrix. The latent factors captured in the U and V matrices represent underlying patterns in user preferences and item characteristics.

The SVD model in this analysis involves selecting a certain number of latent factors to represent the interactions between users and locations. By minimizing the difference between the predicted and actual ratings, the model learns the latent factors that best explain the observed user-item interactions. This approach allows for capturing complex patterns and dependencies in the data, contributing to accurate predictions for user ratings at different locations.

For this model, we explored a range of values for the number of latent factors (K), considering values from 5 to 50, and varied the learning rates from 0.001 to 0.05, while progressively increasing the number of iterations(n_epochs) up to 50. The number of latent factors influences the dimensionality of the latent space while learning rates and iterations control the optimization process. The learning rate determines the step size during optimization, and the number of iterations controls how many times the algorithm processes the entire dataset.

```
param_grid = {'n_factors': [5,10,20,30,40,50],
              'n_epochs': [10,20,30,40,50],
              'lr_all': [0.001,0.05],
              'reg_all': [0.02, 0.1]}
```

3.1.3 Model 3: Non-negative Matrix Factorization (NMF)

Non-negative Matrix Factorization is another matrix factorization technique that decomposes the user-item matrix into two non-negative matrices: W (user-feature matrix) and H (item-feature matrix). The non-negativity constraint makes the factors interpretable and enhances the model's ability to capture meaningful patterns in the data. NMF is particularly useful when dealing with non-negative data, such as counts or ratings.

In the context of location recommendation, NMF aims to discover latent features that explain the observed user check-ins. By iteratively updating the user and item matrices to minimize the reconstruction error, NMF uncovers meaningful representations of users and locations. The resulting factorized matrices can then be used to predict missing values in the original matrix, providing personalized location recommendations for users.

Very similar to SVD, for the NMF model, we explored a range of values for the number of latent factors (K), considering values from 5 to 50, and varied the learning rates from 0.001 to 0.05, while progressively increasing the number of iterations(n_epochs) upto 50. The number of latent factors influences the dimensionality of the latent space while learning rates and iterations control the optimization process. The learning rate determines the step size during optimization, and the number of iterations controls how many times the algorithm processes the entire dataset.

```
param_grid = {'n_factors': [5,10,20,30,40,50],
              'n_epochs': [10,20,30,40,50],
              'lr_all': [0.001,0.05],
              'reg_all': [0.02, 0.1]}
```

3.2 MODEL 4: GEO-SOCIAL LOCATION RECOMMENDER

The proposed model incorporates both geographical and social features to enhance the accuracy of location recommendations. The key components of the model involve collaborative filtering, social collaborative filtering, and personalized geographical influence modeling through kernel density estimation. Let's break down how geo and social features are included and combined:

SOCIAL COLLABORATIVE FILTERING

SCF extends the collaborative filtering paradigm by incorporating social influence. Users often seek recommendations from friends, and SCF aims to capture and utilize this social dimension. The model emphasizes the shared social connections among users as a crucial factor influencing their preferences. The social similarity between users is computed based on their shared social friendships, and this information is integrated into the recommendation process.

$$\text{SocSim}(u_i, u_k) = \frac{|F(u_i) \cap F(u_k)|}{|F(u_i) \cup F(u_k)|} \tag{1}$$

where F(ui) denotes the set of users having social friendships with user ui.

the rating of ui to unvisited location lj , denoted as $\hat{r_{i,j}}$,

$$\hat{r}_{i,j} = \frac{\sum_{u_k \in U \wedge k \neq i} \text{SocSim}(u_i, u_k) \cdot r_{k,j}}{\sum_{u_k \in U \wedge k \neq i} \text{SocSim}(u_i, u_k)} \tag{2}$$

$$\hat{p}_{i,j} = \frac{\hat{r}_{i,j}}{\max_{l_j \in L - L_i} \{\hat{r}_{i,j}\}} \tag{3}$$

PERSONALIZED GEOGRAPHICAL INFLUENCE MODELING

The geographical influence of locations is crucial in the context of location recommendations. GSLR incorporates the personalized geographical influence of locations through kernel density estimation. The model captures the personalized distribution of distances between every pair of locations visited by a user. This is achieved through two steps: distance sample collection and distance distribution estimation. Distance Sample Collection: The model acquires a sample for a user by computing the distance between every pair of locations that the user has checked in. For a cold-start user with only one check-in, the distance between the visited location and the user's residence is employed as the sample. Distance Distribution Estimation: The kernel density estimation process is used to estimate the distribution of distances. A kernel function is applied to derive the density estimator, capturing the personalized influence of geographical distance.

