California’s Personalised Yelp Rating Prediction

ABSTRACT

Predictive modelling has become ubiquitous in anticipating future behaviour outcomes using Machine Learning and data mining techniques. In this study, we use the Yelp dataset for predicting the ratings given by users to certain businesses. We exploit not just user’s and business’ features but also associations between a user and its friends to predict ratings, which provides us with a plethora of personalised user data. Further, to not just build a black box model of a user’s ratings for a particular business but also explain the behind the scenes, we employed SHAP as an intuitive explanation tool. We use a wide range of features for users, businesses and their interactions, along with multiple models, for instance, Linear Regression, Ridge Regression, Decision Trees, RandomForest, Collaborative Filtering, Multi-Layered Perceptron and XGBoost. We compare the performance of these models while deciding the significance of different features. Our model finally predicts the ratings for businesses in California with an MSE of 1.181 on the test data, which performs better than the baseline mean rating MSE of 2.0039.

1 INTRODUCTION

Yelp dataset is a collection of user ratings for businesses across 11 metropolitan areas. Users rate businesses on a scale of 1-5 along with submitting text reviews and tips. Users can also interact by following each other’s reviews and ratings, along with categorising fellow users as Useful, Cool and Funny. This data is available at [14] and we use this after pre-processing and feature extraction.

Recommendation Systems have seen increased popularity with tech giants like Amazon, Meta, Netflix and TikTok utilizing them to propose similarity based suggestions to users. Recommendation approaches also include predicting the rating that a user would give to a particular item and then suggest potentially high rated items to the user.

Widely used baseline for recommender systems is the mean rating and bias-only latent factor models, and further extending it to collaborative filtering approaches. If the data includes features for users and items, regression based approaches can also be utilised. Multiple models including ensemble and Deep Learning based approaches can also be used to propose similarities to users.

2 LITERATURE REVIEW

Review rating prediction is one of the most widely used recommendation tasks. Researchers have taken various approaches to tackle this problem. While some of them worked with the traditional approaches[11][10][9], others tried the learning based methods [7][15][12].

2.1 Dataset Used

We perform the rating prediction task on the Yelp dataset [14]. Using feature engineering, exploratory data analysis and outlier removal after looking at the trends in the data, we extract the effective features.

While the Yelp dataset is popular in academia and personal projects, it has been studied in a variety of different ways. Our problem of predicting what rating a user would provide to a business in California is not widely studied in past literature.

2.2 Similar Datasets

An adaptation of the Yelp dataset for binary sentiment classification was presented by HuggingFace. It is constructed by considering stars 1 and 2 negative, and 3 and 4 positive. This dataset was first used as a text classification benchmark in[16].

2.3 Rating Prediction Using Textual Data

The authors in[2] took upon the challenge of predicting ratings by utilising the user’s textual reviews for that business, naming it as Review rating prediction. Through this, they aimed to identify whether it is enough to look at the star ratings of a product and ignore its textual reviews. Their methodology involved combining four feature extraction methods, (i) unigrams, (ii) bigrams, (iii) trigrams and (iv) Latent Semantic Indexing, with four machine learning algorithms, (i) logistic regression, (ii) Naive Bayes classification, (iii) perceptrons, and (iv) linear Support Vector Classification.

The problem pursued in [3] - Predicting a Business Star in Yelp from Its Reviews Text Alone, also primarily involved textual data. They combined the unigrams model with feature engineering methods such as Parts-of Speech tagging, and used linear regression, support vector regression and decision trees for prediction.

2.4 Sentiment Analysis and its State of the art methods

Owing to the popularity of the task of Sentiment Analysis on Yelp Binary classification, a number of state of the art methods have been presented in recent times. [13] proposed XLNet - a Generalized Autoregressive Pretraining Model for Language Understanding in a recent NeurIPS 2019 conference. It enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order and overcomes the limitations of BERT thanks to its autoregressive formulation.

2.5 Harvard Business School study on impact of reviews on ratings

A research done by Harvard Business School [4] investigated if online consumer reviews affect restaurant demand using a novel dataset combining reviews from the website Yelp.com and restaurant data from the Washington State Department of Revenue. By studying the causal impact of Yelp ratings, it is found that consumer response to a restaurant’s average rating is affected by the number of reviews and whether the reviewers are certified as “elite” by Yelp, but is unaffected by the size of the reviewers’ Yelp friends network.

While this study deems the ‘eliteness’ of a user at a particular instance of time important, we studied the number of times a user has been awarded with the elite tag as a feature. Not so surprisingly, this feature was amongst the top 20 features in our model. This means that a user's liking for a business is affected by the number of times he/she was awarded as an elite user.

2.6 State of the art methods employed in our problem statement

In a problem statement similar to ours on Yelp User Rating Prediction, a hybrid approach of combining collaborative filtering and content-based filtering is employed. Specifically, they trained a KNN model first and obtained predictions for each users based on their neighbours. The prediction was then added to MNOLR as a feature to recalculate the proportional odds model. The prediction was also added to BDTR for comparison. For content-based filtering, Ordinal Multinomial Logistic Regression and Binary Decision Tree Regression are used and for Collaborative Filtering, Item-based K Nearest Neighbors (KNN), User-based K Nearest Neighbors (KNN), Matrix Factorization are tested.

3 DATASET

3.1 AllTrails

Most of us in the project group being hikers, we started off with an idea of building recommender systems for trails based on the difficulty level (Easy, Moderate, Hard), length of trail, the previous ratings, previous number of completions of the trail and a few other intuitive features. For this, we began scraping data from All Trails [1] - a popular website with a huge corpus of data on trails around the world using a Python Web Scraping library called Selenium. However, our efforts were met in vain when the website blocked our IP address and asked us to complete a puzzle for every few records. We tried reaching out to the AllTrails CEO - Ron Schneidermann, to be able to secure some data but didn't hear back.

3.2 Yelp Dataset

We then shifted to the second favourite activity of ours - food. The Yelp Dataset is a subset of the businesses, reviews, and user data on Yelp for use in personal, educational, and academic purposes.

A previous and much shorter version (eliminating a number of features) of this dataset was initially used in a Kaggle competition that Yelp conducted for recruiting in the Yelp data mining team - https://www.kaggle.com/c/yelp-recruiting. However, the problem statement picked up here was completely different from ours. Their prediction task involved finding how many "useful" votes a Yelp review will receive.

The entire yelp dataset consists of 6,990,280 reviews, 150,346 businesses, 200,100 pictures, and has the data from 11 different metropolitan areas. In order to get a reasonable length dataset that can be used for prediction, we filter it to use businesses from California, and consider all users which rate these businesses. Finally our data for California has 348,856 reviews, 5203 businesses, 155,948 users.

The extended and updated version of the dataset we are making use of in our project can be found here [14]. It provides us with 6 files:

3.2.1 *business.json.* Contains business data including

(1) Location Data: address, city, state, postal_code, latitude, longitude
(2) Attributes: ByAppointmentOnly, BusinessAcceptsCreditCards, NoiseLevel etc.
(3) Categories: Restaurants, Shopping, Food, Home Services, Health & Medical etc.
(4) Hours of operation, per day
(5) Stars
(6) Review_count

3.2.2 *review.json.* Contains full review text data including

(1) User_id that wrote the review
(2) Business_id the review is written for
(3) Stars accompanying the review text
(4) How the review was received by others - useful, funny, cool
(5) Date and time the review was written

3.2.3 *user.json.* User data including

(1) User's friend mapping (IDs of friends)
(2) User's fans
(3) All the metadata associated with the user
(4) Yelping_since: The year user joined the platform
(5) Elite_Years in which the user was nominated and awarded to be an elite user
(6) Average_stars: Average stars given by the user to different businesses
(7) Received feedback on reviews 'useful', 'funny', 'cool'
(8) Compliments: 'compliment_hot', 'compliment_more', 'compliment_profile', 'compliment_cute', 'compliment_list', 'compliment_note', 'compliment_plain', 'compliment_cool', 'compliment_funny', 'compliment_writer', 'compliment_photos'

3.2.4 *checkin.json.* Checkins on a business.

