Restaurant Recommender System

CMPE 256 Project Report

Submitted By:

Archita Chakraborty	015224339
Divya Khandelwal	015276885
Priyanka NAM	015287805

CONTENTS

Github Link:https://github.com/divyaKh/CMPE256Project.git

Chapter 1. Introduction **3**

1.1 Project Description 3

1.2 Motivation and Objective 3

Chapter 2. System Design & Implementation details **4**

2.1 Algorithms considered 4

2.2 Technologies and Tools Used 4

2.3 System Design and Implementation Details 4

Chapter 3. Experiments / Proof of concept evaluation **5**

3.1 Dataset 5

3.2 Exploratory Data Analysis 6

3.3 Data preprocessing decisions 7

3.4 Graphs 8

3.5 Analysis of results 9

Chapter 4. Discussion & Conclusions **10**

4.1 Decisions made 10

4.2 Difficulties faced 10

4.3 Things that worked 11

4.4 Conclusion 11

4.5 Future Scope 11

Chapter 5. Project Plan / Task Distribution **11**

Chapter 1. Introduction

1.1 Project Description

Title - Restaurant Recommeder system using Yelp Dataset.
The main purpose of this project is:

1. Examine the Yelp Data set
2. Develop a recommendation system using different algorithms.
3. Predict ratings of the restaurants and recommend top ones to users

1.2 Motivation and Objective

1. To build a recommender system to recommend restaurants to its user
2. To provide relevant suggestions to users based on how the restaurants have been rated by the other users.
3. To thoroughly understand the concepts of Data Mining and Machine Learning. Following are a few:
    - Exploratory Data Analysis : How to understand and visualize the dataset
    - Data cleaning, preprocessing
    - Recommendation system and Sentiment analysis
        - Item Profiles versus User Profiles
        - Content based versus Collaborative Filtering based versus Hybrid recommender system
        - Scaling techniques, Hyperparameter tuning
        - Machine learning models like KNN, SVD, NMF, BaselineOnly, etc

Chapter 2. System Design & Implementation details

2.1 Algorithms considered

KNN : We are using KNN based recommender models from the Surprise library for both the collaborative and content based filtering. KNN models are simple to implement and understand but have a cold start problem for new restaurants and new users. The KNN models are also memory intensive for large datasets.

SVD : SVD and SVDpp models from the Surprise library are used as they perform better compared to KNN models. The only drawback is the execution time when using SVD on large datasets.

BaselineOnly :We are using the BaselineOnly algorithm from the Surprise library which is a basic recommendation model that predicts the baseline estimate for a given user and item.

Hybrid : We combined BaselineOnly and SVD models to create a hybrid model as the drawbacks of one model can be overcome by other models. The effect of any one model would be significantly less in the final recommendations

2.2 Technologies and Tools Used

1. **Python** : Programming language
2. **Jupyter Notebook:** Document-centric platform to display the code, graphical representation
3. **Google Colab**: Browser based document-centric platform
4. **Github**: Source code and version control Manager.
5. **HPC**: High Performance Computation cluster from SJSU

2.3 System Design and Implementation Details

Step 1: Data Preprocessing:

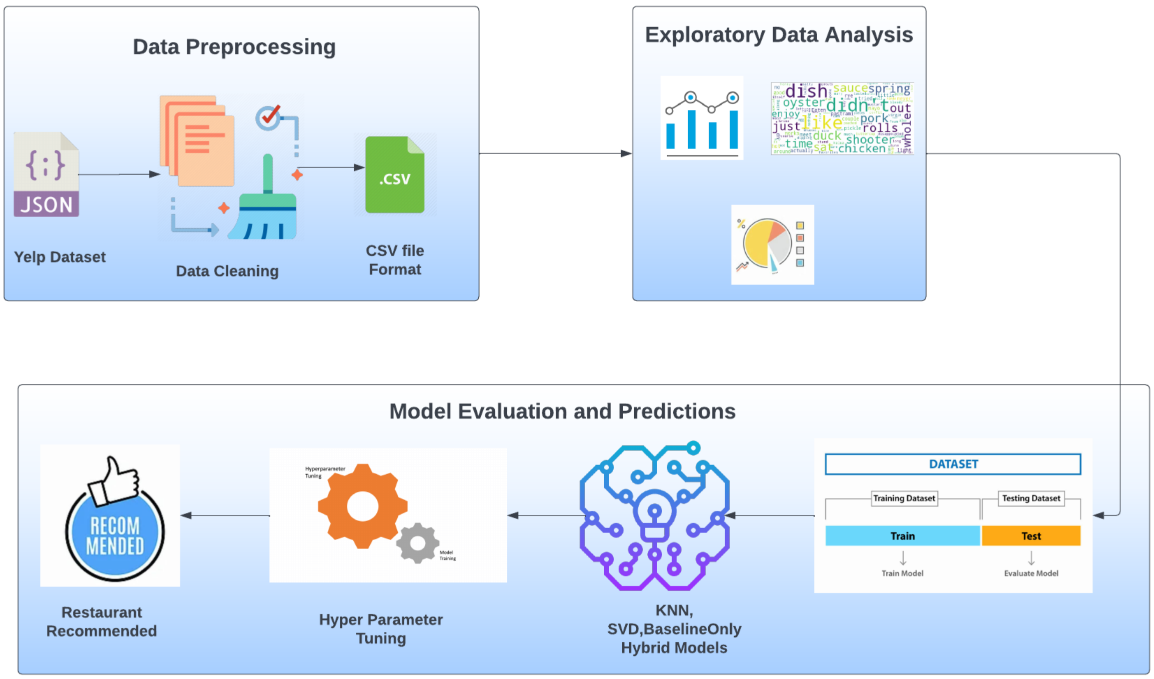
Yelp's dataset is huge, so data preprocessing is the vital part of our project. Yelp dataset contains data files i.e., Business, Users and Reviews in JSON format. We did data preprocessing for all these files. Our data preprocessing includes data cleaning by filling the missing values, eliminating repeated values, data transformation by constructing new attributes from the given data and data reduction by filtering restaurant data only from the state of California. The cleaned data is then stored in separate CSV files for business, users, and reviews.

Step 2: Exploratory Data Analysis

EDA is one of the critical steps in our project. EDA helped us to discover patterns, spot anomalies in the data. We did EDA for business, users, and reviews separately. The processed files from data preprocessing are used for EDA. We have used python libraries like seaborn, Matplotlib, plotly and word cloud to visualize the data and to draw the conclusions from the data.

Step 3: Model Evaluation and Predictions

In this step, we divide the data into test and train and fed the data to different machine learning recommender models. We performed hyper parameter tuning to get the best parameters for the recommender models. We have used RMSE as a performance indicator of our models.



Sytem Design

Chapter 3. Experiments / Proof of concept evaluation

3.1 Dataset

Yelp Dataset: https://www.yelp.com/dataset

Yelp data is derived from Yelp Open Dataset. It contains 1.2 million business attributes,1,987,897 users and 131,930 businesses. We would be using the following json files from the dataset:

1. **Review dataset:** Contains review text data consisting of id for user who wrote the review and the business the review is written for, as well the rating.
2. **Business dataset**: Contains information about the different business's location, business type, attributes, categories, stars rated, etc.
3. **User dataset**: All the user data example user id, user name, reviews given, year, etc.

3.2 Exploratory Data Analysis

The Yelp dataset represents various business data in Fig.2 and their reviews and rating for the North America region. It comprised various businesses like Restaurants, Shopping, Health and Medical, etc. with the Restaurants category being highest in count as shown in Fig. 3.

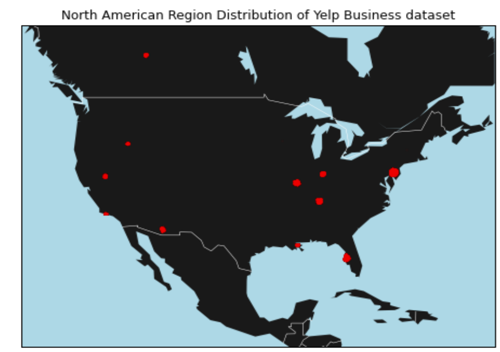


Fig. 2

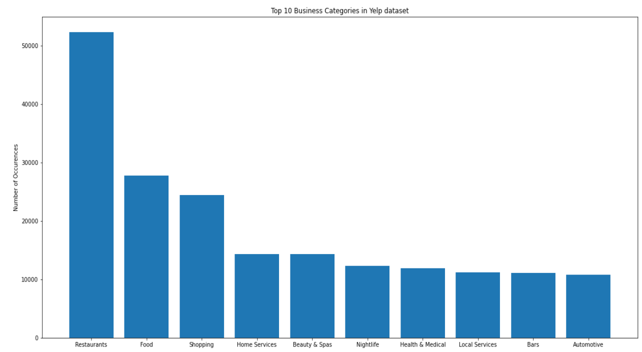


Fig. 3

For this project, we narrowed down our analysis to restaurants in California. From Fig 4. we can see the trend of ratings for restaurants in CA. The majority of users have given ratings equal and higher than 3. Fig.5 shows the distribution of sentiment in the user reviews.It has more positive reviews compared to negative ones.

