Predictive Analysis of Google Business Profiles Ratings Based on GPS Coordinates and Review Text

Introduction

Advances in technology, such as the Google search engine, are great steps forward globally. With this development, people nowadays rely on the internet in giving them recommendations when making choices. In the meantime, the user interactive platform has also facilitated the sharing of information, and thus reviews voluntarily left by some are influencing the others' choices, such as in the settings of purchasing products and visiting restaurants that have higher review ratings. The ratings could directly impact a business' customer flows and economic growth. In this project, thus, we intend to explore how Business profiles ratings derived from Google Map/Local can be related to individual users' and businesses' features. Are certain users more likely to give lower ratings? Are some businesses more likely to have positive reviews and higher ratings? To answer these questions, we conducted exploratory data analysis on the dataset of Google Local Reviews, and built a baseline predictive model of logistic regression classifier with tuned hyperparameter class_weight , which was later evaluated by the performance metric of accuracy, balanced accuracy, and confusion matrix. Subsequently, we also built models such as Multi-layer Perceptron and random forest to see if improvements on models could be made.

Section 1: Datasets and EDA

There are three datasets readily available, respectively reference to user, business, and review details information. The **Users** dataset has 4,567,431 entries, including:

- **userName -** name the user as str
- **jobs -** user occupation
- **currentPlace -** current location stored as cities and GPS coordinates
- **previousPlace -** previous location stored as cities and GPS coordinates
- **education -** education level
- **gPlusUserId -** gps id

The **Businesses**/places dataset has 3,116,785 entries, including:

- **name -** business name on profile
- **price -** None
- **address -** business address
- **hours -** business opening hours
- **phone -** business phone
- **closed -** boolean indicating if the business is closed or not
- **gPlusPlaceId -** gps id

The Reviews dataset consisted of reviews about businesses from Google Maps has 11,453,845 entries, including:

- **rating -** overall rating of the business
- **reviewerName -** username
- **reviewText -** contents of the review
- **categories -** business category
- **gPlusPlaceId -** business gps id
- **unixReviewTime -** when review made
- **gPlusUserId -** user gps id

One special thing about these datasets is that they contain geographic information saved as lists of gps coordinates for each business and each user. This is also the feature we intend to dive into more to investigate if location-based information can be used in review rating prediction. To offer a more specific lens and avoid the misinterpretation of review texts written in other languages, we only focused on reviews of businesses in the United States that are also made by US users in the following tasks. After some research, we have the gps x coordinates in a range of 25 - 50 and y coordinates in a range of -124 - -67 to keep only reviews made in the US and by US users.

In the EDA part, we went through these tasks:

- Summary statistics on the ratings
- Analysis of the geographical data
- Text analysis
- Potential relationship pattern

First of all, there are 740,935 ratings that are made by the US users to businesses located in the United States. As observed in Figure 1, the histogram of rating frequency shows an obvious left skewness, with a mean of 4.152 and standard deviation of 1.21 indicated by Table 1. The second, third and fourth quartiles all correspond to ratings of 5.

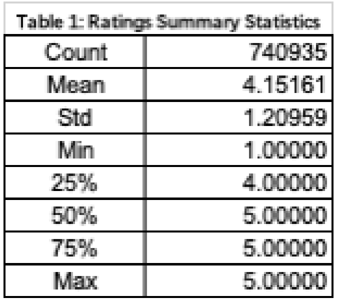

Table 1: Ratings Summary Statistics

Count	740935
Mean	4.15161
Std	1.20959
Min	1.00000
25%	4.00000
50%	5.00000
75%	5.00000
Max	5.00000

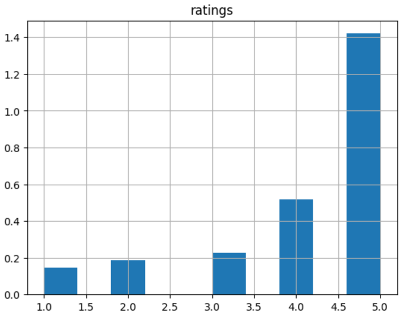

Table 1 & Figure 1: Basic Rating Statistics

To view these statistics geographically, scatter plots were also generated. The Business Average Ratings in Figure 2 has illustrated typically lower ratings in the western and southwestern parts, compared to the rest of the regions. The User Average Ratings in Figure 3 has also embedded a similar pattern: with users from the west inclining to leave bad reviews than users in other parts of the map.

