Google Restaurants Rating Prediction

Part 1

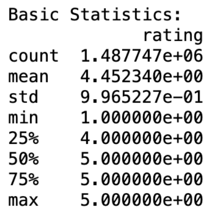
Dataset Overview

The "Google Restaurants" dataset, is an extensive collection of user-generated reviews for restaurants. Each entry in the dataset not only encompasses textual reviews but also includes numerical ratings and a set of images associated with the review, offering a multifaceted view of customer experiences.

Key Fields in the Dataset

- business_id: A unique identifier for the restaurant
- user_id: A unique identifier for the user who wrote the review
- rating: Numerical rating user gave of their experience at the restaurant
- review_text: Textual content of the user’s review
- pics: A list of dictionaries with image IDs and URLs, providing visual of the user's dining experience

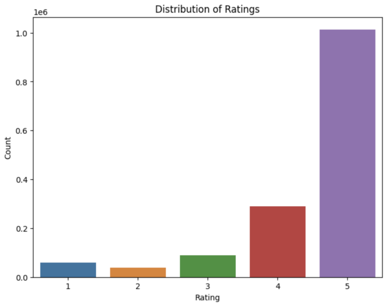
Basic Statistics



```
Basic Statistics:
             rating
count  1.487747e+06
mean   4.452340e+00
std    9.965227e-01
min    1.000000e+00
25%    4.000000e+00
50%    5.000000e+00
75%    5.000000e+00
max    5.000000e+00
```

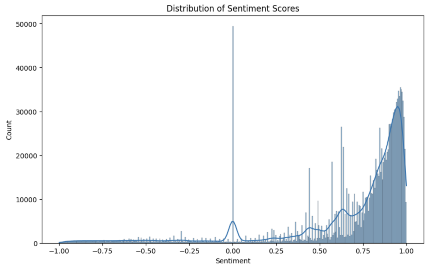
There are approximately 1.49 million reviews in the dataset, indicating a rich dataset. The mean rating is about 4.45 on a scale of 1 to 5. This suggests that the overall sentiment of the reviews is quite positive.There is some variation in the ratings, but it's not extremely high. Most ratings are clustered around the mean.

Rating Distribution:



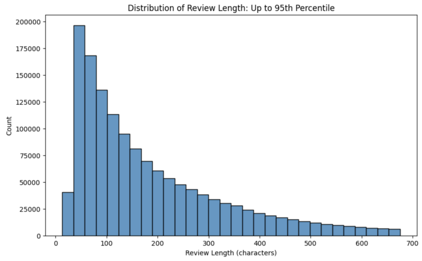
At least 25% of the reviews have a rating of 4 or lower, and over 75% have a rating of 5. This is also the most common rating since it's equal to the median. This further emphasizes the positive skew of the ratings.

Review Sentiment Scores Analysis:



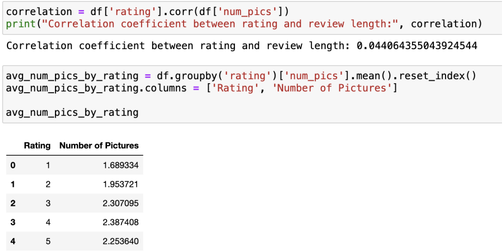
The distribution shows a prominent peak near the score of 1, indicating a significant portion of the text is rated with a high positive sentiment. Notably, there is a considerable number of neutral sentiments scored around 0. However, the positive sentiments are far more frequent than neutral or negative ones, as seen by the right-skewed nature of the distribution.
The correlation coefficient of 0.5277 signifies a moderate positive association between user ratings and sentiment scores, in which reviews with higher ratings tend to have more positive sentiment scores.

Review Length Analysis



The figure focuses on the range up to the 95th percentile, excluding the longest 5% of reviews that're likely outliers, and provides a detailed view of the distribution for the majority of reviews. This graph shows a clear right-skewed distribution, indicating that many users keep their reviews brief and under 100 characters. The correlation coefficient of -0.218 suggests a slight tendency for higher-rated reviews to be shorter. This pattern might imply that users are more verbose when expressing negative experiences, while positive feedback tends to be more succinct.

