Rating Prediction of Restaurants based on Google Local Reviews

Jiake Chen 1195378246@qq.com
Jiang Wang jiw069@ucsd.edu

Abstract

Recently, more and more attention has been given to online review datasets. Local business owners are trying to leverage review data to increase their rating and attract customers. In this paper, we use a subset of the Google Local Review dataset to predict the rating received by restaurants located in California. We believe the geographical locations and categories of the restaurant may impact the average rating received, and review text is the most important information we can use to infer the expected rating. We have implemented similarity-based model, K-Nearest Neighbors and ridge regression to conduct rating prediction. The results by large confirms to our assumption.

Overview

To local business (especially restaurant) owners, online review data by customers is one of the least expensive methods to attract potential customers and increase profits. There are multiple ways restaurant owners can utilize the review data. For example, they can try to maintain a high rating to make people feel that it is a good restaurant. Or they can review the text left by the customers and see what they can improve. Needless to say, the online review data offer a previous tool for restaurant owners to advertise their business and improve their service. In this report, we will utilize the Google Local Review dataset to predict the ratings received by each individual restaurant. We believe the result of this study would offer local business owners of how to effectively brand and advertise their business.

1. The Google Local Review Dataset

1.1 Dataset Sources and Overview

The dataset used in this study is a subset of the google reviews dataset, obtained from professor McAuley's collections. The dataset contains three parts: **Places data**, **User data**, and **Review data**. The rating variable is in the review data, where each entry is stored as a user-place pair.

Instead of using the whole datasets that contains the ratings of places throughout the world, we limit the scope of this study to the restaurants located in California, USA. The reason is twofold. First, we believe places in similar categories would share some common predictive features. Second, we want to take advantage of the review text, which is available in the dataset, but many of which are not written in English.

1.2 List of Features

This section introduces the readers with the features in this dataset, some of which will be used in the following preprocessing step.

The features of the three data sections are listed in Table 1.

Table 1. Features in the Dataset

Features	Description
Shared Features	
gPlusUserId	ID number of the user
gPlusPlaceId	ID number of the place
Place Features	
name	Name of the place
price	Price level of the place (in terms of $, $$, and $$$)
address	Address of the place
hours	Open hours of the place
closed	If the placed has been closed
gps	The [latitude, longitude] pair of the place
User Features	
userName	Name of user
jobs	All previous jobs of user
currentPlace	Current residential place of user
previousPlaces	Previous residential places of user
education	Previous education of user
Review Features	
rating	The assigned rating of place given by the user
reviewerName	The name of the reviewer
reviewText	The review text
categories	The type of the place
reviewTime	Time of the review was created

1.3 Dataset Preprocessing

To extract all the restaurants in California, we work mainly on the Places and Review dataset. The following rules are applied to filter the places in California:

Geography Filter:

The place must be located in the space specified by the given GPS coordinates[1]:

North: 43 N°; South: 24.74 N°
West: 124.78 W°; East: 115 W°

The South and West limits are the US boundaries, and the west and east limits are figured out by heuristics.

Additionally, we check the address of the places, and exclude all places that do not have "CA" in it[2].

Restaurant Filter:

After we extracted all the places in California, we compare the result with the Review data, we exclude those reviews that refer to the places that are not in California.

Next, we examine the category section of the remaining reviews, we exclude all those reviews that does not have "restaurants".

Finally, we extract all the unique places and users in the remaining reviews. We subset the places and users according to the 'gPlusPlaceId' 'gPlusUserId' of the remaining reviews.