(1) Contains a comma-separated list of timestamps for each checkin, each with format YYYY-MM-DD HH:MM:SS

3.2.5 *tip.json.* Tips written by a user on a business. Tips are shorter than reviews and tend to convey quick suggestions. The file contains:

(1) user_id
(2) business_id
(3) text
(4) date of tip
(5) compliment_count

3.2.6 *photo.json.* Contains photo data including the caption and classification (one of "food", "drink", "menu", "inside" or "outside").

Out of these 6 files, we picked up the data on businesses, users, reviews, and tips to work with.

4 EXPLORATORY DATA ANALYSIS

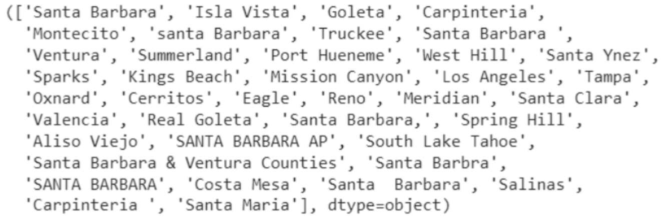
4.1 Businesses Data

4.1.1 *Cities.* The Cities in the dataset, being different in capitalization, spaces, and filled with spelling errors, required some Data Cleaning.

Before cleaning cities data: Figure 1

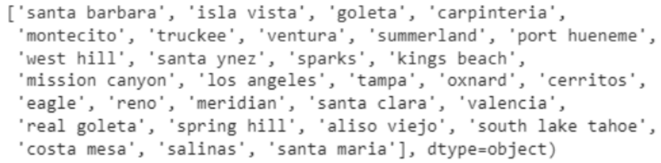
After cleaning cities data: Figure 2

Observing the number of businesses per city data, we could see that the data was highly skewed towards Santa Barbara. For the rest of the cities, there was less or negligible data available. Figure 3



```
(['Santa Barbara', 'Isla Vista', 'Goleta', 'Carpinteria',
  'Montecito', 'santa Barbara', 'Truckee', 'Santa Barbara ',
  'Ventura', 'Summerland', 'Port Hueneme', 'West Hill', 'Santa Ynez',
  'Sparks', 'Kings Beach', 'Mission Canyon', 'Los Angeles', 'Tampa',
  'Oxnard', 'Cerritos', 'Eagle', 'Reno', 'Meridian', 'Santa Clara',
  'Valencia', 'Real Goleta', 'Santa Barbara,', 'Spring Hill',
  'Aliso Viejo', 'SANTA BARBARA AP', 'South Lake Tahoe',
  'Santa Barbara & Ventura Counties', 'Santa Barbra',
  'SANTA BARBARA', 'Costa Mesa', 'Santa  Barbara', 'Salinas',
  'Carpinteria ', 'Santa Maria'], dtype=object)
```

Figure 1



```
['santa barbara', 'isla vista', 'goleta', 'carpinteria',
 'montecito', 'truckee', 'ventura', 'summerland', 'port hueneme',
 'west hill', 'santa ynez', 'sparks', 'kings beach',
 'mission canyon', 'los angeles', 'tampa', 'oxnard', 'cerritos',
 'eagle', 'reno', 'meridian', 'santa clara', 'valencia',
 'real goleta', 'spring hill', 'aliso viejo', 'south lake tahoe',
 'costa mesa', 'salinas', 'santa maria'], dtype=object)
```

Figure 2

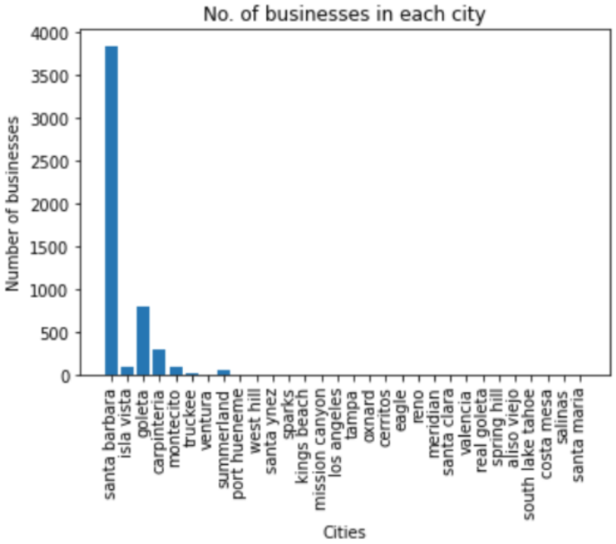


Figure 3

While Santa Barbara had the highest number of reviews, it was observed that a few other cities had the highest rating. This is the result of the lack of data for other cities, where just one rating of 5 is determining the average rating for the whole city.

We thus deemed city to not be a worthy feature in our prediction task. Figure 4, Figure 5

4.1.2 Postal Codes. The postal codes displayed a similar trend as the cities, with most of them being located in Santa Barbara, leading us to not using it as an input feature.

4.1.3 Is Open. We also observed that about 1100 businesses had closed down when the data was collected, and about 4000 were open.

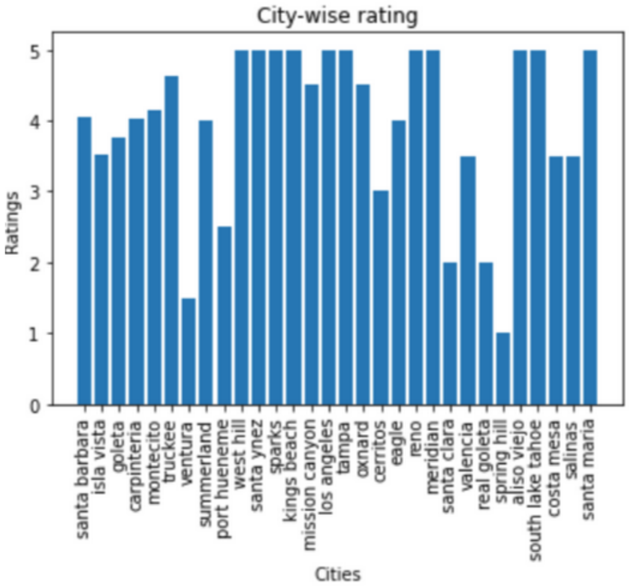


Figure 4

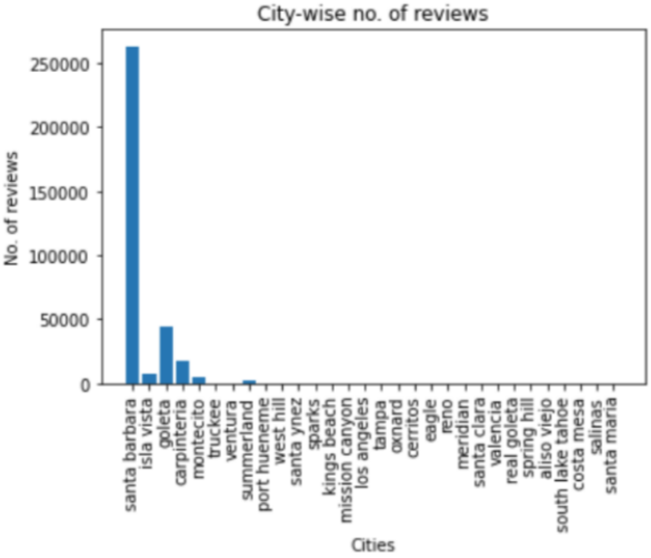
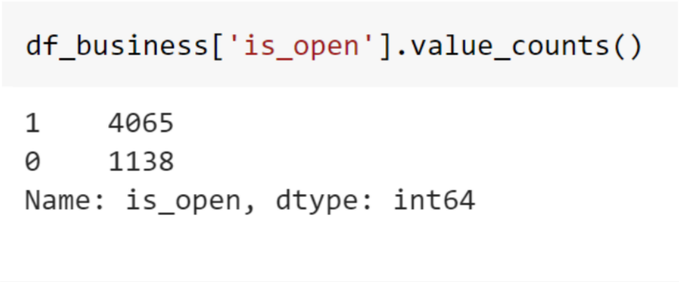


