Jason Tay
CS 249

# Final Project Report: Yelp Rating Prediction

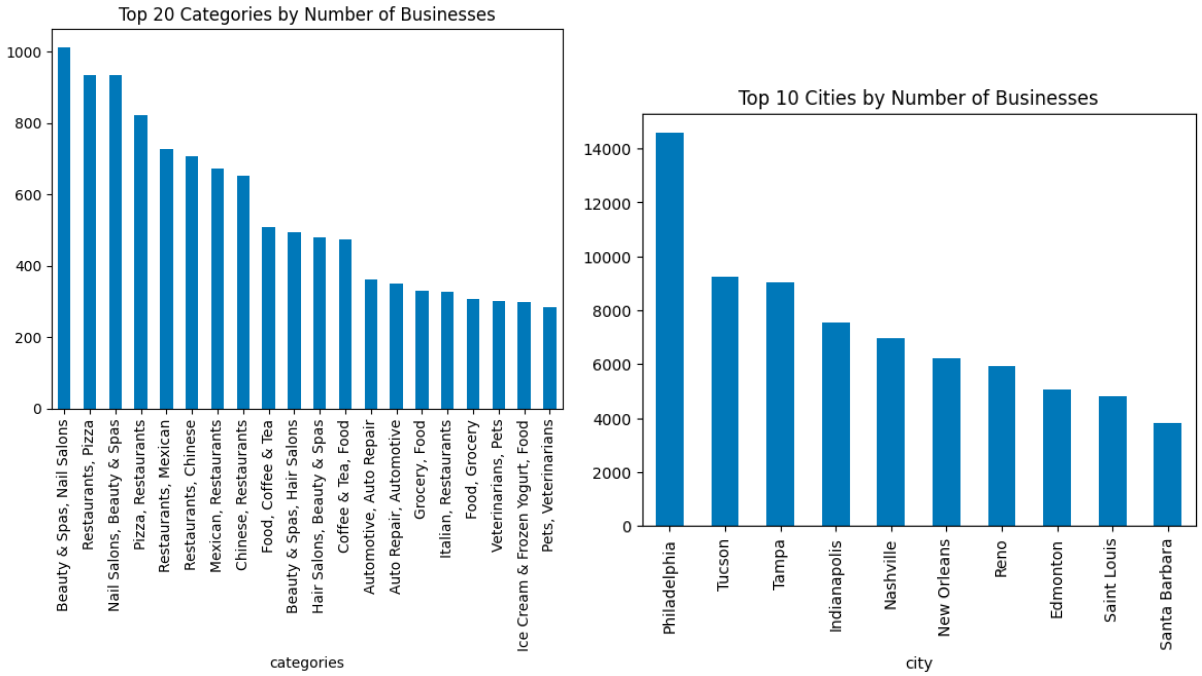
The importance of predicting restaurant ratings lies in its potential to empower restaurant owners, particularly small businesses, with actionable insights to thrive in a competitive market. As the owner of a local curry and ramen eatery in Hawaii, my father's experience inspired me to embark on this project. Restaurants serve as vital community hubs, fostering social interactions and culinary experiences. However, the complexities of running a successful restaurant, from managing operating hours to curating menus, can be overwhelming. By leveraging data analytics and predictive modeling, restaurant owners can gain valuable insights into the factors influencing their ratings. This knowledge enables them to make informed decisions to enhance their offerings and improve customer satisfaction. Ultimately, accurate rating prediction models contribute to the long-term success and sustainability of restaurants by identifying areas for improvement and optimizing operations.

This final project aimed to construct a predictive model for restaurant ratings using the extensive Yelp dataset. Initially conceived as a classifier, the project pivoted towards regression modeling to predict relative rating differences more accurately. This decision was driven by the recognition that restaurant ratings often vary on a continuous scale, making regression a more suitable approach. By predicting these nuanced differences, the model can offer actionable insights to restaurant owners, aiding them in fine-tuning their operations to enhance customer satisfaction and overall performance. The Yelp dataset, comprising restaurant reviews, ratings, and attributes from diverse locations, served as the primary dataset for analysis. With over 8GB of restaurant data, the Yelp dataset provided a wealth of information for modeling, albeit requiring extensive data cleaning and imputation tasks before training the model.

The Yelp dataset consists of various files with the prefix `yelp_academic_dataset`: `business.json`, `checkin.json`, `review.json`, `tip.json`, and `user.json`. For the purposes of my project, I will focus the analysis on first breaking up the features and data in `business.json`, which is a JSON object containing 150,346 businesses. To streamline the analysis, emphasis will be placed on parsing the features and data within `business.json`, housing information on 150,346 businesses. Each entry consists of a unique id which maps to the reviews in `review.json` and includes various identifiers for the establishment's location, the star rating, and a list of attributes. Here's the list of values for each business entry:

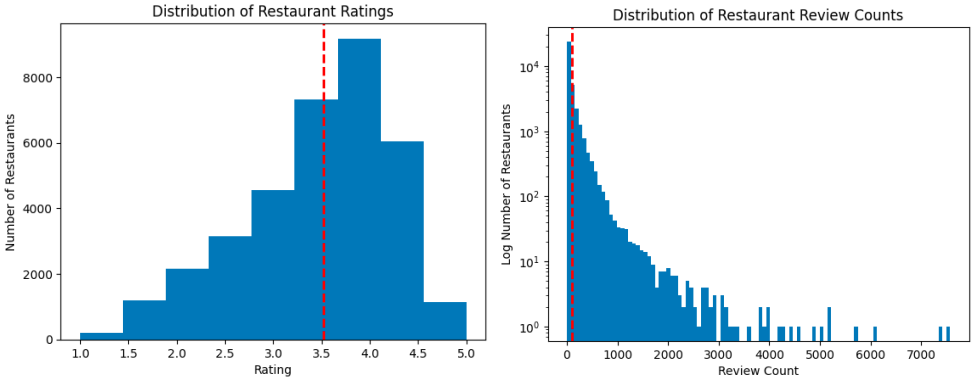
```
business_id      object
name             object
address          object
city             object
state            object
postal_code      object
latitude        float64
longitude       float64
stars           float64
review_count      int64
is_open           int64
attributes       object
categories       object
hours            object
```

In the above list of features for businesses, the target feature is the stars, which represents the ratings (in increments of 0.5) out of 5. Before proceeding with data preprocessing or imputation, I first decided to explore the dataset more. In the below generated graphs, we can see the most popular types of businesses included in the dataset and where the data comes from:




From the graphs above, we can confirm that data in the Yelp dataset comes from metropolitan areas across the US and the diverse types of cuisines for their available restaurants (i.e. pizza, Chinese, Mexican, Italian, etc.). This suggests that the dataset should be great for developing a predictive model for ratings since the data is diverse in cuisine and location to reduce regional biases. For the rest of this project, all further analysis and feature engineering will be focused on the business data in the Yelp dataset.

As a prelimary data cleaning step, I first remove all the data not relevant to restaurants by dropping all rows without the "restaurant" value in `categories`. I also make sure the establishment is still open based on the `is_open` feature since it would not make as much sense to evaluate the ratings through restaurants that no longer operate. There were 440 restaurants with no attributes (a list of tags with various properties about a restaurant) and 3370 with no listed hours. After eliminating these restaurants, we are left with 31,357, which is far less than the original 150,346 business as many are not restaurants. Here are the graphs for the distribution of restaurant ratings and reviews:




The average number of restaurant reviews is 104.143 per restaurant. A log graph is used since majority of restaurants have <1000 reviews.

## Data Preprocessing and Imputation

Data preprocessing and imputation presented the greatest challenge in this project, primarily due to the complex JSON representation of the Yelp dataset's businesses. While this representation does fairly represent the diversity and variance in restaurants, the data preprocessing required careful work to extract usable features. The dataset featured deeply nested objects with inconsistent properties, requiring meticulous cleaning and preparation for training. Notably, the `attributes` feature contains several inconsistent and varied values from establishment to establishment based on what is reported and . These include values such as WiFi avaliablity, whether alcohol is served, general assessments of the ambiance, and whether the restaurant is good for children. Some of these features are further nested like ambiance for example:

```
"Ambience":"{'romantic': False, 'intimate': False, 'touristy': False, 'hipster': False, 'divey': False,
'classy': False, 'trendy': False, 'upscale': False, 'casual': False}"
```

This feature contains many boolean attributes inside the nested object, which represent various aspects of the perceived ambiance of a resturant's interior.

Here are the top unique attributes (number in front indicates the number of restaurants):

```
270 {'RestaurantsDelivery': 'True', 'RestaurantsTakeOut': True'}
69 {'RestaurantsDelivery': 'None', 'RestaurantsTakeOut': 'None}
36 {'RestaurantsDelivery': 'True', 'OutdoorSeating': 'True', 'RestaurantsTakeOut': 'True'}
36 {'BusinessAcceptsCreditCards': 'True', 'RestaurantsDelivery': 'True', 'RestaurantsTakeOut': 'True'}
32 {'RestaurantsTakeOut': 'True'}
```

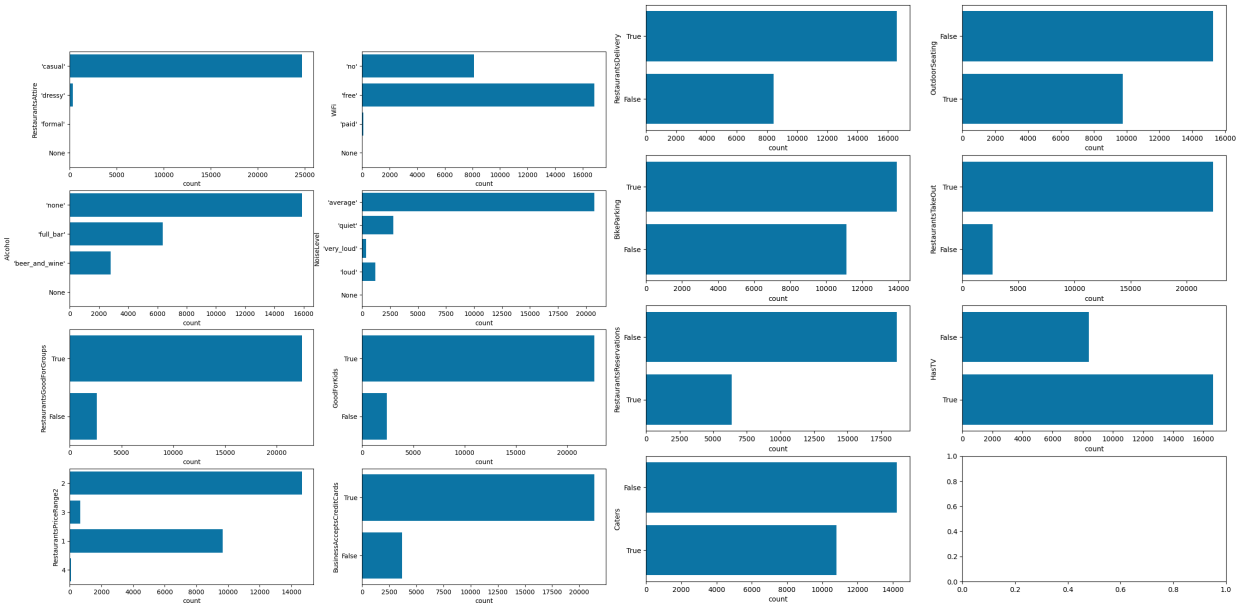
Thus, many restaurants have unique attributes with many values being missing or not represented. I manually removed irrelevant features to restaurant businesses such as `HairSpecializesIn` and `AcceptsInsurance`. These are attributes in the Yelp dataset that are relevant to non-restaurant business such as hair salons and medical treatment centers. In addition, attribute features with over 30% missing values were dropped, these are the remaining features (with the corresponding number of missing values):

```
HasTV: 4457, RestaurantsTakeOut: 1055, RestaurantsDelivery: 1389, Caters: 6979, OutdoorSeating: 4079, Alcohol:
5904, BusinessAcceptsCreditCards: 3097, WiFi: 5828, BikeParking: 6169, Ambience: 4766,
RestaurantsGoodForGroups: 5735, RestaurantsPriceRange2: 3963, GoodForKids: 6034, RestaurantsAttire: 7198,
GoodForMeal: 8759, NoiseLevel: 8556, RestaurantsReservations: 4430, BusinessParking: 2844, Friday: 108,
Saturday: 639, Sunday: 3397, Monday: 2826, Tuesday: 1710, Wednesday: 621, Thursday: 231
```