In terms of the text analysis, the word cloud graph below shows words with most frequencies. Words such as place, great, food, and service are common in these reviews.

From above, we can see that there is an underlying relationship between users' and businesses' geographical information and the profile ratings - businesses that are more likely to receive lower ratings and users that are more likely to leave lower ratings. Accordingly, in following predictive tasks, we want to figure out if knowing geographical information of users and businesses can offer us any useful insights of the ratings they may have.

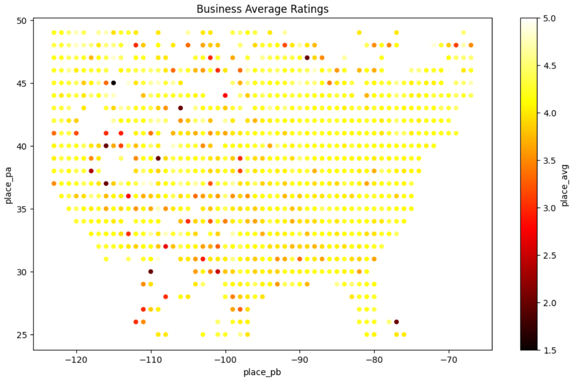

Figure 2: Who Received Bad Reviews More?

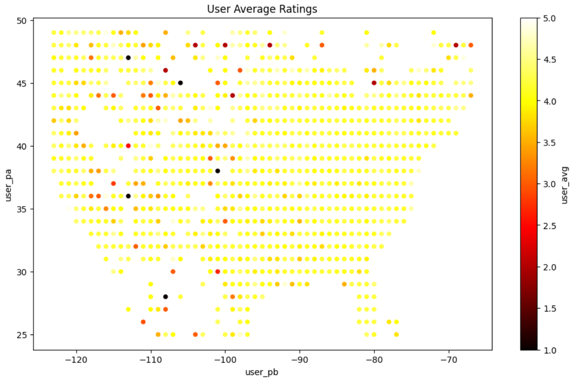

Figure 3: Who Likes to Give Bad Reviews?

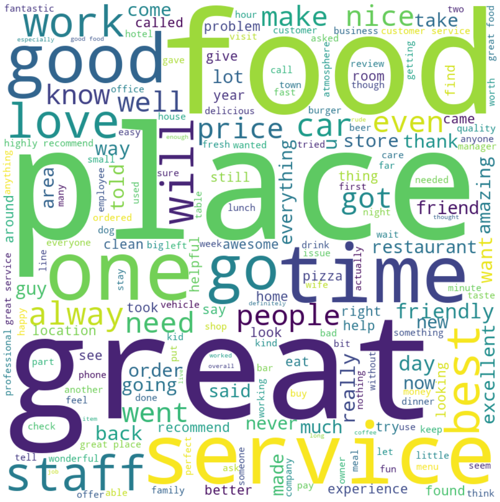

Figure 4: Word Cloud on Review Text

Section 2: Predictive Task

Thinking back to the datasets' attributes, we can see that features such as reviewText and categories from Reviews, business GPS coordinates from Businesses, and user GPS coordinates from Users are fairly justifiable to use to build our models.

1. **reviewText**, which is the review textual contents left by a certain user, saved as strings
2. **categories**, which is the category of the business profile
3. Businesses **gps**, which is a pair of business gps coordinates saved as lists
4. Users **gps**, which is a pair of user gps coordinates saved as lists

These potential variables can be good candidates as our predictors. However, since we wanted to build our models based on users' and businesses' location information and the granularity of the categories variable is also tough to adjust, we decided to only take the gps variables from the two datasets as well as the reviewText variable for further text sentiment analysis. The rating variable in the Reviews datasets serves as our response variable, and it's saved in float yet only has been assigned five discrete numbers from 1 to 5. Therefore, we took this predictive task as a classification problem.