$$f(d_{ij}) = \frac{1}{|D|h} \sum_{d' \in D} K\left(\frac{d_{ij} - d'}{h}\right) \tag{4}$$

Then, the probability of u_i visiting a new location l_j can be obtained by taking the mean probability as follows

$$p(l_j|L_i) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(d_{ij}) \tag{5}$$

Then, the normalized probability from social similarity and geographical similarity is combined with the probabilities as mentioned in above equations 3 and 5 by

$$s_{i,j} = \frac{\hat{p}_{i,j} + p(l_j|L_i)}{2} \tag{6}$$

To assess our model's performance, we employed precision, recall, and F1-score as evaluation metrics. We conducted a comprehensive analysis by calculating precision and recall for varying values of k, ranging from 1 to 10. The resulting plot visually illustrates the dynamic interplay between precision and recall across different k values.

3.3 Model 5 : Two Tower Neural Model

The two-tower recommendation model is a sophisticated architecture widely employed in modern recommendation systems, offering a nuanced approach to understanding and leveraging user-item interactions to provide personalized product recommendations. At its core, our model comprises two distinct towers designed to process user and location features independently, with each tower focusing on learning embeddings that capture the intrinsic characteristics of users and items. The user tower, for instance, delves into various user features, encompassing pins, check-ins, and trips count, while the location-based tower processes features such as number of checkins and pictures captured. During the training phase, the model refines these embeddings based on observed interactions between users and locations. The essence of the model lies in its ability to discern patterns and relationships within the data, effectively encoding user preferences and location attributes in a lower-dimensional space. As a result, the learned embeddings enable the model to generalize from observed interactions and make informed predictions for unseen user-location pairs.Further, the architecture facilitates a collaborative filtering approach, wherein the model implicitly learns similarities between users and items. Users who exhibit similar behaviors or preferences end up with similar embeddings, and likewise for locations. This collaborative nature allows the model to make recommendations based on the preferences of users who share similar characteristics in the embedding space. Covariates in our system are often believed to have a low-dimensional intrinsic representation, and we tried to select only a few important ones. Given a typical recommender system with user covariates $x_u \in \mathbb{R}^{D_u}$ and item covariates $x_i \in \mathbb{R}^{D_i}$, the two-tower model can be written as:

$$R(x_u, \tilde{x}_i) = \langle f(x_u), \tilde{f}(\tilde{x}_i)\rangle \tag{7}$$

where f : $R^{D_u} \rightarrow R^p$ and f : $R^{D_i} \rightarrow R^p$ are two deep neural networks mapping xu and x˜i into the same p dimensional embedded space. The recommendation mechanism of the two-tower model is based on the dot product of f(x_u)) and $\tilde{f}(\tilde{x}_i)$, as illustrated in Figure 4 below

The cost function of optimizing the two-tower model can be organized as

$$\min_{f,\tilde{f}} \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} (k_{ui} - \langle f(x_u), \tilde{f}(x_i)\rangle)^2 + \lambda\left(J(f) + J(\tilde{f})\right) \tag{8}$$

where J(·) can be the L1-norm or L2-norm penalty for preventing the deep neural network from overfitting. Moreover, it is interesting to note that (2) reduces to the classical SVD-based collaborative filtering method, when x_u and $\tilde{x_i}$ contain only one-hot encodings for users and items.

Our two-tower architecture comprises distinct branches dedicated to processing user and item features independently. The user tower extracts and learns embeddings representing user characteristics, while the item tower focuses on capturing unique features associated with each item. Additionally, we introduced innovative design by developing an ensemble of two-tower models with

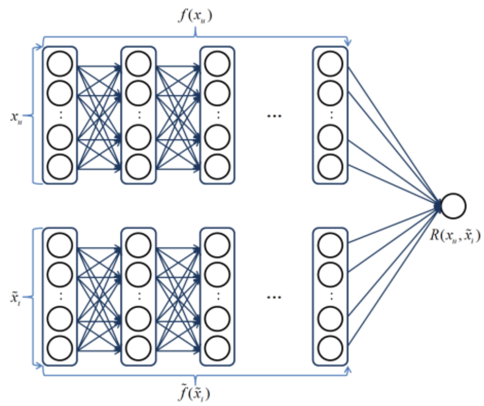


Figure 4: NN structure of two-tower model

different architectures or hyperparameters to combine diverse perspectives and enhance the overall recommendation performance.

