Technical Report: Travel Recommendation System

Google Colab Link:https://colab.research.google.com/drive/1NvXjx6-k6K69ie_-ccQa_9Q2Ictbk6K#scrollTo=02aad131

1. Introduction

The objective of this project is to design a personalized travel recommendation system that integrates multiple data sources and machine learning algorithms. The system provides destination suggestions using both content-based and collaborative filtering methods, and predicts destination popularity based on user and destination features using a regression model.

2. Dataset Overview

The project uses four interrelated datasets:

1. Destinations Dataset: Contains information on 1000 Indian tourist destinations, including DestinationID, Name, State, Type, Popularity, and BestTimeToVisit.
2. Users Dataset: Comprises 999 users and their preferences, demographics (Gender, NumberOfAdults, NumberOfChildren), and travel interests.
3. Reviews Dataset: Includes user ratings (1–5 scale) and textual reviews for destinations.
4. User History Dataset: Tracks previous visits, including VisitDate and ExperienceRating.

3. Data Preprocessing and Merging

- Checked for missing values and duplicates (none found).
- Cleaned and merged datasets into a unified dataframe df with 993 entries and 21 features.
- Created a new features column by concatenating Type, State, and BestTimeToVisit to represent a destination's characteristics.

4. Exploratory Data Analysis

- Ratings Distribution: Most users gave ratings in the mid-range (1 to 3).
- Destination Popularity: Visualized using bar plots.
- Destination Types: Equally distributed among Historical, Beach, City, Nature, and Adventure.
- Best Time to Visit: Count plot showed most popular visiting seasons.

5. Content-Based Filtering (KNN)

- Used Count Vectorizer to convert destination features into numerical vectors.
- Trained a K-Nearest Neighbors (KNN) model using cosine similarity to recommend similar destinations.
- Example Output: Given a destination, returned 5 most similar ones based on feature similarity.

6. Collaborative Filtering

- Created a user-item matrix using pivot() from the User History dataset.
- Applied KNN-based collaborative filtering to recommend destinations based on user similarity.
- Function collaborative_recommend_knn(user_id) returns unseen destinations rated highly by similar users.
  - If UserID not found in the matrix, a user-not-found warning is returned.

7. Popularity Prediction Using Random Forest

- Selected features: Name, State, Type, BestTimeToVisit, Preferences, Gender, NumberOfAdults, and NumberOfChildren.
- Encoded categorical features using LabelEncoder.
- Split data into training and test sets (80-20 split).
- Trained a Random Forest Regressor to predict Popularity.

Model Performance:
Mean Squared Error (MSE): 0.30
R^2 Score: 0.07

8. Deployment Function: Custom Destination Predictor

- Implemented a function recommend_destinations() that:
- Takes user input as a dictionary.
- Encodes values using trained label encoders.
- Uses the trained model to predict destination popularity.

9. Conclusion

- *This system combines content-based filtering, collaborative filtering, and machine learning for a hybrid travel recommendation engine. While basic recommendation functionality works well, improvements can be made:*
- *Better NLP on review text for sentiment extraction.*
- *Matrix factorization or deep learning models for more accurate collaborative filtering.*
- *Feature engineering or external data (e.g., climate, cost, accessibility) to improve popularity prediction.*

10. Tools and Libraries Used

- ✓ Python, Pandas, NumPy
- ✓ Scikit-learn (KNN, RandomForestRegressor)
- ✓ Seaborn, Matplotlib
- ✓ CountVectorizer (text feature extraction)

Diwakar Kumar
Doordarshita Purohit