Number of Pictures Analysis



```
correlation = df['rating'].corr(df['num_pics'])
print("Correlation coefficient between rating and review length:", correlation)
```

```
Correlation coefficient between rating and review length: 0.044064355043924544
```

```
avg_num_pics_by_rating = df.groupby('rating')['num_pics'].mean().reset_index()
avg_num_pics_by_rating.columns = ['Rating', 'Number of Pictures']

avg_num_pics_by_rating
```

	Rating	Number of Pictures
0	1	1.689334
1	2	1.953721
2	3	2.307095
3	4	2.387408
4	5	2.253640

Most reviews contain a relatively small number of pictures, with a very high count for reviews with a single picture. As the number of pictures increases, the frequency of such reviews decreases substantially. This could suggest that users are less inclined to upload multiple pictures with their reviews, or it might reflect a platform design where uploading images is less emphasized. The steep drop-off after the first few pictures indicates that while pictures are a component of reviews, the majority of users do not engage heavily in visual documentation of their experiences.

The correlation coefficient of 0.044 is low, suggesting no strong linear relationship between rating and the number of pictures they post in their reviews. In summary, the lack of a strong correlation and the non-linear pattern in the average number of pictures per rating suggest that the decision to include pictures in a review is likely independent of the numerical rating given.

Part 2

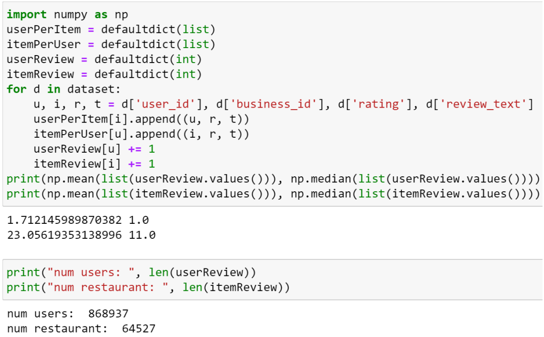
Predictive Task

The dataset contains over 1.49 million interactions between users and restaurants that represents users' preferences toward restaurants. The predictive task is to predict a given user's preference toward a given restaurant (in terms of the rating the user would likely give), based on the past interactions between user and restaurants and past rating the restaurant received. This could help with recommending restaurants to users by recommending the restaurant users would likely give a higher rating.

Evaluation Metrics

For evaluating the performance of the models, several different metrics could be used, including overall precision, precision on predicting 5 star ratings, and MSE of the model. Since the prediction of integer rating should range from 1 to 5, it can be seen as a multi-class classification. Hence, it would be helpful to evaluate the model using the overall precision of the model. The particular process would be to evaluate the model's precision on each class of rating, then combine to assess the overall precision by taking the weighted average of those metrics according to rating distribution. We take the weighted average because the majority of the reviews clustered around 5, and do not follow a normal distribution. Additionally, since the objective of the prediction task is supposed to help with recommending users with restaurants that they would likely to rank highly, it would be helpful to get the precision on the 5 star rating the model predicts to evaluate the accuracy of the model's 5 star predictions. Lastly, since the predictive task is mostly done through regression models instead of classification models, we also include MSE as the metric for evaluating regression performance.

Baseline Model



```
import numpy as np
userPerItem = defaultdict(list)
itemPerUser = defaultdict(list)
userReview = defaultdict(int)
itemReview = defaultdict(int)
for d in dataset:
    u, i, r, t = d['user_id'], d['business_id'], d['rating'], d['review_text']
    userPerItem[i].append((u, r, t))
    itemPerUser[u].append((i, r, t))
    userReview[u] += 1
    itemReview[i] += 1
print(np.mean(list(userReview.values())), np.median(list(userReview.values())))
print(np.mean(list(itemReview.values())), np.median(list(itemReview.values())))
```

```
1.712145989870382 1.0
23.05619353138996 11.0
```

```
print("num users: ", len(userReview))
print("num restaurant: ", len(itemReview))
```

```
num users:  868937
num restaurant:  64527
```

According to the statistics, the interactions between users and restaurants in dataset is very sparse in which users on average have left reviews for about 2 restaurants, and restaurants on average received around 10 reviews. In previous exploration, it occurred that over 75% of the ratings are clustered around 5. Hence we could set the baseline model based on the distribution of the restaurant review ratings. In the baseline model, it predicts a user's rating to a given restaurant by randomly selecting a rating in the set {1, 2, 3, 4, 5} with the probability of choosing each class of rating equal to its proportion in the dataset.

Relevant Features

Some relevant features that could improve the performance of model could be the following:

- User id: represent each user and can be obtained from the feature "user_id" directly
- Restaurant id: represent each restaurant and can be obtained from the feature "business_id" directly
- Restaurant average rating: obtained through iteratively getting average rating for each restaurant.
- Number of Reviews: obtained through counting historical reviews for each restaurant.
- Sentiment score (correlation to rating at 0.5277): obtained by applying the polarity_scores function of the SentimentIntensityAnalyzer library on the feature "review_text".
- Review length (correlation to rating at -0.218): obtained by applying length function to feature "review_text" to obtain the number of tokens in the review.

