Factors Influencing Ratings for Consumer Businesses

Advaith Ravishankar, Ketki Chakradeo, Arnav Talreja

Introduction

Overview

With the rise of online platforms like google maps and amazon, user-generated reviews and ratings have become a valuable resource for understanding user preferences. According to Forbes[1], positive reviews with ratings between 4.2 and 4.7 and reaching over 101 reviews can boost sales by 250%, which tell us that reviews have significant influence of user likability

Such a drastic increase in sales makes it necessary to understand which factors are responsible for positive reviews. If this can be extracted, we can tailor a recommendation system for users by predicting which businesses they will rate highly, and return the top K ratings for them.

This research focuses on a dataset derived from Jiacheng Li et al.[2] and An Yan et al.[3], which includes comprehensive review information from Google Maps for the United States. Given the scope of the original dataset, which spans millions of businesses and users across the United States, we have constrained our analysis to a subset of the data focused on businesses located in Hawaii. This subset provides a manageable yet representative sample, comprising 1.5 million reviews, over 21,000 businesses.

We propose four predictive modeling approaches: a mean rating and Jaccard Collaborative Filtering as baselines, a naive random forest regression model using TF-IDF and metadata, and a XGBoost. Each model is evaluated using Mean Squared Error (MSE) and R-squared , providing insights into both performance and generalizability.

With these approaches, we will be able to identify which factors play a critical role in ratings so we can use them to suggest businesses they will rate highly.

Previous Work

Similar datasets includes Yelp's data[4] about businesses and their information. This dataset consists of similar information to the google maps dataset but lacks the spatial information through latitude and longitude data. It also focuses on restaurants rather than on business.

Prior approaches make use of collaborative filtering and SVD[5], to predict restaurants users can attend by predicting the top restaurants they would rate. We will be building on this by using ensemble learning.

Research Question

Can we predict what rating (1 to 5) a user will give to a business based on user interaction (the users rating history, and review), and business metadata (location, average_rating, hours)?

Given a strong prediction, what factors are most responsible for ratings so that we can use them to create a recommendation system that suggests businesses that they are likely to rate highly.

Dataset

Description

The data set is taken from Jiacheng Li et al (2022)[6] and An Yan et al[7] which contains review information from Google Maps (ratings, text, images, etc) and business

[1] https://www.forbes.com/councils/forbesbusinesscouncil/2023/11/15/the-power-of-reputation-how-to-get-more-reviews-from-customers/

[2] https://arxiv.org/abs/2202.13469

[3] https://arxiv.org/abs/2207.00422

[4] https://www.yelp.com/dataset

[5] https://towardsdatascience.com/yelp-restaurant-recommendation-system-capstone-project-264fe7a7dea1

[6] https://arxiv.org/abs/2202.13469

[7] https://arxiv.org/abs/2207.00422

metadata (address, geographic info, descriptions, category information, price, open hours, etc.), and links (related businesses) up to Sep 2021 in the United States. The full dataset consists of 666,324,103 Reviews, 113,643,107 Users and 4,963,111 Businesses. Due to computational and storage constraints for a class project, we are taking a subset of this dataset by constraining the location to only Hawaii. The Hawaii dataset consists of:

1. 1,504,347 Reviews (0.22% of the full dataset)
2. 64,231 Users (0.05% of the full dataset)
3. 21,507 businesses (0.43% of the full dataset)

The data is formatted to:

Label	**Format**	**Description**
user_id	string	ID of the reviewer
gmap_id	string	ID of the business
rating	integer	rating of the business by user
text	string	text of the review by user
num_of_reviews	int	number of reviews for the corresponding gmap_id
latitude	float	latitude of the business
longitude	float	longitude of the business
avg_rating	float	average rating of the business
hours	[[day_1, hours], …]	open hours formatted
name	string	name of the reviewer
time	int	time of the review (unix time)
pics	URL	pictures of the review
resp	string	business response to the review including unix time and text of the response
address	string	address of the business
description	string	description of the business
category	list	category of the business
price	int	price of the business
MISC	string	MISC information
state	string	the current status of the business (e.g., permanently closed)
relative_results	list	relative businesses recommended by Google
url	URL	URL of the business

Table 1: Dataset Keys

There are 23 variables in the dataset, however, we filtered the data set to only include the data points directly related our task (user_id, gmap_id, rating, text, num_of_reviews, latitude, longitude, avg_rating, hours)

User_id and gmap_id are the independent variables and rating is the outcome we are predicting, text is important as it carries sentiment describing the review, num_of_reviews is significant as more ratings tend to attract more people, latitude and longitude give geospatial information, avg_rating gives the current score of the business which might influence people's thoughts and hours as the longer something is open, the more engagement it will receive from users.