For the remaining features, I have to make decisions for each of them to decide how to impute the data. The first set of features are those where I determined that missing the label is equivalent to lacking the feature and features in this set were imputed by replacing missing values with "`False`". For example, it makes most sense to label restaurants that are missing the `RestaurantsDelivery` feature with "False" since not providing that information makes the most sense for restaurants that do not offer that feature. This set includes the following features: `RestaurantsDelivery, OutdoorSeating, BikeParking, RestaurantsTakeOut, BusinessAcceptsCreditCards, RestaurantsReservations, Caters`. Some of the boolean features for this set contained "`None`" values, which I replaced with "`False`" so that the features can be later transformed to booleans.

The next set of features were more subjective assessments of the restaurant and thus difficult to simply assess by setting the feature to "`False`". This set included: `RestaurantsGoodForGroups, GoodForKids, RestaurantsPriceRange2`. I imputed these values by replacing any missing values with the mode of the feature. For the restaurant hours (e.g. `Monday`), I filled any missing values with the notation "`0:0-0:0`", which indicates that the restaurant is not opened that day. This notation is present in the dataset when a restaurant's Yelp page marks it as "closed" on a given day of the week. I made the decision to impute missing values as closed because if restaurant does not report its opening or closing hours for a given day, it can be naturally assumed that the restaurant is not opened. The categoric features: `RestaurantsAttire, WiFi, Alcohol, NoiseLevel, RestaurantsPriceRange2` required more thorough preprocessing as the labels inconsistently had a "u" prefix (e.g. `u'full_bar'` for `Alcohol`),

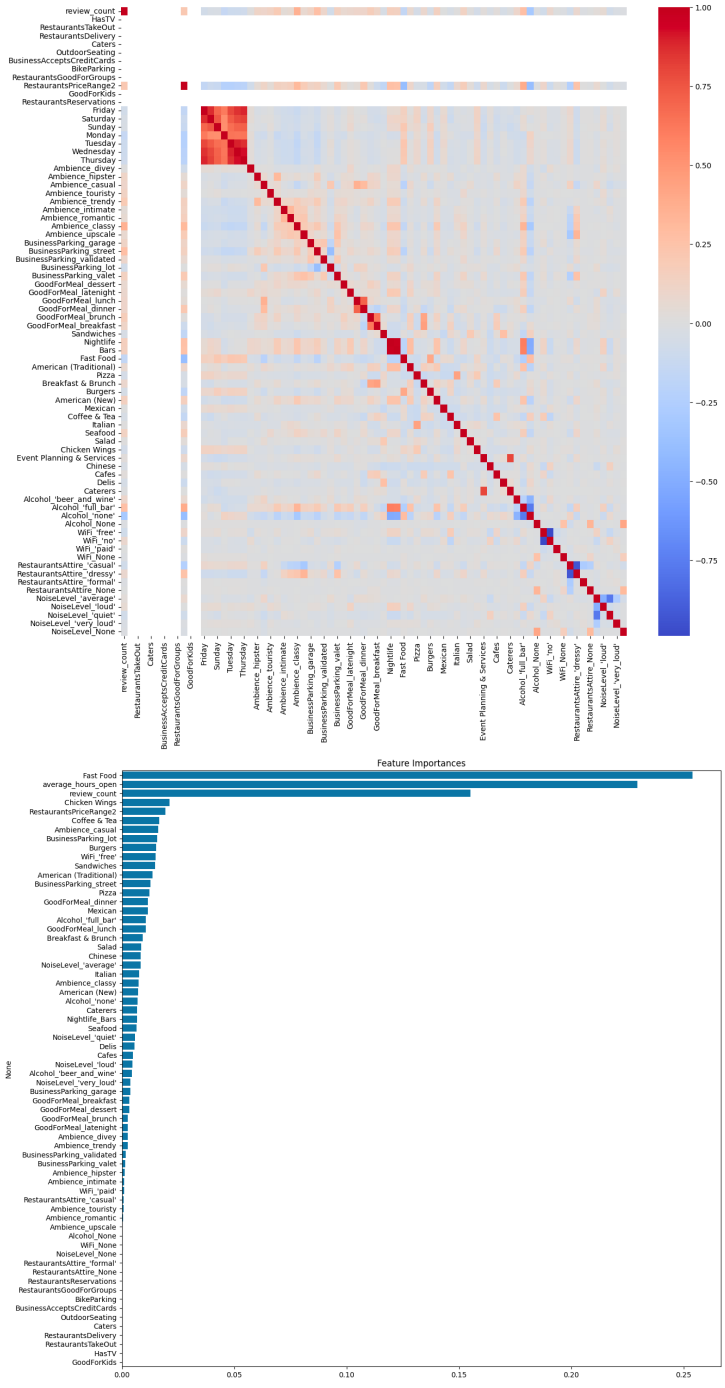
which results in inconsistent string representations for the same feature values. To resolve this, I processed all categoric labels and manually removed the prefix. All missing values were replaced with the mode for the feature. Here's what the final output for all the attribute features looks like after preprocessing and imputation:



The nested JSON object properties required further preprocessing. This includes `Ambience, BusinessParking, GoodForMeal`. These features were objects with several boolean labels. For example, the `Ambience` for Golden Chopstick Chinese Restaurant is: `"{'touristy': False, 'hipster': False, 'romantic': False, 'divey': None, 'intimate': False, 'trendy': False, 'upscale': False, 'classy': False, 'casual': None}"`. To handle these features, I had to unpack them and construct new features with a prefix (e.g. `Ambience_`) for each of the boolean labels. I also replaced all missing or None values with the mode in this case as it would not be fair to label restaurants without these values as not having these properties. Instead, by using the mode, I could preserve most of the information contained in these labels. For `categories,` there were over 629 different unique labels. I removed the "Restaurant" and "Food" labels since I had already filtered for "Restaurants". Then I simply counted number of categories to find the top 20 to keep for training. This was the final ouput of this process:

```
[('Sandwiches', 4556), ('Nightlife', 4367), ('Bars', 4217), ('Fast Food', 4200), ('American (Traditional)',
4161), ('Pizza', 3473), ('Breakfast & Brunch', 3378), ('Burgers', 3248), ('American (New)', 2738), ('Mexican',
2365), ('Coffee & Tea', 2306), ('Italian', 2234), ('Seafood', 1857), ('Salad', 1746), ('Chicken Wings', 1729),
('Event Planning & Services', 1705), ('Chinese', 1522), ('Cafes', 1302), ('Delis', 1256), ('Caterers', 1131)]
```

To finish the data processing, all categoric variables were One-Hot encoded, I extracted the time open from days of the week, and Boolean attributes were accordingly set. I generated a diagram for the correlation between all features and the feature importance (attached below) by fitting the training data with a RandomForestRegressor.




These features: `{('Wednesday', 'Tuesday'), ('Caterers', 'Event Planning & Services'), ('Bars', 'Nightlife'), ('Saturday', 'Friday'), ("RestaurantsAttire_'dressy'", "RestaurantsAttire_'casual'"), ('Thursday', 'Saturday'), ('Thursday', 'Friday'), ('Wednesday', 'Friday'), ("WiFi_'no'", "WiFi_'free'"), ('Thursday', 'Wednesday'), ('Thursday', 'Tuesday')}` were the most highly correlated (greater than 0.8). I dropped `'Event Planning & Services'`, `'Nightlife'`, and `"WiFi_'no'"` since they presented redundant information from the other features. To avoid collinearity with the hours opened per day, I engineered a new feature representing the average hours open for each restaurant. Finally, I dropped the bottom 15 least important features based on the RandomForestRegressor to lower the dimensionality of the data and keep more relevant features.

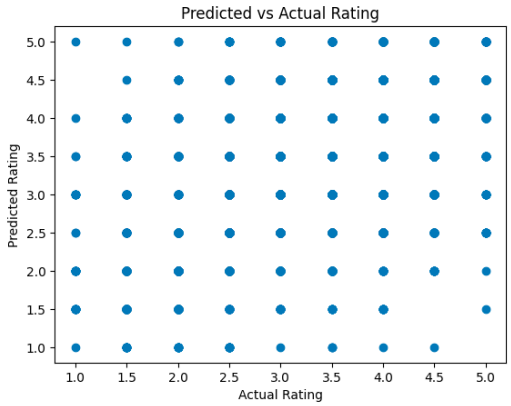
The final training data now has the following 51 features:

```
'review_count', 'RestaurantsPriceRange2', 'Ambience_divey','Ambience_hipster', 'Ambience_casual',
'Ambience_touristy','Ambience_trendy', 'Ambience_intimate',
'Ambience_romantic','Ambience_classy', 'Ambience_upscale',
'BusinessParking_garage','BusinessParking_street',
'BusinessParking_validated','BusinessParking_lot', 'BusinessParking_valet',
'GoodForMeal_dessert','GoodForMeal_latenight', 'GoodForMeal_lunch',
'GoodForMeal_dinner','GoodForMeal_brunch', 'GoodForMeal_breakfast', 'Sandwiches','Fast Food',
'American (Traditional)', 'Pizza', 'Breakfast & Brunch','Burgers', 'American (New)', 'Mexican',
'Coffee & Tea', 'Italian','Seafood', 'Salad', 'Chicken Wings', 'Chinese', 'Cafes',
'Delis','Caterers', 'Alcohol_'beer_and_wine'', 'Alcohol_'full_bar'','Alcohol_'none'',
'WiFi_'free'', 'WiFi_'paid'','RestaurantsAttire_'casual'',
'NoiseLevel_'average'','NoiseLevel_'loud'', 'NoiseLevel_'quiet'',
'NoiseLevel_'very_loud'','average_hours_open', 'Nightlife_Bars'
```