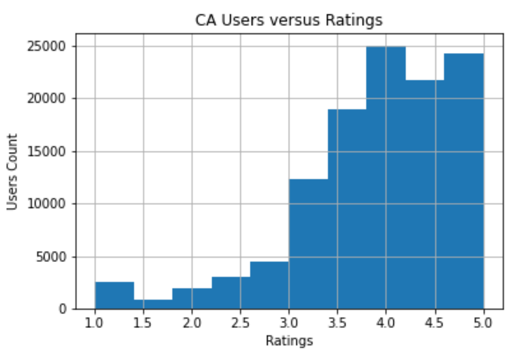


Fig. 4

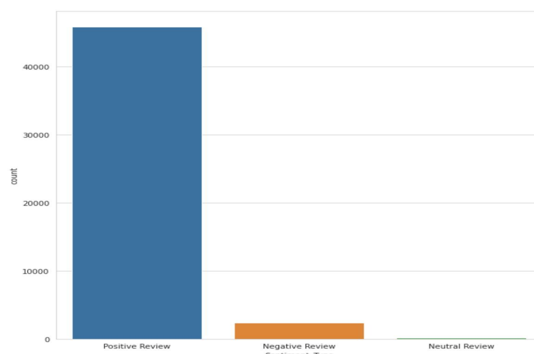


Fig. 5

Fig. 6

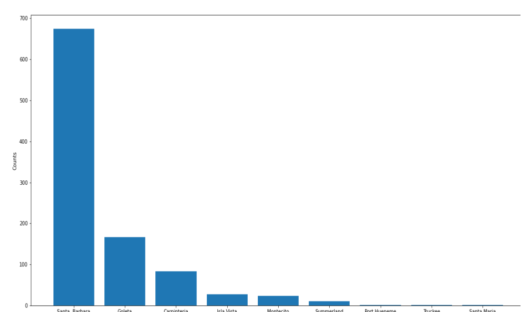


Fig. 7

From the graph in Fig.7, we could analyze that the restaurant data in California is from nine cities with Santa Barbara holding the majority of business data.

3.3 Data preprocessing decisions

The data set was loaded using the json library available in python. There were no duplicate details found in the business dataset. Filtered out only business data related to the Restaurant and Food business category as Yelp dataset had many other business categories.From the restaurant business, we only did the recommendation for state California in the United states. Reviews dataset was large as it not only contains the star rating but also the user review justifying that star rating. We filtered the reviews related to California restaurant businesses as we are limiting our recommendation models to that specific state. Similarly we also filtered the users from the Users dataset based on whether they gave a review for any restaurant in California. Even the filtered reviews were large for the recommendation models, so we added a user elite filter for the reviews. We also performed a sentiment analysis on the user review text and filtered the reviews based on their sentiment and subjectivity for better performance.

3.4 Methodology followed

For the recommendation models, we preprocessed and split the dataset into test and training sets. Based on the below table, we are able to pick the top performing for our hybrid model and then used GridSearchCV function from the surprise library that provides an easy way to tune and select the best hyperparameters to achieve best RMSE Scores.For the GridSearchCV, we provided a set of options for each model parameter and the function checks all the parameter combinations for the best parameters. We evaluated three different KNN models namely, KNNBasic, KNNWithMeans, KNNBaseline, BaselineOnly, SVD and SVDpp for the restaurant recommendation system. We are using the K-Fold validation technique to eliminate any bias in test and training set choices.