Model Validity Assessment

To assess the validity of the model's prediction on rating, we will be using k-fold cross validation with a k value of 5 on the model. The dataset will be shuffled and randomly separated into training set and testing set 5 times, and each time the model will be trained on the training set and evaluated on the testing set based on the metrics as mentioned in the previous section to assess the robustness of the model overall, which would spot any sign of overfitting in the models.

Part 3

Relevant Model Considerations

Restaurant average model: This model predicts the rating by taking the historical average rating of given restaurants.

Linear regression model: This model predicts the rating using a linear regression model that takes different combinations of features, including user id, restaurant id, number of past reviews for restaurant, historical average rating of restaurants.

Similarity-based model with collaborative filtering approach: In addition to the average rating of the given restaurant, this model accounts for user’s preference to this type of restaurant by taking into account the contribution of the user's past rating weighted by the past restaurant’s similarity with the current restaurant using Pearson similarity. Then round the result rating to the nearest integer.

$$\text{Similarity}(i,j) = \frac{\sum_{u\in U}(r_{u,i}-\bar{r}_i)(r_{u,j}-\bar{r}_j)}{\sqrt{\sum_{u\in U}(r_{u,i}-\bar{r}_i)^2 \cdot \sum_{u\in U}(r_{u,j}-\bar{r}_j)^2}}$$

$$r(u,i) = \bar{R}_i + \frac{\sum_{j\in I_u\setminus\{i\}}(R_{u,j}-\bar{R}_j)\cdot \text{Sim}(i,j)}{\sum_{j\in I_u\setminus\{i\}}\text{Sim}(i,j)}$$

Latent factor model: this model converts the given data point into a feature vector consisting of one-hot-encoding representations of users and restaurant, and assign each entry a gamma value to account for the weighting factor that captures the preference of user and interactions with latent factors. The predictive result is also rounded to the nearest integer.

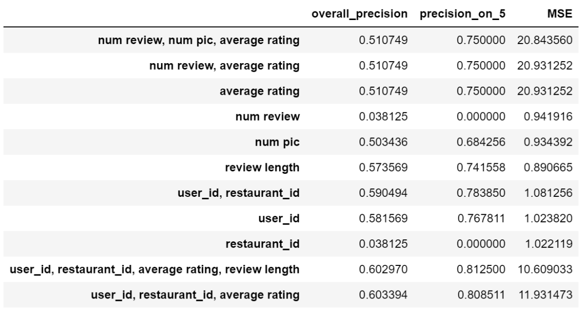
$$\arg\min_{\alpha,\beta,\gamma}\sum_{u,i}(\alpha+\beta_u+\beta_i+\gamma_u\cdot\gamma_i-R_{u,i})^2+\lambda\left[\sum_u \beta_u^2+\sum_i \beta_i^2+\sum_i \|\gamma_i\|_2^2+\sum_u \|\gamma_u\|_2^2\right]$$

$$f(u,i) = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i$$

Model Optimization

Two of the models require optimization on hyperparameters before proceeding to comparison. The process is shown below:

Linear Regression Model: The optimization is done through choosing features most relevant and gives better performance. The following combinations of features were attempted and evaluated. Ultimately, the best set of features concluded from the metrics is using one-hot-encoding of user Id and restaurant id, and the historical average review of the restaurant and the length of review the restaurant received.



	overall_precision	precision_on_5	MSE
num review, num pic, average rating	0.510749	0.750000	20.843560
num review, average rating	0.510749	0.750000	20.931252
average rating	0.510749	0.750000	20.931252
num review	0.038125	0.000000	0.941916
num pic	0.503436	0.684256	0.934392
review length	0.573569	0.741558	0.890665
user_id, restaurant_id	0.590494	0.783850	1.081256
user_id	0.581569	0.767811	1.023820
restaurant_id	0.038125	0.000000	1.022119
user_id, restaurant_id, average rating, review length	0.602970	0.812500	10.609033
user_id, restaurant_id, average rating	0.603394	0.808511	11.931473

Latent Factor Model: In this model, the optimization is mainly focused on choosing the best lambda regularization terms. Due to the extreme sparsity of user-item interaction on the dataset where users on average have left reviews for about 1 to 2 restaurants, and restaurants on average received around 10 to 20 reviews, we optimize by assigning different lambda regularization to betaU and betaI in addition to the other lambda assigned to the gamma terms. We also optimize the parameters with alternating least squares to minimize the objective. And in prevention of overfitting training data, early stop is implemented once the validation MSE stops decreasing. To find the best hyperparameter lambda values, we experiment with different lambda values over the range [1, 10] increment by 0.2 each time. According to the metrics, the best hyperparameter is λ1=8, λ2=1.7, λ3=10.

Strength and Weakness of Models

The strength and weakness of each model in comparison to the baseline is as follows:

Restaurant Average Model: The dataset is very popularity-based in which users tend to go to popular restaurants, with the top 3.4% most popular account for more than 28% of total reviews. Moreover, the interactions between users and restaurants in the dataset is very sparse in which users on average reviewed 1 to 2 restaurants, and restaurants on average received 10 to 20 reviews. Hence, the historical average rating of the restaurants would be a better estimator for rating than the baseline that uses merely rating distributions. However, this model has limited improvement upon the baseline since it does not take features from the user, which is not customized for predicting the user's preference.

Pearson Similarity Model (with collaborative filtering): This model improves Restaurant Average Model by introducing pearson similarity between the predicting restaurant and user's past restaurants to account for user's preference, which has slightly higher performance in precision compared to merely predicing restaurant average. However, this model only uses interaction data and does not take into account any additional features of restaurants, and works slowly with a large dataset of this size.

Linear Regression Model: In comparison to all previous models along with the baseline, this model takes into account many different features, including both from the user and the restaurants, making its prediction more customized to the user. There is some improvement in precision as compared to other models, but the MSE is not the best so far. Moreover, since this dataset is highly sparse and large at the same time, the one-hot-encoding in this model works very slowly, which could be a concern.

Latent Factor Model: In comparison to the similarity model and the baseline, the latent factor model deployed supervised machine learning. Moreover, unlike the previous models who treat the user and restaurant separately, by introducing the latent factors and get dot product of the gamma terms for users and restaurants, it makes sure the predictive result is more customized to user's preference, which is more preferable for this predictive task. However, just like with the similarity model, it could improve by taking more features into the model.

Part 4

Related Literature Review

In our project, we've utilized the "Google Restaurants" dataset to predict user preferences for restaurants based on historical interactions. This dataset is extensive, featuring over 1.49 million user-restaurant interactions. It includes ratings, textual reviews, and images, providing a comprehensive view of customer experiences. The dataset used in this study is originated from a large-scale multi-modal dataset,
namely Gest, which is collected from Google Local Restaurants
including review text and corresponding pictures generated by the study "Personalized Showcases: Generating Multi-Modal Explanations for Recommendations" [6].
To understand our project's context and benchmark our approaches, it's insightful to explore similar studies and datasets:

"Google Local Dataset Analysis" by Xinchi Gu and Long Jin [1]: This study uses similar datasets as ours, focusing on location data and review text for rating prediction. They developed a location-based latent factor model, akin to SVD++, showing the significance of spatial context in rating predictions. Their method resonates with our use of latent factors but differs in emphasizing location data.

"Rating Prediction on Google Local Data" by Shiyi Hua, Akshay Kotha, Ziyuan Yan[2]: This GitHub project also analyzed Google Local data, employing various models including Ridge regression with bag-of-words and sentiment-based predictions. Notably, both our project and this study utilized one-hot encoding techniques to enhance model performance – in our case, for user and business identifiers, and in their case, for features like price level and hour. Their approach of incorporating textual review content and additional features like price level and review length offers a broader scope than our primarily interaction-based models.

"Predicting Restaurant Ratings from Yelp Reviews" by Zefang Liu [3]: Liu's work on Yelp data used machine learning and transformer-based models, achieving notable accuracy with models like XLNet and Logistic Regression. This study's exploration of text-based features and advanced models provides insights into alternative methodologies that could enhance our project's predictive capabilities.

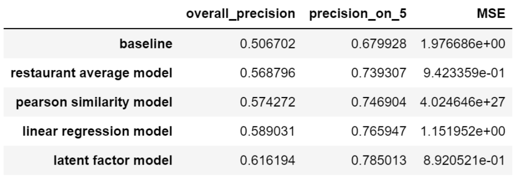
"Predicting Ratings and Popularity Change of Restaurants" by Yiwen Guo, Anran Lu, and Zeyu Wang [4]: Focusing on Yelp data, this study employed methods including logistic regression, Naive Bayes, and SVM. Their findings, particularly the effectiveness of logistic regression, align with our exploration of linear models but suggest further scope in experimenting with diverse algorithms.

"Movie Rating Prediction" by Sherin Claudia [5]: Although focusing on movie ratings, this study's use of diverse models like Decision Trees, SVMs, and Naive Bayes offers a perspective on the applicability of different techniques in rating prediction. The methodology provides a broader view of possible approaches.

In comparison, our project utilizes models like similarity-based predictions, linear regression, and latent factor. While we've achieved significant results, these studies suggest potential areas for improvement, such as incorporating location data, textual analysis, and more advanced machine learning techniques. The conclusions from existing work indicate the complexity of rating prediction and the need for a multifaceted approach, incorporating various data aspects to enhance our own findings and methodologies.

Part 5

Model Results

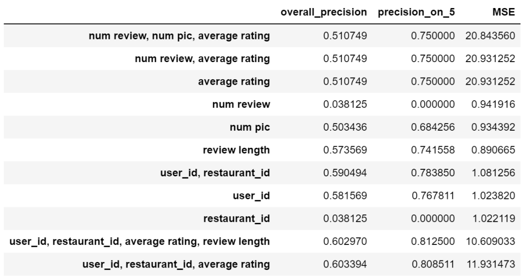


	overall_precision	precision_on_5	MSE
baseline	0.506702	0.679928	1.976686e+00
restaurant average model	0.568796	0.739307	9.423359e-01
pearson similarity model	0.574272	0.746904	4.024646e+27
linear regression model	0.589031	0.765947	1.151952e+00
latent factor model	0.616194	0.785013	8.920521e-01

According to the metric result evaluated on each model attempted, the latent factor model performs the best with the lowest MSE and highest precision both overall and on predicting 5 star ratings. On the other hand, the baseline model of random prediction with probability equal to distribution performs significantly less accurately as compared to the other models in terms of precision and MSE. Other models have moderate performance, with the restaurant average model performing relatively worse and linear regression performing relatively better.

Features and Parameters Interpretation

During the optimization process on models, especially on linear regression models that attempted various different combinations of features, we determined that features like average rating of the restaurant, number of pictures in the review, length of review and user id to be predictive. Other features such as number of historical reviews and restaurant id to be not very predictive.



	overall_precision	precision_on_5	MSE
num review, num pic, average rating	0.510749	0.750000	20.843560
num review, average rating	0.510749	0.750000	20.931252
average rating	0.510749	0.750000	20.931252
num review	0.038125	0.000000	0.941916
num pic	0.503436	0.684256	0.934392
review length	0.573569	0.741558	0.890665
user_id, restaurant_id	0.590494	0.783850	1.081256
user_id	0.581569	0.767811	1.023820
restaurant_id	0.038125	0.000000	1.022119
user_id, restaurant_id, average rating, review length	0.602970	0.812500	10.609033
user_id, restaurant_id, average rating	0.603394	0.808511	11.931473

In the latent factor model that performs best in terms of both precision and MSE, the lambda regularization value for beta U is significantly greater than the lambda value for beta I, indicating that the rating on a restaurant is related more to user's preference instead of the restaurant, which agrees with our analysis that restaurant id is not very predictive but user id is very predictive as shown in the table above.

Citations

[1] Xinchi Gu, Long Jin, "Google Local Dataset Analysis", UCSD, 2015. https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/wi15/Xinchi_Gu_Long_Jin.pdf

[2] Shiyi Hua, Akshay Kotha, Ziyuan Yan, "Rating Prediction on Google Local Data", GitHub Repository. https://github.com/akshayreddykotha/rating-prediction-google-local/tree/master

[3] Zefang Liu, "Predicting Restaurant Ratings from Yelp Reviews", arXiv, 2020. https://arxiv.org/pdf/2012.06690.pdf

[4[Yiwen Guo, ICME, Anran Lu, ICME, and Zeyu Wang, "Predicting Ratings and Popularity Change of Restaurants", Stanford University, 2017. https://cs229.stanford.edu/proj2017/final-reports/5244334.pdf

[5] Sherin Claudia, "Movie Rating Prediction", Kaggle. https://www.kaggle.com/code/sherinclaudia/movie-rating-prediction

[6] An Yan, Zhankui He, Jiacheng Li, Tianyang Zhang, Julian McAuley, “Personalized Showcases: Generating Multi-Modal Explanations for Recommendations”, UC San Diego, 2023. https://arxiv.org/pdf/2207.00422.pdf