Understanding the interplay between rating, sentiment score and category for restaurants in Google Local Review

ABSTRACT

With the ever-increasing demand for a spontaneous response for queries raised by users, the need for robust systems to predict outcomes is imperative. The modern world powered by the internet has a high impact on the lives of people on a day-to-day basis ranging from internet searches to suggesting products on e-commerce websites. Consequently, online review forums and blogs act as a critical factor for a user to choose an outcome. In recent times, people tend to evaluate a business based on the reviews it has received on Google. The task of personal or relevant recommendation and determining the ratings or quality of business based on the engagements is the need of the hour. In this work we propose an experiment on how information from multiple points of interests such as restaurant information, GPS, user review sentiments could contribute to the task of rating prediction and suggesting personalized cuisines. We conduct experiments on the Google local reviews dataset. Our proposed strategy exhibits less than 11% inaccuracy in the prediction task, thus showing the effectiveness of the feature extraction strategy.

Keywords

Recommendation, prediction, regression, sentiment analysis.

1. INTRODUCTION

From online platforms such as Google, Netflix, Foursquare to online e-commerce sites such as Amazon, Uber, etc., rating prediction and recommendation algorithms have become a vital part to the success of these businesses. Many traditional and machine learning approaches have been employed to predict the rating of a user for an item based on other user-item interactions or based on popularity of the item. Though these models seem to perform well on user prediction, they fail to account for various contextual information than can otherwise help improve the model's performance.

Using contextual information in the model, adds extra useful features to the model, thus increasing the relevance of the recommendation or rating based on circumstances such as location, time of the day, price of the item, etc. In certain datasets, we even have access to the review text provided by the user. This can provide valuable information such as the average sentimental score, that can improve the model by letting it better understand the dynamics of the user. Though this feature seems to be redundant when the business could directly ask for rating feedback, text in general can provide additional valuable information for the business to consider to improve than just using the average rating.

In this paper, we explore restaurants and user reviews from the Google Local Reviews Dataset. We perform exploratory analysis on each of the features to understand the relevance for the prediction output. This information is used to carefully do feature engineering to better suit the model. The selected features are trained to perform two predictive tasks namely, rating prediction of a user for a restaurant and top categories prediction based on likeliness for a user using various models. The results have been verified using standard performance metrics such as Root Mean Squared Error.

2. RELATED WORK

2.1 Collaborative Filtering

Many machine learning models adapt collaborative filtering methods such as user-based or item-based K-Nearest Neighbors to predict rating. [1] compares these item-item and user-user K-NN algorithms with similarity measures such as Cosine or Pearson and various other matrix factorization methods and report the performance measures of the models using Root Mean Squared Error and Mean Average Error. Though the model works well on predicting the rating of an item or by a user with a good accuracy, it does not take into picture various other information associated with a user or item such as location, price, category of the item, etc. In this paper we explore how these various features can affect the performance of the rating prediction task.

[2] also discusses how the ranking problem can help improve the performance of the rating prediction task of an item. This takes into consideration of all items in the collection, irrespective of whether the item has been rated by the user or not in the model. This method can be extended to our model to consider various other features while predicting the rating that could be potentially given to a restaurant even though there has been no interaction.

2.2 Big Data Review Analysis

Products like Google, Amazon, etc., engages a large number of customers for reviews and ratings for places and items. Every day, exabytes of data are transmitted in the internet, which makes it challenging problem to analyze useful information for business from the reviews. [3] works on helping business by providing a method to analyze large-scale datasets from Yelp and Google using Big Data Hadoop technology owing to its scalability. They analyze various reviews and find useful information such as the most popular category, the most popular location, etc. This information is viable for business to make decisions about spending more money on a particular sector or a specific location. The proposed solution can be further improved by adopting more information from various other features and model them to get a better recommendation.

[4] proposes a restaurant recommendation model which uses location, time and preference of the user. It considers the current geospatial location, historically visited places and the recommendation request hour. All these features give realistic recommendations. Hence similar features can be considered in our work with various modeling strategies to make better restaurant recommendations.

3. DATASET

In this paper, we perform analysis and modeling on the Google Local Reviews Dataset introduced by [5]. This dataset consists of a large collection of users, places and business reviews and rating. In this work, we evaluate datasets containing restaurants.

3.1 Dataset Review

The Google Local Dataset used in this experiment contains information about various business around the world, the reviews and ratings given for those businesses by different users and the corresponding users information. The information is captured in three files with features namely:

- **users**: userName, jobs, currrentPlace, previousPlace, education, gPlusUserId
- **places**: name, price, address, hours, phone, closed, gPlusUserId, gps
- **reviews**: rating, reviewerName, reviewText, categories, gPlusUserId, unixReviewTime, reviewTime, gPlusUserId

The number of datapoints available are shown in Table 1. The gPlusUserId and gPlusPlaceId column can be used to create a join between the three datasets after processing. The dataset can be cleaned based on the required information for the predictive task and also removing datapoints with missing information required for modeling.