Algorithm	test_rmse	fit_time	test_time
BaselineOnly	0.941237	0.032786	0.042431
SVD	0.943450	1.540388	0.074710
SVDpp	0.943587	8.875154	0.340566
KNNBaseline	0.987051	2.336614	0.965319
KNNBasic	1.030607	2.328661	0.862901
CoClustering	1.035614	0.885702	0.050126
KNNWithMeans	1.048164	2.370150	0.895000
NMF	1.094173	1.769459	0.065591
NormalPredictor	1.346152	0.027148	0.075412

Table 1. Algorithm Comparison based on RMSE Scores

Sentiment Analysis

We also performed sentiment analysis of the reviews rather than just taking their star ratings for the restaurants. We used the textblob library for sentiment analysis which gives two types of scores for every user review. One is the sentiment score and the other is the subjectivity or polarity score. The sentiment score is in the range of -1 to 1, where -1 represents a very negative sentiment and vice versa. The polarity scores are in the range of 0 to 1, where 1 means the review is very subjective. We filtered out the reviews that have more than 0.6 polarity and with sentiment scores between -0.2 to 0.2. The intuition is that reviews with too much polarity are biased and reviews that are mostly neutral will not have much value in the recommendation model.

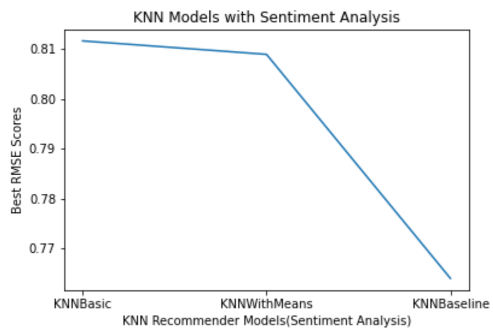


Fig.8

Comparison of RMSE scores for various KNN models after Sentiment Analysis

3.4 Graphs

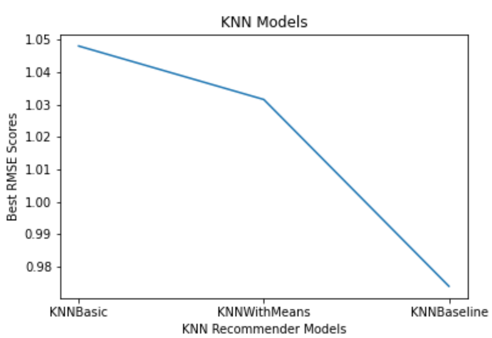


Fig.9

This graph shows the comparison of RMSE scores for various KNN models

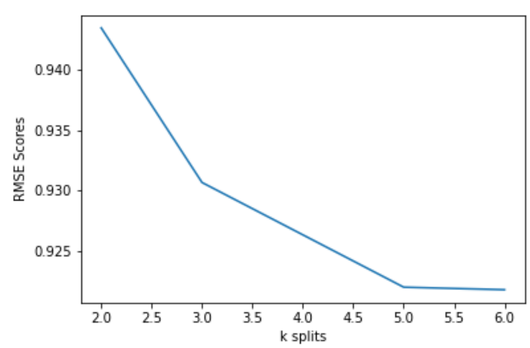


Fig.10

This graph shows the comparison of RMSE scores for SVD for different kfold values

All Models

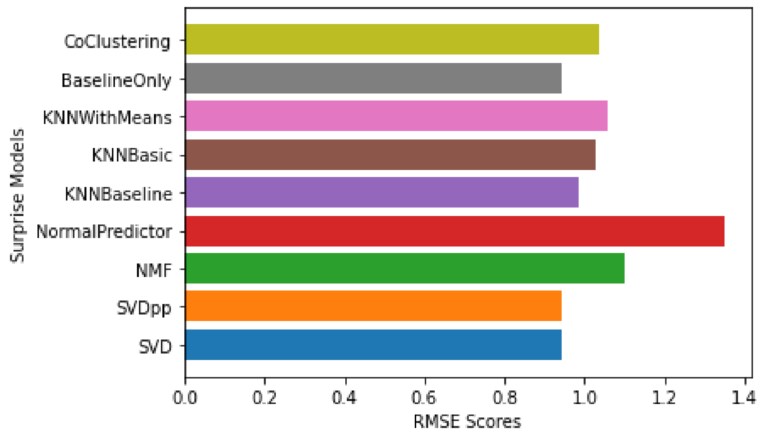


Fig.11

This graph shows the comparison of RMSE scores for all models

3.5 Analysis of results

We used the top models from Table 1 to achieve RMSE Scores for KNNBasic, KNNWithMeans, KNNBaseline, BaselineOnly, SVD models with best parameters and below table shows the comparison of all the models with best RMSE Scores.