Data Preprocessing

Overview

As *text* and *hours* are the only non numeric variables apart from *user_id* and *gmap_id*, we processed them to be usable for a recommender system.

Term Frequency-Inverse Document Frequency (TF-IDF)

For the text dataset, we extracted the TF-IDF from the entire dataset. We used the following formulae:

$$tf(t, d) = f_{t,d} / \sum_{t' \in d} f_{t', d}$$

$$idf(t, D) = log(N / |\{d: \in D \, and \, t \in d\}|)$$

Where t is the term, d is the document, $f_{t,d}$ is the frequency of t in d, N is the total number of documents and D is the set of of all documents

With this we obtain a quantifiable metric for each word in the reviews. We then extract the tf-idf of the top 3 occurring terms in the review and have them as variables. We use this as an alternative as extracting a vector the size of the corpus is too computationally expensive for our resources.

Days Open

The hours are formatted to a nested list. To quantify this information, we took the number of days the business is open as a variable which is extracted by the formula

$$days_{open} = | hours |$$

Extracted Variables

After applying these transformations we have added the following variables

Label	**Format**	**Replaces**	**Description**
tfidf_1	float	text	TF-IDF of most frequent term
tfidf_2	float	text	TF-IDF of second most frequent term
tfidf_3	float	text	TF-IDF of third most frequent term
days_open	float	hours	The number of days a business is open

Table 2: Extracted Variables

Exploratory Data Analysis

Missing Text Data

With reviews, the text input may be left blank. Therefore, when we look at the distribution of reviews with and without text:

	With Text Review	**Without Text Review**
Number of Observations	852,596	651,751
Percentage (%)	56.7	43.3

Table 3: Missing Text Data

This means that 43.3% of the observations have no text which leads to a tf-idf of zeros. As half the data is missing, this might make it difficult for the model to correlate text in the reviews to ratings.

Data Distributions

As all the data points are now numeric, we can visualize their distributions to understand outliers (we ignore latitude and longitude information as the values close to each other due to the proximity in Hawaii).

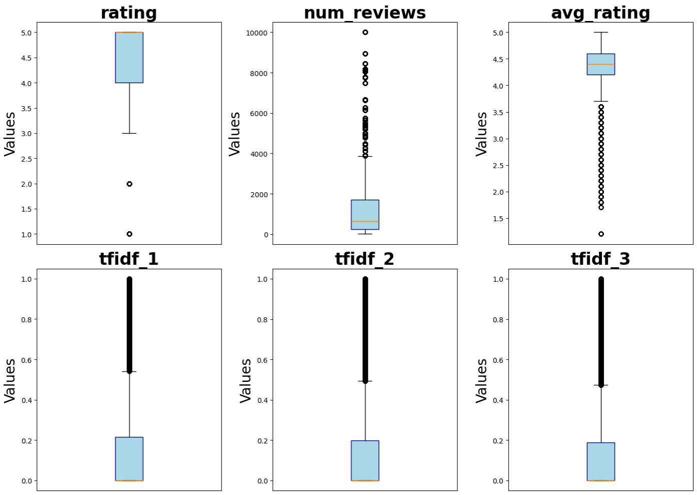

Figure 1: Boxplot of Variables

From the distributions, ratings are concentrated between 4 and 5 with lower quartiles being until 3. The number of reviews for each business are concentrated around 1000 and the ratings for each business are around 4.5. This shows a bias where businesses in Hawaii tend to receive a higher rating.

All three tf-idf data points are concentrated near 0 with outliers above. As 43.3% of the data is missing with TF-IDFs of 0, it biases the data towards 0.

Correlation Matrix

We computed the correlation of each factor with one another to see which factors are strongly correlated with ratings.