Table 1. Basic Statistics of the Dataset

Dataset	Number of entries
Users	3,747,937
Places	3,114,353
Reviews	11,453,845

3.2 Dataset Cleaning

The information from each dataset can be cleaned or removed based on relevance for the required task.

3.2.1 Users

The recommendation task involves predicting information for a restaurant i.e., place. Thus, we can remove all the column information from the users data frame except for the userName and gPlusUserId column.

3.2.2 Places

The relevant columns for the recommendation task from places.json are shown in Table 2. The GPS column can be split into two columns: latitude and longitude after dropping the rows which does not contain a GPS value. The resultant dataset contains 3,087,402 datapoints.

Table 2. Data formula of the places data frame

Name	Description
name	The name of the place
price	Price range of the place ($, $$, R, QQ, etc.)
hours	Hours when the place is open on each weekday
gPlusPlaceId	Unique ID of the place
gps	Location of the place (Latitude & Longitude)

The price column is filtered for $, $$ and $$$ prices. The rows where the hours column is empty can be dropped from the data frame to reduce the size of the data frame to 322,765 rows. The hours column can be processed to get more information for each weekday. The hours column is converted into 7 columns for each weekday and is assigned a value of either 0 (Morning restaurant), 1 (Night restaurant) or 2 (Both morning and night restaurant). The rows where these weekdays columns are NaN can be dropped to ease the predictive task. The resultant data frame contains 264,832 rows. The gPlusPlaceId from the new data frame can be used to filter the reviews dataset. The number of places for each price is shown in Figure 1.

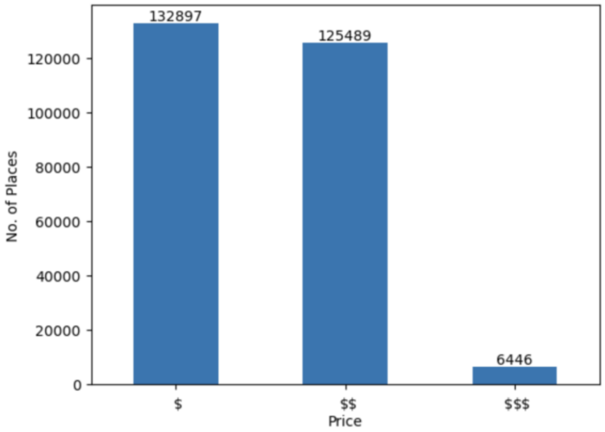

Figure 1. Number of places per price range

3.2.3 Reviews

The reviews dataset is processed initially to load only the entries for the gPlusPlaceId filtered from the places data frame. The reviewerName column is dropped from the data frame since this information is available in the users data frame. The data frame contains reviews and ratings from 1990 to 2014 as shown in Figure 2. The reviewTime column can be used to filter more recent data. In this paper, we only consider the reviews from 2012, 2013 and 2014 which contains 76.52% of the total reviews.

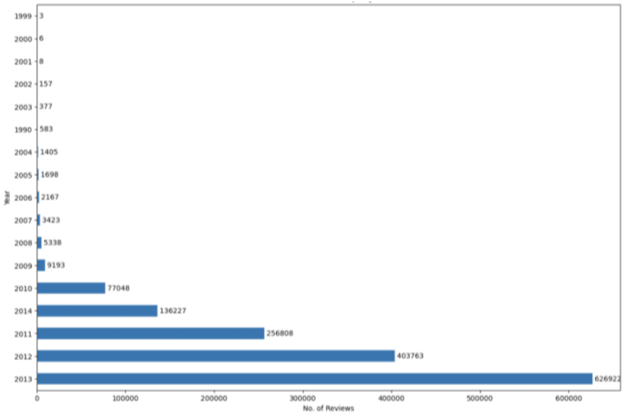

Figure 2. Number of reviews per year

The categories column is processed to filter the reviews with a category with the name restaurant in it. The reviewText information can be used to infer sentiment information for each review, and thus the rows where the reviewText is English are only used. Finally, only the reviews with categories that has a total occurrence of 2500 and above are considered. The total count of reviews of the top 10 categories from the filtered list is shown in Figure 3. The categories column in the data frame is processed to only retain the entries from the filtered top 40 categories. The filtered data frame contains a total of 643,195 rows.