Figure 5

4.1.4 Categories. The business to category was a one to many relationship in the dataset, with 985 unique category values. We observed the category-wise count of those categories that contain at least 150 businesses, in the form of bar plots.

We also observed that about 1100 businesses had closed down when the data was collected, and about 4000 were open.

It is evident that restaurants are the most popularly reviewed business category on Yelp, followed by Shopping, Food, and Home Services. Figure 6



```
df_business['is_open'].value_counts()
```

```
1    4065
0    1138
Name: is_open, dtype: int64
```

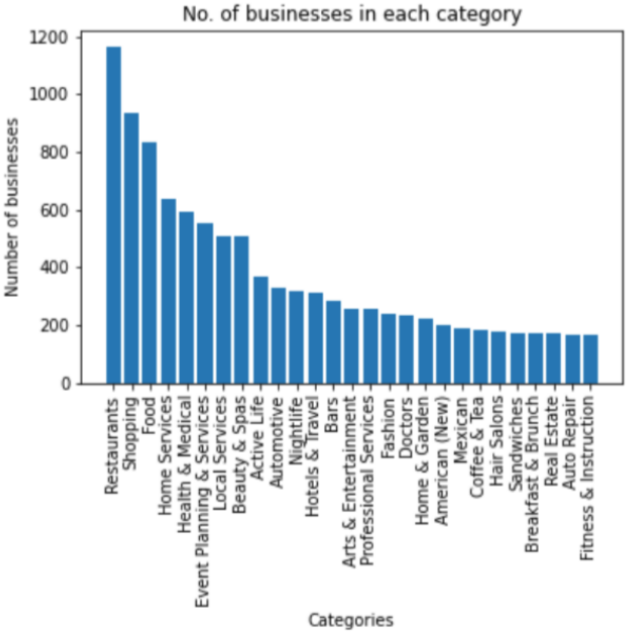


Figure 6

Most of the business categories have an average rating between 4 and 4.5 and total review count less than or equal to 1000. Figure 7, Figure 8

4.1.5 Ratings/Stars. Majority of the records were of a rating of 4.5, which was quite surprising as we expected the ratings to follow a normal distribution that peaks around 3. Figure 9

4.1.6 Ratings vs Reviews. It was also observed that the rating of 4 was accompanied by the most number of textual reviews by the users. Figure 10

4.2 Reviews Data

4.2.1 Rating Distribution. The distribution of the star ratings is depicted in Figure 11. Interesting thing to note is that the star distribution is skewed towards 5 star rating.

4.2.2 Average Review Length for Different Star Ratings. Average review length for different star ratings is plotted in Figure 12. Interesting thing to note is that average review length is high for less rated comments. We can conclude that people who are unsatisfied with the service tend to write more.

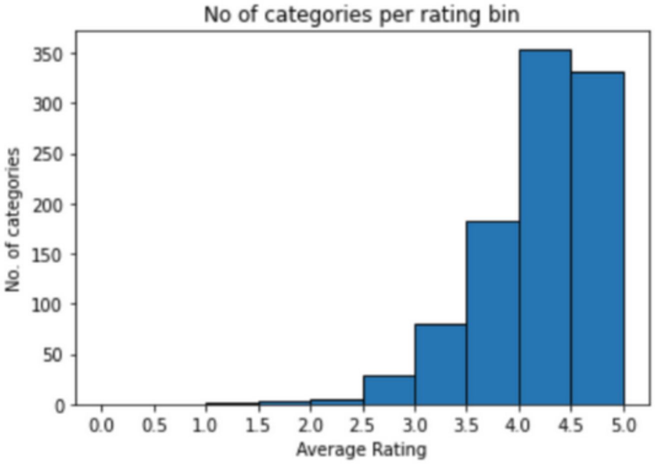


Figure 7

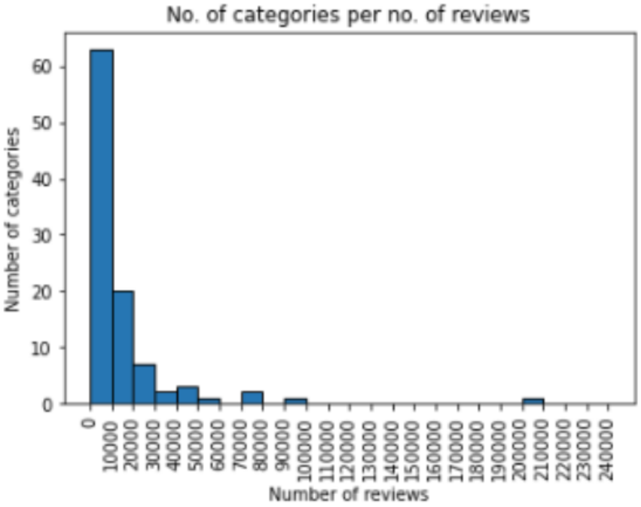


Figure 8

4.2.3 Number of Reviews for over years. Number of reviews over different years is plotted in Figure 13. Interesting thing to note is that number of reviews was increasing up till 2020, and then number of reviews started dropping. This can be correlated with the onset of COVID-19 pandemic - as lesser people visited businesses because of the lockdowns.

4.2.4 Number of reviews for different hours. Total number of reviews at different hours of the day (Figure 14). It can be deduced that people tend to enter reviews between late night and early morning. This could be an important observation as Yelp can scale their backend dynamically to support customer traffic.