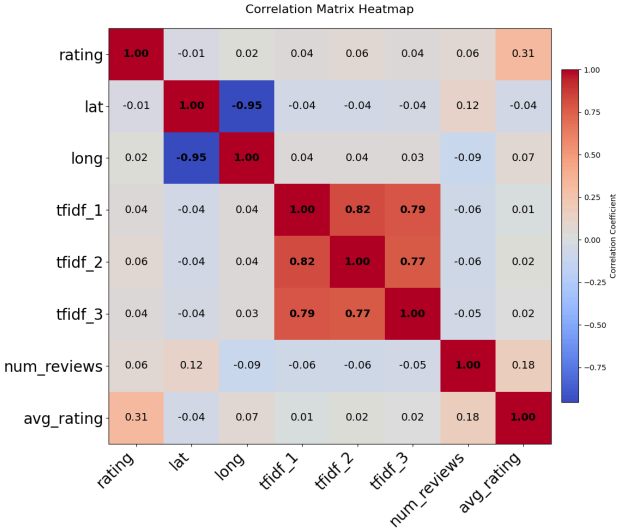

Figure 2: Correlation Heatmap

Average rating has a correlation of 0.31 with rating, showing a positive relationship and influence with ratings.

Every other factor has a near 0 correlation. This means that alone, they do not have statistically significant relationships with ratings. However, there still might be a multivariate relationship which is lost. For example, tf-idfs for 1 word wont alone have meaning but with the top 3 together, might correlate with rating.

Geospatial Distribution

As our data has latitude and longitude information, we wanted to visualize the distribution in a geospatial manner.

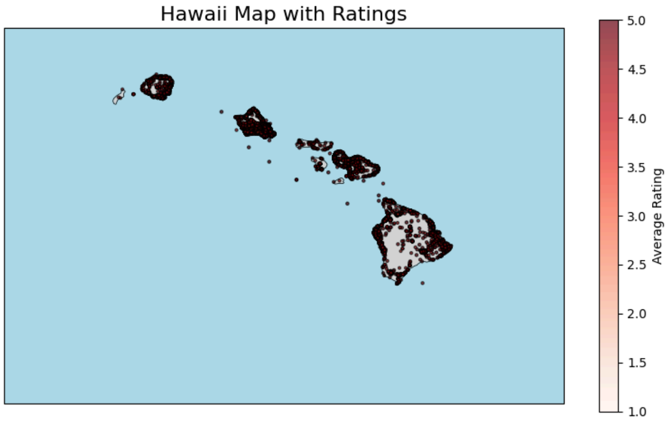

Figure 3: Ratings Across Hawaii

The coast of each island is covered densely with rating points. From the data distribution section, we know that the ratings are skewed toward 4 to 5 which fits the distribution in the map.

The middle of each island is densely covered in foliage[8], which is why there are no business ratings in that area. This shows us that the data is well distributed and representative of hawaii.

[8] https://gisgeography.com/wp-content/uploads/2013/02/Hawaii-Satellite-Map.jpg

Predictive task

Overview

The study aims to first predict the rating a user will give to a particular business (1 to 5). Based on the model, it will then delve into which feature is responsible for positive ratings.

For this task, we will predict ratings in 4 ways:

1. Mean only prediction (Baseline 1)
2. Jaccard Collaborative Filtering (Baseline 2)
3. Random Forest using TF-IDFs, average reviews, and location.
4. XGBoost

Other Considered Approaches

We did not create a latent factor model as initializing an adjacency matrix was too computationally expensive.

Validation

For each model, we will validate our approach by mean squared error on the train set (for performance), mean squared error on the test set (for generalizability) and R-squared (explained variability).

$$Mean\ Squared\ Error\ =\ (1/\ N)\ *\ \sum_{i=1}^{N}\ (y_i - \widehat{y_i})^2$$

$$R^2\ =\ 1\ -\ (SS_{residual}/SS_{total})$$

Approach 1 and approach 2 serve as baselines for the task as it is the simplest form of prediction. We will treat the other approaches to be valid if they have a lower MSE relative to approach 1 & 2.

We trained each model on 1,002,548 samples and tested it on 250,638 samples (80-20 split).

Models

Approach 1 & 2: Baselines

The mean approach serves as a baseline where we took the mean of all ratings and compared them with our train and test set.

The collaborative filtering is also a simple model from class. We used the following formulation

$$\hat{r}_{u,i}\ =\ ri_{avg}\ +\ \sum_{j \neq i} (r_{u,i} - ri_{avg}).\ sim(U_i,\ U_j)/sim(U_i,\ U_j)$$

These were used to compare the effectiveness of approach 3 and approach 4 prediction.

Approach 3: Random Forest Regression

Random Forest Regression was chosen for ensemble learning. By combining several naive classifiers, we can construct a strong predictor of ratings. We defined the following parameters: n_estimators=100, random_state=42, min_sample_split=2 and min_samples_lead = 1.

We did perform a grid search for the number of estimators (70, 80, 90, 100) but all results were similar (we encountered issues)

Approach 4: XGBoost

Like approach 3, we wanted to leverage ensemble learning so we designed an XGboost on a linear classifier.

The model was set to gb-linear and trained reg:linear. We optimized the hyperparameters to n_estimators=1600, max_depth=10 using gridsearch and found optimality.

Results

Model Performance

For each model, the performance on the train set, test set and r-squared on the test set is reported.

Model	MSE Train	MSE Test	R-Squared

Mean Only	0.86	0.86	0.00
Jaccard Collaborative Filtering	0.92	0.92	0.02
Random Forest	0.37	0.78	0.09
XGBoost	0.81	0.81	0.10

Table 4: Model Performance

The mean only model has an MSE of 0.86 on both the test and train set. This means that the model, on average, is about 0.92 points off the rating (RMSE of 0.92). The R-squared is 0 which is by definition with predicting by mean.

The collaborative filtering model has an MSE of 0.92 on both the test and train set. This means that the model, on average, is about 0.96 points off the rating (RMSE of 0.96). The R-squared is 0.02 which tells us that 2% of the variability in rating is explained by the model.

With the Random Forest Model, the MSE on the train set is 0.37 and 0.78 on the test set. This is an improvement on the baseline, showing that this model is a viable predictor. It also shows that on average, the model is 0.61 points off on train set ratings and 0.88 points off for test set ratings. This difference highlights an overfit in the model, showing that the model has memorized the train set. Furthermore, the model has an R-squared of 0.09 which shows that the model only explains 9% of the variance in ratings, which is a modest improvement over the baseline but still leaves much of the variability unexplained.

Finally, with the XGBoost Model, the MSE on the train set and the test set is 0.81. This is a slight improvement on the baseline, showing that this model is a viable predictor. It also shows that on average, the model is 0.90 points off the rating. The R-squared is 0.10 which tells us that 10% of the variability in rating is explained by the model.

Feature Importance: Random Forest Regression

We extracted all features; influence on the predictions.

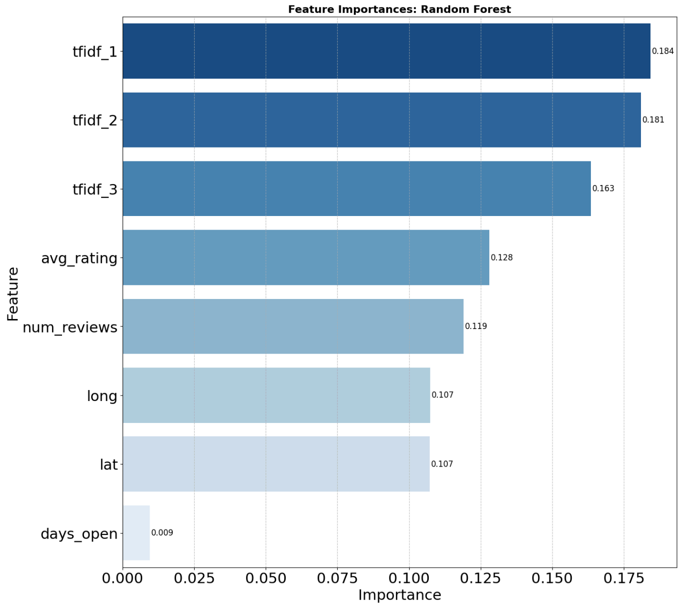

Figure 4: Random Forest Importance

All the importance scores are close to 0.1 except open day which means on an absolute scale all factors have weak influence on rating.

When evaluating on a relative scale. The top 3 factors are the tf-idfs with importance of 0.184, 0.181 and 1.63 which tell us that the text for the review is the strongest factor for the rating.