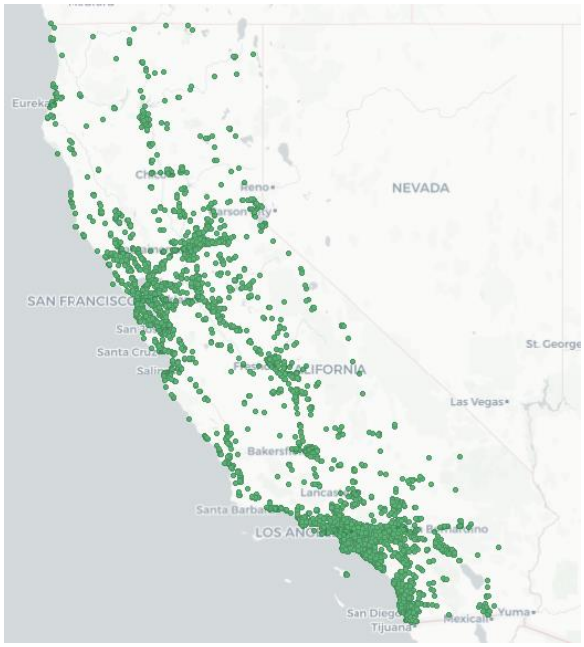
[1] If a place does not have GPS information, then it is excluded from our study

[2] This implies that we also exclude all those places that maybe located in CA but has an incomplete address

1.4 Dataset Description

We obtained a (collection of) datasets that includes 45,522 restaurants, 92,571 users, and 291,733 interactions of user-restaurants. The geographical distribution of these restaurants is displayed in Graph 1. As shown in the following, all restaurants are located in the state of California.

Graph 1. Geo-Distribution of Restaurants



By observing the geographical distribution, it appears that most of the restaurants are located in Los Angeles, San Diego, and the Bay area. It might be interesting to examine how restaurants in these areas (and also other places) differ in terms of their ratings.

1.5 Exploratory Data Analysis

1.5.1 Rating Distribution – User and Place

Overall, the rating exhibits a centering trend: Most of the ratings are distributed around values 3-5, with a few outliers with values of 1 or 2. The distribution of rating, either grouped by restaurant, or grouped by user, is provided in Table 2.

Table 2. Statistics of Rating

Level	Statistics	Value
Global	Mean	4.001
	Min	0.0
	Max	5.0
	STD	1.083
User Mean	Mean	4.044
	Min	1.0
	Max	5.0
	STD	1.127
Restaurant Mean	Mean	3.922
	Min	1.0
	Max	5.0
	STD	0.844

Overall, the table shows that ratings grouped by restaurant tend to have lower variance than ratings grouped by user. This may imply that "good" restaurants in general receive better ratings, independent of personal taste. This may also imply that prediction using restaurant mean is perhaps a better model than the prediction using user mean.

1.5.2 Rating Distribution – Geographical Areas

In previous sections, we discussed the possibility that geographical locations may also have an impact on the ratings of each restaurant. In this part, we mainly focus on extracting the distribution of rating by where the restaurant is located. We focus on the following areas: San Diego, Los Angeles, and the Bay area. The (box) filters of the geographical comparison are displayed in Table 3.

Table 3. Geographical filters

Area	Coordinate	Boundary
San Diego	Latitude	31.72 N°, 33.72 N°
	Longitude	116.16 W°, 118.16 W°
Los Angeles	Latitude	33.05 N°, 35.05 N°
	Longitude	117.24 W°, 119.24 W°
Bay Area[3]	Latitude	36.82 N°, 38.82 N°
	Longitude	120.88 W°, 123.29 W°

And the distribution of ratings (in terms of average rating per restaurant) is given in Table 4.

Table 4. Statistics of Rating by Region

Area	Statistics	Value
San Diego	Mean	3.966
	Min	1.0
	Max	5.0
	STD	0.8559
Los Angeles	Mean	3.931
	Min	1.0
	Max	5.0
	STD	0.8760
Bay Area	Mean	3.890
	Min	1.0
	Max	5.0
	STD	0.780

As observed in Table 4 (and when we compare it with Table 2), we notice that all these three metropolitan areas have lower mean and standard deviation rating when compared with global statistics. This implies that incorporating geographical factors in our models is likely to yield a smaller error rate.

1.5.3 Rating Distribution – By Category

Finally, we are interested in how the category of a restaurant may help us predict the rating. We extracted the categories of all the restaurants in our dataset. Each category is assigned with one list of ratings. Please note that one restaurant may have multiple categories, so that one rating may appear in multiple lists. The statistics of the ratings of some of (the most interesting) restaurant categories are given in Table 5.