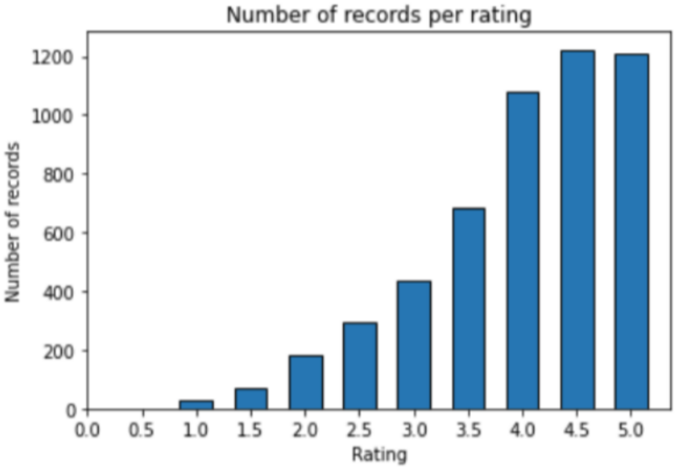


Figure 9

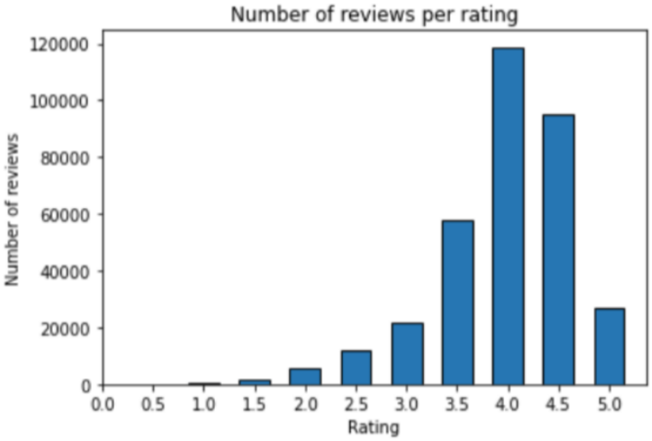


Figure 10

4.2.5 Average Star Rating for Useful/Funny/Cool Comments. Average star rating for useful/funny/cool and not useful/funny/cool comments (Figure 15).

4.2.6 Average Review Length for Useful/Funny/Cool Comments. Average star rating for useful/funny/cool and not useful/funny/cool comments (Figure 16).

4.2.7 Average Rating for different hours. As shown in (Figure 17), people who review late night or early morning, tend to give higher ratings.

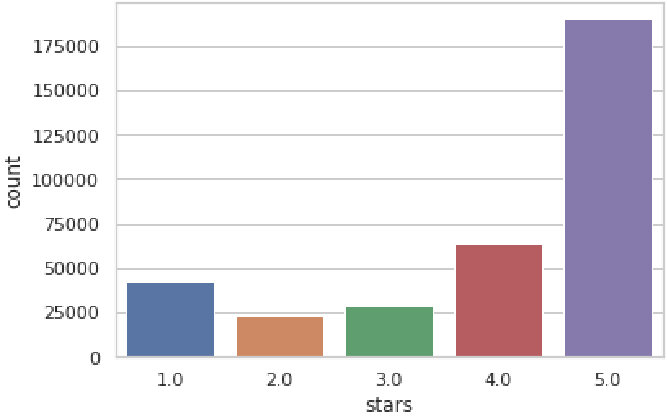


Figure 11

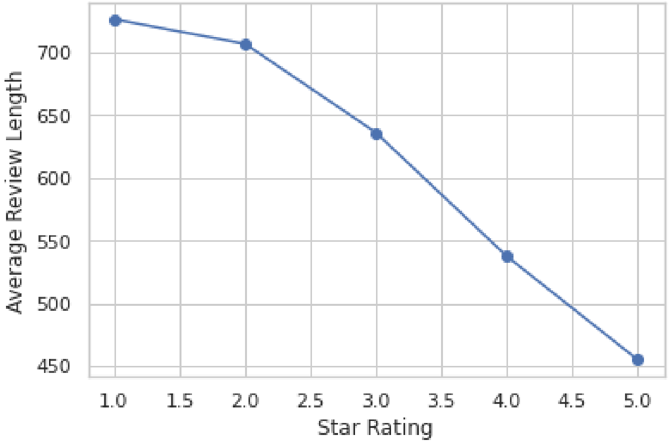


Figure 12

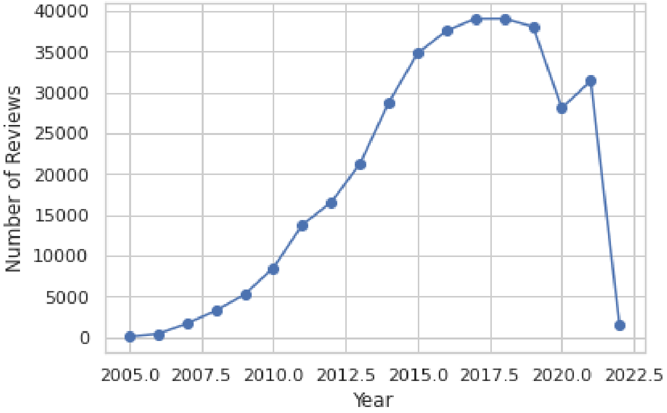


Figure 13

4.3 Tips Data

Tips are considered as short reviews that a user might give to a business. As shown in the Figures 18 19 20 the pattern followed by tips is similar to that observed in the reviews. With one interesting thing to note is that in Figure 20 the number of tips for business categories is significantly larger for categories that include food, shopping, etc which is also intuitive because there are only a few

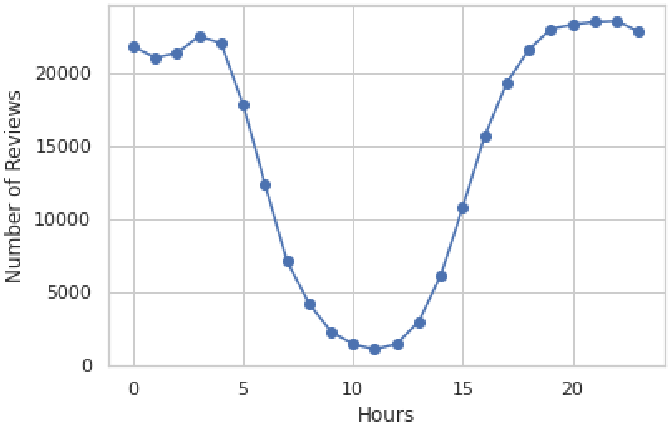


Figure 14

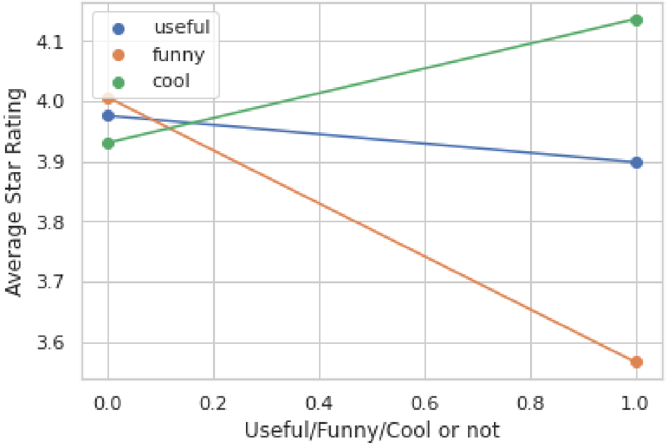


Figure 15

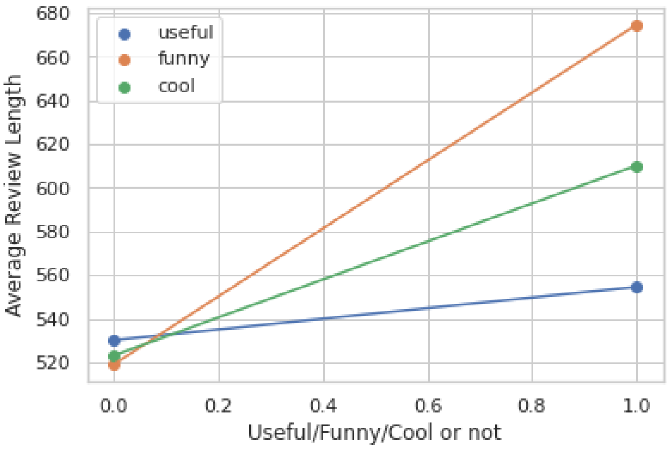


Figure 16

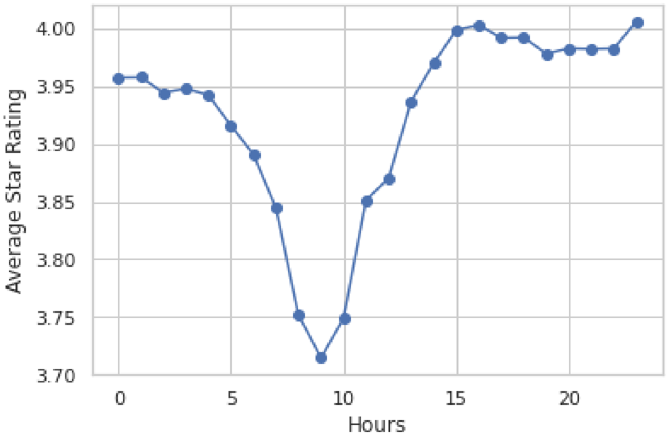


Figure 17

instances when a person writes up a tip for business categories like hospitals etc.