I applied an 80-20 train-test split and ensured that all imputation processes were applied to the testing set separately to ensure no data leakage. This is notable since many of the imputations use the mode to replace missing attributes. All the following models used this processed, one hot-encoded data.
The test and train dataset have 6305 and 25052 entries, respectively.

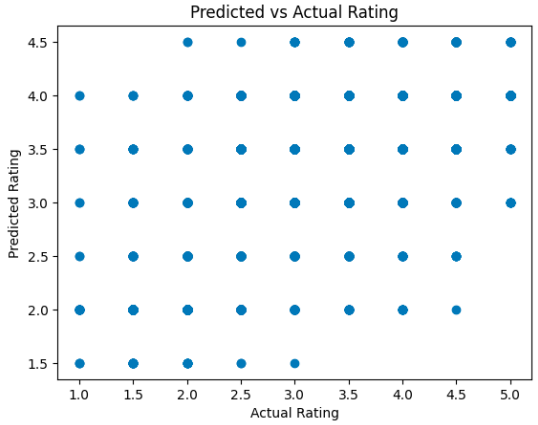
## Modeling and Results

For this project, I opted to utilize decision tree models due to their high interpretability, which aligns well with the goal of gaining insights into the factors contributing to a restaurant's success in terms of their Yelp ratings. Decision tree models offer transparency in their decision-making process, allowing for the identification of key features and their respective importance in predicting restaurant ratings. This interpretability is crucial for understanding the nuanced relationships between various attributes, such as operating hours, food categories, and customer reviews, and how they influence ratings. Additionally, decision trees are robust to nonlinear relationships and can handle both numerical and categorical data effectively. By leveraging decision tree models, I aimed to develop a predictive model that not only provides accurate ratings but also offers actionable insights for restaurant owners to optimize their operations and enhance customer satisfaction. All models were trained and then the predictions were generated by rounding the output of the regression models to the nearest 0.5, since this is how the Yelp stars data is presented (rounded values based on all the reviews for a restaurant). The baseline decision tree model without any hyperparameter tuning had a mean squared error (MSE) of 0.69952.

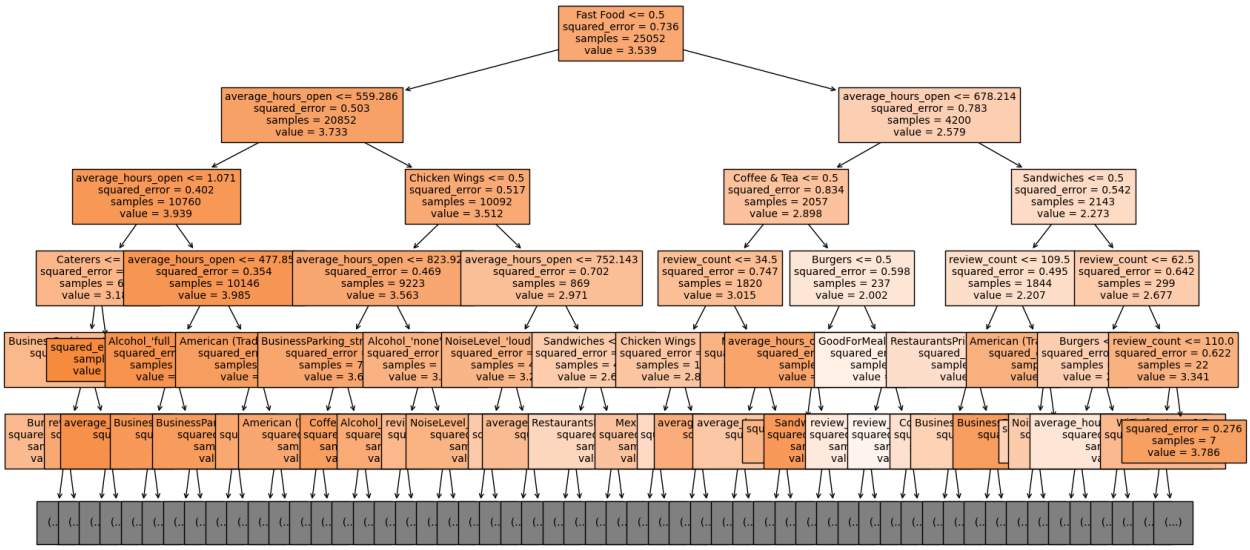



From the graph above, we can see that the results are not great and appear to be more randomly distributed. We need to do more hyperparameter tuning, ensembling, and more advanced techniques to achieve better results. Using GridSearchCV, we can use 5-fold cross validation with hyperparameter tuning for the depth of the tree, minimum samples split, and minimum sample leaf. The best parameters I

found were: `{'max_depth': 9, 'min_samples_leaf': 7, 'min_samples_split': 9}`. Using these refined parameters with cross-validation, I was able to significantly reduce the MSE to 0.42387 and the resulting predictions were somewhat improved with no low rating businesses below 2.0 having high predicted ratings and likewise for the higher ratings with no low predicted ratings.




