# CSE 258: Recommender Systems Assignment

**Group:** Amber Trujillo, Sreyashi Bhattacharya, Keerthana Raviprasad, Mohib Mohyuddin

---

## 1. INTRODUCTION

This research assignment aims to perform a comparative analysis of various predictive modeling techniques that we have studied as part of the CSE 258 Recommender Systems course.

## 2. DATA SET

The data set we are working with is the Google Local Dataset (link: https://datarepo.eng.ucsd.edu/mcauley_group/gdrive/googlelocal/#subsets). Given the time and space constraints of working with extremely large datasets, we chose the **Alabama 10 core** version which contains **5,146,330** user reviews and has the following fields:

| user_id | name | time | rating | text | pics | resp | gmap_id |
|---|---|---|---|---|---|---|---|

The data is in JSON format and was originally collected by Julian McAuley et al [1] for similar work in topic mining and research in recommender systems [2]. The fields correspond to the following definitions:

{

user_id - ID of the reviewer
name - name of the reviwer
time - time of the review (unix time)
rating - rating of the business
text - text of the review
pics - pictures of the review
resp - business response to the review including unix time and text of the response
gmap_id - ID of the business

}

### 2.1 GOALS AND OBJECTIVES

Our primary objective is to predict the star ratings of this Google Local Dataset using prediction techniques studied in class. For this exercise, we will be experimenting with regression techniques, latent factor models, classification, text-based classifiers, and sentiment analysis techniques to predict the star rating of the business based on features extracted from user reviews. We will compare the performance, accuracy, and other strengths and weaknesses of various approaches, based on which we will make a recommendation on the approach and type of model to use for similar analyses and datasets.

## 3. EXPLORATORY ANALYSIS

In this phase we did a deep dive into the data to better understand the nuances in the data, the spread, and the statistical metrics - we would then use these learnings to inform our methodology and model(s) logic for the implementation phase. We used the first 100,000 data points for the exploratory data analysis phase, but trained and tested our model(s) on the full data set of 5,146,330 data points.

Since review and response text would be key features in our feature matrix for the models we want to implement, we explored how many fields include the text fields contained non-null values to see if including these features was even viable to begin with.

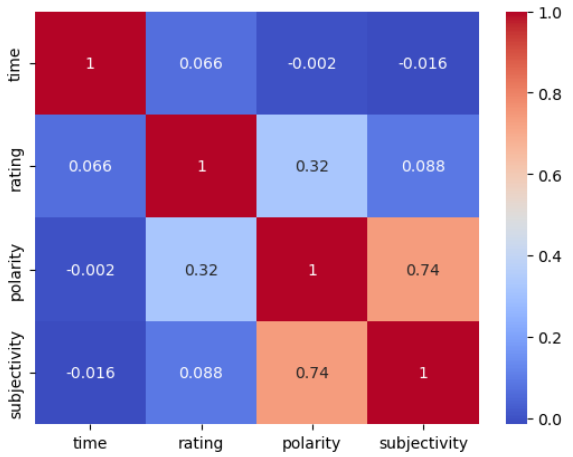
| **Column** | **Non-Null Count** |
|---|---|
| user_id | 50,000 |
| Name of reviewer | 50,000 |

| | |
|---|---|
| time | 50,000 |
| rating | 50,000 |
| text | 27,714 |
| pics | 994 |
| response | 7,402 |
| gmap_id | 50,000 |

Next, we wanted to explore the viability of performing sentiment analysis on the review and response text for which we wanted to get a sense of the words/tokens being used in the text most often. For this analysis, we plotted a word cloud of the most frequently used words, after removing the stop words. The results are as follows:

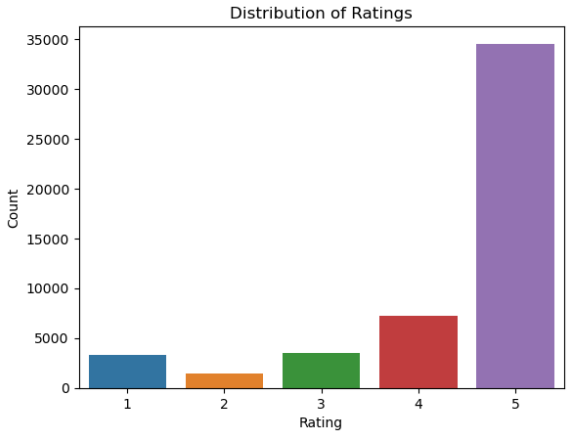

| **Word** | **Frequency** |
|---|---|
| None | 22,342 |
| Great | 8,125 |
| Place | 4,223 |
| Service | 4,038 |
| Good | 3,969 |
| Friendly | 3,386 |
| Staff | 3,158 |
| Nice | 2,978 |
| Get | 2,450 |
| People | 2,442 |