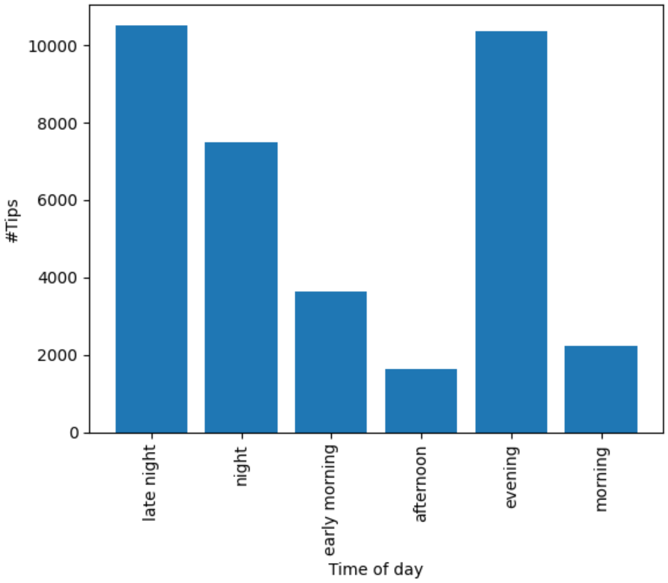


Figure 18

4.4 User Data

In this section, we plot the user data to understand the different features and the range in which the particular features vary. We thus determine the appropriate number of bins of these features. Following are the figures that we plot.

Fine binned histogram for number of friends: (Figure 21)
Coarse binned histogram for number of friends: (Figure 22)
Fine binned histogram for number of reviews: (Figure 23)
Coarse binned histogram for number of reviews: (Figure 24)
Average Rating given by users per year: (Figure 25)

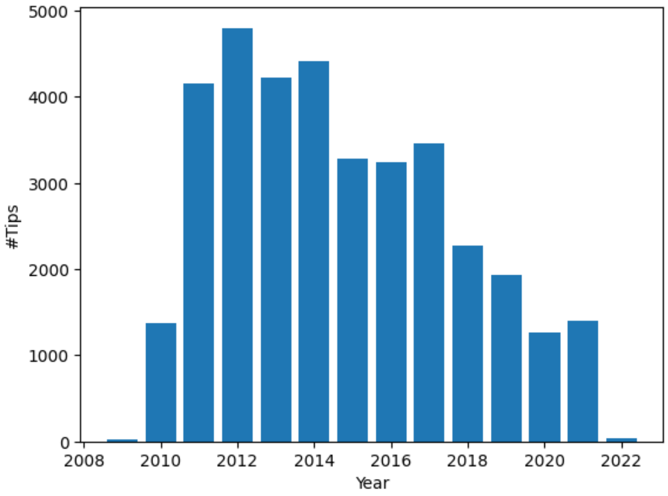


Figure 19

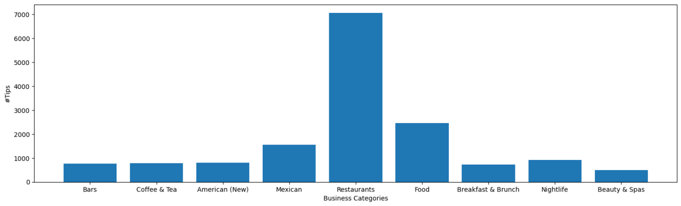


Figure 20

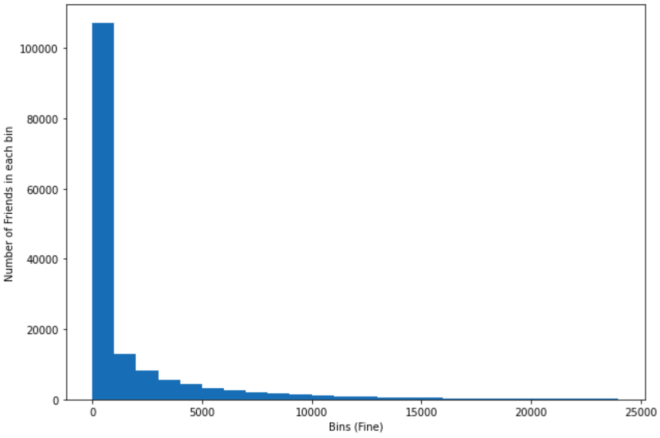


Figure 21

Fine binned histogram for number of Fans: (Figure 26)
Coarse binned histogram for number of Fans: (Figure 27)
Number of users per year: (Figure 28)

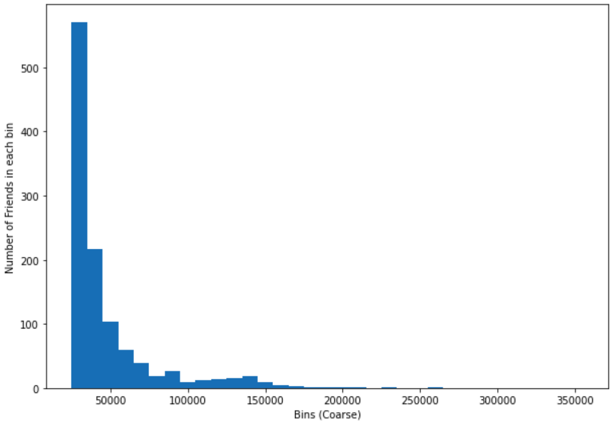


Figure 22

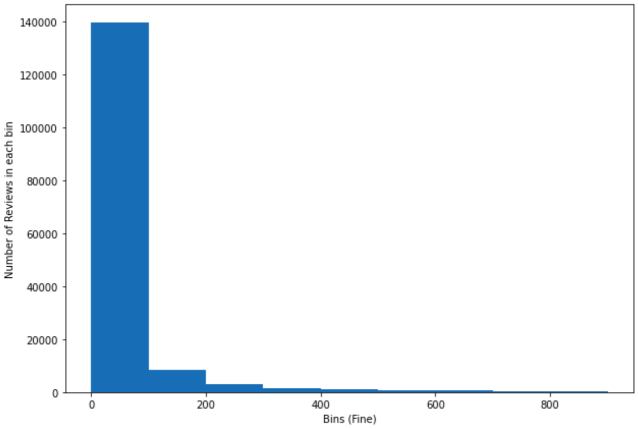


Figure 23

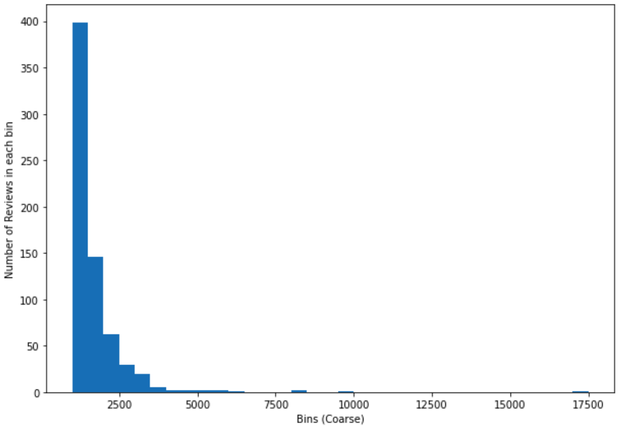


Figure 24

5 PREDICTIVE TASK

The predictive task picked up for our project is to predict the rating a particular user will give to a particular business. This can further be extended to recommend businesses to a user depending on the highest ratings for the category of business the user wished to