Before deciding which models as our classifiers, we have few performance metrics in mind for later evaluation. First of all, **accuracy**, the ratio of sum of true positives and true negatives out of all the predictions made, can give us a direct note of the model's performance. In consideration of the clustered independent values of ratings as introduced in part 1, where its frequency exhibits a strong left skewness, accuracy itself is not sufficient. Therefore, to avoid giving out recommendations that are biased toward the good, we also used **balanced accuracy**, which is commonly used in multi-class classification. Balanced accuracy is the arithmetic mean of sensitivity and specificity when encountering imbalanced domains. Other than that, we also encompass **confusion matrix**, a kind of contingency table. It visualizes and summarizes the performance of a classification algorithm. Its i-th row and j-th column entry indicates the number of samples with the true label being i-th class and predicted label being j-th class. Our confusion matrix is supposed to be a 5 x 5 table because we have 5 classes, from 1 to 5.

Recapping from the models we learned previously in and outside class, there are three we want to build for this project:

- **Logistic regression classifier**, which is a supervised learning that utilizes logistic functions
- **Multi-layer Perceptron Classifier**, MLP, which is a Neural Network algorithm that learns the relationships between linear and non-linear data, or a complexion of logistic regression

- **Random forest classifier**, which is an ensemble learning method that fits a number of decision tree classifiers on various sub-samples of the dataset

Familiarized in Assignment 1, we decided to go with logistic regression classifier as our baseline model. MLP and random forest are also built to see rooms for improvement. More model details and implementation are enclosed in the next section to predict business profile ratings based on users' and businesses' gps locations.

Section 3: Model

3.1 Feature Selection

In the prediction task, our interest is the latent information encoded by implicit data, thus we directly use geographical data and review text as features for predictions. As introduced above, the features we took into account are users' gps coordinates and businesses' coordinates. With these two predictors, we expect to see if we can make predictions of a user's rating/preference knowing his/her location and the business's location, thus making further recommendations.

3.2 Data Preprocessing

For the geographical data, we convert them into correct longitude, latitude format with correct range. Next, we use a standard scaler to normalize it since we might try models such as neural networks that require normalization. For the text data, we use tf-idf vectorizer to extract the vector information from each vector, then we use the method from principal component analysis, the truncatedSVD method, on the sparse vector result from tf-idf vectorizer to perform the latent semantic analysis (LSA).

3.3 Model Selection

For the model selection, with large data and abstract tasks (analysis of geographical data and text) in mind, we forwent models like support vector machine (SVM) and K-nearest neighbor (KNN), which require a large amount of computation power. And the ideal model in our mind is the complex neural network which can handle abstract tasks.

Baseline: Logistic Regression

The baseline model we chose is the logistic regression, the most basic linear classification model. It is functioned by applying an activation layer on the linear model output. The activation layer in logistic regression is the sigmoid function. Since the sigmoid function results in a value between 0 and 1, it is used to approximate the probability of one class, or logit value. Moreover, since we are doing the multi-class classification problem, our logistic regression is actually using the method called softmax regression, which uses sum and normalization on probability value to make predictions. Our hyperparameters: set balanced class weight.

	Train	Test
Accuracy	0.7699	0.4794
Balanced Accuracy	0.6849	0.3428

Table 2: logistic regression result

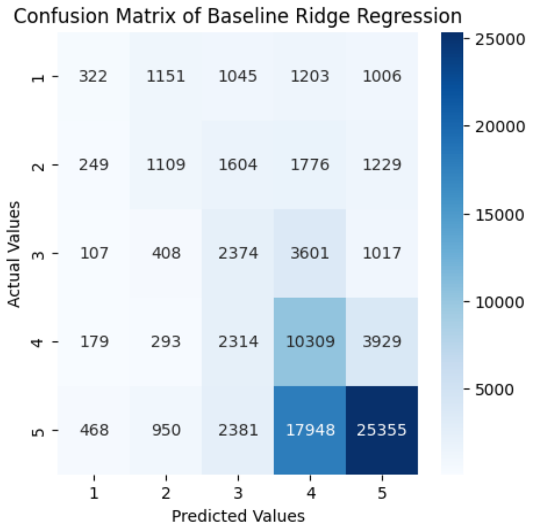

Figure 5: baseline confusion matrix

Random Forest Classifier

The Random Forest algorithm constructs multiple individual decision trees and uses a voting mechanism to decide the result. The significance of random forest classifier is that it is built based on decision trees which is a powerful algorithm dealing with tabular data. Moreover, it constructs multiple trees which enable efficiency with parallelism and prevent overfitting by limiting the depth of individual trees. However, since our task is abstract tasks we are not expecting the random forest classifier to perform very well. Our hyperparameters: set balanced class weight and limit max tree depth to ten.

