YELP RECOMMENDER SYSTEM.

TABLE OF CONTENTS.

1.Introduction.

2.Prerequisites

3.Dataset

4.Data Pre-processing

5.Collaborative Filtering

6.Recommendation Generation

7.Evaluation

8.Usage

9.Conclusion

10.License

INTRODUCTION.

- This project implements a **collaborative filtering-based recommendation system** to recommend businesses (restaurants, services, etc.) to Yelp users. The recommendations are based on user ratings of companies, and the system predicts which businesses a user would likely enjoy based on their past reviews.
- The recommendation system employs **collaborative filtering** techniques, specifically using **Pearson correlation** to calculate the similarity between users or items (businesses). By analyzing the relationships between users or businesses, the system is able to recommend businesses that a user has not yet rated but might enjoy. The system is evaluated using standard metrics such as **precision**, **recall**, and **Root Mean Squared Error (RMSE)**.

2. PREREQUISITES

- Before running the project, ensure you have the necessary tools and libraries to execute it. This project requires a Python environment, and the core libraries used include **Pandas** for data manipulation, **NumPy** for numerical operations, and **Scikit-learn** for evaluation metrics.

3. DATASET

- This recommender system uses the **Yelp dataset**, which includes extensive data about user reviews, business information, and ratings. The key files from the Yelp dataset are:
- **Reviews.csv**: Contains user reviews for various businesses, including the ratings given by users.
- **Users.csv**: Contains information about users such as user IDs and the number of reviews they have written.
- **Businesses.csv**: Contains details about businesses, including business IDs, names, categories, and locations.

DATASET STRUCTURE:

- Reviews.csv: Contains the following relevant columns:
  - user_id: ID of the user who left the review.
  - business_id: ID of the business being reviewed.
  - rating: Rating is given by the user for the business.
- Users.csv: Contains columns like:
  - user_id: Unique identifier for each user.
  - name: Name of the user.
  - review_count: Number of reviews the user has written.
- Businesses.csv: Contains columns like:
  - business_id: Unique identifier for each business.
  - name: Name of the business.
  - categories: Categories associated with the business(e.g. restaurants, shopping etc).

This project mainly processes the reviews.csv file, as it contains the ratings users have given to businesses. The goal is to build a user-item matrix, where the rows represent users, the columns represent businesses, and the values are the ratings provided by the users for those businesses.

4. DATA PRE-PROCESSING.

- The first step in building the recommender system is **data pre-processing**. The main objective is to transform the raw Yelp review data into a **user-item rating matrix**. This matrix represents users' ratings for businesses, and missing values indicate that a user has not rated a particular business.
- Key steps:
  1. Load the Data: Load the reviews.csv file which contains the user review data.
  2. Filter necessary columns: Extract the columns user_id, business_id and rating to focus on user-business interactions.
  3. Create User-item Matrix: Construct a matrix where:
     - Each row represents a unique user.
     - Each column represents a unique business.
     - The values in the matrix represent ratings given by users to businesses.
  4. Handle missing values: Missing ratings are represented as NaN, indicating that the user has not rated a particular business.
- This matrix is the core data structure used for collaborative filtering.

5. COLLABORATIVE FILTERING.

- Collaborative filtering is a technique used to generate recommendations based on the historical preferences of users. In this project, the system uses **Pearson correlation** to calculate the similarity between users or items (businesses).
- **Pearson Correlation Similarity**
- The **Pearson correlation coefficient** measures the linear relationship between two variables, which in this case are the ratings of users or businesses. If two users or businesses have similar rating patterns, their Pearson correlation will be high, indicating that their preferences are likely to align.
- The similarity calculation process involves:
    1. **Normalization**: Subtracting the average rating of a user (or item) from their individual ratings to account for varying rating scales.
    2. **Similarity Calculation**: Using the Pearson correlation formula to compute similarity scores between users or between items (businesses).
    3. **Creating a Similarity Matrix**: Storing the similarity scores between users or items in a matrix that can be referenced when making predictions.

6. RECOMMENDATION GENERATION

- The core function of the recommender system is to generate **personalized recommendations** for each user. Based on the ratings that the user has already given, the system predicts which other businesses the user is likely to enjoy.

Steps for Generating Recommendations:

1. **Identify Rated Items**: Identify the businesses that the user has already rated.
2. **Predict Ratings**: For each unrated business, predict the rating the user would give, based on the similarity matrix.
3. **Sort and Recommend**: Rank the businesses by their predicted ratings and recommend the top 5 businesses that the user has not yet rated.

- The system can recommend businesses even if the user has not rated them in the past. The recommendations are based on the similarity between the user and others or between the businesses themselves.

7. EVALUATION.

- The recommender system's performance is evaluated using several key metrics:
  1. **Root Mean Squared Error (RMSE)**: This metric measures the difference between the predicted ratings and the actual ratings. A lower RMSE indicates better predictive accuracy.
  2. **Precision**: Precision is the fraction of recommended items that are actually relevant (i.e., items the user is likely to enjoy). High precision means the system is good at recommending relevant businesses.
  3. **Recall**: Recall measures the fraction of relevant items that the system successfully recommended out of all possible relevant items. High recall means the system is good at suggesting businesses that the user would like.
- These evaluation metrics help in assessing the effectiveness of the recommendation system and can be used to fine-tune the model for improved performance.

8. USAGE.

- Once the dataset is pre-processed and the collaborative filtering model is built, the system can be used to generate recommendations for a specific user.
- For example, the system will take in a **user_id** and produce the top 5 recommended businesses for that user. This allows for personalized, data-driven suggestions that improve user experience by helping them discover new businesses based on their preferences.

9. CONCLUSION.

- The collaborative filtering-based recommender system successfully predicts which businesses a user is most likely to enjoy, based on their historical ratings and the ratings of similar users or businesses. By leveraging **Pearson correlation** to measure similarity, the system is able to make accurate recommendations.
- The system's effectiveness is evaluated using key metrics like RMSE, precision, and recall, which provide valuable insights into the accuracy of the recommendations. This project offers a solid foundation for building personalized recommendation engines, and future improvements could involve exploring more sophisticated algorithms like **matrix factorization** or incorporating additional features like **item-based collaborative filtering**.