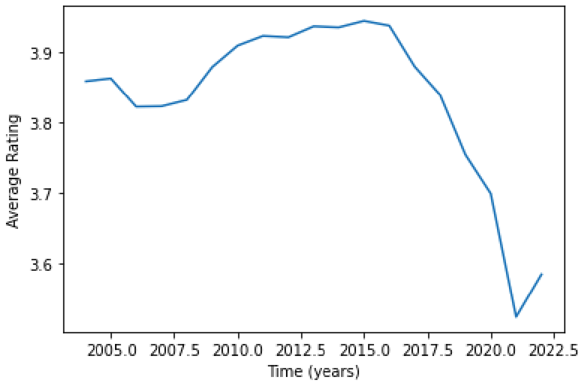


Figure 25

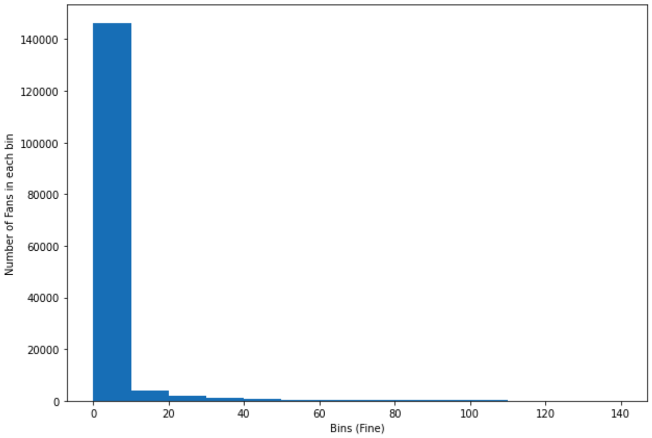


Figure 26

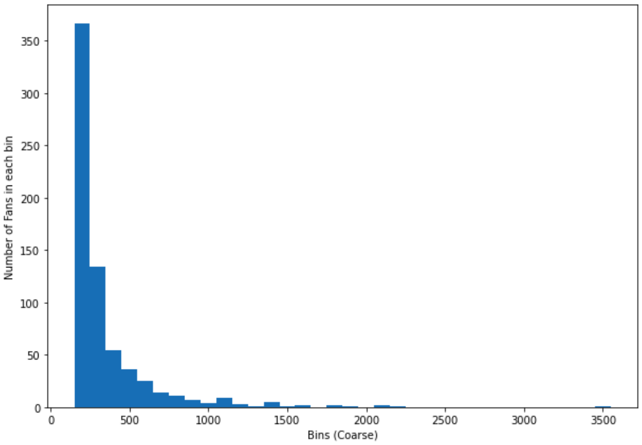


Figure 27

visit i.e if a user wishes to visit a restaurant, he/she will only be suggested the predicted top ratings in the restaurant category.

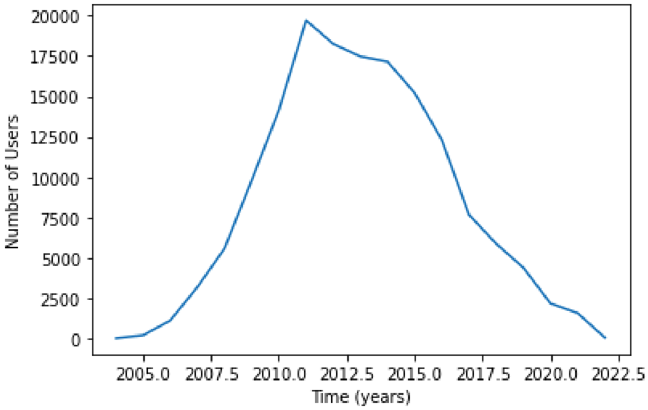


Figure 28

Taking just the Business ID and User ID, our task is to predict the rating/ stars that the user will allot to the business. Through this, we'll be able to determine whether a user will like a business or not and whether he/she should be recommended the same.

5.1 Evaluation Metrics

We use MSE loss for training and validation of our models. According to research on evaluating recommendation systems [8], RMSE and MAE are two popular metrics used for evaluating recommendation systems. We thus observed the results of our model on the 3 metrics, MSE, MAE and RMSE, and reported the results in 8.

$$MSE : \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{1}$$

$$MAE : \frac{1}{N}\sum_{i=1}^{N}|x_i - y_i| \tag{2}$$

$$RMSE : \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(x_i - y_i)^2} \tag{3}$$

5.2 Relevant Baseline

The baseline for our model for predicting business ratings out of 1 to 5 is a model that only predicts the mean rating of 3.95 each time.

5.3 Assessing the validity of model's predictions

We divided our data into Train, Validation and Test sets where after training our model on the Train Test, we optimised it by performing hyperparameter tuning on the validation set and finally evaluated the model and assessed it's prediction's validity on the Test set.

6 FEATURE ENGINEERING

In this section, we discuss the features we designed, and the heuristics we take into consideration to find optimal features for prediction.

- **Review Count (Business)** (int) Depicts the total number of reviews for the particular business in consideration.

- **Category (Business)** (one hot encoded vector) We have a number of categories provided by users for each business, for instance “Restaurant”, “Shopping”, “Health and Medical”, etc. There are a total of 27 categories, each with at least 150 records, which we use as features of our model by one-hot encoding.
- **Average Stars (Business)** (float) The average star rating for the business in consideration.
- **Operating Hours Per day (Business)** (datetime) Depicts the operating hours of the business in consideration for a week. We tried using the number of hours of operation as a feature. However, out of the total 5200 businesses, more than 900 have nan as the value, so we dropped this feature.
- **Year of Review (Interaction)** (one hot encoded vector) We use the year in which the reviews are given as a one hot encoded vector from the datetime object. Years are in the range 2005-2022, which is a total of 18 values. We also observed the yearly temporal data trend through this feature.
- **Month of Review (Interaction)** (one hot encoded vector) We use the month in which the reviews are given as a one hot encoded vector from the datetime object. For this, we divide the months of the year into 4 quartiles. We also observed the month-wise temporal data trend through this feature.
- **Review Count (User)** (int) Depicts the total number of reviews for the particular user in consideration.
- **Average Stars (User)** (float) The average star rating for the user in consideration.
- **Yelping Since (User)** (int) We try to introduce the notion of expertise of a user by using the (review year - entry year) of the user on yelp. Depicts the reliability of the user while rating businesses.
- **Elite user in current year (User)** (binary value 0-1) Depicts if the user in question is elite for the year of the review. However, this value was 0 for most data samples, so we dropped this feature from consideration.
- **Number of Friends (User)** (int) We have a list of friends of a particular user, and we use the length of this list as a feature.
- **Number of Fans (User)** (int) Depicts the number of fans of a particular user.
- **Useful Reviews (User)** (int) Depicts the number of reviews categorised as ‘Useful’ of a user by other users. This should portray the reliability of the user in rating businesses.
- **Cool Reviews (User)** (int) Depicts the number of reviews categorised as ‘Cool’ of a user by other users. This should portray the reliability of the user in rating businesses.
- **Funny Reviews (User)** (int) Depicts the number of reviews categorised as ‘Funny’ of a user by other users. This should portray the reliability of the user in rating businesses.
- **Number of Elite years (User)** (int) This is the length of the list which gives the number of years in which the user was chosen as elite.
- **Average rating of immediate friends (User)** (float) Depicts the average rating of the user’s immediate friends.
- **Average rating of connected component of friends (User)** (float) Depicts the average rating of the user’s friends in the connected components of the graph.
- **Standard Deviation of connected component of friends (User)** (float) Depicts the standard deviation of the user’s friends in the connected components of the graph.