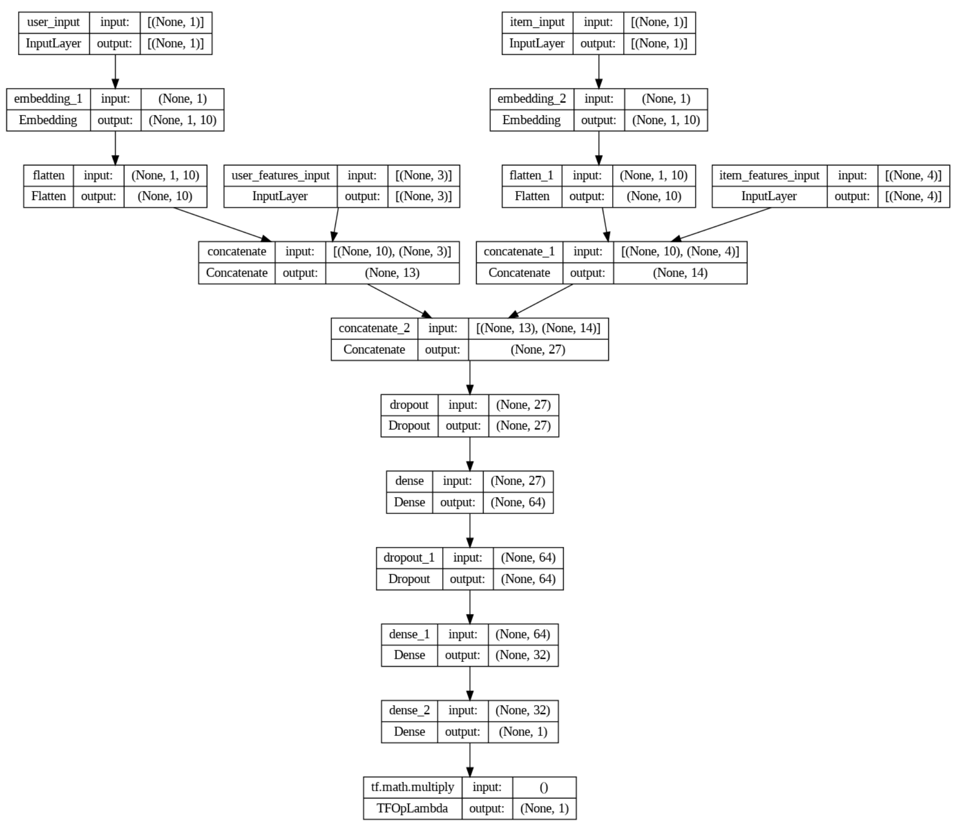


Figure 5: Best Model Architecture

We systematically explored the hyperparameter space to optimize the model's performance. We considered tuning to include the learning rate (learning_rate) from 0.0001 to 0.1, the number of epochs (epochs) from 5 to 50 to progressively check the best performance, then the batch size (batch_size) as 16, the number of embedding dimensions (embedding_dim) as 10, and the number of hidden units in dense layers (hidden_units) from 32, 64 and 128 respectively.

A glimpse of our best-performing architectures can be seen below in Figure 5. The user and location input layers have been flattened to start with 10 layers, then the corresponding 3 and 4 dimensions have been concatenated to the user and location to make them 13 and 14, which in turn gets concatenated at layer 2 to become 27 dimension vectors. We now employ multiple layers of hidden and dropout layers to have a final fully connected layer of 32 dimensions. In the end, we used a sigmoid function to get an output of the predicted rating (1 dimension), which when multiplied by 10 rescales it back to the original rating ranges.

4 Experimental Results

To assess the performance of the models, a few different evaluation metrics were employed. The effectiveness of the recommendation system is rigorously validated through various evaluation metrics, including RMSE, MAE and metrics such as Precision@K, Recall@K, and F1@K.

Precision at k is calculated using the formula:

$$Precision@k = \frac{\text{Number of relevant items in top } k}{k}$$

Recall at k is calculated using the formula:

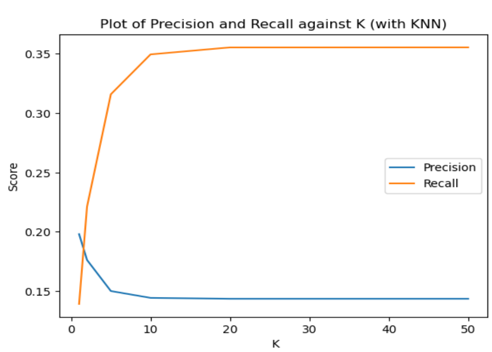
$$Recall@k = \frac{\text{Number of relevant items in top } k}{\text{Total number of relevant items}}$$

These metrics offer insights into the accuracy and reliability of the recommender system, ensuring that the recommendations align with user preferences and are relevant to their travel experiences.