Table 5. Statistics of Rating by Category

Category	Statistics	Value
Fast Food	Mean	3.801
	Std	1.230
Jamaican	Mean	4.381
	Std	0.7815
Australian	Mean	4.0
	Std	1.414

Table 5 mainly shows three types of restaurants. The restaurants with a moderate mean and a high variance. These are the most popular categories. The restaurants with a high mean and a low variance. These are types of minority categories (e.g. Jamaican restaurants). Not many people will visit these restaurants, but in general the people visiting these restaurants will give a high rating. The last category would be another type of minority restaurant. These are the restaurants that not many people would visit,

[3] Includes San Francisco and San Jose Area

but unlike the second category not the majority of customers would give a high rating.

2. The Predictive Task: Rating Prediction

As indicated in our title, the theme of this report is to predict the rating of each user-restaurant pair. For each user-restaurant pair, we would have the list of features list in Table 1, except the rating[4].

2.1 Baseline Model

As indicated in Table 2 of the previous section, we decided to use restaurant average and global mean as the baseline, since using simple heuristics we believe using restaurant average is better than using the user average, which tends to have a higher variance.

The baseline model can be described as:

```
Input: user-restaurant pair
if restaurant is observed in training set:
    return average rating of that restaurant;
else:
    return global mean;
```

2.2 Evaluation Metric and Justification

Considering that rating itself is a continuous variable, we decide to use **mean-squared-error** (MSE) as the metric to evaluate our models.

There used to be some other ideas in other group. Considering that ratings would only take a finite number of values, we can also treat it as a classification problem. However, later we dropped this idea since we realize the sequence of the value matters, namely 5 is better than 4 and 4 is better than 3. We want to preserve the ordering in the variable rating.

Hence, we treat rating as a continuous variable, and we want to minimize our MSE as much as possible.

2.3 Data Split and Validation

We treat the first 80% as the training set, next 10% as the validation set, and the last 10% as the test set. Note that the dataset is shuffled before being split. We have also considered cross-validation techniques, but we feel the validation process may be a bit slow with the available computation power we have.

2.4 Model and Feature Selection

2.4.1 Model selection

The main model we use is OLS and in a general sense, ridge regression to fit our model. We will try the model with different combination of features to examine the performance.

2.4.2 Preprocessing of Rating

Note that regression usually requires the dependent variable to be normally distributed, however, this is not the case for our rating variable, which appears to be right-skewed. Therefore, contrary to the way we are taught in

[4] The tasks of our report would be by and large similar to the assignment 1 of CSE 258, except that we use a different subset of the data.

this class (set rating directly as y), we also training another series of model that takes the natural logarithm of rating as y.

However, note that we are still keeping the original way of fitting the regression (set rating directly as y), because using the logarithm as the dependent variable will increase the error margin for outliers, which may or may not penalize the model's performance depending on the data's distribution.

We will compare the two series of models and select the ones that performs better.

2.4.3 Feature selection – Known Estimates

Notice that in the baseline model, we derive two estimates: the global mean, and the restaurant mean from the training set. Here we are doing something very similar. We will use the user's mean and restaurant's mean together to make the prediction. The feature generation step can be described as follows (by append, I mean append to the feature vector):

```
Input: user-restaurant pair
if user is observed in training set:
    append average rating of that user;
else:
    append global mean;
if restaurant is observed in training set:
    append average rating of that restaurant;
else:
    append global mean;
```

Due to the large training set we have, we are not very much worried about the collinearity of the user-based prediction and the restaurant-based prediction (because many of which has already been covered in the dataset).

2.4.4 Feature Selection – Price Level

We observe that price level may also have an impact on people's ratings, as customers usually have a higher expectation of restaurants that charge more.