To get a sense of what the polarity and subjectivity scores were for each non-null review, we plotted the following heat map:

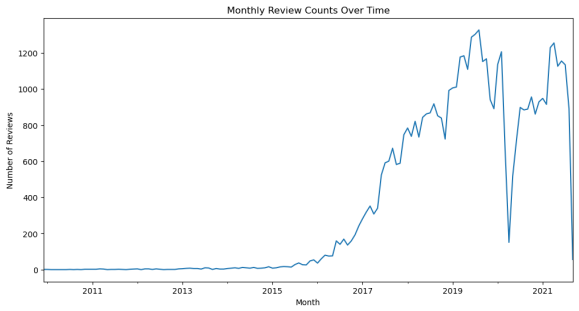



At this point we were fairly confident that by applying sentiment analysis on the review text and using that as a feature would yield reasonably accurate predictive model(s).

Next, we explored the frequency distribution of the ratings.




And also uncovered the reviewing trend in general to see how the monthly review count has evolved over time




## 4. LITERATURE REVIEW

The advent of the internet has meant that businesses are ever reliant on third-party reviews as revenue generators and engagement tools [3]. These reviews act as a gateway for the businesses to interact with and elicit feedback from their consumer base [4]. The importance of reviews can be gauged by an analysis done by Brightlocal which reports that 97% of customers check online reviews and read an average of about seven reviews before committing to a business or making frequent transactions. Liu et al [5] via their analysis has suggested that customer review platforms such as google reviews and the likes are sound sources to discern what leads to customer satisfaction and consequently motivates a purchase decision which in turn can be leveraged by businesses to adjust offerings as per the demand [6].

Implementing recommender systems to predict star ratings can oftentimes be challenging. Most recommender systems rely on historical user trends/behaviors (ratings) and are classified as collaborative filtering models (CF). There are salient approaches to CF; (i) the neighborhood approach and (ii) the latent factor approach. Neighborhood approaches are fairly commonplace and make use of similarities among users or items. Latent factor models attempt to model items and users as vectors in the same 'latent factor' space and the star rating is elicited by means of proximity between the related latent factor vectors.

Diving deeper into neighborhood models, [7] suggests that the item-item similarity approach fares better than the user-user approach in terms of RMSE and also being more scalable - in our case this would mean a gmap_id to gmap_id similarity model would offer strong performance. Prior to comparing ratings it is a good practice to eliminate biases that obscure the foundational relationship between items (gmap_ids). Such biases can include instances where certain gmap_ids have a trend of receiving higher ratings than others or similarly the tendency of certain users perpetually giving high ratings to certain items [8]. Paolo et al. [9] has suggested an approach where they remove the denominator of the Correlation Neighborhood method (item-item kNN) method:

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in D^k(u;i)} d_{ij} \left(r_{uj} - b_{uj}\right)}{\sum_{j \in D^k(u;i)} d_{ij}}$$

The modified approach becomes:

$$\hat{r}_{ui} = b_{ui} + \sum_{j \in D^k(u;i)} d_{ij} \left(r_{uj} - b_{uj}\right)$$

This approach allowed them to rank items based solely on their appeal/relevance to the users where rui is not a rating per se but illustrates the relationship between the user and the item. [8,10] have also deployed similar non-normalized recommender models in their experiments.

Latent factor models are also increasingly popular in this domain, in particular the Singular Value Decomposition models, informally referred to as SVD models which are based on factoring user-items ratings matrices [11]. However, a limitation of SVD has often been cited that it is undefined when faced with missing ratings (some reviews are logged without ratings) and some researchers [12] have suggested filling the missing ratings with a baseline estimate, however a prominent criticism of this workaround is that this gives rise to a

rather dense feature matrix which is computationally taxing to process. More contemporary solutions involve training factor vectors on known ratings on the basis of an adequate objective function that is set up to minimize the error - a technique referred to as gradient descent [13] is deployed to minimize the objective function being used. [11] have also experimented with a powerful matrix factorization model which represents users as a combination of item-related features and reported an RMSE of 0.900 on the Netflix Dataset - this technique is referred to as the *Assymetric-SVD* (AsySVD).

## 5. METHODOLOGY AND RESULTS

To predict ratings of the Google Local Dataset, we experimented with four different models and a few supplementary ones and compared performance and accuracy across the board. The four main models make use of concepts taught in class and iteratively become more advanced from Model 1 to Model 4 to achieve quantifiable improved performance. We judged the performance of our models based on MSE, Accuracy, and Balanced Error Rate (BER)

### Model 1: Linear Regression

To establish a baseline model, we implemented a simple linear regression of the form:

$$X\theta = y$$

The feature matrix consisted of one-hot encodings of the weekday, month and hour which we extracted from the unix formatted time stamp, length of the review as a ratio of the max review length, length of the response from the business, and binary representation of if pictures were attached. Our prediction variable '*y*' was the star rating the user shared for the business.

After tuning the parameters, the best *C, Validation and Test BER* were as follows:

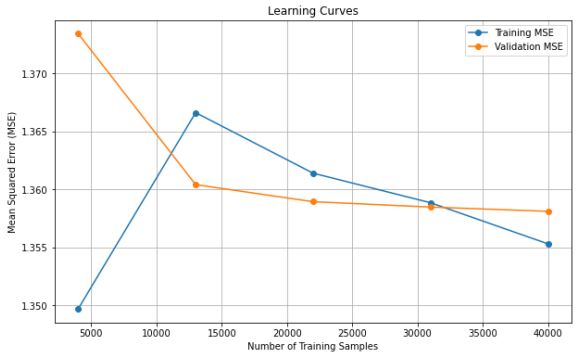
| C | Validation BER | Test BER |
|---|---|---|
| 100 | 0.363 | 0.363 |

The MSE of Model 1 was reported to be: 1.3881

Extensions of Model 1 included the same Linear Regression model but implemented with a Lasso and Ridge regularization technique. The results are as follows:

| Regularizer | Alpha | MSE |
|---|---|---|
| Lasso | 0.1 | 1.284 |
| Ridge | 1.0 | 1.083 |

Below is a graph that illustrates the learning curve of our first model. From the plot, you can see that as we continue to train and validate the data, the MSE begins to drop.




### Model 2: Sentiment Analysis

To implement Model 2, we performed sentiment analysis on the review and response text and assigned a sentiment score to both using NLTK's vader library. We then implemented a Ridge regularized Regression model, achieving the following train and validation MSE scores:

| | **MSE** |
|---|---|
| Training Set | 0.9097 |
| Validation Set | 0.9109 |

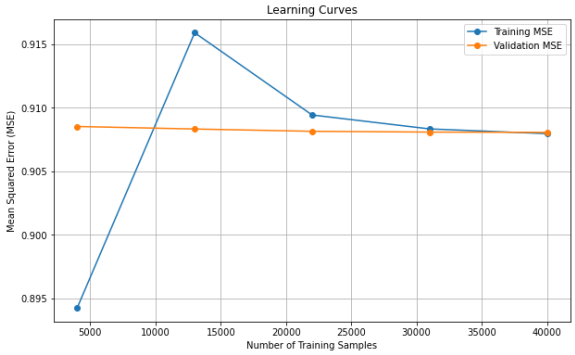
Next, we tuned the hyperparameters and recorded the following scores for *C, Best Validation BER, and Best Test BER:*

| **C** | **Validation BER** | **Test BER** |
|---|---|---|
| 0.01 | 0.1880 | 0.1882 |

Model 2 offered an improved performance over Model 1 and reported the following accuracy scores:

| **Model 2** | **Accuracy Score** |
|---|---|
| Validation Accuracy | 0.91448 |
| Test Accuracy | 0.9141 |

The learning curve of our model 2 exemplifies the improvement of our model's accuracy. The validation MSE follows a horizontal slope very closely. This demonstrates that our training data was successful in narrowing the model's MSE.




**Model 3: Combining significant coefficients from Model 1 with Sentiment Scored from Model 2**

Model 3 is an extension of Model 2 and incorporates statistically significant features from Model 1, in terms of coefficients, with the sentiment scores from Model 2 to improve accuracy and performance. For Model 3, the feature matrix consisted of sentiment scores for review and response variables, review length (normalized by max review length), one-hot encoding of the 'hour' extracted from the date, and a binary representation of if pictures were attached. We performed a Ridge Regularization on the Regression model and achieved the following scores for training and validation MSE

| **Model 3** | **MSE** |
|---|---|
| Training Set | 0.8386 |
| Validation Set | 0.8354 |

After tuning the hyperparameters, we achieved the following *C, Best Validation BER, and Best Test BER:*

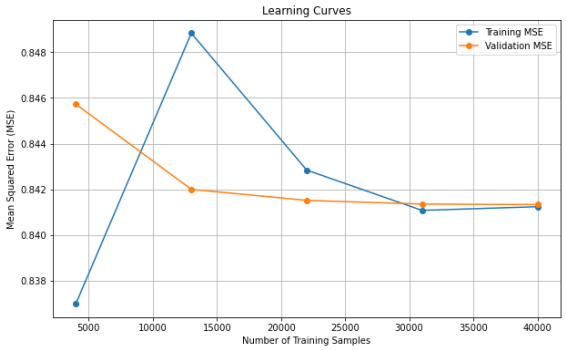
| **Best C** | **Best Validation BER** | **Best Test BER** |
|---|---|---|
| 100 | 0.1625 | 0.1637 |

The Accuracy scores for Model 3 were reported as follows:

| **Model 3** | **Accuracy Score** |
|---|---|
| Validation Accuracy | 0.91752 |
| Test Accuracy | 0.91676 |

Although Model 3 had slightly better performance scores, our validation dataset still had more of a learning curve than Model 3. This

is likely due to the increased number of predictor variables.




**Model 4: Appending Jaccard Similarity between businesses to Model 3**

Model 4 builds on the improved performance of Model 3 and appends item-item similarity between businesses to the feature matrix. The item-item similarity was computed using Jaccard similarity of the form:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

For this calculation, we leveraged the 'gmap_id' column in the dataset which represents a unique identifier of the businesses listed in the dataset. We appended the Jaccard similarity of the businesses to the feature matrix from Model 3 and achieved the following results:

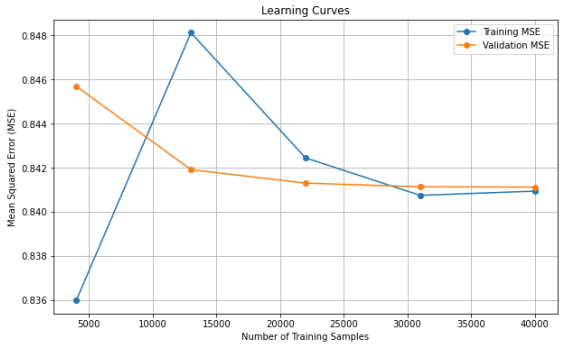
| **Model 4** | **MSE** |
|---|---|
| Training Set | .8384 |
| Validation Set | .8352 |

| **Best C** | **Best Validation BER** | **Best Test BER** |
|---|---|---|
| 100 | 0.1623 | 0.1638 |

The Accuracy scores for Model 4 were reported as follows:

| **Model 4** | **Accuracy Score** |
|---|---|
| Validation Accuracy | .91738 |
| Test Accuracy | .91664 |

The learning curve of Model 4 closely resembles the plot from Model 3. This is expected considering the accuracy, MSE, and BER scores were almost identical. The resemblance of the results and plots implies that Jaccard similarity is not a significant predictor of ratings if sentiment scores were already included in the model.




## 6. CONCLUSIONS

From the four models we tested, we can conclude that sentiment scores were by far the most successful in predicting business ratings. When implementing additional predictor variables and Jaccard similarity to our sentiment model, we found that the performance scores did improve, but not at a significant amount. Upon further investigation of our Model 2, we discovered the length of response, month, and weekday had low magnitudes and significance when predicting ratings. Future explorations would include determining if a bias led to such low errors and high predictions. Although we shuffled the data to try to minimize order bias, it is possible our sample was unbalanced.

## REFERENCES


[1] UCTopic: Unsupervised Contrastive Learning for Phrase Representations and Topic Mining
Jiacheng Li, Jingbo Shang, Julian McAuley
Annual Meeting of the Association for Computational Linguistics (ACL), 2022

[2] Personalized Showcases: Generating Multi-Modal Explanations for Recommendations
An Yan, Zhankui He, Jiacheng Li, Tianyang Zhang, Julian Mcauley
The 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2023

[3] Fang, B., Ye, Q., Kucukusta, D., Law, R., 2016. Analysis of the perceived value of online tourism reviews: influence of readability and reviewer characteristics. Tour. Manag. 52, 498–506.

[4] Kozinets, R.V., 2016. Amazonian forests and trees: multiplicity and objectivity in studies of online consumer-generated ratings and reviews, a commentary on de Langhe, Fernbach, and Lichtenstein. J. Consum. Res. 42 (6), 834–839.

[5] Liu, B., 2012. Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol. 5 (1), 1–167.

[6] Liu, Y., Teichert, T., Rossi, M., Li, H., Hu, F., 2017. Big data for big insights: investigating language-specific drivers of hotel satisfaction with 412,784 user-generated reviews. Tour. Manag. 59, 554–563

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. 10th Int. Conf. on World Wide Web, pages 285–295, 2001

[8] Y. Koren. Collaborative filtering with temporal dynamics. In KDD ’09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 447–456, New York, NY, USA, 2009. ACM

[9] Cremonesi, Paolo & Koren, Yehuda & Turrin, Roberto. (2010). Performance of recommender algorithms on top-N recommendation tasks. RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems. 39-46. 10.1145/1864708.1864721.

[10] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS), 22(1):143–177, 2004.

[11] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. IEEE Computer, 42(8):30–37, 2009

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of Dimensionality Reduction in Recommender System-A Case Study. Defense Technical Information Center, 2000.

[13] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. Proceedings of KDD Cup and Workshop, 2007.