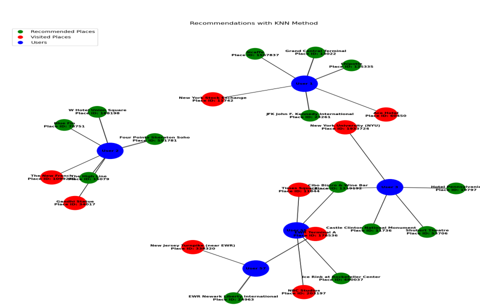
This involved experimenting with different values for parameters such as the number of neighbors (K) in KNN, the number of latent factors in SVD, and the rank in NMF. The models were trained and evaluated multiple times using various hyperparameter configurations to find the combination that resulted in the best performance.

We have summarized the results from all 4 models below:

- **Collaborative Filtering using KNN**: The optimal model for collaborative filtering (that was found using grid search on multiple parameters described in model training) using KNN using cosine distance as a distance metric achieved testing RMSE of **1.45**. We also performed 5-fold cross-validation on the optimal model which gave a mean RMSE of **1.45** and a mean MAE of **0.65**. The recommendations using KNN and the variation of recall and precision as we increase the number of recommendations are shown below in figures 6a and 6b



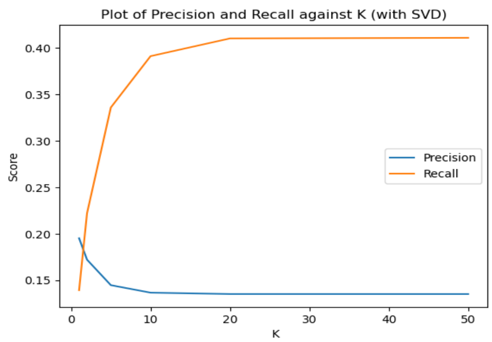
(a) Precision and recall of KNN vs K



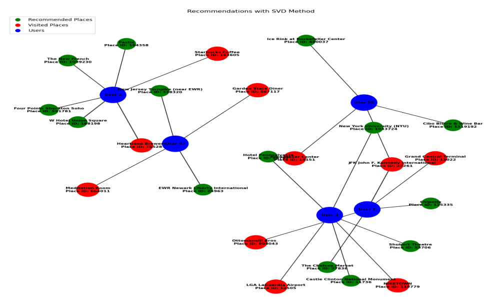
(b) Recommendation graph using KNN

Figure 6: Performance and Insights from KNN

- **Collaborative Filtering using SVD**: The optimal model for collaborative filtering (that was found using grid search on multiple parameters described in model training) using SVD achieved testing RMSE of **1.31**. 5-fold cross-validation on the optimal model which gave a mean RMSE of **1.31** and a mean MAE of **0.6529**. The recommendations using SVD and the variation of recall and precision as we increase the number of recommendations are shown below in the figures 7a and 7b



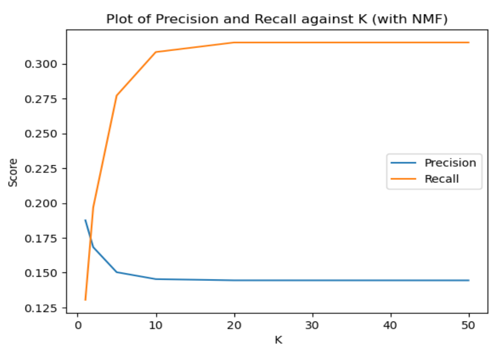
(a) Precision and recall of SVD vs K



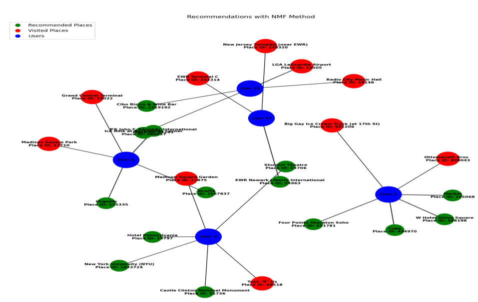
(b) Recommendation graph using SVD

Figure 7: Performance and Insights from SVD

- **Collaborative Filtering using NMF**: The optimal model for collaborative filtering (that was found using grid search on multiple parameters described in model training) using NMF achieved testing RMSE of **1.42**. 5-fold cross-validation on the optimal model gave a mean RMSE of **1.43** and a mean MAE of **0.69**. The recommendations using NMF and the variation of recall and precision as we increase the number of recommendations are shown below in the figures 8a and 8b



(a) Precision and recall of NMF vs K



(b) Recommendation graph using NMF

Figure 8: Performance and Insights from NMF

- **Geo-Social Recommender**: The optimal model we achieved using both geographical (location procimity) and social similarity of the users achieved a precision@10 of **0.13** and recall@10 of **0.17**. The variation of recall and precision as we increase the number of recommendations is shown below in figure 9

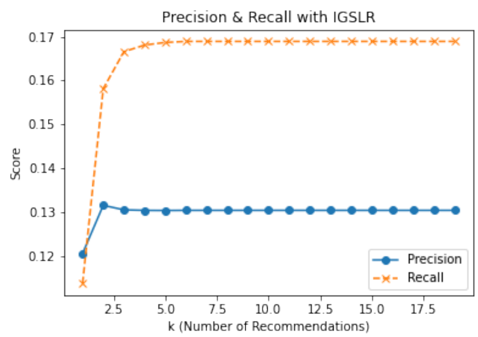


Figure 9: Precision and recall of Geo-Social recommender vs K

- **Two Tower Neural Model**: To include more user and location features into the model, we used the Two Tower Model with both user features in one tower and location features in the other. The features we used and the hyperparameters we varied in order to select the optimal model are described in the model training section. The optimal model achieved a precision@10 of **0.13** and a recall@10 of **0.41**.The variation of recall and precision as we increase the number of recommendations is shown below in figure 10

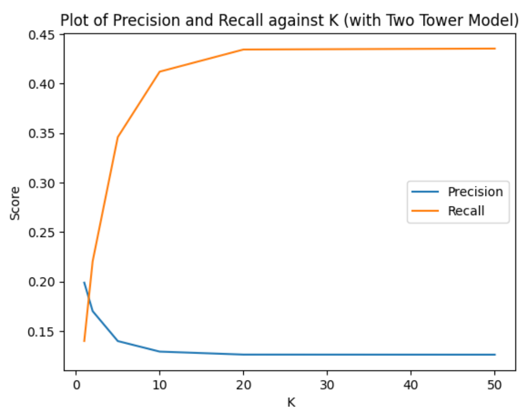


Figure 10: Precision and recall of Two tower model vs K

Model Comparison

We finally compare the performance of each of the 5 models i.e The collaborative filtering models (KNN, SVD and NMF), the Geo-Social recommendation model which considered both location proximity and social similarity of users, and finally the more sophisticated Two Tower deep learning model which considered both user and location features.

The comparative performance of all 5 models based on precision@10, recall@10 and F1 score @10 are shown in the table below:

Model	Precision@10	Recall@10	F1 score @10
KNN	0.14	0.35	0.20
SVD	0.14	0.39	0.21
NMF	0.15	0.31	0.20
Geo-Social	0.13	0.17	0.15
Two Tower	0.13	0.41	0.20

Table 1: Comparison of Model performance

We can see from the comparison of the models that all of them have similar performance with regard to precision. However, the recall for the two-tower model is much higher at **0.41** compared to the other models hence it can cover a much more comprehensive list of relevant POIs.

5 Insights and Contributions

The project makes significant contributions to the travel recommendation landscape by systematically evaluating and comparing diverse models, each addressing specific challenges. Precision and recall metrics serve as pivotal indicators of recommendation system effectiveness. Collaborative filtering models, including KNN, SVD, and NMF, showcase their utility in predicting user preferences. The Geo-Social Recommender achieves a precision@10 of **0.13** and a recall@10 of **0.17**, indicating its capability to provide accurate and relevant recommendations. The Two-Tower Neural Model, with its innovative architecture, further refines recommendations based on learned embeddings, with performance measures of precision@10 of **0.13** and a much superior recall@10 as **0.41**, indicating the contribution to the project's goal of delivering personalized and memorable travel experiences. The combination of insights from geographical analysis, and user activity metrics enriches the understanding of user behaviors, laying the foundation for targeted and effective recommendation strategies. Overall, the project not only introduces cutting-edge models but also substantiates their effectiveness through meticulous evaluation, paving the way for a new era in personalized travel decision-making.

6 Conclusion

In summary, The work overall showcases a multifaceted approach to the travel recommendation problem. Starting with classical collaborative filtering methods, including Singular Value Decomposition (SVD), K-Nearest Neighbors (KNN), and Non-Negative Matrix Factorization (NMF). With meticulous hyperparameter tuning to optimize performance. Non-negative matrix Factorization (NMF) offers a valuable contribution, particularly effective for non-negative data, by factorizing the user-item matrix into interpretable non-negative matrices for personalized location recommendations.

Additionally, the incorporation of a specialized Geo-Social Location Recommender (GSLR) stands out as an innovative stride. By integrating geographical and social features, GSLR recognizes the significance of both spatial and social dimensions in influencing user preferences. The model not only taps into collaborative filtering but also considers users' social/friendship connections and personalized geographical influences, offering a nuanced understanding of individualized travel behaviors.

Furthermore, the project introduces a cutting-edge Two-Tower Neural Recommender, a sophisticated architecture designed to comprehensively process user and location features independently.

This approach leverages the power of deep neural networks to learn embeddings that encapsulate the intrinsic characteristics of users and locations. The dual-tower structure enables the model to discern complex patterns and relationships, leading to more informed predictions for user-location pairs. Notably, the ensemble of Two-Tower models with diverse architectures or hyperparameters reflects an innovative strategy, harnessing multiple perspectives to boost recommendation performance. This fusion of traditional collaborative filtering, advanced neural network architectures, and novel ensemble techniques underscores the project's commitment to pushing the boundaries of recommender system innovation in the realm of travel decision-making.

REFERENCES

Mera Kartika Delimayanti, Mauldy Laya, Bambang Warsuta, Mas Brillianesa Faydhurrahman, Mohammad Azri Khairuddin, Hubbaka Ghoyati, Anggi Mardiyono, and Rizqi Fitri Naryanto. Web-based movie recommendation system using content-based filtering and knn algorithm. *2022 9th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pp. 314–318, 2022. URL https://api.semanticscholar.org/CorpusID:253122936.

Huiji Gao, Jiliang Tang, and Huan Liu. Addressing the cold-start problem in location recommendation using geo-social correlations. *Data Mining and Knowledge Discovery*, 29:299–323, 2015. URL https://api.semanticscholar.org/CorpusID:13364278.

Jiangli Jiao, Xueying Zhang, Fenglian Li, and Yan Wang. A novel learning rate function and its application on the svd++ recommendation algorithm. *IEEE Access*, 8:14112–14122, 2020. URL https://api.semanticscholar.org/CorpusID:210970488.

Xiangfu Meng and Jinfeng Fang. A diverse and personalized poi recommendation approach by integrating geo-social embedding relations. *IEEE Access*, 8:226309–226323, 2020. URL https://api.semanticscholar.org/CorpusID:230512742.

Zhaobo Wang, Yanmin Zhu, Qiaomei Zhang, Haobing Liu, Chunyang Wang, and Tong Liu. Graph-enhanced spatial-temporal network for next poi recommendation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16:1 – 21, 2022. URL https://api.semanticscholar.org/CorpusID:247085095.

Yuxia Wu, Ke Li, Guoshuai Zhao, and Xueming Qian. Personalized long- and short-term preference learning for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34:1944–1957, 2022. URL https://api.semanticscholar.org/CorpusID:226628837.

Le Yan, Zhen Qin, Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork. Revisiting two-tower models for unbiased learning to rank. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022. URL https://api.semanticscholar.org/CorpusID:248482955.

Yantao Yu, Weipeng Wang, Zhoutian Feng, Daiyue Xue, Meituan, and Beijing. A dual augmented two-tower model for online large-scale recommendation. 2021. URL https://api.semanticscholar.org/CorpusID:252924234.

Jia-Dong Zhang and Chi-Yin Chow. Igslr: Personalized geo-social location recommendation: A kernel density estimation approach. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'13, pp. 334–343, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450325219. doi: 10.1145/2525314.2525339. URL https://doi.org/10.1145/2525314.2525339.

Pengpeng Zhao, Haifeng Zhu, Yanchi Liu, Ziting Zhou, Zhixu Li, Jiajie Xu, Lei Zhao, and Victor S. Sheng. A generative model approach for geo-social group recommendation. *Journal of Computer Science and Technology*, 33:727–738, 2018. URL https://api.semanticscholar.org/CorpusID:49725911.