# Star Rating Classification of Yelp Businesses

Emily Nomura, Brown University, DATA 1030 Final Project
[https://github.com/emilynomura1/1030FinalProject]

## Introduction

The Yelp business dataset is composed of 179,344 unique businesses located in the United States and Canada.[1] The data can be downloaded from the Yelp Open Dataset website. The businesses included in the dataset range from locally owned small shops to widely-known corporate-owned businesses. There are a total of 60 features in the raw, unprocessed data. The final cleaned dataset contained 178,372 observations and 43 features before preprocessing.

The dataset is composed of continuous and categorical numeric variables. The majority of the features are text-based, including a list of categories that the business fits under and various business attributes. The number of attributes that a business has ranges depending on if the business' customers or the business owners themselves input information into Yelp's software. A detailed description of the variables present in the cleaned dataset can be found in the report folder of the project GitHub repository. All features aside from name, star-rating, review count, categories, and city contain missing data.

The problem I aim to solve in this project is classification. More specifically, can the model accurately predict the star rating of a business given a set of predictors? This type of problem is useful to answer because when any food-ordering or business-reviewing software is supplied with new business data, there is no information about the business other than what the business owners input themselves. Once customers start to visit the business, write reviews, and answer questions about the business, this type of prediction model can help to categorize businesses in order to better aid consumer and user decisions.

### Literature Review

The Yelp business dataset is free and easy to acquire online. The implementation of machine learning algorithms in order to predict something about the review or the business given a particular review text is common. The Yelp Github account linked to the Yelp Dataset Challenge provides an example model with the goal of predicting likely business categories given review text.[2]

A report by Li et al. implements support vector machines and random forests in order to predict if a review was voted "useful" or not given the review text. They concluded that a random forest model with 190 trees on six categories returned the highest accuracy of 0.689.[3] Another project by Nabiha Asghar aims to predict the user's star rating on a one point scale from one to five given the review text. Asghar's best model consisted of logistic regression using features composed of the top 10,000 Bigrams and Unigrams, which achieved an accuracy of 64%.[4]

## Exploratory Data Analysis

In the Yelp business dataset, an average star rating of 3 or 4 are the most common, while a star rating of 1 is least common. There is no single predictor that explains the star rating of a business with great certainty. Because the distribution of the target variable is imbalanced, caution should be taken when performing data splitting and data preprocessing.

[1] "Frequently Asked Questions." *Yelp Dataset Documentation*, Yelp.
[2] Clark, S., Artem, A., & Chambers, B. *Dataset Examples - Yelp*, Github.
[3] Li, Y., Liu, Y., Chiou, R., & Kalipatnapu, P. *Prediction of Useful Reviews on Yelp Dataset*.
[4] Asghar, N. *Yelp Dataset Challenge: Review Rating Prediction*, University of Waterloo.

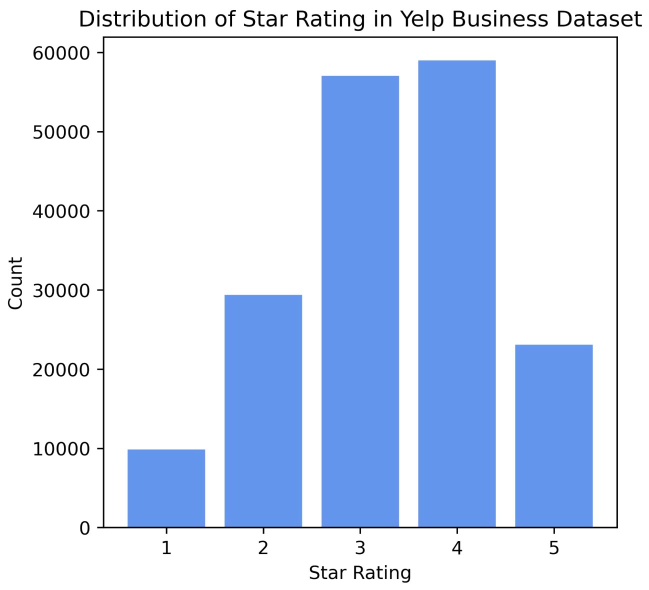


Fig. 1. The average star rating target variable is composed of 5 classes; class 3 and 4 are the most populous.

Some features were removed if 95% or more of their data was missing. After data cleaning, the remaining features that contained missing data were all categorical business attributes.