Here's the actual decision tree with the tuned hyperparameters:

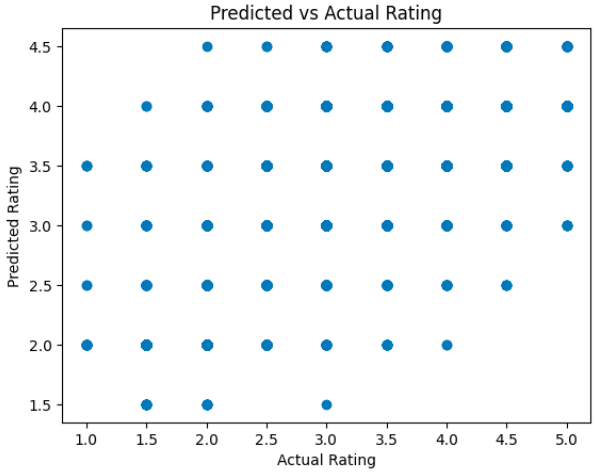



To further improve the results of the model, I decided to employ ensembling techniques with RandomForest and XGBoost. Random forests are powerful ensemble methods that build multiple decision trees during training and output the average prediction of the individual trees. They are effective in handling high-dimensional datasets with complex interactions between features, as they reduce overfitting and provide robust predictions. Additionally, random forests can handle missing values and maintain good accuracy even with many trees. On the other hand, XGBoost, an implementation of gradient boosting, offers enhanced performance compared to traditional gradient boosting methods. XGBoost iteratively builds decision trees in a sequential manner, with each tree attempting to correct the errors of the previous ones. It employs regularization techniques to prevent overfitting and is known for its efficiency and scalability. XGBoost's ability to capture complex nonlinear relationships and handle large datasets makes it particularly suitable for this project, where we aim to predict restaurant ratings based on diverse and potentially high-dimensional features. By leveraging these advanced ensemble

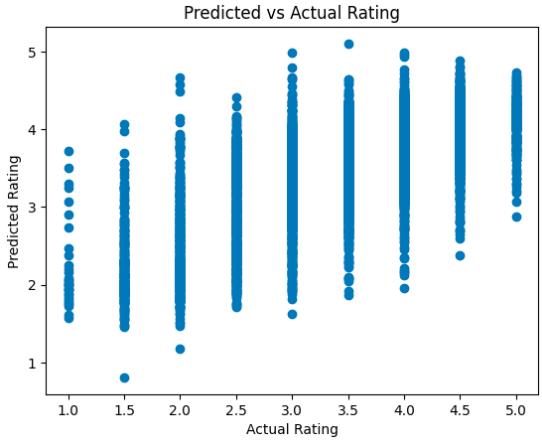
techniques, we can achieve higher predictive accuracy and robustness in our model, ultimately leading to more accurate ratings predictions for restaurants [1].

Despite the decreased interpretability associated with ensemble methods like RandomForest and XGBoost, we can still extract valuable insights regarding feature importance. Feature importance refers to the extent to which each feature contributes to the model's predictive performance. In the context of decision trees, feature importance is often determined by analyzing how much each feature reduces the impurity or error in the model when making decisions. For example, in RandomForest, features that lead to greater reductions in impurity across the ensemble of trees are considered more important. Similarly, in XGBoost, features that are frequently used in decision splits or contribute more significantly to reducing the loss function during training are deemed more important. By examining the relative importance of features, we can gain insights into which factors have the most significant influence on the predicted restaurant ratings. This understanding allows us to prioritize and focus on the most impactful features when making decisions or optimizations related to restaurant operations and customer satisfaction.

With a baseline RandomForest (RF) model without an hyperparameter tuning, I could lower the MSE to 0.38902, which further reduces the MSE below even the tuned decision tree. I was only able to slightly reduce the MSE further to 0.3889 with GridSearchCV (5-fold cross validation). Doing some analysis on this model, I found that the mean absolute error (MAE) is 0.4585249801744647. This gives a clearer picture of how the model performs relative to the ratings. On average, the predicted ratings are slightly below half a rating away from the actual rating. The $R^2$ of the model 0.46783722425237817.




The XGBoost model achieved the best results with an MSE of 0.360519, MAE of 0.46692 and improved $R^2$ at 0.5066708069243135. This improved performance is reflected visually below by the improved predicted vs actual ratings graph. While there still is a wide disparity between actual ratings and their predictions, the model performs much better with distinguishing higher and lower rated restaurants. The decision tree diagram for this model is located in the appendix [A].




Listing out the feature importance, we find that the following features are most important for the predictions with the best performing model:

| | |
|---|---|
| Fast Food | 0.442395 |
| Chicken Wings | 0.042502 |
| Coffee & Tea | 0.038004 |
| average_hours_open | 0.026253 |
| NoiseLevel_'very_loud' | 0.024578 |
| Alcohol_'full_bar' | 0.023862 |
| Burgers | 0.022308 |
| BusinessParking_street | 0.022032 |
| Pizza | 0.021309 |
| Chinese | 0.020868 |
| American (Traditional) | 0.019364 |
| Mexican | 0.016535 |
| BusinessParking_validated | 0.014318 |
| Alcohol_'none' | 0.013984 |
| Ambience_classy | 0.012167 |
| Salad | 0.011632 |
| NoiseLevel_'loud' | 0.011273 |
| Caterers | 0.011239 |
| Sandwiches | 0.010771 |
| Cafes | 0.009458 |
| RestaurantsPriceRange2 | 0.009396 |
| Ambience_trendy | 0.008300 |
| Italian | 0.008084 |
| NoiseLevel_'quiet' | 0.007999 |
| Ambience_divey | 0.007730 |
| Nightlife_Bars | 0.007650 |
| GoodForMeal_breakfast | 0.007647 |
| GoodForMeal_dinner | 0.007436 |
| review_count | 0.007422 |
| Breakfast & Brunch | 0.007399 |