Feature Importance: XGBoost

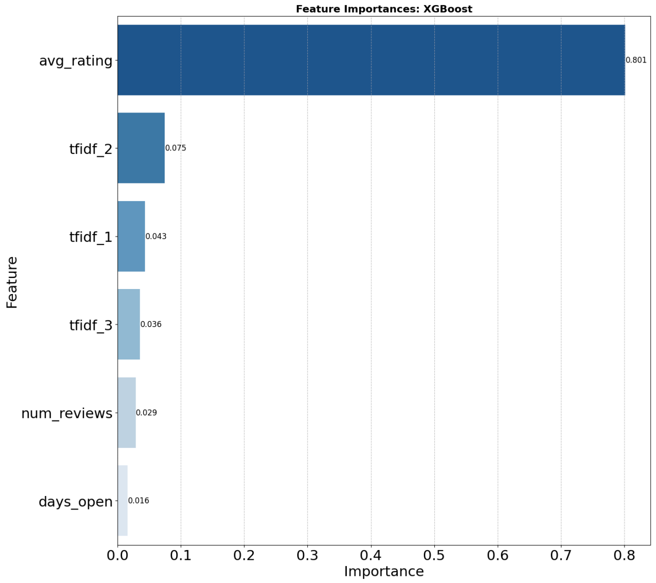

Figure 5: XGBoost Importance

Avg_rating had the highest importance of 0.801 showing that it was significant in predicting ratings. The tf-idf variables followed this with importance of 0.075, 0.043, 0.036 showing that text was the next important factor but as the values are close to 0, they do not have significance.

Discussion

The Random Forest model shows promise as a viable predictor, significantly reducing the MSE on the train set to 0.37 compared to the baseline models, demonstrating its ability to capture complex patterns. However, the test MSE of 0.78 indicates some degree of overfitting, as the model performs better on training data than on unseen data. Its R-squared value of 0.09, suggests it captures 9% of the variance in ratings, which is a notable improvement over simpler approaches like the mean-only and collaborative filtering models.

The XGBoost model stands out for its consistent performance across train and test sets, achieving an MSE of 0.81 for both. This indicates better generalizability and reduced overfitting compared to Random Forest. Additionally, its R-squared value of 0.10, while still low, marks the highest variability explained among the models tested. Feature importance analysis from XGBoost further reveals that avg_rating is a key predictor, contributing significantly to the model's performance.

Prior Work Comparison

As we used a different dataset, we could not compare results. We attempted to find a similar task using the google maps dataset, but we could not find anything.

Limitations and Improvements

When training the random forest model, the runtime was 18 minutes. This made it difficult to run gridsearch to optimize the model. With more computational strength, we can leverage parallel computing to find the optimal tree.

In addition, this model will work in Hawaii but as it is not exposed to data outside of the state, it will fail to generalize for out of state businesses. With more storage capacity, one could train on the entire dataset and make it generalize for any business.

Another issue is that R-squared for the models are below 10% which indicates poor explanation for rating's variance. One way to improve it is to add more information which correlates strongly with ratings.

Conclusion

The results demonstrate that while Random Forest and XGBoost models provide improvements over baseline approaches of mean only and jaccard collaborative filtering, there is significant room for refinement. The Random Forest model successfully captures complex patterns with a low train MSE of 0.37, but its higher test MSE of 0.78 and R-squared of 0.09 highlight overfitting and limited generalizability. On the other hand, the XGBoost model offers better consistency across train and test sets, achieving an MSE of 0.81 and an R-squared of 0.10, the highest among the tested models. Feature importance analysis identifies **avg_rating** for ratings as the most influential variable for predicting what rating a user will give, followed by **review text features**, showing that these factors influence a business's rating.

Therefore, to create a strong recommendation system, we can use the sentiments from the text and average rating for the business as indicators to predict what a user will rate a business and suggest their top K predicted ratings.

References

[1] *https://www.forbes.com/councils/forbesbusinesscouncil/2023/11/15/the-power-of-reputation-how-to-get-more-reviews-from-customers/*

[2] https://arxiv.org/abs/2202.13469

[3] https://arxiv.org/abs/2207.00422

[4] https://www.yelp.com/dataset

[5] https://towardsdatascience.com/yelp-restaurant-recommendation-system-capstone-project-264fe7a7dea1