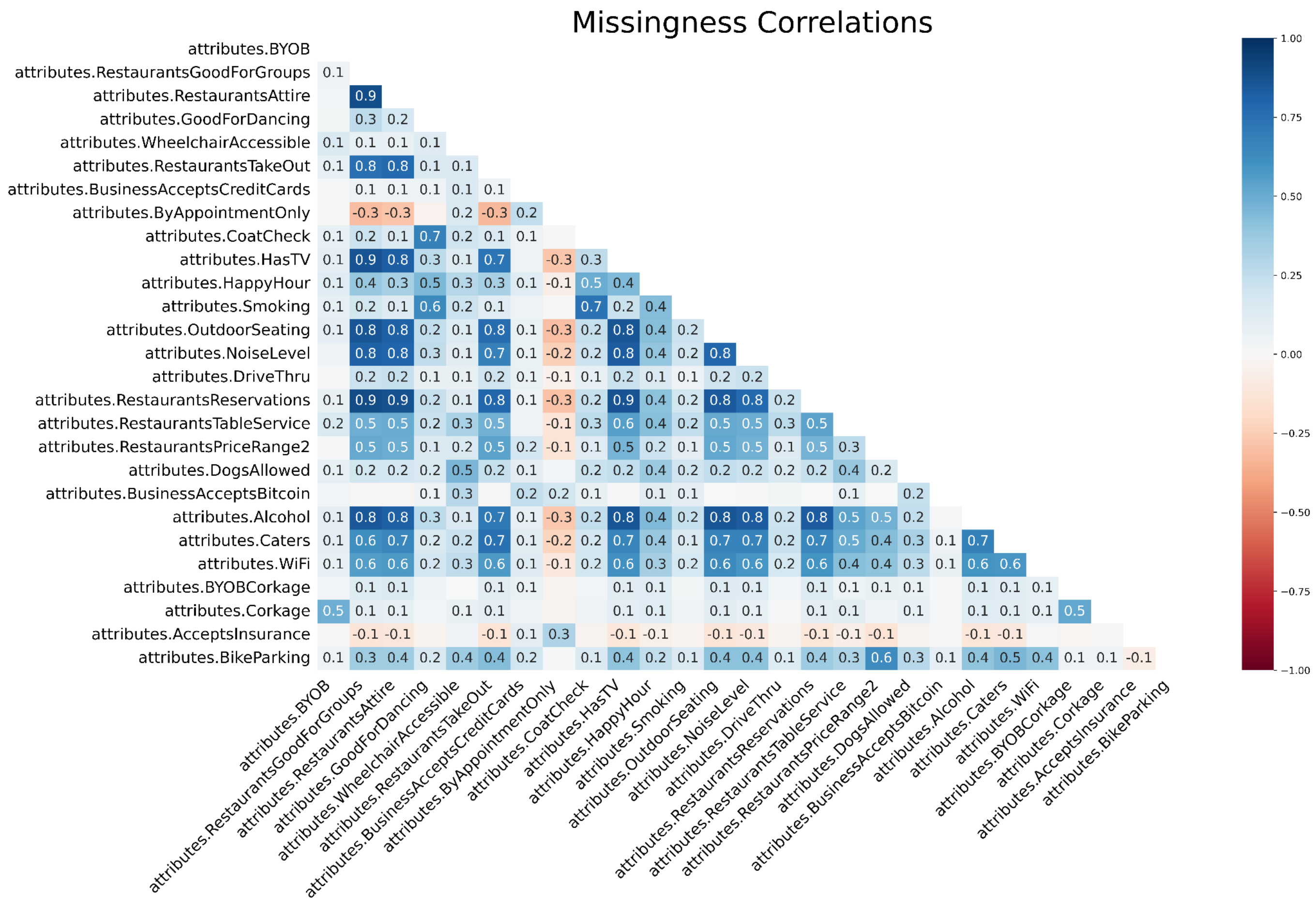


Fig. 2. A missingness correlation matrix was created using the missingno library. Higher values indicate that if data was missing in a particular feature row-wise, data was also highly likely to be missing in a particular feature column-wise.

## Methods

Each observation represents a unique business; the dataset is independent and identically distributed. It does not have a group structure or consist of any type of time-series. The data was split using the train test split function from the scikit-learn preprocessing package. 80% of the feature matrix was split into training, 10% in validation, and 10% in testing. The stratification argument was specified during splitting to ensure stratified splits on the imbalanced data.