## Conclusion

Despite achieving enhanced predictive accuracy and robustness with RandomForest and XGBoost, it's essential to acknowledge the inherent challenges in predicting restaurant ratings accurately. While both models reduced the mean squared error (MSE) significantly, yielding values of 0.38902 for RandomForest and 0.360519 for XGBoost, there remains a notable variance between the actual and predicted ratings. Moreover, the R-squared ($R^2$) value, a measure of how well the model explains the variability in the data, was relatively low, < 0.7 even in the best-performing model. However, it is important to recognize that predicting human behavior, such as restaurant ratings, is inherently complex, and achieving a high $R^2$can be challenging. In fact, $R^2$values below 0.5 are common in models based on human behavior, especially in social sciences [2]. Despite these limitations, the models still provide

valuable insights into the trends behind the ratings. By identifying influential factors and their relative importance, restaurant owners and stakeholders can gain actionable insights to optimize their operations and enhance customer satisfaction. Therefore, while the models may not perfectly predict individual ratings, they offer valuable guidance for improving overall restaurant performance or general trends in Yelp restaurant reviews.

Although both advanced models, RandomForest and XGBoost, resulted in lower interpretability, but still allowed for the extraction of feature importance, shedding light on the key factors influencing restaurant ratings on Yelp. The feature importance analysis revealed that attributes such as "Fast Food," "Chicken Wings," and "Coffee & Tea" held significant sway over ratings predictions, with respective importances of 0.442395, 0.042502, and 0.038004. Moreover, operational factors such as "`average_hours_open`" and "`BusinessParking_street`" also emerged as noteworthy contributors, underscoring the importance of accessibility and convenience to customers. Furthermore, aspects of the dining experience, such as noise levels ("`NoiseLevel_'very_loud'`") and available amenities ("`Alcohol_'full_bar'`"), demonstrated observable impacts on ratings. By leveraging these insights, restaurant owners can strategically optimize their operations and offerings to enhance customer satisfaction and drive higher ratings. The integration of advanced ensemble techniques not only facilitated accurate ratings predictions but also empowered restaurant stakeholders with actionable insights, thereby fostering informed decision-making and competitive advantage in the dynamic restaurant industry landscape.

Additionally, moving forward, one promising avenue for enhancing our understanding of restaurant ratings on Yelp involves leveraging semantic analysis techniques on the review text data. By analyzing the sentiment, tone, and topics discussed in the reviews, we can gain a more holistic understanding of the factors driving customer perceptions and ratings. Incorporating sentiment analysis can provide valuable insights into the emotional responses and experiences expressed by reviewers, allowing us to identify common themes and sentiments associated with positive and negative ratings. Moreover, exploring word embeddings techniques can help capture the semantic relationships between words in the review text, enabling us to extract nuanced contextual information and better understand the underlying reasons behind specific ratings. Furthermore, it's crucial to address the challenges posed by the dataset's structure, particularly the need for extensive attribute imputation due to missing values. While decision tree models provided a baseline for predictive modeling, the complex JSON representation of the Yelp dataset's businesses introduced significant preprocessing and cleaning challenges. These challenges highlight the importance of exploring more sophisticated data preprocessing techniques and feature engineering strategies to handle missing data effectively. Additionally, incorporating domain-specific knowledge and external datasets could help enrich the feature space and improve model performance.