Model Name	**Parameters**	**Best RMSE Scores**
KNNBasic	{'bsl_options': {'method': 'als', 'n_epochs': 5}, 'k': 5, 'sim_options': {'name': 'msd', 'min_support': 7, 'user_based': True}}	1.048
KNNWithMeans	{'bsl_options': {'method': 'sgd', 'n_epochs': 15}, 'k': 5, 'sim_options': {'name': 'pearson_baseline', 'min_support': 7, 'user_based': False}}	1.031
KNNBaseline	{'bsl_options': {'method': 'sgd', 'n_epochs': 15}, 'k': 5, 'sim_options': {'name': 'cosine', 'min_support': 7, 'user_based': True}}	0.974
SVD	{'n_factors': 20, 'n_epochs': 30, 'lr_all': 0.005, 'reg_all': 0.07} With k = 6	0.921
BaselineOnly	{'method': 'als', 'n_epochs': 2, 'lr_all': 0.001, 'reg_all': 0.01, 'reg_u': 5, 'reg_i': 2}	0.916

We have picked the top two performing models, SVD and BaselineOnly and have built a hybrid model by averaging the predicted ratings from the two models. The following Fig. 12 shows the output of our recommendation system. It takes in an user_id as an input to our function and returns the top 10 restaurants rated and we have taken a threshold value of 4.5 i.e. top 10 restaurants rated above 4.5 are returned as shown below. (The rating scale is 1-5 in the dataset)

```
Recommendations are listed below for user  Heidi

 Top 10 Recommended Restaurants
```

Restaurant name
Daves Dogs - Cart
Wine Edventures
Backyard Bowls
Jump On The School Bus
Santa Barbara Certified Farmers Market
Third Window Brewing
Taqueria Cuernavaca
Topa Topa Brewing Company
Intermezzo By Wine Cask
Plaza Deli

Fig.12

Chapter 4. Discussion & Conclusions

4.1 Decisions made

We used sentiment analysis to predict the rating rather than just using star ratings. Finally we have selected SVD and BaselineOnly, our top two models and the final predicted rating of the restaurant is provided by averaging the estimated rating from the two models.

4.2 Difficulties faced

1. RAM limitations and system incompatibilities for heavy computations.
2. Working with a large yelp dataset.
3. Dealing with reviews based on different cities, which was highly skewed.
4. Processing review text for sentiment analysis and identifying restaurant specific vocabulary.
5. Tuning parameters for recommender models on huge datasets is a tedious task.
6. Using all restaurant reviews of California state for recommender models was not feasible. So we had to filter these reviews based on some attributes like eliteness, timestamp etc.

4.3 Things that worked

We picked simpler recommendation models like KNN, SVD and BaselineOnly as our main goal was to create the best recommender model with the combination of simpler models.

4.4 Conclusion

1. Without hyperparameter tuning, the top performing model was BaselineOnly with RMSE as 0.941.
2. With hyperparameter tuning, the top performing model was also BaselineOnly with RMSE as 0.916. It was observed that SVD also gave a good RMSE score of 0.921.
3. Sentiment analysis of user reviews give better performing models than simply relying on user star ratings.

4.5 Future Scope

1. To combat cold-start problems, we can implement content based filtering models as well.
2. A web application could be developed for the ease of users to interact with the system and give a list of restaurant recommendations based on the user's location.

Chapter 5. Project Plan / Task Distribution

All* = Archita Chakraborty, Divya Khandelwal, Priyanka NAM

Component	**Person assigned to**	**Person who did the task**
Project Proposal	All	All
Data Preprocessing and EDA	All	All
KNNBasic, KNNBaseline, KNNWithMeans-User Based and Item Based	Priyanka NAM	Priyanka NAM
SVD, SVDpp	Divya Khandelwal	Divya Khandelwal
BaselineOnly & SVD Hybrid Model, Final Recommendation system	Archita Chakraborty	Archita Chakraborty
Project Presentation, Git Readme	All	All
Project Report - Section 1		
Motivation	Divya Khandelwal	Divya Khandelwal

Objective	Divya Khandelwal	Divya Khandelwal
Project Report - Section 2		
System Design and Implementation Details	Priyanka NAM	Priyanka NAM
Algorithms considered selected and why	Priyanka NAM	Priyanka NAM
Technologies and Tools Used	Archita Chakraborty	Archita Chakraborty
Project Report - Section 3		
Dataset	Archita Chakraborty	Archita Chakraborty
Algorithms and Comparison	All	All
Analysis of Results	All	All
Project Report - Section 4		
Decisions made and Difficulties faced	All	All
Things that worked, Things that didn't work	All	All
Conclusion	All	All