A new category for each categorical feature with missing data was imputed. It is difficult to determine whether this pattern of missingness is missing at random or missing not at random. For example, if a user decides not to answer an attribute-related question regarding a business, it could be because they aren't familiar with Yelp's mobile application or because they don't have the time. This would imply that the missing data in business attributes is missing at random. However, if a user decides not to answer a question because they are upset about the experience they had, this would imply that the missingness patterns in certain features are missing not at random.

Five main preprocessing functions were used: ordinal encoder, one hot encoder, min max scalar, standard scalar, and multi-label binarizer. There was only one ordinal feature - the feature indicating restaurant price range. All other attribute features as well as city and state were transformed with the one hot encoder. The review count feature was transformed with standard scalar because the number of reviews for each business was not well-bounded. The features describing the total number of hours a business is open each day were supplied the min max scalar. The final data after preprocessing was composed of 2518 features.

The machine learning pipeline developed was tested on four main algorithms: random forest classifier, k-nearest-neighbors classifier, multi-class linear SVC, and a ridge regression classifier.[5] The parameters tuned for each model varied according to the algorithm used. The values used for parameter tuning and the best model parameters for each algorithm are provided.

| | Parameters tuned | Values tried | Best parameter(s) |
|---|---|---|---|
| Random forest classifier | max_depth,<br>min_samples_split | [50, 55, 57, 60]<br>[7, 8, 9] | 60<br>8 |
| Ridge regression classifier | alpha | [1e-3, 1e-2, 1e-1, 1] | 1 |
| K-nearest-neighbors classifier | n_neighbors | [50, 60, 70] | 50 |
| Multi-class linear SVC | C | [1e-3, 1e-2, 1e-1, 1] | 1 |

Fig. 3. Parameters were tuned using GridSearchCV. Testing values taken from course lecture examples and scikit-learn documentation.

The evaluation metric measured was the weighted F1 score. The reasons for choosing this metric are twofold. Firstly, because the classification problem is multi-class, the F1 score is preferable since we want to compute a balanced measure between precision and recall. Secondly, because the target variable is imbalanced, we want to weigh the metric by class distribution. Thus, the weighted F1 score fits the needs of this imbalanced multi-class classification problem best.

## Results

There are various methods of calculating the baseline accuracy of an imbalanced muli-class classification problem. In a zero rate classifier, the model will always predict the value of the most populous class.[6] The baseline accuracy of a zeroR classifier on this dataset is 0.331. Another method of calculating baseline

[5] *Multiclass and multioutput algorithms*, Scikit-learn.
[6] Lee, Aaron. *Choosing a Baseline Accuracy for a Classification Model*. Towards Data Science.

accuracy is a random rate classifier which makes predictions based on class weights.[7] The random rate classification baseline accuracy for the Yelp business dataset is 0.259.

All four machine learning algorithms achieved a weighted F1 test score above the zero rate classification baseline and the random rate classification baseline. Even the worst model achieved a test score 2.38 standard deviations above the zeroR baseline accuracy. Standard deviations measuring model variability were calculated by running the same model on five different random states. A summary of model performance and uncertainty using different algorithms is provided.

| | AVG of test scores | STDEV of test scores |
|---|---|---|
| Random forest classifier | 0.4476029106 | 0.001320365996 |
| Ridge regression classifier | 0.3759220088 | 0.006656577412 |
| K-nearest-neighbors classifier | 0.3581909249 | 0.01399839945 |
| Multi-class linear SVC | 0.3527848132 | 0.009244487319 |

Fig. 4. The average weighted F1 test score and standard deviation of test scores for each classifier. Five random states were looped through and a variety of parameters were tuned for each model using a grid search.

The most predictive machine learning algorithm was the random forest classifier with a max depth of 60 and a minimum samples split value of 8. In the final model, if a feature had a Gini importance or mean accuracy decrease less than or equal to zero, it was dropped from the feature matrix. The final reduced model was composed of 1828 features and achieved a weighted F1 test score of 0.454.

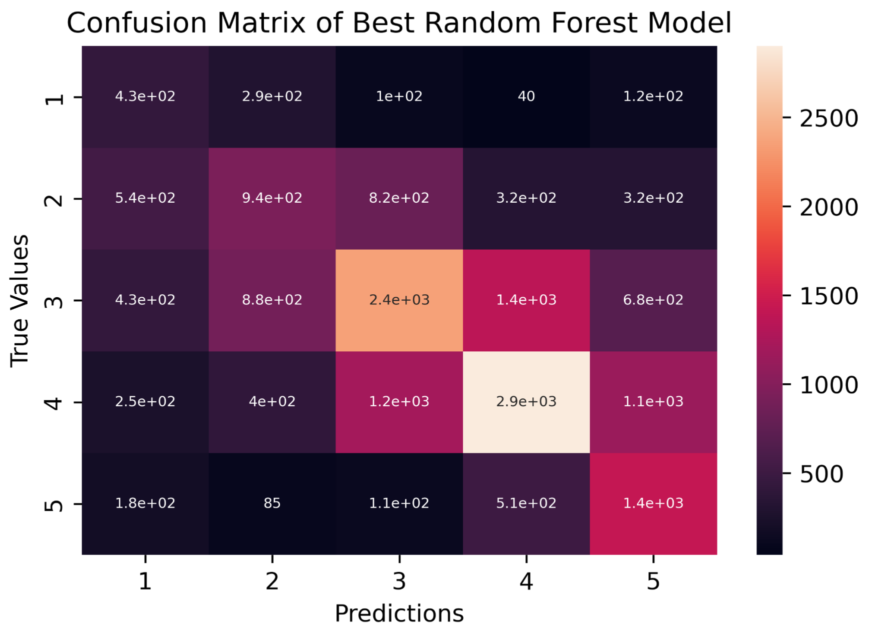


Fig. 5. More populous classes were correctly categorized by the model more frequently than classes with less observations.

## Feature Importance

Two different random forest classification models were trained on subsets of the data in order to determine global and local feature importances. Global feature importance was calculated in three different manners: using the built in scikit-learn Gini feature importances from the random forest classifier, permutation importance, and shapley additive explanations (SHAP).

[7] Ibid.

Each method of calculating global feature importances resulted in the same top feature: review count. The remaining most important features using Gini importance were the seven features specifying the number of hours the business was open each day and the ordinal feature price range. For features that take many unique values, Gini importance can be misleading and difficult to interpret.[8] Contrastingly, global feature importances calculated using permutation are highly interpretable. A predictor is removed from the model and the average decrease in accuracy is calculated. Features with a higher mean accuracy decrease are thus deemed more important.

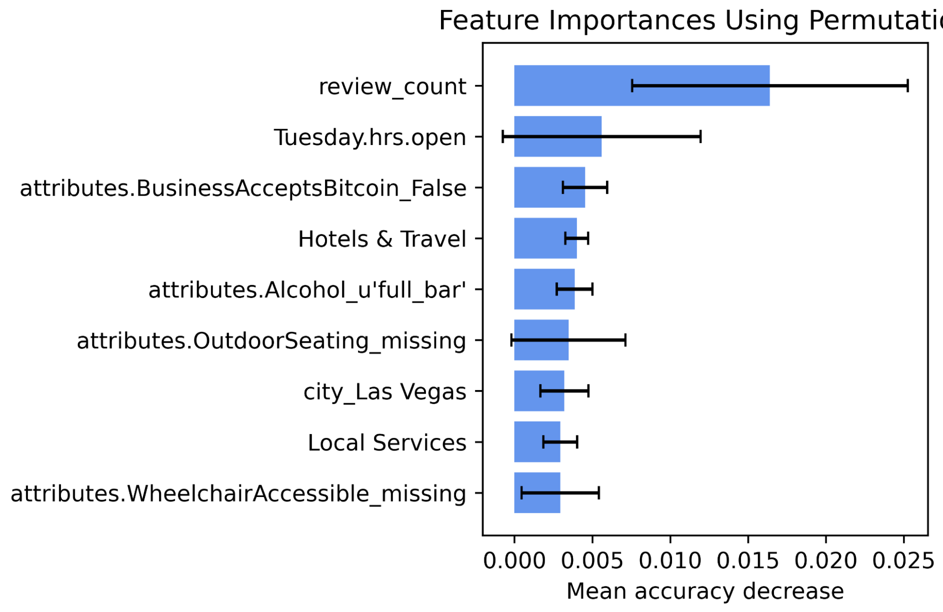
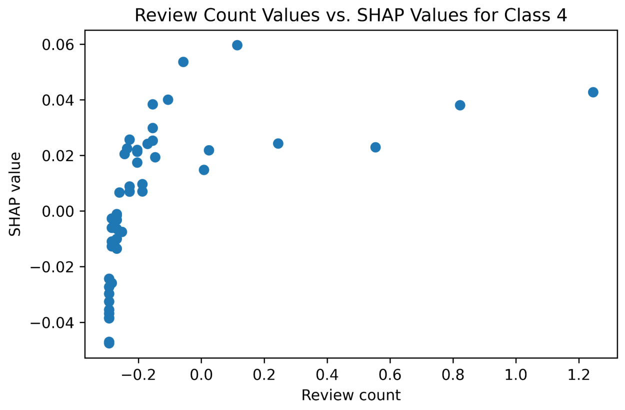


Fig. 6. Review count and the number of hours a business is open on Tuesday were the two most important features according to permutation importance.