The reviews and places data frame are merged by performing a right join on gPlusPlaceId. The rows with no categories are dropped. The shape of the resultant data frame is (643195, 16).

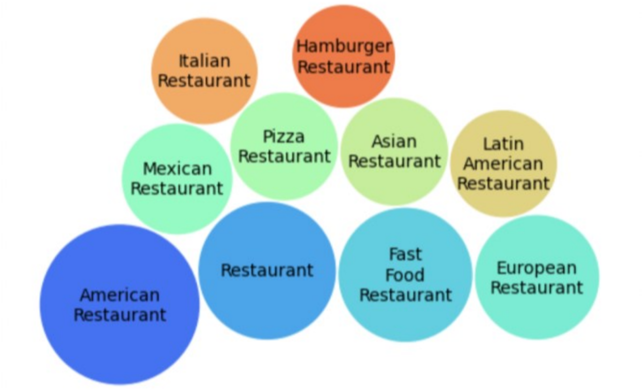

Figure 3. Top 10 categories by number of reviews

4. PREDICTIVE TASK

We will predict the rating of a user for a restaurant given the features and predict the most likelihood category for a given user. Let $U = \{u_1, u_2, u_3, \ldots, u_U\}$ denote the set of users, $P = \{p_1, p_2, p_3, \ldots, p_P\}$ denote the set of places considered and$R = \{r_1, r_2, r_3, \ldots, r_N\}$ denote the reviews made by users in U on Places P. For the experiments only restaurants were considered. Given a user u_i, let $P(i) \subseteq P$ be the set of restaurants visited and reviewed by u_i and $R(i) \subseteq R$ be the reviews made by the user on $P(i)$. A review is a sentence or paragraph written by a user on the restaurant expressing his views based on his experience. Let $P(i) = [p_1^{(i)}, p_2^{(i)}, p_3^{(i)}, p_4^{(i)}, \ldots, p_{f1}^{(i)}]$ denote the features considered for each place which includes, prices, geographical position as mentioned in Section 3.2.2. Similarly let $R(i) = [r_1^{(i)}, r_2^{(i)}, r_3^{(i)}, r_4^{(i)}, \ldots, r_{f2}^{(i)}]$ denote the features considered for each review including average sentiment score, rating, category etc. Each user $u \in U$ is associated with a d-dimensional feature (e.g., user ID, current place, education, jobs etc.) vector $X \in R^{1 \times d}$. The predictive task is divided into two subparts as follows. left.

4.1 Ratings Prediction

This involves predicting the rating of a place based on 'Rating', 'Categories', and 'Price' feature inputs to the model. $Y = \{1, 2, 3, 4, 5\}$ denotes the predictions made by the proposed model where each value indicates the rating given by the user.

Given U, P, and R, this work aims to determine the ratings provided by the user as illustrated in Equation (1).

$$F: \Delta(U, P, R) \rightarrow Y_r \qquad (1)$$

4.2 Categories Prediction

We also try to find similarities between different existing 'Restaurant' categories and trying to recommend the user a category which best suits his taste based on his history of places visited, using 'Rating', 'Categories', and 'Price'. The predicted category would still be of 'Restaurant' type and has 40 possible values. Let the preferred place category for the user recommended by the model be represented as

$$Y_c = \{'American\ Restaurant', 'Asian\ Restaurant', \ldots\}$$

Mathematically, Given U, P, and R, this work aims to determine the preferred categories of the user as illustrated in Equation (2).

$$F: \Delta(U, P, R) \rightarrow Y_c \qquad (2)$$

In the following sections we will discuss further about the features, model and its performance.

5. FEATURE ANALYSIS

For the experiments carried out in this work, we choose relevant features/information from the dataset. To reinforce the claim of choosing a particular feature, the following section talks about the extensive feature analysis carried out on each of the datasets (users, places and reviews) available. In the subsections we analyze significance and intuition behind choosing the features and their relevance to the problem that is focused in this work. that is being solved such as the geographical position of the restaurant (latitude and longitude), user's review about the restaurant, Ratings (Distribution), Users and Places (Normal count) – Anything that would be used significantly in the model should go in here.

5.1 Users and Reviews Information

As mentioned in section 3.2.1, in the user dataset we extract the user Google+ user ID to monitor and map the user information for modeling. There are a total of 3,747,937 unique users found in the dataset. However, for this work, we consider the users who have reviewed a restaurant using English vocabulary. Thus, the subset of users is considered as mentioned in Equation 1. Consequently, 404,901 users were extracted.

$$USERS(data) = \{u \mid u \in U \,\&\, lang(u_{review}) = English\} \qquad (3)$$

5.1.1 Language

The Google Local Reviews dataset contains information collected from various parts of the world. Since, wide range of people from different nationalities and languages are part of this dataset, the data requires cleaning. This work primarily focuses on using the reviews that are in English. To identify the language of the text, we implemented the PolyGlot [6] Python library and obtained the distribution of languages across the entire dataset. Figure 4 depicts the distribution of all the languages and highlights the top 10 most used languages in the review's dataset. Since over 73% of the text are written in English, only these subsets of reviews are considered.