7 EXPERIMENTAL EVALUATION

7.1 Model and Methodology

After applying feature engineering we obtain a dataset of 348856 data points (review data) with 89 features per datapoint. This is divided into 80%, 10%, 10% split for training, validation and test sets. We model the star rating prediction task as a regression problem and tried several models. The validation and test MSE of all these models are reported in the Results section. We discuss few of the models below:

Baseline

The mean star rating on the train dataset is used as our baseline model. The mean rating on the train dataset is 3.95.

Interaction Based Models

These models do not use user or item features, and make use only of interactions to predict star rating. We use the following models:

- Collaborative Filtering: We used collaborative filtering with different similarity definitions.
  - Jaccard Similarity
  - Cosine Similarity
  - Pearson Similarity
- Bias Only Latent Factor Model

Few of the above models perform poorly as compared to baseline. Number of users are approximately 150k and the number of reviews are approximately 350k - that implies we have 2.33 reviews per user on average. We have very few interactions per user. Hence this might be the reason why models such as above (that use only interactions) perform poorly.

- Strength: Uses interaction between users and businesses
- Weakness: Does not use features of users and businesses, which in this case severely impacts performance

Regression Based Models

We also tried regression models provided by sklearn

- Linear Regression
- Ridge Regression
- Decision Trees

- Strength: It utilises features of the users and businesses and explores the linearity in the data. Ridge regression or L2 Regularization, adds the “squared magnitude” of the coefficient as the penalty term to the loss function. Decision trees are easy to read and interpret.
- Weakness: Since it’s a linear model, it does not make use of interaction between user and businesses. Decison Trees are highly unstable and highly likely to overfit.

Ensemble Based Models

- XG Boost
- Random Forest

- Strength: Ensemble models makes use of multiple models while prediction. Ensemble models offer a solution to overcome the technical challenges of building a single estimator.

It also reduces the spread or dispersion of the predictions and model performance. Hence it is more robust to overfitting and has better generalisation.

- Weakness: These models are difficult to interpret and not easily explainable i.e they suffer from lack of comprehensibility.

Deep Learning

We use Multilayer perceptron (MLP) as a representative of deep learning models. The input feature of size 89 is mapped to hidden layer of size 100 which is then mapped to output. We employed 'relu' activation for this purpose.

- Strength: It learns a nonlinear function and hence incorporates interaction and features of users and businesses.
- Weakness: Neural Networks are prone to overfitting and are not as interpretable as linear models.

Proposed Model

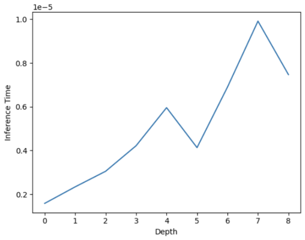
We propose Random Forest as it outperforms all other models on the unseen test dataset and is robust to outliers. Since it's an ensemble based model, combining decisions from multiple models improves the performance.

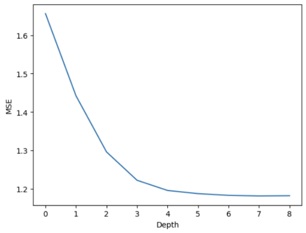
Using interaction based collaborative models was an unsuccessful attempt for this dataset due to reasons described above.

In terms of scalability issues, the initial dataset was too large to load and work with. Thus, instead of loading all at once, we loaded it line-by-line and filtered out the required data. We also created our own subset of the data relevant to the problem at hand. Applying random forest on our cleaned and structured dataset did not cause us any further scalability problems.

Further, overfitting was avoided because of the robust nature of random forest. We also made use of hyperparameter tuning to overcome overfitting.

To optimize our model, we found a balance between the depth of RandomForest and Inference time. As visible in the below figures, the difference in MSE in depth 5 and 8 is insignificant while the inference time is almost double. So to optimize our model we chose the depth to be 5.





7.2 Challenges

The Yelp dataset is an extremely informational dataset with vast opportunities to explore. Due to the presence of multiple within the same dataset, large size of each file, and multiple possibilities of intersections amongst those data files, extracting information from the dataset posed many difficulties. Some of the major challenges are listed below:

(1) The size of the dataset was huge and couldn't be loaded into any of our personal laptops so we had to restructure the dataset and restrict it to keep data related to California only.
(2) After reducing the size there were some columns like postal code and City that were skewed and contained information regarding only 1 postal code and one city i.e. Santa Barbara which led us to drop these columns altogether. As we wanted to use the demographic data for prediction, which could have been a really good feature.

8 RESULTS

8.1 Model Evaluations

We report Validation and Test MSE of all the models we experimented with. Random Forest outperforms all other models on MSE loss as seen in 1

Table 1: MSE Loss (Validation and Test Sets)

Model	Validation MSE	Test MSE
Baseline (mean rating)	1.9686	2.0039
Linear Regression	1.2313	1.236
Ridge Regression	1.2313	1.236
Jaccard Similarity	2.0452	2.1567
Cosine Similarity	2.0529	2.22167
Pearson Similarity	2.9831	3.1514
Bias Only Latent Factor Model	1.6985	1.8892
XG Boost	**1.196**	**1.1842**
Multi Layer Perceptron	1.219	1.25
Random Forest	**1.194**	**1.181**
Decision Trees	2.421	2.4928

In addition to MSE, we also reported MAE and RMSE loss of models on the test set in 2

Table 2: RMSE and MAE Loss (Test Set)

Model	RMSE	MAE
Baseline (mean rating)	1.41560	1.16549
Linear Regression	1.1117	0.8474
Ridge Regression	1.1117	0.84750
Jaccard Similarity	1.4685	1.0846
Cosine Similarity	1.4905	1.1233
Pearson Similarity	1.7752	1.5698
Bias Only Latent Factor Model	1.3745	1.1210
XG Boost	**1.0901**	**0.7919**
Multi Layer Perceptron	1.1180	0.8417
Random Forest	**1.0870**	**0.7811**
Decision Trees	1.5788	0.9693

As expected and further discussed in 7.1, we observe that the similarity based models such as Jaccard, Cosine and Pearson do not perform well on this dataset due to less number of interactions per user. We conclude that the RandomForest model has the lowest loss for all the three metrics used in this study.

8.2 Model Explainability using SHAP

We employed SHAP (Shapley Additive Explanations) [5] - a unified approach to explain the output of any machine learning model, to analyze our best performing rating prediction model.

A summary plot shows the most important features and the magnitude of their impact on the model. The Figure 30 shows the SHAP feature importance for the random forest trained for predicting the rating a user will give to a business.