The two most globally important features calculated using SHAP were review count and price range, which is not surprising. Aside from star rating, these features are also some of the first attributes a user typically looks for when viewing a business on any food-reviewing application.

Local feature importance was visualized with SHAP values by specifying both a particular feature and specific class. The model was more likely to place businesses with a high review count in class 4 than businesses with a low review count. Intuitively, this makes sense since highly-reviewed businesses are often popular because they receive a great deal of positive reviews. Interestingly, for class 5, businesses with a high review count had negative SHAP values, which implied that the model was likely to *not* place these businesses in class 5. This could be due to the viral nature in which some businesses receive a large number of negative reviews at once because of a bad customer experience that may have been posted online. However, this behavior could also be attributed to the uncertainty in splitting and the fact that only the first fifty observations in the test set are displayed. For example, fast food restaurants often have many reviews but a low star rating, so if most of the businesses displayed were fast food restaurants, this behavior of the data would not be surprising.

[8] *Random Forest Classifier Feature Importances*, Scikit-learn.



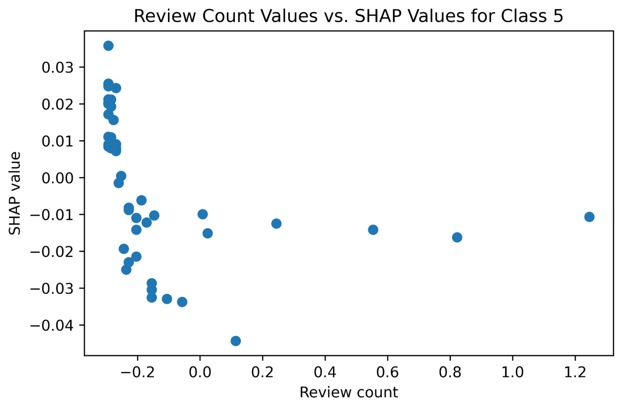


Fig. 7/8. SHAP values for the first 50 observations in the test set by review count. Higher SHAP values indicate that the model is more likely to categorize an observation into the class specified.

The least important features were variables transformed with the one hot encoder or multi label binarizer that did not appear often in the data. For instance, the city Sunnyslope only appears one time in the dataset. This feature and similarly infrequently-appearing ones were some of the least important features.

## Outlook

A classification problem with five different classes is very difficult to solve. If the problem were changed to instead classify businesses as high or low rated, the model would have a higher evaluation test score. In addition, while the machine learning algorithms implemented are well-suited for large datasets, a gradient boosting method such as XGBoost could have increased the predictive power of the model even more so than a random forest classifier.

One of the most significant limitations of this project is the nature of the Yelp business dataset. The star rating of a business often has little to do with business attributes, which make up the majority of predictive features. In order to achieve a higher test score, additional data from other sources such as Yelp reviews could be supplied to the model. For example, if a simple count of "positively-associated" and "negatively-associated" words from business reviews were provided, the model would likely classify with a much higher accuracy.

## References

1. "Frequently Asked Questions." *Yelp Dataset Documentation*, Yelp.
2. Clark, S., Artem, A., & Chambers, B. *Dataset Examples - Yelp*, Github. November 7, 2014.
3. Li, Y., Liu, Y., Chiou, R., & Kalipatnapu, P. *Prediction of Useful Reviews on Yelp Dataset.*
4. Asghar, N. *Yelp Dataset Challenge: Review Rating Prediction*, University of Waterloo. May 17, 2016.
5. *Multiclass and multioutput algorithms*, Scikit-learn.
6. Lee, Aaron. *Choosing a Baseline Accuracy for a Classification Model*, Towards Data Science. May 7, 2021.
7. *Random Forest Classifier Feature Importances*, Scikit-learn.
8. Xu, Ji. *Sklearn ColumnTransformer with MultiLabel Binarizer*, StackOverflow.