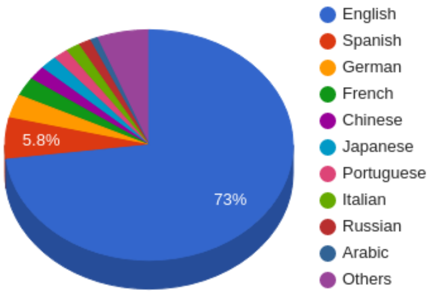

Figure 4. Language distribution across the dataset

5.1.2 Sentiment Score

In [7], the authors use the sentiment scores associated with products sold on e-commerce websites to predict the ratings of the product. Thus, we implement the idea proposed in [7], we analyze the sentiments associated with the review the user had given for the restaurant. To carry out this task, we employ the library functions in TextBlob [8] Python library. The sentiment function returns the polarity of the text. The range of values returned is [-1.0, 1.0] where negative value indicates the sentiment is negative and vice versa. On analysis of the reviews considered we discover that 82.14% of the reviews considered (over 500k reviews) have a positive sentiment. Thus, there is a class imbalance between positive and negative sentiments and the proposed model's robustness can be evaluated on its ability to extrapolate the information from this imbalance. Figure 5, illustrates the average sentiment score for each day of the week.

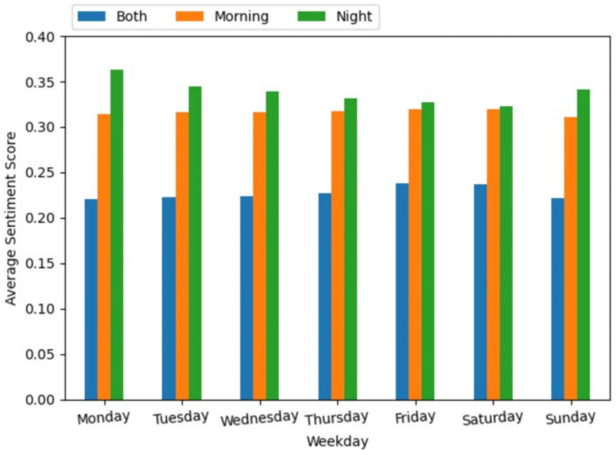

Figure 5. Average Sentiment Score for each day of week

5.1.3 Rating

Figure 6 illustrates the distribution of the ratings data considered. From this, we can infer that there are sufficient data points for each review in the dataset.

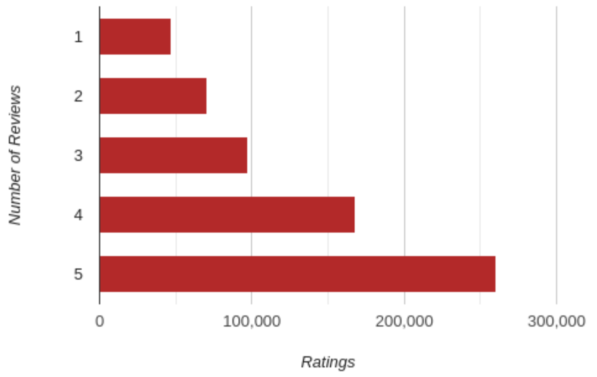

Figure 6. Average User Rating for each day of week

5.1.4 Categories

The Google reviews dataset records place Ids 'gPlusPlaceId' and 'gPlusUserId' and corresponding review details for given by each user to each place. But from the data we see that maximum number of users have rated each place at most once. Hence, to deal with this data sparsity, we group places into their categories and do predictions based on the number of reviews for each category and the number of reviews given by each user. 73.96% of the data corresponds to 'Restaurants', and hence we choose this category as our initial dataset which contributes to 862,661 reviews.

5.2 Restaurant Information

To leverage the restaurant information and its importance to the model we analyze the distribution of its data using plots and figures as discussed in sections below.

5.2.1 GPS (Latitude and Longitude)

The latitude and longitude coordinates provided in the dataset is a crucial feature in determining the rating of a restaurant. This is due to the fact that restaurants that are located in localities accessible by the general public are more likely to be recognized and reviewed. Figure 7 illustrates the distribution of the restaurants found in the dataset based on their location and a world map of the restaurants is sketched.