We use two ways to code this variable. First, we use one-hot encoding to code this variable. The reason is that we assume the relationship between the rating and the price level is nonlinear[5]. Second, we use numerical values to code this variable, which shows

2.4.5 Feature Selection – Category

As discussed in the exploratory analysis section, we observed different distribution of ratings for different categories. We believe restaurant's category may impact the rating for that restaurant. Similar to price level, we use one-hot coding and recode this variable into 50 mostly-visited restaurant categories we have. Please note that one restaurant may have multiple categories.

2.4.6 Feature Selection – Geography

We have emphasized that geographical factors may be important for rating prediction. To evaluate the importance of such variables, we first run K-means on our training set based on

[5] Due to the large number of missing values in this field, we decide to code the variable into three binary variables.

the GPS locations of the restaurants. We want to cluster the restaurants that are close enough to each other. Our assumption is that people who visit restaurants very close are likely to have similar tastes, so that they will share some common ground when they rate. To code this geographical features, after using K-means, we will create a series of binary variables to represent clusters. Restaurants that lie in the same cluster would have the same feature (sub)vector.

2.4.7 Feature Selection – Review Text

Similar to what we have done in class, we will use the review text to predict the rating of the restaurant. Since we have already preprocessed the data, most of the entries in the Review data would either have None or English text, making it easier for us to deal with. We extracted the top 1,000 values with their TFIDF values in their feature vector.

3. Models

In this part, we will discuss the multiple ways we have tried to predict the rating. The main things we have tried are similarity-based model, KNN-based model, and ridge regression.

3.1 Baseline

The naïve baseline model has been discussed in part 2.1. To training the model we simply generate the global mean and the mean for each individual restaurant. This model appears to have moderate performance which yields an MSE value of 1.52.

3.2 Similarity-based Model

In this model we use Jaccard similarity between users and places to predict the ratings. We think if place A is similar to place B, the rating for place A will be close to the expected rating for place B. Additionally, if user A is similar to user B, the user A's rating will be close to the expected rating from user B's. We think two places are similar when they have been rated by a same user and two users are similar when they have been to a same place.

When we predict the rating given (user, place), we first find the most similar user U to this user and the most similar place P to this place[6]. Then we use U's average rating and the average rating of P as features and fit them into ridge regression to predict the rating. Note that it is quite possible we would encounter a similarity of 0, due to the unbalanced natural of our dataset. When such case happens, similar to the base model, we use the average rating of all reviews to make the prediction.

Performance

The results are as following:

1) when we only use the Jaccard similarities between users, the MSE of the predictions is 1.12605.

[6] We simply mean the user/place that has the highest Jaccard similarity value.

2) when we only use the Jaccard similarities between places, the MSE of the predictions is 1.14687.
3) when we only use both of the Jaccard similarities, the MSE of the predictions is 1.12695.

3.3 KNN-based model to Incorporate Temporal Order

Following the idea of similarity of the previous part, we believe the KNN method is also a measure of similarity. We come to think about using KNN-method when making the prediction.

We believe the temporal order of reviews matter. That is, if two reviews are very close to each other, then their ratings are very likely to be close. We derive a distance measure based on the distance of the absolute value of the timestamp difference. With a KNN-based model. The way of making the prediction is quite simple, for each input timestamp, we compare it against the timestamp of all the entries in our training sets to yield a prediction. We simply extract the first K matches and make the prediction[7].

One caveat of this method is how to take advantage of the additional information we have to make a more precise prediction. Notice that user and place ID are given, so that we can first compare the user and place IDs and then calculate the distance against the subset of reviews that share the same user/place ID.

Therefore, instead of directly comparing the datapoint with the whole training set, we first search for the user ID in our training set and compare the datapoint against the reviews of that restaurant. If we cannot find the user ID in our training set, we refer to the place ID. If we cannot find either place ID or user ID in the training set, we simply use the global mean[8]. The key idea is very simple: Instead of computing the distance on the whole dataset, we just compute the distance on the subset of the dataset.

Performance Deterioration

One issue with the KNN method is that the prediction time is linear to the number of training set and the number of datasets. It matters less in our study because we are only calculating the distance based on timestamp, and we have taken advantage of the user, restaurant information we have to reduce our workload. However, the method scales pretty badly as we increase the validation set, or we increase the number of features we want to use.