Model Result:

	Train	Test
Accuracy	0.4581	0.4424
Balanced Accuracy	0.4672	0.4349

Table 3: random forest result

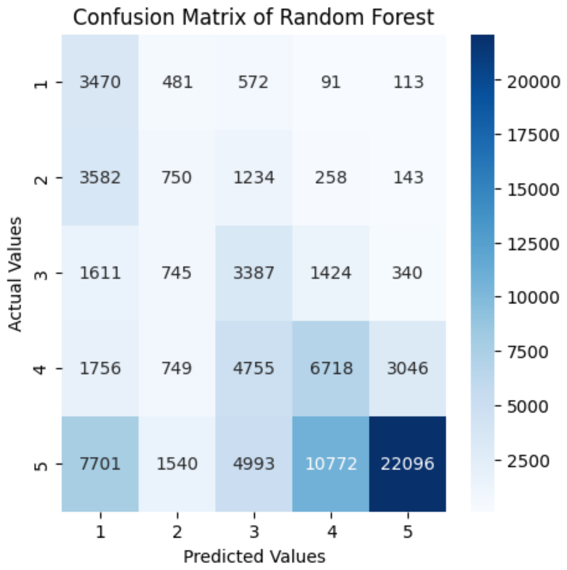

Figure 6: random forest confusion matrix

We can see that the random forest classifier has a relatively lower accuracy on the test set compared to our baseline model (0.02) from Table 3. However, we observed that it has a better balance accuracy rate compared to baseline model. In addition, the proximity in accuracy between train and test set indicates our model is not overfitting. The higher balanced accuracy might help us make our predictions more safe from the lower rating predictions, which is, wrongly predicting a bad rating as a good rating.

Multi-layer Perceptron Classifier (MLP)

One of our expected models that can perform well in this task is the multi-layer perceptron classifier. It is a simple neural network that can learn both linear representations and non linear representations with its non linear activation function. We believe this model can handle the complex geographical and text data better. Our hyperparameter: we design a two layer neural net with first layer of 64 nodes and second layer with 16 nodes, all using ReLU activation function.

Model Result:

	Train	Test
Accuracy	0.6579	0.6537
Balanced Accuracy	0.4601	0.4505

Table 4: multi-layer perceptron result

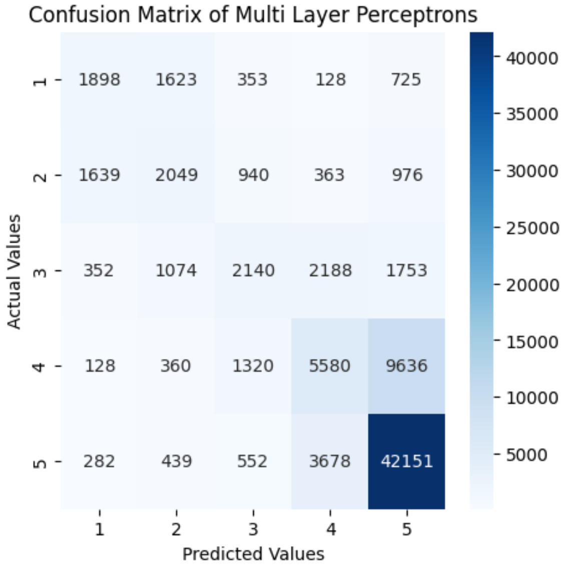

Figure 7: MLP confusion matrix

The results support our hypothesis. We can see that the multi-layer perceptron classifier reaches a high accuracy of 0.65 (0.18 higher vs. baseline) and balanced accuracy of 0.45 (0.11 higher vs. baseline).

3.4 Problems and Unsuccessful Attempts

In the process of feature engineering and model testing, we meet some challenges. Firstly, we try to directly use the result from the tf-idf vectorizer. However, this is a huge sparse matrix. Therefore, we use the PCA model to reduce its dimension such that our model can make predictions within a reasonable amount of time. Secondly, overfitting is a problem when we are using decision tree based models. We tried both decision tree, random forest, and XGBoost models. However, our highly imbalanced data make those decision tree models easily overfit. Therefore, we choose the random forest classifier with our hyperparameters tuning that limits the model's complexity and choose that as a representation of decision tree models on our task.

3.5 Validity

The most foundation method we use to validate our model is the train test split. We split our data into 90:10 size. The train dataset size comes to 740,935 and the test counterpart has a size of 82,327. In the whole data preprocess and training process, our model can't see the test data to make our model's result generalize better. Also we used the random train test splitter that can eliminate sample bias.

Section 4: Related Literature

The datasets enclosed in this project are collected by Professor Julian McAuley and his team for research purposes. His team had worked on the topic of sequential recommendation algorithms using these datasets.[1,2]

A location-based review rating prediction is a popular topic recently. Studies led by a group of student researchers have also looked into the recommender system by user information such as social links and geolocations.[3] With a more comprehensive datasets Yelp reviews, this study is able to dig deeper into the function of users' regular location of places that they "checked in" around and the real-social-linked friends' choices. A point-of-interest concept is also included in this research to assist the geolocations in predicting the Star Rating and the user's preference.[3] Their choices of models are Random Forest, Gradient Boosting Machine, and eXtreme Gradient Boosting, with evaluation metrics different from ours as MAE and RMSE, partially due to their treatment of rating as a continuous variable.

Since our dataset includes the text data, relative research includes using those text data for classification and translation, we have to mention the famous paper in the deep learning field: *Attention Is All You Need*.[4] This paper uses a structure called transformer to process text data and is reaching state-of-the-art performance on text related tasks. Our attempt on using neural networks is trying to get a model complexity close to this one but there is a large gap between performance results.

Section 5: Results & Conclusion

Model Name	Test Accuracy	Test Balanced Accuracy
Logistic Regression	0.4794	0.3428
Random Forest	0.4424	0.4349
MLP	0.6537	0.4505

Table 5: multi layer perceptron result

The data from three distinct models, including their test accuracy, and test balanced accuracy statistics, are shown in Table 5. The final model uses a multi-layer perceptron classifier, as was described above in the section on model analysis, and it performs at an accuracy level of 65.37%, the highest of the 3 models. A reliable indicator of the importance of the findings can be found below in Figure 8.

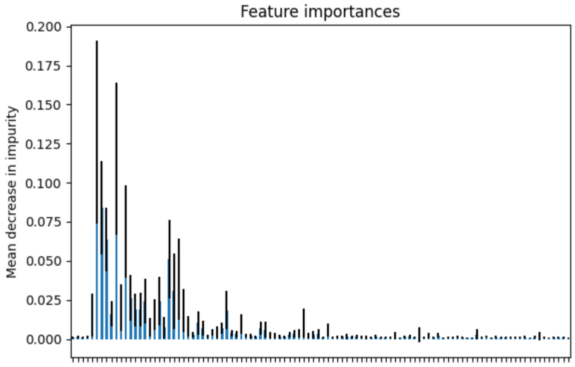

Figure 8: Feature Importance

In Figure 8, the first four features are the position data and the rest one hundred features are the result from the PCA. From the figure, we can see that our random forest classifier is mainly using those encoded text data rather than the geometric data. One thing to note is that our best model, multi-layer perceptron, is a complex neural network that can't easily be used to analyze feature importance. Therefore, we use our random forest to analyze the feature importance. We can't make inferences about whether a multi-layer perceptrons classifier uses that information or not.

In our work, we tried to predict ratings based on geographical and text data. We wish in practice our model can be used to predict and make recommendations to google map users on the business / places they should try to explore. However, our approach still has a lot of room to improve. From the geographical data, we are hoping that the model can understand the geographical networks while we wish the model to do the language processing work, too. To further improve our model, we will choose to separate our task into two models dealing with those tasks differently and use a third model to learn those two models' output to return the final output (ensemble learning), inspired by the literature introduced above.

References

1. **Translation-based factorization machines for sequential recommendation**, Rajiv Pasricha, Julian McAuley, RecSys, 2018.
2. **Translation-based recommendation,** Ruining He, Wang-Cheng Kang, Julian McAuley, RecSys, 2017.
3. **Review Rating Prediction on Location-Based Social Networks Using Text, Social Links, and Geolocations**, Yuehua WANG, Zhinong ZHONG, Anran YANG, and Ning JING, IEICE TRANS. INF. & SYST., 2018.
4. **Attention Is All You Need**, Google, 2017, https://arxiv.org/abs/1706.03762