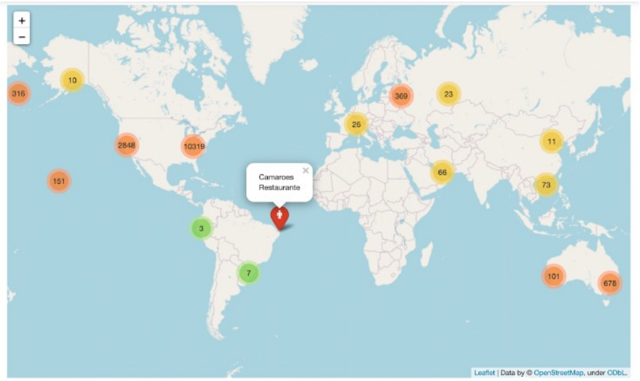

Figure 7. Cluster map with restaurants categorized by price

From Figure 7, we can infer that the restaurants are distributed all over the world and are not concentrated in one particular location. This sparsity is useful in analyzing user and review information specific to the geographic location.

5.2.2 Hours

Another useful feature we hypothesize could be critical is the opening and closing times of the restaurants. The intuition behind using this information as a feature for modeling is that, in countries like India, there are numerous restaurants that run exclusively in the night. The opening and closing timings of the restaurant can potentially influence the users who visit the restaurant. For instance, night shift workers, drivers tend to rate the night-based restaurants more often than the general public. Figure 8 illustrates the distribution of restaurants based on the time its open. From Figure 8, we can infer that the majority of the restaurants are morning-based restaurants.

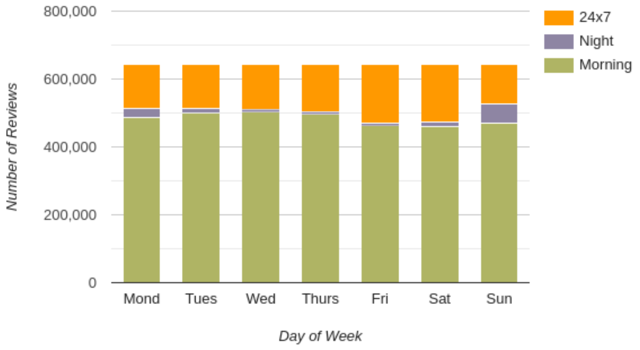

Figure 8: Restaurant timing distribution

5.2.3 Category Tag Information

For this task, we utilized the category information available for each restaurant. Each restaurant can fall under multiple categories. For example, a restaurant has the following categories tags: 'European restaurant', 'Italian restaurant' and 'Pizza restaurant'. On exploratory analysis, we discover that the tags are more generalized for the initial tags and become specific in later tags. In the above example ‘European restaurant’ tag is more general and ‘Pizza restaurant’ is more specific. As mentioned in section 3.2.3, in this work, we consider the businesses that are restaurants and have at least 2500 user reviews. Thus, the number of categories filters down to 40 categories. Now, modeling on these categories becomes a challenging task.

We aim to enable efficient data merging between places features and reviews. We utilize the strategy employed by the authors of [9]. Instead of using data points with the same set of features except one (in this case the places have the same features when we create additional data points with only difference among them being the category type). For instance, a restaurant with 3 categories will have 3 data points instead of 1 with redundant information such as price, latitude and longitude. This could induce some bias in the model. To counteract, we employ the PIVOT operation as suggested by [9] where additional dimensions (columns) are created for each data point (row in data frame). A category column is assigned value 1 if the corresponding restaurant contains the category tag and 0 otherwise. Thus, we obtain a denser representation compared to extrapolating the categories into multiple data points, increasing the time complexity of the model.

6. EVALUATION METRICS

We propose using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to evaluate the performance of the rating predictor model.

RMSE is the square root of the sum of the squared difference between the predicted and actual target variables, divided by the number of data points.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\big(y(i) - y(p)\big)^2} \tag{4}$$

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It’s the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |y(i) - y(p)| \tag{5}$$

7. EXPERIMENTS AND RESULTS

The objective of this work is to predict the ratings of the restaurant based on its features and the reviews it had received from users who visited the restaurant. For carrying out the experiments we implement machine learning models on the dataset. We did not consider using deep learning models such as Recurrent Neural Networks (RNNs), Artificial Neural Networks (ANNs) in this work since they perform better than machine learning models when the dimensionality of the data is high. In this work, we implemented linear model algorithms such as Linear Regression, Lasso Regressions and Ridge Regression. We also implemented Random Forest Regressor to test out the performance of ensemble methods on this data.

7.1 Ratings Prediction

7.1.1 Linear Models

Let X be the set of features and Y be the set of corresponding outputs of X. Linear regression models aim to fit data with coefficients $w = (w_1, w_2, w_3, \dots, w_n)$ to minimize the residual sum of squares between the observed targets Y in the dataset and targets predicted by the linear approximation. In general, it solves Equation 3, where Y is the predicted value, X is the input and b is the bias term. W is the set of parameters or coefficients learnt.

$$Y = W^T X + b \tag{4}$$

Further, regularization can be applied to linear models with lasso regression applying L1 prior as regularizer and Ridge regression applying L2 prior as regularizer. The alpha parameter for the lasso and ridge regressors are 0.1 and 0.5 respectively.

7.1.2 Tree Based Models

Random Forest Regression is a supervised learning algorithm that uses ensemble learning methods for regression. In ensemble learning, predictions from multiple ML algorithms are combined to make a more accurate prediction than a single model. The max_depth parameter for the model was set to 4. Tree based models were considered since they capture the non-linear relationships between variables better than linear models.

7.1.3 Sampling

We know that there is native class imbalance in the dataset from Figure 6. To improve the performance of the model, oversampling and down sampling strategies are applied and the results thus obtained are compared.

Sampling is an active process of gathering observations with the intent of estimating a population variable. In this work, we implement SMOTE oversampling and Near Miss under sampling. The test scores were evaluated and the best model suitable for the given data set was determined from both sampled and unsampled data.

In Over Sampling the minority sample count is increased to match the sample count of the majority sample and in under sampling the majority sample count is brought down to match the minority sample count. In SMOTE the k nearest neighbors of the minority class of each sample is identified and new samples are generated along the lines joining the neighboring point and minority sample.

The NearMiss under sampling method calculates the distances between all instances of the majority class and the instances of the minority class. K instances of the majority class that have the smallest distances to those in the minority class are selected. If there are n instances in the minority class, the nearest method will result in k*n instances of the majority class.

7.1.4 Results

Since we are focused on minimizing error rate, a low RMSE and MAE is desirable. From the Tables 3, 4 and 5, we can infer that the results of both over-sampled and under-sampled data are similar and do not show any improvement. In fact, the model trained on unsampled data performs better than the sampled ones. Unsampled data with random forest regressor gives a RMSE of 0.759 and MAE of 0.581.

Table 3. Unsampled Data Results

Model	RMSE	MAE
Linear Regression	1.044	0.855
Lasso Regression	0.859	0.677
Ridge Regression	0.780	0.599
Random Forest Regression	0.759	0.581

Table 4. Over-sampled Data Results

Model	RMSE	MAE
Linear Regression	1.231	1.044
Lasso Regression	0.876	0.706
Ridge Regression	0.843	0.645
Random Forest Regression	0.836	0.648

Table 5. Under-sampled Data Results

Model	RMSE	MAE
Linear Regression	1.231	1.044
Lasso Regression	0.886	0.723
Ridge Regression	0.857	0.661
Random Forest Regression	0.849	0.650

On the other hand, sampled data run on the same algorithm gives an RMSE of 0.836 and 0.648. Out of the models run, unsampled data Random Forest Regressor performed the best. With an RMSE of 0.759 and MAE of 0.581, and given that the target ratings range from 1 to 5, the error rate relative to the range is 11.62%. Linear models with regularization perform better than non-regularized linear regression models as they regularize the parameters and force the weights of the uninformative features close to or exactly zero (L2 and L1 respectively). Thus, tree-based regression models perform best on this dataset

7.2 Categories Prediction

The objective of this task is to predict the top preference categories of the user based on features like 'Rating', 'Sentiment score', 'Price'. As part of experiments, we tried to one-hot encode the available categories and find similarities between them by using methods of clustering like k-means clustering and recommend categories which fall within the same cluster. We also experimented with Regression methods like Logistic Regression by considering the categories as multi-classes. Finally, we introduced personalization in these predictions by using SVD models and used variations of ratings and categories probability received from the Logistic Regression model to determine the top preferred categories of the user. Additionally, we also used a mathematical model which could be used for predicting startup ideas for categories no present within a determined 'x' km radius. In all the experiments mentioned in this sub-section, the training and test data split is 80:20

7.2.1 Clustering Methods

In this experiment, we use the places and reviews dataset and right merge the reviews with places. We drop rows which don't have categories and use the remaining 643,195 rows for the experiment. We one-hot encode all the 40 unique categories using MultiLabelBinarizer. The classes are shown in the Figure (9).

Using k-means clustering and evaluating using silhouette scores gave 32 as the optimal number of clusters. But we ran into the class imbalance issue where categories like 'American Restaurant' dominated over other categories. To deal with this, we incorporated 'KMeansConstrained' algorithm which ensures that all clusters formed have the minimum cluster size specified number of entries. We used PCA to reduce the number of input features and made 10 clusters but there was not much similarities between the clusters and they showed up as distinct groups as shown in the scatter plot on Figure 10.

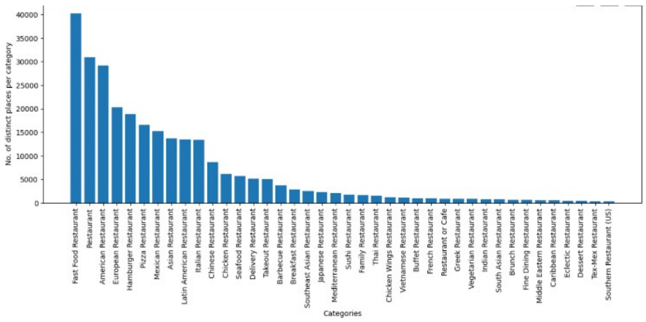

Figure 9: Selected top category classes

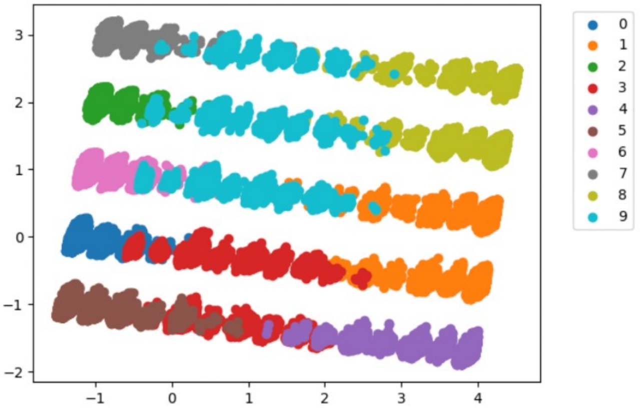

Figure 10: Scatter plot of the k-means category clusters

7.2.2 Regression Methods

As the clustering between categories did not giving promising results, we filtered users who have given at least 10 ratings (10,459 unique users) and considered all their place ratings to recommend them top categories. This dataset with all place ratings for each user had 228,605 rows. We label encoded the categories and used 'rating', 'sentimentScore', 'price' as features into the 'multinomial' Logistic Regression model.

7.2.2.1 Evaluation

Based on the category probabilities predicted by the 'multinomial' Logistic Regression model, we take the 5 most probable categories and if any of these matches with the actual category of the place visited as per the test set, we consider it a correct prediction. With this evaluation strategy, our model predicts with an accuracy of 52.6%.

7.2.3 Collaborative filtering methods

We had an expectation that introducing user as a feature in the model might help in more personalized recommendations for the user. Thus, we used an SVD Machine Learning model 'gPlusUserId' as indexes, place categories as columns and mean of all values corresponding to this user, place category combination. Here, two different variations of values are considered. First, the actual rating given by the user to recommend the most probable categories for the user. The values filled in the pivot table required as input by the SVD model become the mean ratings for each user, place categories. Second, the category probability value received from the Logistic Regression model. Here the values are the mean ratings for each user, place categories. We evaluate our model using the same evaluation strategy as Logistic Regression. If the actual category is within the top 5 recommended categories, we consider it a correct prediction. The results did not turn out to be promising as the number of ratings for each category by the user was sparse and the number of unique users was also less. Fine tuning the number of categories to be considered and balancing the dataset across all available categories can help to improve the predictions. As of now, most of the ratings belongs to the 'American Restaurant' category and hence, the predictions are skewed.

7.2.4 Mathematical models

7.2.4.1 Global Categories recommendation

There are some categories which are not at all present within a 20km neighborhood. Users wanting to visit such category places might have to travel far and hence if such a category place is opened in the vicinity, it might get good ratings and become a unicorn of the area. The startup suggesting idea is extended from the idea suggested by [10]. Thus, corresponding to each entry in the test set, we filter out the categories which are not present in the 20km radius of the test point and recommend these categories as startup ideas in this particular GPS location.

7.2.4.2 Competing categories

The different competing strategies mentioned in [11] show that startups should consider their competitors when trying to enter into the market. As in the previous strategy, there are no categories within the radius, thus there is no competition and there is a possibility that the business will fail to attract customers due to lacking market. But if we recommend categories which are already present in the vicinity which occur the least and have the lowest maximum rating among others, this can be a good assessment of competitive markets and might give an advantage to the new venture. Thus, we recommend such categories as startup ideas with an expected rating as the mean of all restaurants of this category in the vicinity. Figure 11 shows the presence of restaurants in the 5 km radius of a random input point and the recommended categories are ['Southern Restaurant (US)', 'Tex-Mex Restaurant', 'Dessert Restaurant'].