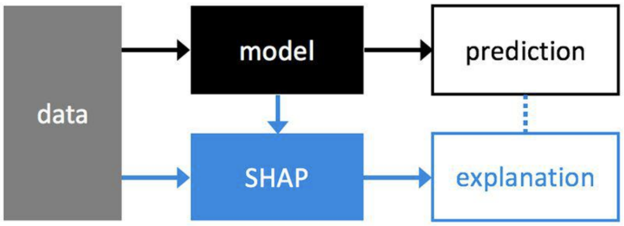


Figure 29
SHAP Working[6]

The Average Stars of a user (Feature 5) was the most important feature, changing the predicted rating probability on average by 45 percentage points (0.45 on the x-axis). It was followed by Average Business Stars, which changes the probability by 31 percentage points (0.31 on the x-axis).

Top_k (sorted) = ['Average Stars of Users', 'Average Business Stars', ' Useful lookup', 'Business reviews', 'User Reviews', 'Fans', 'Cool Reviews', 'Funny Reviews', Average Restaurant Rating', 'Friends', 'Average Friends Rating', 'Yelping since', 'Elite years', 'Average Food Category Rating', 'Average Breakfast And Brunch Category Rating', 'Average Sandwich Category Rating', 'Average Home and Garden Rating', 'Average Event Planning and Services Rating', 'Average Bars Rating', 'Year 2019']

The Figure 31 shows SHAP explanation force plots for a datapoint from the Yelp dataset. It indicates the features and their positive or negative contribution towards the prediction. The user of the data point has a predicted rating of 4.21.

Significance of the results: These results can be interpreted as thresholds for whether or not to recommend a business to a user.

Feature representations that worked well and those that did not: To represent months, we divided them into 4 quartiles of the year and that worked really well for us. Using the Year as a feature was a bit of a task as we debated on created 11 features for the same. After some trials and errors, we figured that One hot encoding the months is the best way to go forward with it.

9 CONCLUSION

To conclude, we finalised the task of rating prediction on Yelp dataset. The main aim of this paper is to create a recommendation system for a user based on its previous interactions. After testing multiple models for e.g. Random Forest, Decision Trees, Regression model, Multi Layer Perceptron etc.with baseline model being the global average, we finally propose a Random Forest model that gives best performance. We use features of businness, features of users and interaction between users and business to build a personalized recommender system. Further, with the help of SHAP, we overcame the interpretability issues of ensemble models and visualized importance of each feature of the Random Forest model for the predictive task. We also provide multiple ablation studies to show hyperparameter tuning and optimization on Random Forest model.

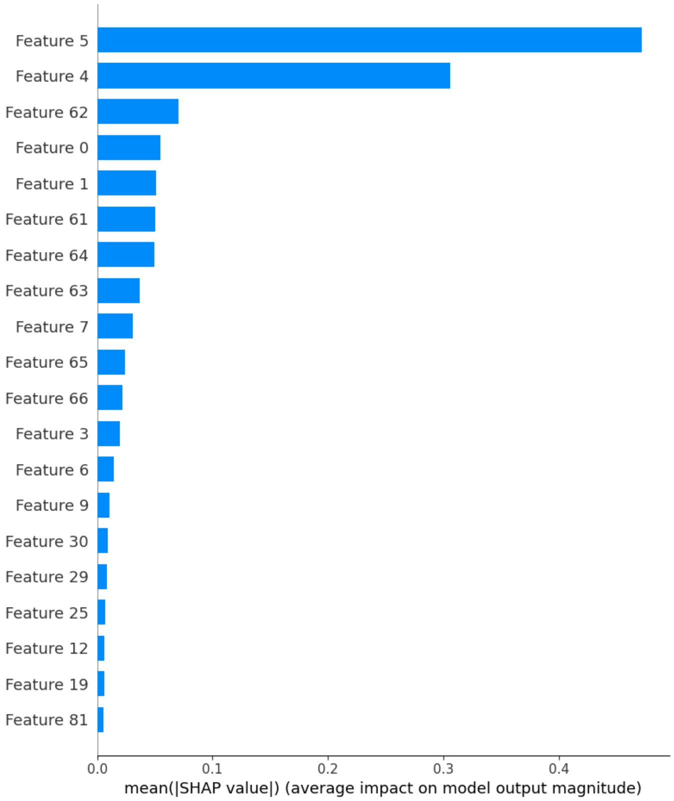


Figure 30

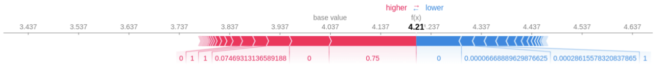


Figure 31

10 FUTURE WORK

The reviews and tips text data holds immense valuable information and can be exploited for the prediction and recommendation tasks in the Yelp dataset. Sentiment Analysis of such text can unearth its associations with ratings. For instance, we can determine what emotions a user expresses for a category of business and use that information to predict ratings for other businesses.

We also saw immense potential in creating a graph network of a user's friends and exploiting the variance between different friend groups to determine the rating.

REFERENCES

[1] allTrails. 2018. all Trails. https://alltrails.com/ [].
[2] Nabiha Asghar. 2016. Yelp Dataset Challenge: Review Rating Prediction. *ArXiv* abs/1605.05362 (2016).
[3] Mingming Fan and Maryam Khademi. 2014. Predicting a Business Star in Yelp from Its Reviews Text Alone. *ArXiv* abs/1401.0864 (2014).
[4] Michael Luca. 2011. Reviews, Reputation, and Revenue: The Case of Yelp.Com. *SSRN Electronic Journal* (09 2011). https://doi.org/10.2139/ssrn.1928601
[5] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[6] Medium. 2018. SHAP Values : The efficient way of interpreting your model. https://medium.datadriveninvestor.com/shap-values-the-efficient-way-of-interpreting-your-model-7de632ed7d2d [].

[7] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (Como, Italy) *(RecSys '17)*. Association for Computing Machinery, New York, NY, USA, 297–305. https://doi.org/10.1145/3109859.3109890

[8] Guy Shani and Asela Gunawardana. 2011. *Evaluating Recommendation Systems*. Springer US, Boston, MA, 257–297. https://doi.org/10.1007/978-0-387-85820-3_8

[9] Wararat Songpan. 2017. The analysis and prediction of customer review rating using opinion mining. In *2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA)*. 71–77. https://doi.org/10.1109/SERA.2017.7965709

[10] Bingkun Wang, Yulin Min, Yongfeng Huang, Xing Li, and Fangzhao Wu. 2013. Review Rating Prediction Based on the Content and Weighting Strong Social Relation of Reviewers. In *Proceedings of the 2013 International Workshop on Mining Unstructured Big Data Using Natural Language Processing* (San Francisco, California, USA) *(UnstructureNLP '13)*. Association for Computing Machinery, New York, NY, USA, 23–30. https://doi.org/10.1145/2513549.2513554

[11] Bingkun Wang, Shufeng Xiong, Yongfeng Huang, and Xing Li. 2018. Review Rating Prediction Based on User Context and Product Context. *Applied Sciences* 8, 10 (2018). https://doi.org/10.3390/app8101849

[12] Chuhan Wu, Fangzhao Wu, Junxin Liu, Yongfeng Huang, and Xing Xie. 2019. ARP: Aspect-Aware Neural Review Rating Prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 2169–2172. https://doi.org/10.1145/3357384.3358086

[13] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf

[14] Yelp. 2018. Yelp Dataset. https://www.yelp.com/dataset [].

[15] Zhigang Yuan, Fangzhao Wu, Junxin Liu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2019. Neural Review Rating Prediction with User and Product Memory. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 2341–2344. https://doi.org/10.1145/3357384.3358138

[16] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf