Personalized Location Recommendation : Exploiting Social and Geographical Influence

ISYE 6740-B Team -3
Aravind Rajeev Nair
Naveen Sethuraman
Manikant Thatipalli
Prasanthi Mounika Toram

Contents

Introduction

Data and Preprocessing

Exploratory Data Analysis

Data Processing

Collaborative Filtering

Modelling

Model Comparison

Conclusion and Future work

Introduction

Objective

Purpose:

- ❖ Develop an advanced recommendation system leveraging the Gowalla dataset to suggest the next location for users based on historical check-ins and preferences.

Goals:

- ❖ Implement collaborative filtering and other advanced recommendation techniques to understand user preferences and patterns.
- ❖ Evaluate the effectiveness of the recommendation system through metrics such as accuracy, precision, and user feedback.

Objective

Outcomes:

- ❖ Provide valuable insights into user behavior and preferences within the context of a popular LBSN.
- ❖ Improve user engagement and enhance user experience by providing accurate and context-aware suggestions for their next destination.

Data and Preprocessing

Data :

Our analysis leverages real-world user check-in data from Gowalla, a popular location-based social network. This data includes information about users, places, and individual check-in details.

- 319063 Unique users
- 2724891 Unique places

Source :
https://drive.google.com/u/0/uc?id=0BzpKyxX1dqTYRTFVYTd1UG81ZXc&export=download

Diving into the Gowalla Data

Check Ins
UserID
PlaceID
Check In Datetime

User Info
id
trips_count
friends_count

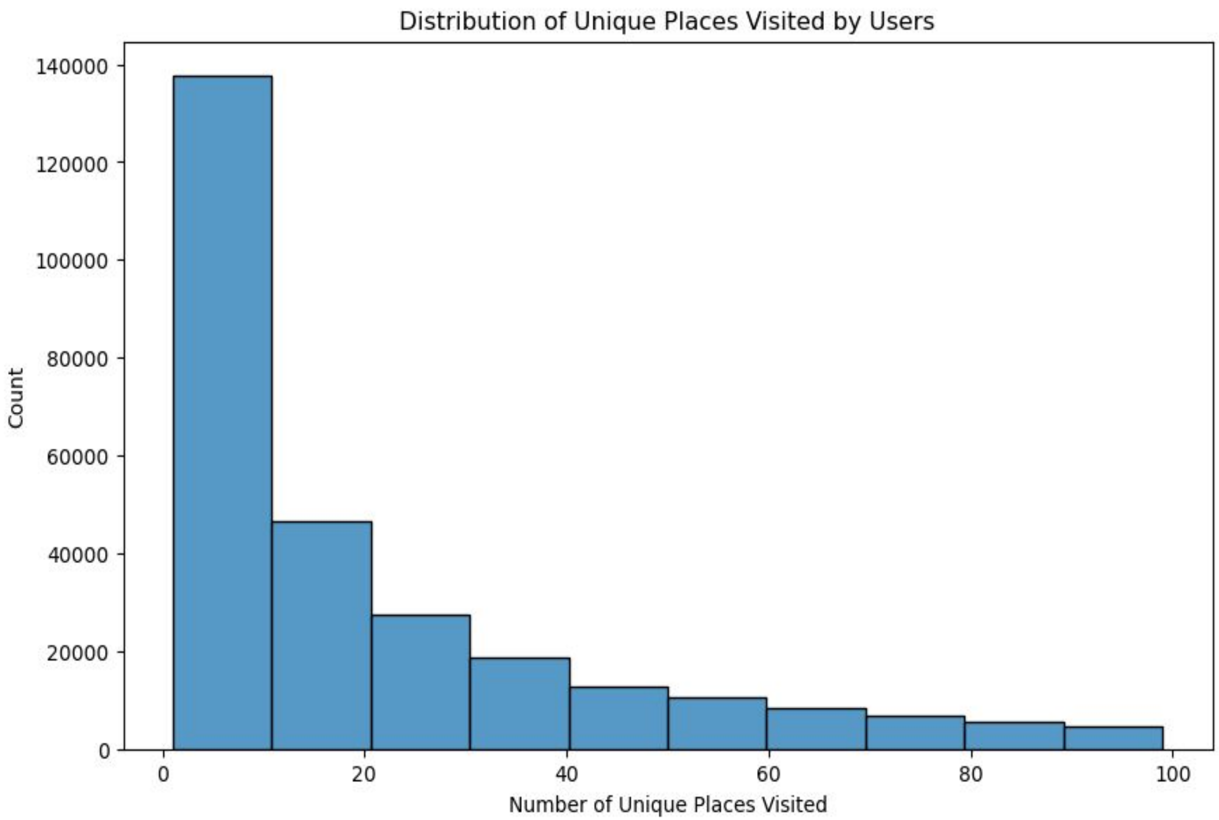
Friendship
User Id
Friend's User ID

Spots
id
created_at
lng
lat
users_count
radius_meters
spot_categories

NY Spots
spotid
spotname
geo-coordinates

Extracted and merged key data points from various sources into a single, unified view for further analysis.

Distribution of No.of unique places visited



- ❖ To enhance recommendation accuracy and computational efficiency, users with fewer than 10 unique places, who constitute a significant portion of the user base, will be filtered out.

Preprocessing

Focused on users with 10-100 unique places to avoid sparse data and reduce potential bias in recommendations.

→

Narrowed the scope to places within the New York area, enhancing recommendation relevance and computational efficiency.

→

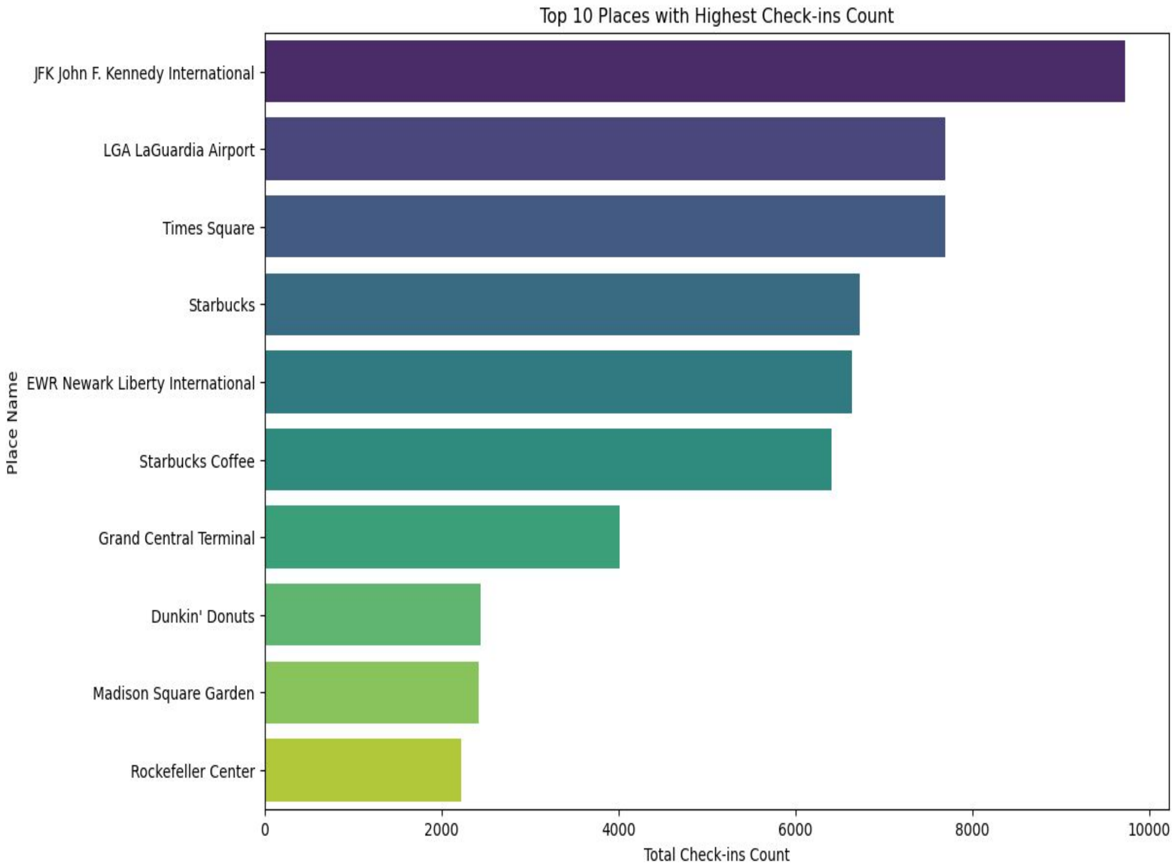
Outcome: 142,000 data points for personalized travel recommendations of ~21k places for ~5k unique users

Processed Data

userid	placeid	datetime	lng	lat	place_photos_count	place_checkins_count	place_radius_meters	user_pins_count	user_friends_count	user_checkin_num	user_trips_count	place_name
116691	11835	2011-04-08T19:00:31Z	-73.982180357	40.7532308669	91	1302	150	86	48	5281	0	The New York Public Library
43632	11835	2011-04-09T02:29:46Z	-73.982180357	40.7532308669	91	1302	150	44	1	3644	0	The New York Public Library
112025	11835	2011-04-28T07:02:50Z	-73.982180357	40.7532308669	91	1302	150	261	1709	5679	12	The New York Public Library
1531870	11835	2011-04-28T05:44:40Z	-73.982180357	40.7532308669	91	1302	150	255	1393	13384	4	The New York Public Library
533055	11835	2011-04-08T19:00:28Z	-73.982180357	40.7532308669	91	1302	150	66	10	6963	0	The New York Public Library
120	11835	2010-10-07T22:21:26Z	-73.982180357	40.7532308669	91	1302	150	294	249	12568	6	The New York Public Library
2080407	11835	2011-04-08T18:33:07Z	-73.982180357	40.7532308669	91	1302	150	54	6	3612	0	The New York Public Library
117848	11835	2011-04-08T19:00:29Z	-73.982180357	40.7532308669	91	1302	150	67	24	3303	2	The New York Public Library
264675	11835	2011-04-15T20:26:52Z	-73.982180357	40.7532308669	91	1302	150	208	805	2968	0	The New York Public Library
264675	11835	2010-08-04T16:31:59Z	-73.982180357	40.7532308669	91	1302	150	208	805	2968	0	The New York Public Library
168935	11835	2011-02-19T16:08:41Z	-73.982180357	40.7532308669	91	1302	150	100	66	3587	1	The New York Public Library
5339	11835	2010-11-22T14:49:07Z	-73.982180357	40.7532308669	91	1302	150	99	35	3516	4	The New York Public Library
68089	11835	2010-11-07T18:30:21Z	-73.982180357	40.7532308669	91	1302	150	67	10	1408	0	The New York Public Library
16982	11835	2010-11-14T21:09:26Z	-73.982180357	40.7532308669	91	1302	150	87	129	4403	0	The New York Public Library

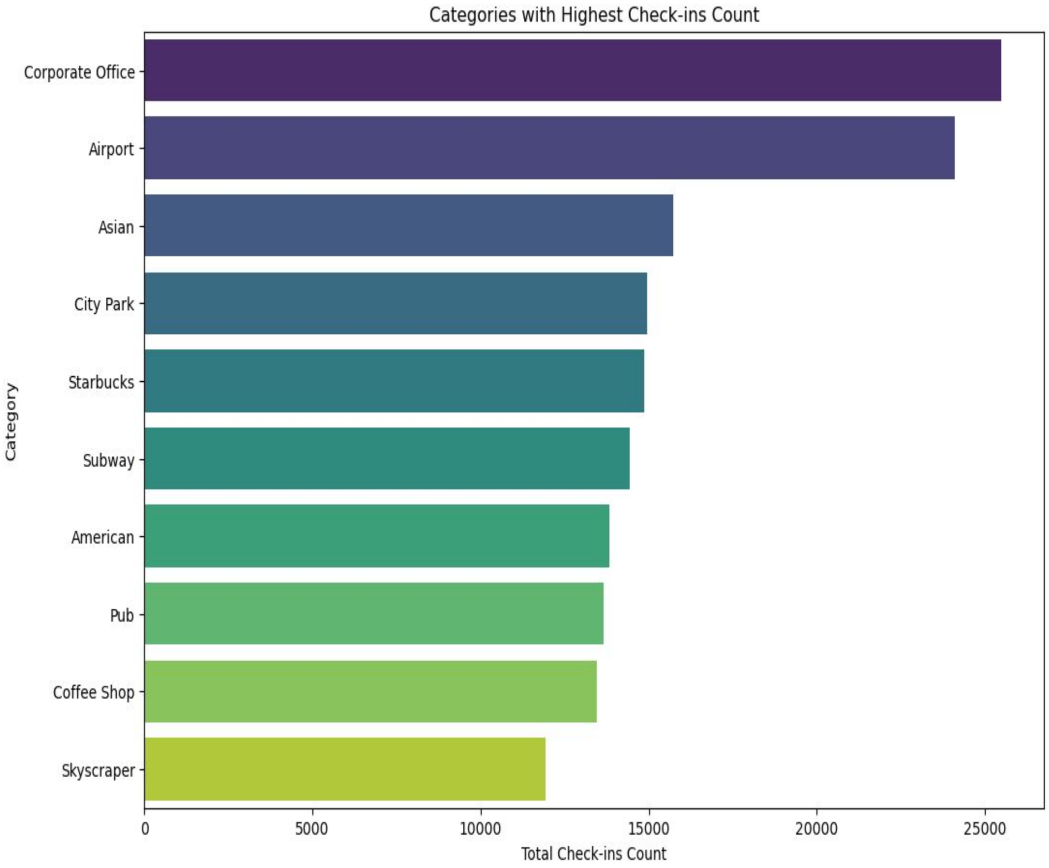
Exploratory Data Analysis

Most Popular Destinations



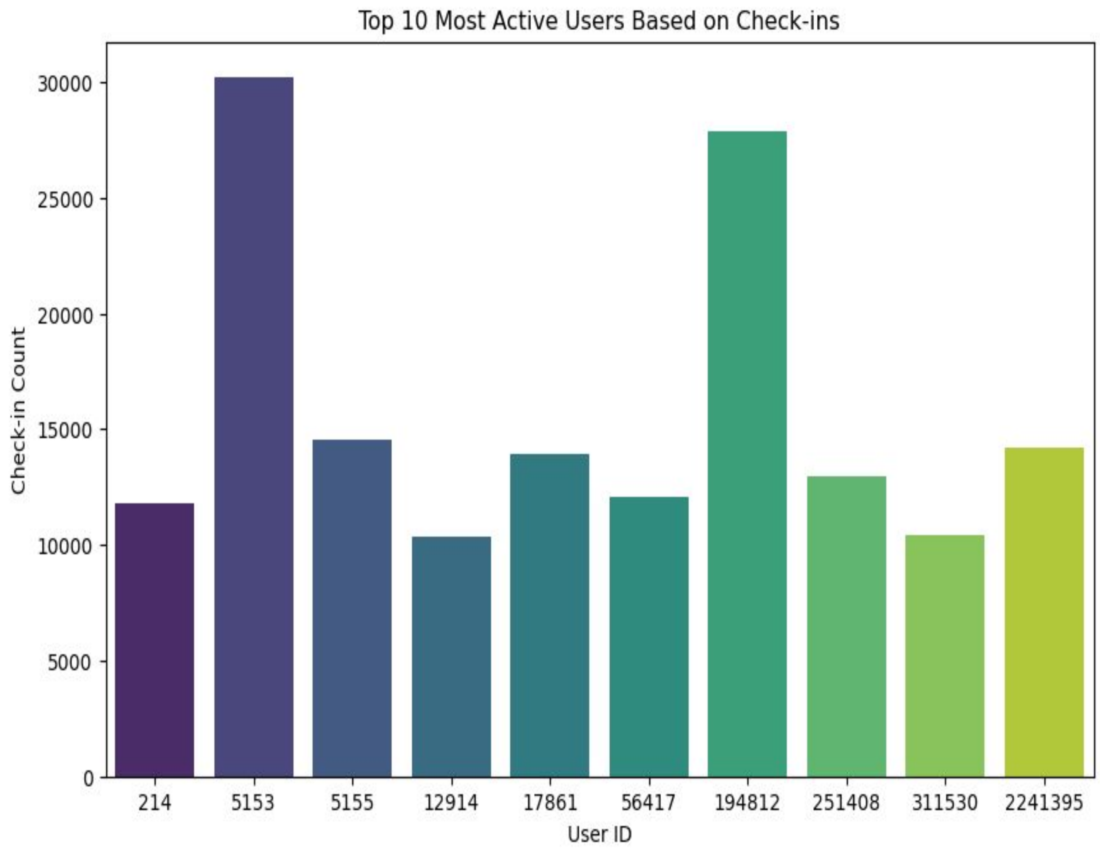
- ❖ Airports, and Times Square lead the way with most number of check in’s, followed by Starbucks.

Most Popular Categories



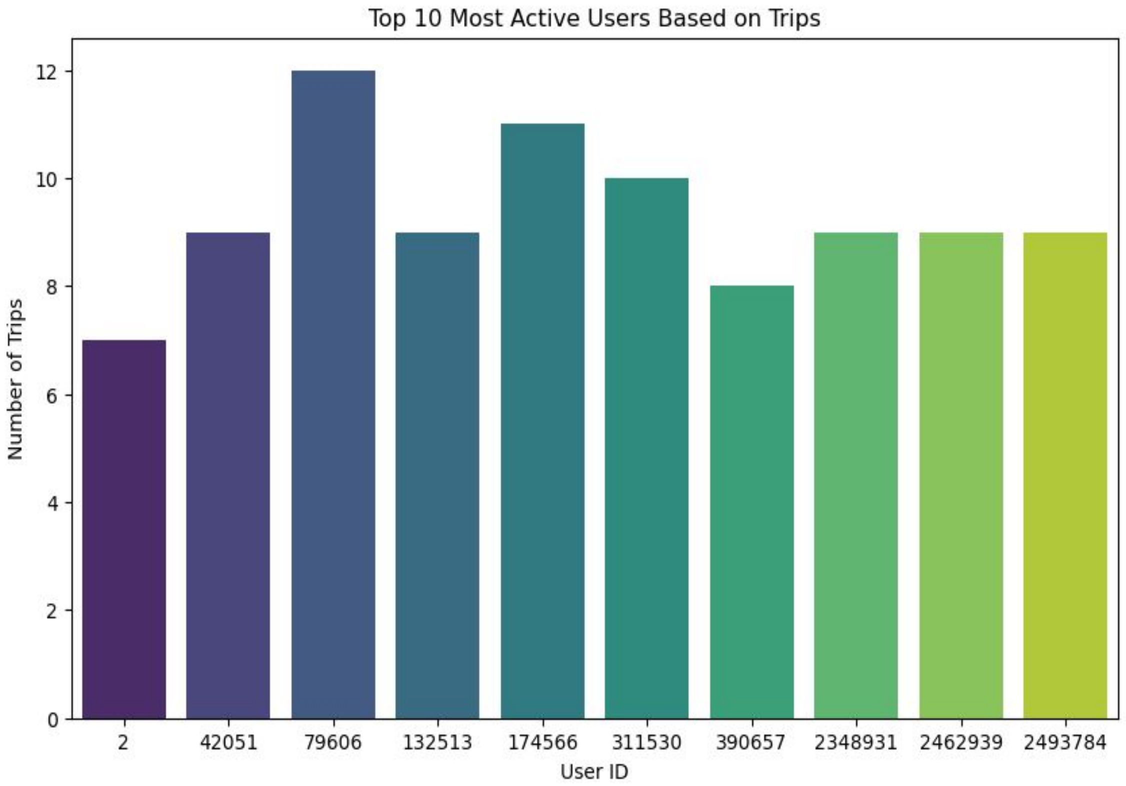
- ❖ Office spaces, Airports, parks and food spots stand out to be popular choices

Highly active users



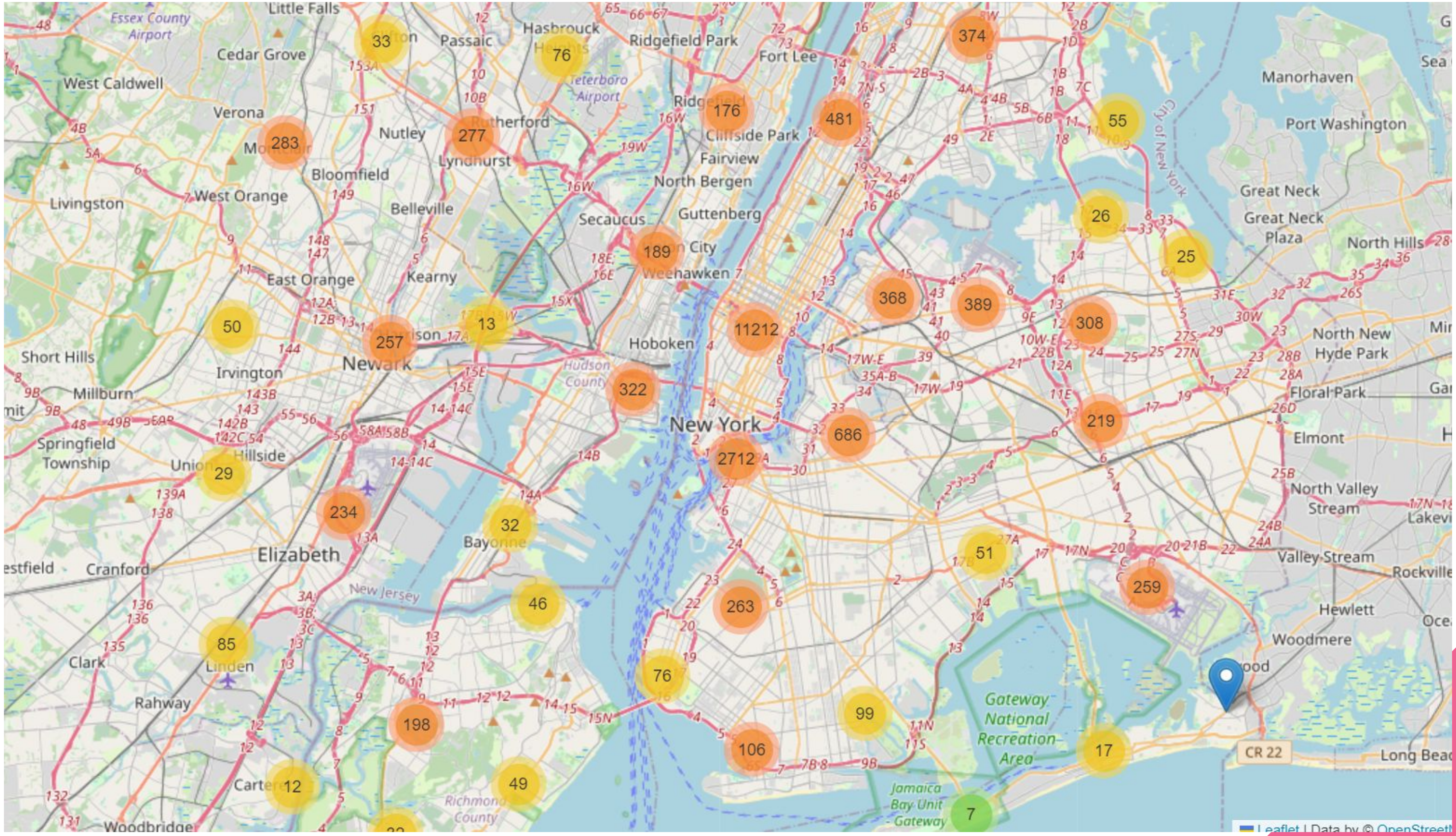
❖ The most active users registered upwards of 10,000 check-ins.

Most Frequent Travelers

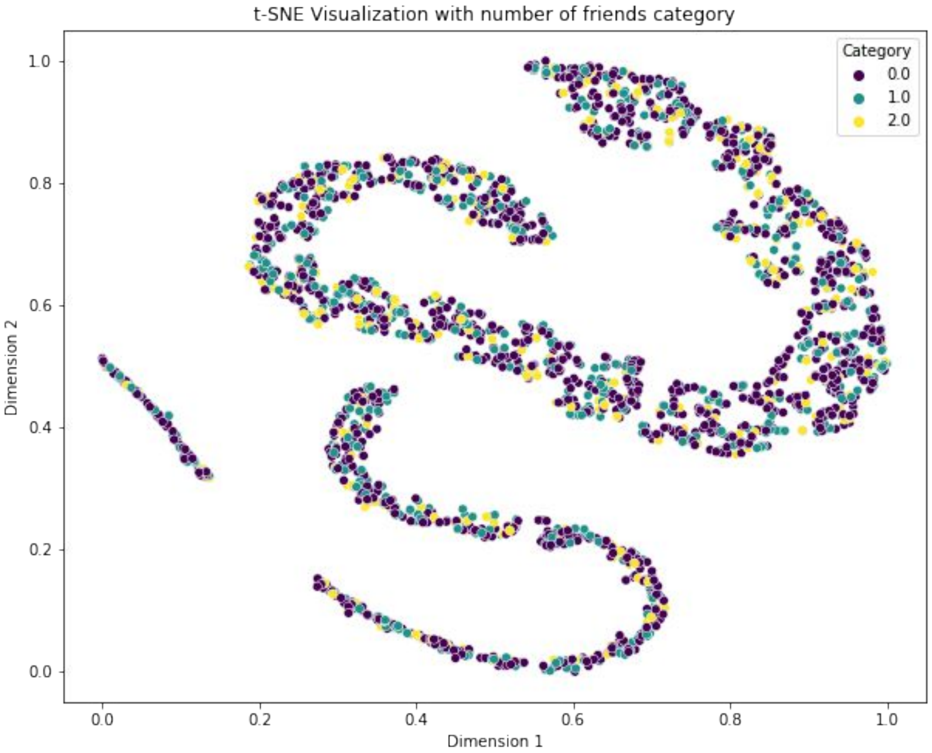


❖ The most frequent travelers have around 7-12 trips.

Heap Map of check-in's across NY



Analyzing user based on friends count (t-sne)



No real patterns were observed to enable categorization of users based on their friendships.

Cat 1 : <25 friends
Cat2 : <40 friends
Cat3 : <50 friends

Data Processing

Rating Derivation

- ❖ Utilized the frequency of check-ins as the basis for calculating ratings.
- ❖ Applied a hyperbolic tangent function to map the calculated rating values between 0 and 1 and transposed them to 0-10.

```
nydf1 = pd.merge(nydf[['userid','placeid']], df_locations['id'],left_on="placeid",right_on="id",how="left")
nydf1 = nydf1.dropna()
nydf1=nydf1.groupby(['userid', 'placeid'])["id"].count().reset_index(name="frequency")
fmin = nydf1["frequency"].min()
fmax = nydf1["frequency"].max()
nydf1["ratings"] = nydf1["frequency"].apply(lambda x: 10*np.tanh(x-fmin/(fmax-fmin))) # update the frequencies ir
```

Collaborative Filtering

Collaborative Filtering

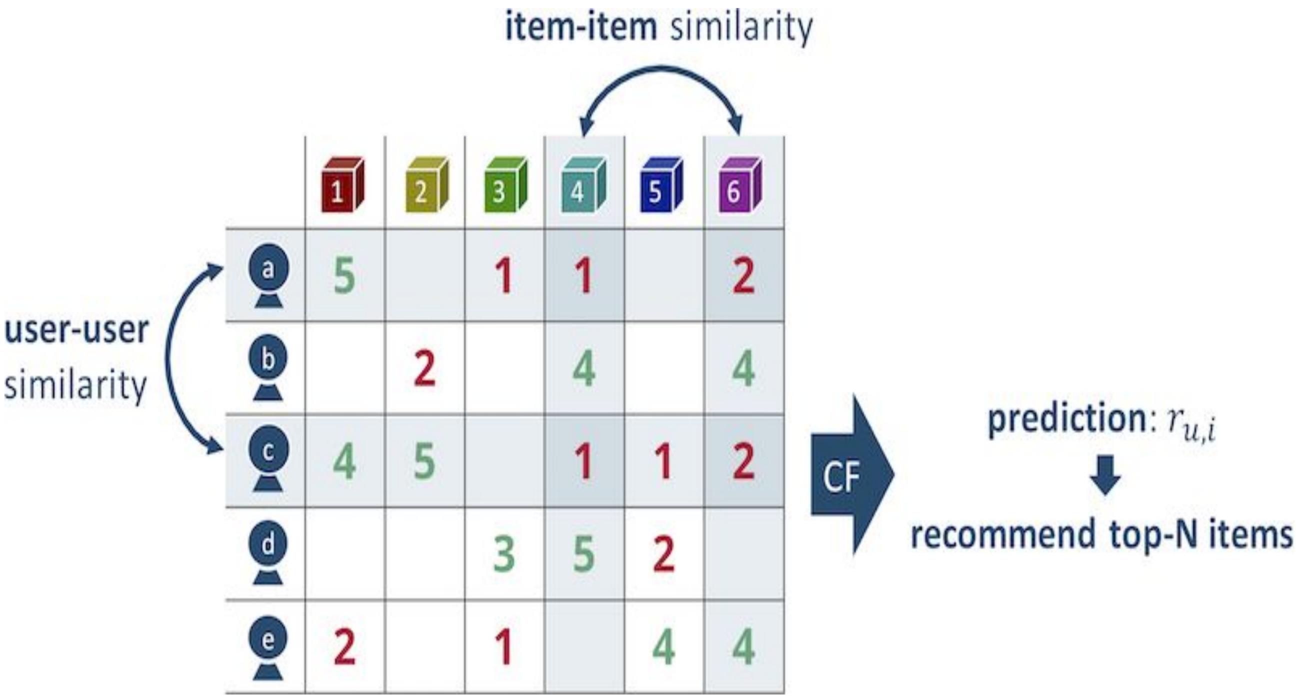
Collaborative Filtering (CF) focuses on understanding user preferences by exploring similarities between users based on their historical interactions and behaviors.

User-User Collaborative Filtering: Recommends items to a user based on the preferences and behaviors of other users who are similar to them.

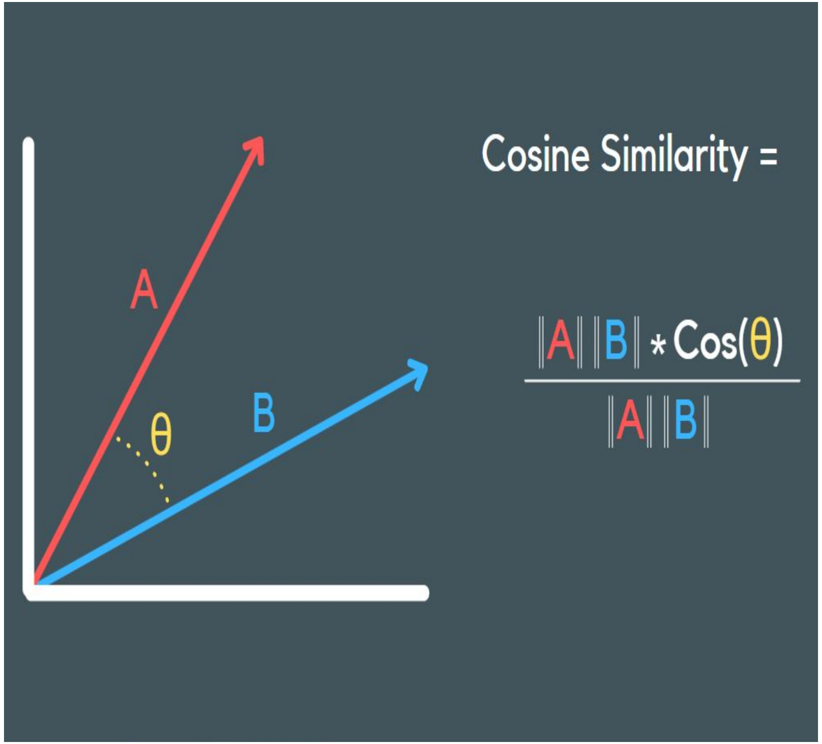
Item-Item Collaborative Filtering: Recommends items to a user based on the similarity between items that the user has interacted with or liked.

User-Item Collaborative Filtering: Predicts or recommends items for a user based on their historical interactions and preferences.

Collaborative Filtering



Cosine Similarity :



$$w(a,i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}}$$

All the models throughout this work use Cosine similarity

Effectiveness of recommender systems

Precision@K

➢ Measures the proportion of recommended items that are relevant to the user. It answers the question: "Out of the top K recommendations, how many are actually good?"
➢ # Relevant items in the top K / K
➢ A higher precision at K indicates that a larger proportion of the recommended items are relevant to the user.
➢ Precision is crucial for user satisfaction. Users are more likely to be satisfied with recommendations if they are mostly relevant.

Effectiveness of recommender systems

Recall@K

➢ Measures the proportion of relevant items that were recommended to the user. It answers the question: "Out of all the relevant items, how many were recommended in the top K?"

➢ # Relevant items in the top K / # Total relevant items

➢ A higher recall at K suggests that a larger proportion of the relevant items have been successfully captured in the top K recommendations.

➢ Recall is important for maximizing coverage. The model should be able to recommend a diverse range of relevant items to the user.

Cross Validation

- **K-fold Cross-Validation** : This is a more robust approach where the data is split into K folds, and each fold is used for testing once. The final performance metric is averaged across all folds.
- Ensures the model learns generalizable patterns and avoids focusing on specific training data points.
- Gives a reliable measure of how well the model will perform on unseen user data.

F1 Score :

- F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

Data after ratings:

1	userid	placeid	lng	lat	place_photos_count	place_checkins_count	place_radius_meters	user_pins_count	user_friends_count	user_checkin_num	user_trips_count	place_name	ratings
2	1	11742	-74.01136279	40.70741722	45	493	75	85	372	1766	2	New York Stock Exchange	0
3	1	11834	-73.98361802	40.75381604	157	1475	250	85	372	1766	2	Bryant Park	0
4	1	11844	-73.98622513	40.75687997	593	7689	300	85	372	1766	2	Times Square	0.25834043784337474
5	1	12313	-73.9857316	40.74844366	152	2204	200	85	372	1766	2	Empire State Building	0
6	1	12973	-73.98945451	40.7413882	62	980	50	85	372	1766	2	Flatiron Building	0
7	1	13022	-73.97725582	40.75279199	168	4009	150	85	372	1766	2	Grand Central Terminal	0
8	1	14148	-73.98016065	40.7601774	124	1545	50	85	372	1766	2	Radio City Music Hall	0
9	1	14151	-73.97857547	40.75871257	161	2228	50	85	372	1766	2	Rockefeller Center	0
10	1	14520	-73.980833	40.760278	11	285	50	85	372	1766	2	Time & Life Building	0
11	1	15079	-74.00754333	40.74239617	147	1316	250	85	372	1766	2	The High Line	0.25834043784337474
12	1	15169	-74.01195287	40.70707082	30	231	100	85	372	1766	2	Trinity Church	0
13	1	16397	-73.99756551	40.73086864	141	1268	150	85	372	1766	2	Washington Square Park	0
14	1	16907	-73.98810522	40.74137425	75	1192	50	85	372	1766	2	Shake Shack	0
15	1	17710	-73.98799539	40.74220108	126	930	150	85	372	1766	2	Madison Square Park	0
16	1	19822	-74.01054543	40.70717272	19	129	75	85	372	1766	2	Federal Hall National Memorial	0
17	1	22806	-74.0049684	40.73588474	8	216	75	85	372	1766	2	Magnolia Bakery, Downtown	0
18	1	23261	-73.7828064	40.64388454	236	9729	1500	85	372	1766	2	JFK John F. Kennedy International	0.773644743
19	1	27278	-73.99025917	40.75609165	35	1134	100	85	372	1766	2	The New York Times	0
20	1	27836	-74.00603056	40.74251159	39	1056	300	85	372	1766	2	The Chelsea Market	0
21	1	34484	-73.9755	40.75150762	28	592	150	85	372	1766	2	Chrysler Building	0
22	1	34817	-74.00416353	40.73419338	8	57	75	85	372	1766	2	Westville	0
23	1	60450	-73.98822069	40.74581014	56	1166	75	85	372	1766	2	Ace Hotel	0
24	1	78751	-73.98542296	40.75936365	12	169	75	85	372	1766	2	Blue Fin	0

Modelling

Recommender Models

Model 1
Cross Validated KNN

Model2
SVD with Cross Validation

Model3
NMF with Cross Validation

Model 4
GEO-SOCIAL COLLABORATIVE FILTERING.

Model 5
Two tower Neural Model

Model 1 - KNN

- Similarity between users is calculated based on their ratings for items and users with similar rating patterns are considered neighbors.
- predicted_rating = Σ(w_i * rating_i) / Σ(w_i)
  - w_i is the weight assigned to neighbor i
  - rating_i is the rating given by neighbor i to the item
- Ranking items based on their predicted ratings for the target user.
- Algorithm recommends the top K items with the highest predicted ratings.

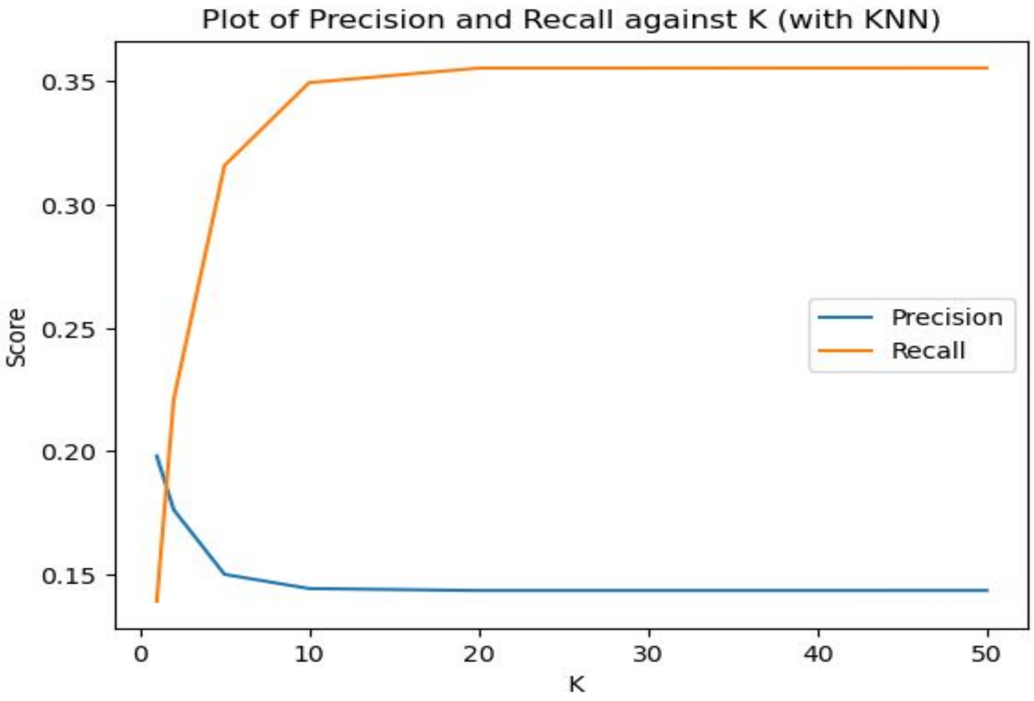
KNN

- Utilized Grid Search with 5-fold Cross-Validation to optimize hyperparameters, enhancing reliability and mitigating overfitting risks.
- Conducted systematic tests with varied K values (5 to 50) to determine the most effective number of neighbors.
- Compared user-user similarity using cosine and Pearson methods, assessing their impact on model performance.
- Evaluated the effects of both uniform and distance-based weightage methods on the contribution of neighbors to the KNN model.

KNN Basic - RMSE, MAE

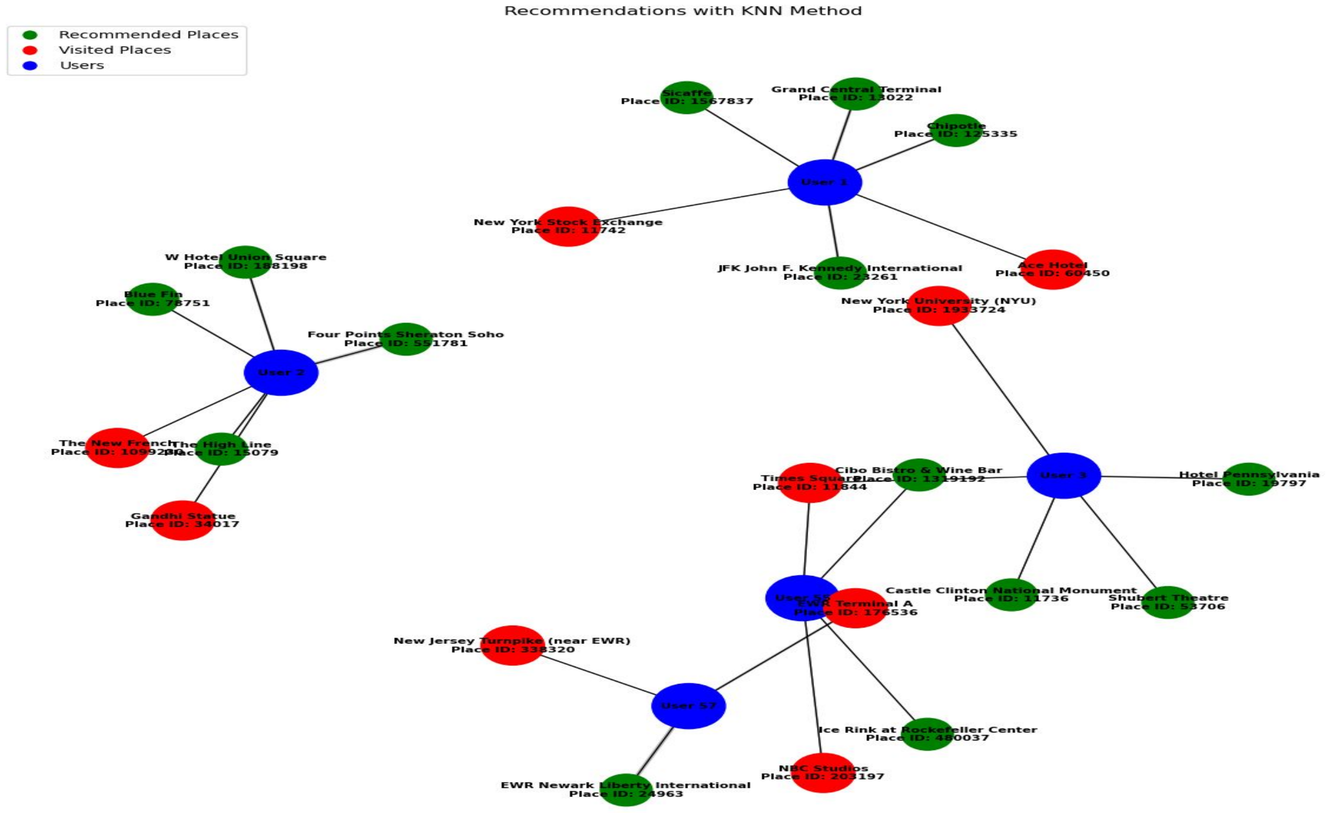
```
Evaluating RMSE, MAE of algorithm KNNBasic on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    1.4699  1.4247  1.4599  1.4661  1.4517  1.4545  0.0161
MAE (testset)     0.6496  0.6375  0.6482  0.6491  0.6442  0.6457  0.0045
Fit time          1.05    0.77    0.70    0.68    0.64    0.77    0.15
Test time         0.92    0.89    0.89    0.87    0.91    0.90    0.02
```

Recall and Precision at K



For KNN with k=10:
Precision : 0.14
Recall : 0.35
F1 Score : 0.2

Recommendation by KNN



Model 2 : SVD

- SVD decomposes the user-item interaction matrix U (m x n) into three matrices:
- **User matrix P (m x k)**: Represents user latent factors, reflecting their underlying preferences and interests.
- **Diagonal matrix S (k x k)**: Contains the singular values, indicating the importance of each latent factor.
- **Item matrix Q (k x n)**: Represents item latent factors, reflecting the characteristics and attributes of each item.
- predicted_rating_ui = P_u * Q_i^T

SVD

- Explored a spectrum of latent factor values (K) from 5 to 50 and investigated how the number of latent factors influences the model's latent space dimensionality.
- Varied learning rates from 0.001 to 0.05 to observe their impact on optimization.Increased the number of iterations (n epochs) up to 50 to control the algorithm's dataset processing.

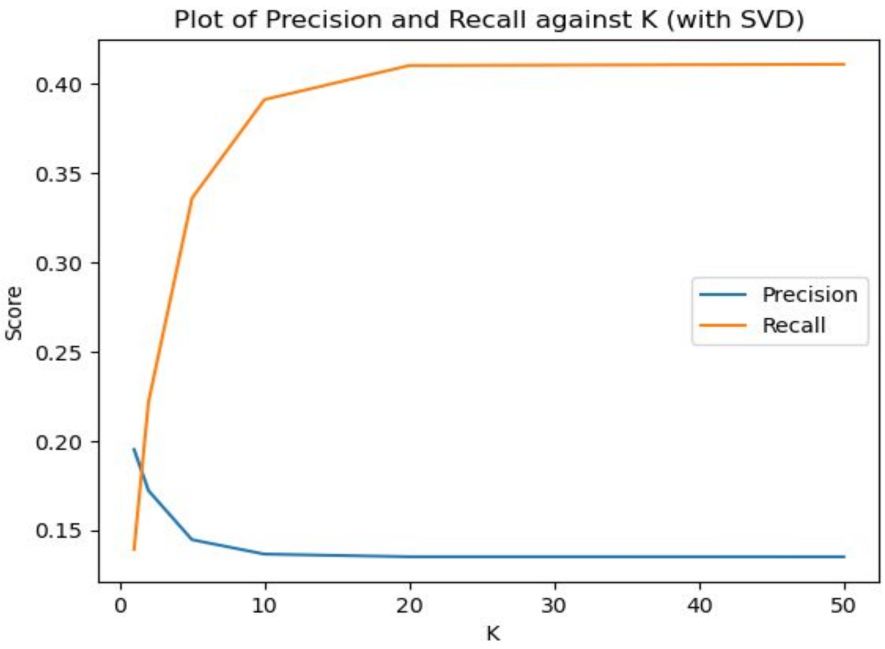
```
param_grid = {'n_factors': [5,10,20,30,40,50],
              'n_epochs': [10,20,30,40,50],
              'lr_all': [0.001,0.05],
              'reg_all': [0.02, 0.1]}
```

SVD RMSE, MAE

Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

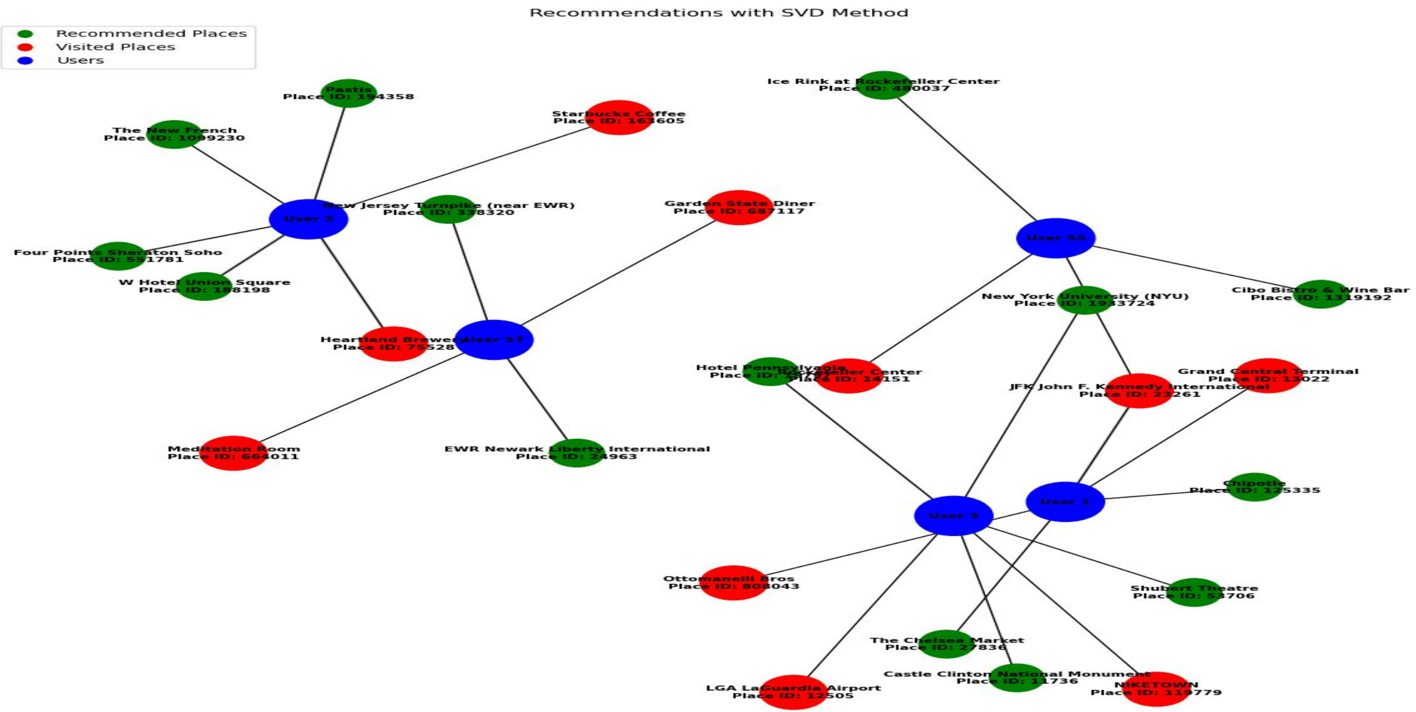
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	1.2947	1.3144	1.3184	1.3115	1.3161	1.3110	0.0084
MAE (testset)	0.6494	0.6496	0.6530	0.6541	0.6586	0.6529	0.0034
Fit time	0.88	0.88	0.88	0.96	1.25	0.97	0.14
Test time	0.07	0.06	0.46	0.06	0.13	0.16	0.15

Precision vs Recall - SVD



For SVD with k=10:
Precision : 0.14
Recall : 0.39
F1 Score : 0.21

Recommendation by SVD



Model 3 - NMF

Factorizes the user-item interaction matrix into non-negative matrices, capturing user preferences and item attributes in a more interpretable way compared to SVD.

NMF decomposes the user-item interaction matrix U (m x n) into two non-negative matrices:

User matrix W (m x k): Represents user preferences as non-negative contributions of latent factors.

Item matrix H (k x n): Represents item attributes as non-negative contributions of latent factors.

predicted_rating_ui = W_u * H_i^T

NMF

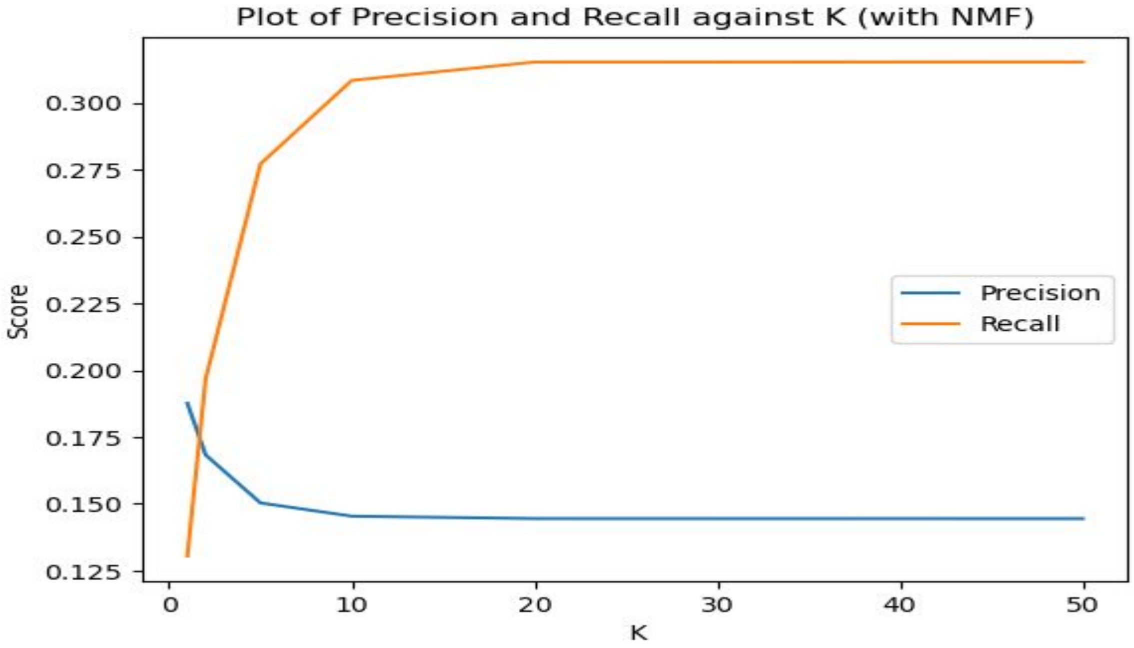
- Investigated a range of latent factor values (K) from 5 to 50 and how the number of latent factors impacts the dimensionality of the latent space in the NMF model.
- Varied learning rates from 0.001 to 0.05 to observe their impact on optimization and increased the number of iterations (n epochs) up to 50 to control the algorithm's dataset processing.
- Parameter grid included values for latent factors, epochs, learning rates, and regularization factors for systematic exploration.