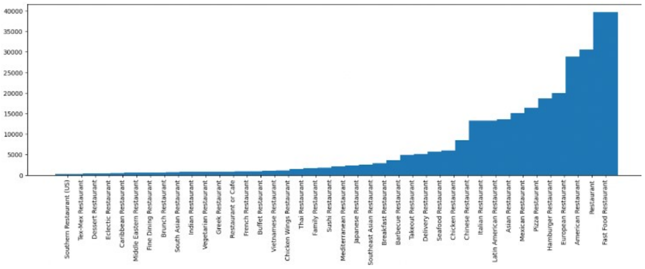

Figure 11: Category distribution in 5km radius of a random sample input

This model was used to predict venture ideas for random input points and the recommendation was considered successful if any of the top three recommended restaurants were present in the actual categories present.

8. CONCLUSIONS AND FUTURE WORK

We performed a predictive task of rating prediction of a user for a item with linear and tree based models with and without sampling. We also predicted top categories likelihood prediction for a user using various traditional machine learning models such as clustering and regression and other collaborative filtering and mathematical models. We evaluated the rating prediction task using standard metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) and found that the random forest regression model performed the best on unsampled data. The category prediction task was evaluated with custom matching of the predicted categories. Both the models seem to give convincing results. The performance of both the predictive task can be improved by better engineering additional features and by incorporating better modeling strategies.

Future research directions include (1) Incorporate dimensionality reduction techniques to improve the modeling for rating prediction. (2) Take time into consideration when predicting, i.e., if there is a trend of declining ratings, or negative sentiments in recent reviews, penalize this restaurant and avoid recommending it and suggest a startup idea of this category. The unix timestamp available in the review dataset can be used for this. (3) Evaluate the performance of the category prediction model on other datasets such as yelp and compare the accuracy. (4) Evaluating the impact of timing of the restaurant on category prediction for the user.

9. REFERENCES

[1] Nudrat, S., Khan, H. U., Iqbal, S., Talha, M. M., Alarfaj, F. K., & Almusallam, N. (2022). Users' Rating Predictions Using Collaborating Filtering Based on Users and Items Similarity Measures. Computational intelligence and neuroscience, 2022, 2347641. https://doi.org/10.1155/2022/2347641

[2] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In Proceedings of the 7th ACM conference on Recommender systems (RecSys '13). Association for Computing Machinery, New York, NY, USA, 213–220. https://doi.org/10.1145/2507157.2507160

[3] Ruchi Singh, Yashaswi Ananth, and Dr. Jongwook Woo. 2017. Big data analysis of local business and reviews. In Proceedings of the International Conference on Electronic Commerce (ICEC '17). Association for Computing Machinery, New York, NY, USA, Article 3, 1–5. https://doi.org/10.1145/3154943.3154946

[4] Md. Ahsan Habib, Md. Abdur Rakib, and Muhammad Abul Hasan. 2016. Location, time, and preference aware restaurant recommendation method. In 2016 19th International Conference on Computer and Information Technology (ICCIT), 315–320.
DOI: https://doi.org/10.1109/ICCITECHN.2016.7860216

[5] Ruining He, Wang-Cheng Kang and Julian McAuley. 2017. Translation-based Recommendation. In RecSys.

[6] Yanqing Chen and Steven Skiena. 2014. Building sentiment lexicons for all major languages. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), 383–389.

[7] Jiangtao Qiu, Chuanhui Liu, Yinghong Li, and Zhangxi Lin. 2018. Leveraging sentiment analysis at the aspects level to predict ratings of reviews. Information Sciences 451–452, (2018), 295–309.

[8] Steven Loria. 2020. textblob Documentation. Release 0.16 (2020). URL: https://github.com/sloria/TextBlob

[9] Sai Vishwanath Venkatesh, Prasanna D. Kumaran, Joish J Bosco, R PravinKumaar, and Vineeth Vijayaraghavan. 2021. A Non-Intrusive Machine Learning Solution for Malware

Detection and Data Theft Classification in Smartphones. In ICCS. DOI: https://dx.doi.org/10.1007/978-3-030-77967-2_17

[10] Xiaopeng Lu, Jiaming Qu, Yongxing Jiang, and Yanbing Zhao. 2018. Should I Invest it? Predicting Future Success of Yelp Restaurants. In Proceedings of the Practice and Experience on Advanced Research Computing (PEARC '18). Association for Computing Machinery, New York, NY, USA, Article 64, 1–6. https://doi.org/10.1145/3219104.3229287

[11] Martin S. Bressler. 2015. How small businesses master the art of competition through superior competitive advantage.