Performance

The result of KNN is okay but not as good as we expected. With a K=1 we obtained an MSE of 1.35, lower than the value of the baseline model. Further increasing the K value does not decrease the error much. After a careful

[7] In our implementation the prediction is based on the mean value of the first K matches, rather than based on the majority vote.

[8] This is because we believe there exists no temporal order if user or place is not the same

inspection of the predictions, we find that the distribution of the dataset acts in no favor of this methods. This is because most user only left 1 or 2 reviews in the dataset, and some of them are outliers. Since there are not many reviews in the subspace we are comparing nearest neighbor with, it is expected that our prediction is not accurate, and increasing K does not decrease error much.

3.4 Ridge Regression

The last model we used is the ridge regression model. This comes with a natural choice as the rating prediction task is very similar to the requirement of HW4. Also, we believe regression is a technique that we have gained confidence all along this course.

We implemented a series of features are test the performance difference of different models. Among all the features we have used, we believe review text is the most predictive feature.

3.4.1 Baseline Regression

We fit the model using the user mean rating and restaurant mean rating from the training set. In the user or restaurant is not found in the training set, we simply substitute it with the global mean. The model yields an error rate of 1.110.

3.4.2 Regression with General Features

In addition to the previous two features, we added the category of the restaurant (using one-hot coding), review length, and review time (using one-hot coding). The model yields an error rate of 1.090.

We also incorporate the price level into the model, price level is treated either as a continuous variable (1,2,3) or a series of binary variables using one-hot coding. For missing values, we substitute the value using 2 (which we assume as mean from population) in the first implementation, or [0,0,0] (which indicates that they have no price tag) in the second implementation. Overall the two implementations yield similar error rate, which is 1.0883.

3.4.3 Regression with Clustered Geo-features

Next, we cluster the restaurants using their GPS data (latitude and longitude). We then convert the K clusters into K-1 binary features. For the values of K, we have tried 2 to 512, incremented by a power of 2. The result shows that the lowest error is achieved when K=32, and the error rate is 1.0881.

3.4.4 Regression with TF-IDF measures[9]

Finally, we have tried the "a-bag-of-words" model. We extracted the 1000 most popular unigrams from the training set and generated their IDF score across the whole dataset. Next, for each document we simply count the TF (number of appearances) of the top 1000 words, and times the value by the corresponding IDF score to get the feature vector. Text feature only

[9] The computation of TF-IDF is just the same with what we've learned in course, and the implementation in HW4.

yields a lower error rate of about 0.823, while text features combined with other features yield a validation error of 0.813.

4. Literature Review

The dataset we used is the Google Local Reviews, and it has been used extensively in academia. To our interest, we only study part of the datasets, but other researchers have discussed other parts of the dataset.

There are some other classical papers detailing and applying the google local review data. Many of the paper try to use NLP and statistical methods to detect the sentiments from the data.

The classic one is the paper published by Google, which tries to detect the sentiment by automatically analyzing the text (Goldensohn et al., 2008). Some other researches also complete similar tasks using topic models (Titov & McDonald, 2008). In general, most of previous researches tend to focus on review texts that try to extract effective information from reviews and therefore can help local business to improve rating.

There are many similar datasets, just like the Google Local Reviews. For example, the Yelp Challenge dataset includes user-business reviews, geographical information, ratings, just like the Google dataset. However, the Yelp Challenge dataset also includes check-in, tip, and photo data so that we can dive deeper.

There are multiple study topics on the Yelp dataset, for example, rating prediction (Asghar, 2016), business star prediction (Fan & Khademi, 2014), finding local experts (Tanvi, 2015), etc. One very interesting research conducted by Huang, Rogers, and Joo (2014) tried to mine latent subtopics (those are highly influential to ratings) using the Latent Dirichlet Allocation (LDA) algorithm. They were able to mine out some important subtopics such as “service” and “value”.

Considering that similar datasets, including the Yelp Challenge dataset and Google Local Reviews, consists the timestamp data, and that the temporal sequence matters for each individual user, the state-of-the-art method of such datasets is, without doubt, a combination of metric-based methods and temporal-based prediction.

Two of the papers come from Professor Julian McAuley’s lab utilizes the timestamp data to understand the sequential behavior of users. Based on where the user has previously gone, McAuley and his colleges’ models can predict what will be in more favor of the user in the future. In essence, there is an item-item interaction, in addition to the user-item interaction, and it is the item-item information that we have long neglected.

One paper of McAuley’s lab proposes a method called *TransRec* that creates translation vectors to model such behavior (He, Kang & McAuley, 2017). Another paper from their lab proposes *TransFM*, that tries to incorporate the common-sense metric-based recommendation to the translation-based recommendation (Rajiv &

McAuley, 2018). Moreover, they have established convincing results that the method is very fast and can be optimized using classical methods.

After studying the existing researches and studies, we can conclude that review text is the most common tool for predicting rating, which we have included in our study. Our results show that review text, or the "a-bag-of-words" model is the most predictive model across all the models we have used.

5. Results and Conclusions

The performance of our models is summarized as follows:

Table 6. Performance of Models

Model	Validation Error	Test Error
Baseline	1.5246	1.514
Similarity-Based Model	1.1265	1.1367
K-Nearest Neighbors	1.3515	1.372
Regression (General)	1.08982	1.09988
Regression (K-Means)	1.08811	1.08828
Regression (TF-IDF)	0.8131	0.8193

Based on the results, we can conclude that the ridge regression with TF-IDF features gives the best performance. This means the review text provides much more information than other features (geographical information, for example). The results conform to our discussion in the previous section that when predicting the ratings, most researches focus on the sentiment analysis of review text.

Observe that the baseline regression model, where we have incorporated the distribution of ratings among users and restaurants, as well as timestamp, price level, and review length, outperforms the baseline model. This indicates that first, user mean may be a more effective predictor than restaurant mean, as people usually assign ratings based on an individual, uniform score. The low efficiency of restaurant mean also implies that different people may observe different aspects of a restaurants and thus assign ratings based on different scales.

All throughout this paper, we have also discussed the possibility that rating distributions are different across various regions, as different people from different areas have different tastes. We use K-means to classify restaurants' geolocations to different clusters to build the feature vector. The result confirms to our assumption as the incorporation of geo-features decrease the error of the model.

Finally, review texts appear to be the dominant factor of predicting the rating, as the error rate decrease significantly when we introduced the TF-IDF model. We believe this is due to the fact that texts give significantly more information than other features. For example, the appearance of "worst" would indicate a low rating, and the appearance of "awesome" would

indicate a low rating. Unlike visit prediction and category prediction, the influence of text in rating prediction is very obvious.

In conclusion, we have tried through the methods we are taught in class and was able to generate some valid models. Regression-based model appear to yield the best performance, and review text is the most informative feature we can use.

References

Asghar, Nabiha. "Yelp dataset challenge: Review rating prediction." *arXiv preprint arXiv:1605.05362* (2016).

Blair-Goldensohn, S., Hannan, K., McDonald, R., Neylon,T., Reis, G. and Reynar, J., 2008. Building a sentiment summarizer for local service reviews.

Fan, Mingming, and Maryam Khademi. "Predicting a business star in yelp from its reviews text alone." arXiv preprint arXiv:1401.0864 (2014).

He, Ruining, Wang-Cheng Kang, and Julian McAuley. "Translation-based recommendation." In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 161-169. ACM, 2017.

Huang, James, Stephanie Rogers, and Eunkwang Joo. "Improving restaurants by extracting subtopics from yelp reviews." *iConference 2014 (Social Media Expo)* (2014).

Jindal, Tanvi. "Finding local experts from Yelp dataset." PhD diss., 2015.

Pasricha, Rajiv, and Julian McAuley. "Translation-based factorization machines for sequential recommendation." In *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 63-71. ACM, 2018.

Titov, I., & McDonald, R. (2008, April). Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web* (pp. 111-120). ACM.