NMF RMSE, MAE

Evaluating RMSE, MAE of algorithm NMF on 5 split(s).

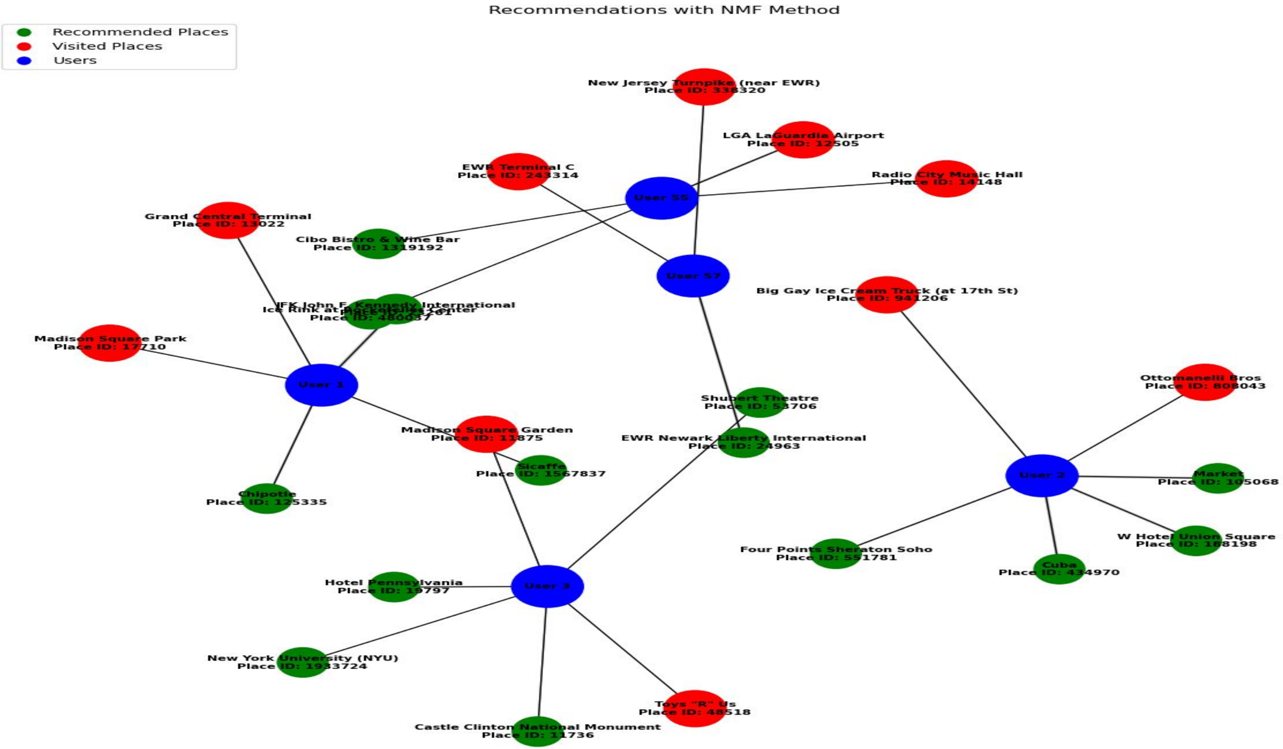
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	1.4383	1.4421	1.4284	1.4020	1.4275	1.4277	0.0140
MAE (testset)	0.6928	0.6932	0.6922	0.6847	0.6932	0.6912	0.0033
Fit time	1.97	2.01	1.92	1.91	1.98	1.96	0.04
Test time	0.06	0.06	0.05	0.05	0.05	0.05	0.00

Precision vs Recall - NMF



For NMF with k = 10:
Precision : 0.15
Recall : 0.31
F1 Score : 0.20

Recommendation by NMF



Geo-Social Collaborative Filtering (GSCLF)

- ❖ Integrating both geographical and social influence for a comprehensive collaborative filtering approach aimed towards personalization.
- ❖ Utilizing kernel density estimation to tailor geographical influence individually, enhancing geo similarity in location recommendations.
- ❖ Incorporating a weighted average of social similarity, providing a nuanced and personalized layer to collaborative filtering for more accurate recommendations.

Social Influence

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

F(ui) denotes the set of users having social friendships with user ui.
SocSim(ui,uk) Social similarity between user ui and uk

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

hat($r_{i,j}$) - Predicted rating of location j for user i

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

hat($p_{i,j}$) - Normalized probability

Geographical Influence

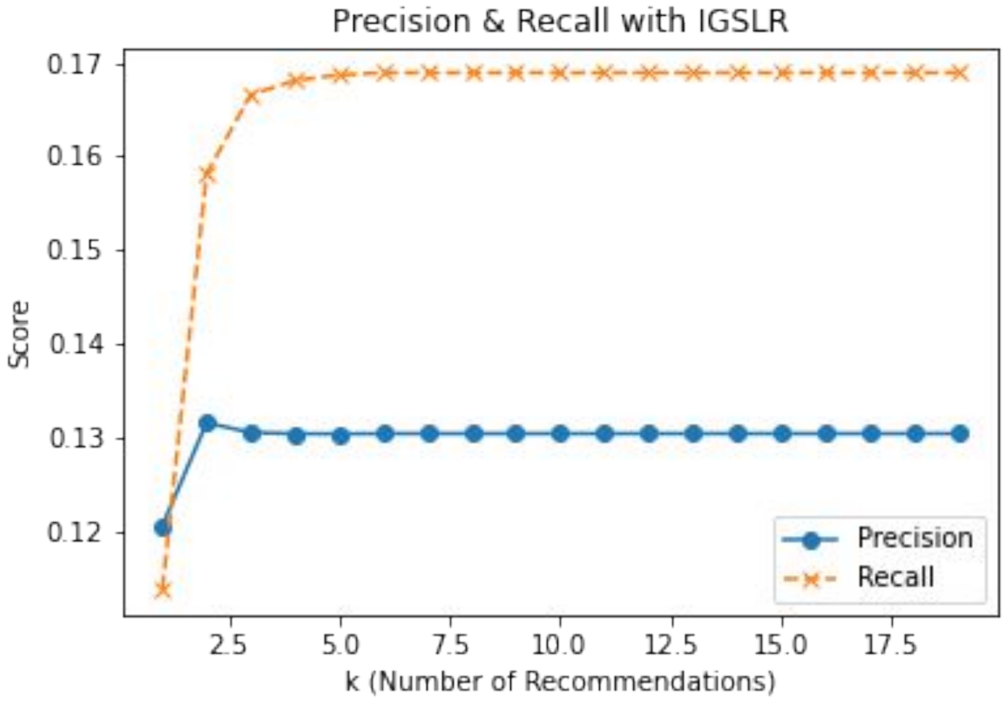
$$\hat{f}(d_{ij}) = \frac{1}{|D|h} \sum_{d' \in D} K\left(\frac{d_{ij} - d'}{h}\right).$$

$d_{i,j}$ - distance between location li,lj
hat(f($d_{i,j}$)) - Probability distribution of $d_{i,j}$

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

p(lj/li) - Probability of ui visiting a new location lj can be obtained by taking the mean probability as follows

Precision vs Recall - Geo Social CF



For GSCLF with k=10:
Precision : 0.13
Recall : 0.17
F1 Score : 0.15

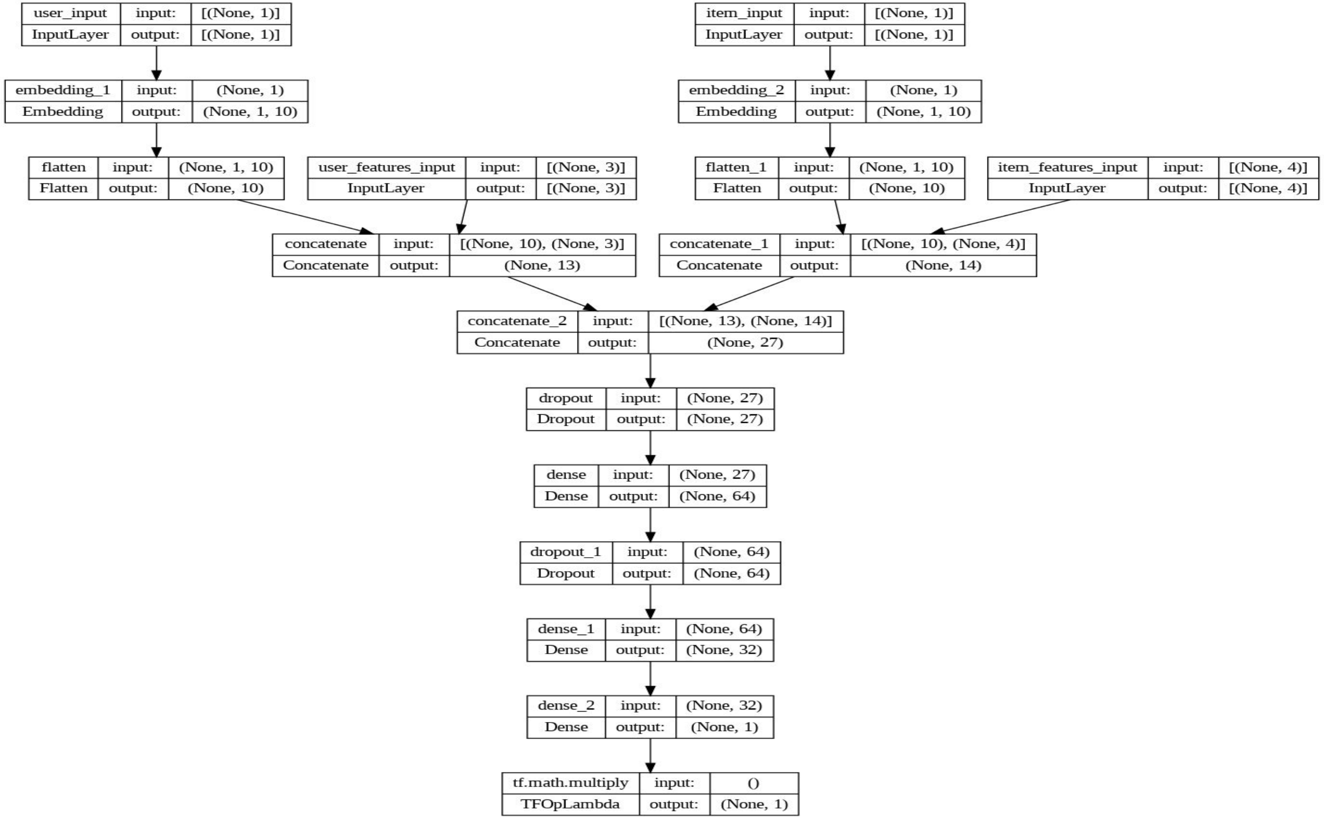
Two Tower Neural Model

- Modern recommendation system where towers process user and location features independently, capturing intrinsic characteristics.
- Employing an implicit collaborative filtering approach, the model determines recommendations based on the dot product of user and item embeddings
- Various models with diverse architectures were implemented, and as a result the best one had 3 hidden layers with 64 units along with 2 dropout layers.
- Our tuning process primarily involved exploring learning rates from 0.001 to 0.05, batch sizes ranging from 16 to 64, and progressive training procedures spanning 10 to 50 epochs.

Two Tower Neural Model Implementation

```
# Define embedding dimensions
embedding_dim = 10

# User input
user_input = Input(shape=(1,), name='user_input')
user_embedding = Embedding(input_dim=data['userid'].max()+1, output_dim=embedding_dim)(user_input)
user_embedding = tf.keras.layers.Flatten()(user_embedding)
user_features_input = Input(shape=(3,), name='user_features_input')

# Item input
item_input = Input(shape=(1,), name='item_input')
item_embedding = Embedding(input_dim=data['placeid'].max()+1, output_dim=embedding_dim)(item_input)
item_embedding = tf.keras.layers.Flatten()(item_embedding)
item_features_input = Input(shape=(4,), name='item_features_input')

# Concatenate user and item embeddings with user and item features
user_concat = Concatenate()([user_embedding, user_features_input])
item_concat = Concatenate()([item_embedding, item_features_input])

# Merge towers with a dense layer
merged = Concatenate()([user_concat, item_concat])
merged = Dropout(0.5)(merged)
merged = Dense(64, activation='relu')(merged)  # Additional dense layer
merged = Dropout(0.3)(merged)
merged = Dense(32, activation='relu')(merged)

# Output layer
output = 10*Dense(1, activation='sigmoid')(merged)
```

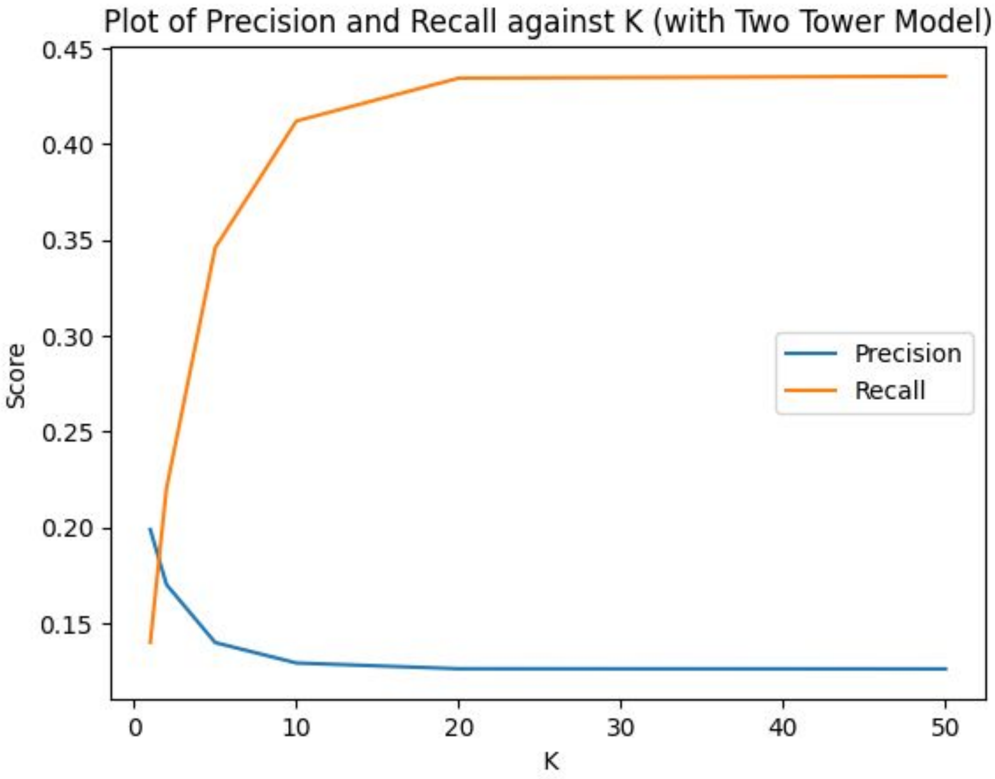
Two Tower Neural Model Architecture



Two tower Model Recommendations

```
Top 5 Recommendations for User 9458:
Place ID: 7224507, Place Name: Bona Fides
Place ID: 292657, Place Name: Salon De Ning
Place ID: 27836, Place Name: The Chelsea Market
Place ID: 284775, Place Name: Clinton St. Baking Company

Top 5 Recommendations for User 74284:
Place ID: 1391695, Place Name: Totale Pizza
Place ID: 579505, Place Name: Robongi
Place ID: 11720, Place Name: Yankee Stadium
Place ID: 167378, Place Name: Tapéo 29
Place ID: 59533, Place Name: Katsu-Hama

Top 5 Recommendations for User 185554:
Place ID: 12505, Place Name: LGA LaGuardia Airport
Place ID: 11738, Place Name: Battery Park
Place ID: 11978, Place Name: Statue of Liberty Pier
Place ID: 11975, Place Name: Statue of Liberty

Top 5 Recommendations for User 309599:
Place ID: 177792, Place Name: Midtown Tunnel
Place ID: 286318, Place Name: Jeffrey New York
Place ID: 194358, Place Name: Pastis
Place ID: 611928, Place Name: Starbucks
Place ID: 748402, Place Name: Angel Orensanz Foundation

Top 5 Recommendations for User 300839:
Place ID: 65871, Place Name: Prime Meats
Place ID: 234307, Place Name: LGA Marine Air Terminal
Place ID: 19762, Place Name: Apple Store, SoHo
Place ID: 486265, Place Name: Coco Roco
Place ID: 440680, Place Name: Washington Square Arch
```

Precision vs Recall - Two tower Model



For Two tower model with k=10 :
Precision : 0.13
Recall : 0.41
F1 Score : 0.20

Model Comparison

Performance Comparison

Algorithm	Recall@k(k=10)	Precision@k(k=10)	F1 Score
KNNBasic	0.35	0.14	0.20
SVD	0.39	0.14	0.21
NMF	0.31	0.15	0.20
GeoSocial	0.17	0.13	0.15
Two tower	0.41	0.13	0.20

Conclusion

- Ref . Slide 57, Performance of Two Tower model is superior to others since its recall@10 = 0.41 is higher compared to other models
- IGSLR is able to capture geographic and social aspects, but it needs more out-of-sample and A/B testing to induce a better performance.
- Addition of user and location based features will help reduce the cold-start problem.
- By incorporating user-item demographics alongside collaborative filtering, the two-tower model paints a richer picture of user similarities, potentially unlocking the door to more personalized and relevant recommendations

Future Work

Temporal Dynamics and Evolving User Preferences :

- Incorporate temporal dynamics to adapt to changes in user preferences over time.

Hybrid Models and Content-Based Filtering :

- Explore hybrid models that integrate content-based filtering and deep learning techniques for a more comprehensive understanding of user preferences.

User Feedback Integration and Continuous Improvement:

- Develop mechanisms to actively gather and incorporate user feedback for continuous refinement of recommendation algorithms.

Scalability for Larger Datasets:

- Investigate approaches to scale the recommendation system for larger datasets and growing user bases.

Thank You