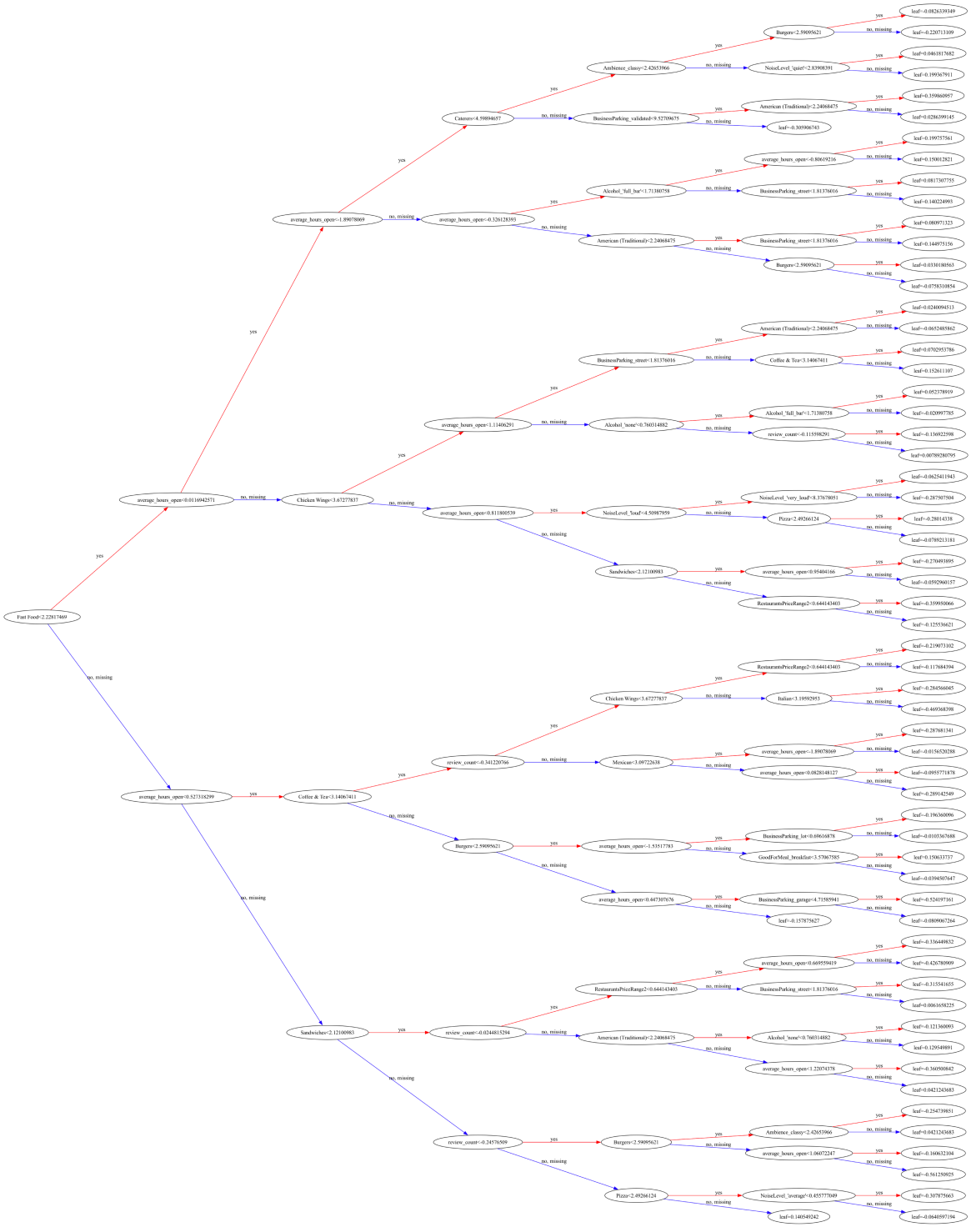
Building upon this baseline research into performant decision trees for the Yelp database, future efforts can focus on developing more advanced models that integrate sentiment analysis and word embeddings within a multinomial framework. By combining structured features with textual data-derived features, such as sentiment scores and word embeddings, we can construct a comprehensive predictive model that captures both quantitative and qualitative aspects of restaurant ratings. Unfortunately, due to time constraints, semantic analysis of the reviews in the dataset was outside the scope of this project. An integrated approach using the Yelp business features and reviews not only enhances the predictive accuracy of the model but also provides deeper insights into the nuanced factors shaping customer perceptions and preferences. The dataset also contains a large number of features, so further work can be done on more feature engineering to develop more meaningful representations of business features. Lastly, if interpretability is of lesser importance, other types of models such as SVM and dimensionality reduction techniques with PCA can be employed to further improve the performance of the model.

**References:**

**[1]** https://www.spiceworks.com/tech/artificial-intelligence/articles/xgboost-vs-random-forest-vs-gradient-boosting/

**[2]** https://towardsdatascience.com/an-ode-to-r-squared-804d8d0ed22c

# Appendix:
# [A]


Fast Food<2.22617469
yes
no, missing
average_hours_open<0.0116942571
yes
no, missing
average_hours_open<-1.89078069
yes
no, missing
Caters<4.59894657
yes
no, missing
Ambience_classy<2.42653966
yes
no, missing
Burgers<2.59095621
yes
no, missing
leaf=-0.0826339349
leaf=-0.220713109
NoiseLevel_'quiet'<2.83908391
yes
no, missing
leaf=0.0461817682
leaf=-0.199367911
BusinessParking_validated<9.52709675
yes
no, missing
American (Traditional)<2.24068475
yes
no, missing
leaf=0.359860957
leaf=0.0286399145
leaf=-0.305906743
average_hours_open<-0.326128393
yes
no, missing
Alcohol_'full_bar'<1.71380758
yes
no, missing
average_hours_open<-0.80619216
yes
no, missing
leaf=-0.199757361
leaf=0.150012821
BusinessParking_street<1.81376016
yes
no, missing
leaf=0.0817307755
leaf=-0.140224993
American (Traditional)<2.24068475
yes
no, missing
BusinessParking_street<1.81376016
yes
no, missing
leaf=0.0809713225
leaf=-0.144975156
Burgers<2.59095621
yes
no, missing
leaf=0.0330180563
leaf=-0.0758310854
Chicken Wings<3.67277837
yes
no, missing
average_hours_open<1.11406291
yes
no, missing
BusinessParking_street<1.81376016
yes
no, missing
American (Traditional)<2.24068475
yes
no, missing
leaf=0.0240094513
leaf=-0.0652485862
Coffee & Tea<3.14067411
yes
no, missing
leaf=0.0702953786
leaf=-0.152611107
Alcohol_'none'<0.760314882
yes
no, missing
Alcohol_'full_bar'<1.71380758
yes
no, missing
leaf=0.0523788919
leaf=-0.020997785
review_count<-0.115598291
yes
no, missing
leaf=-0.136922598
leaf=0.00789280795
average_hours_open<0.811800539
yes
no, missing
NoiseLevel_'loud'<4.50987959
yes
no, missing
NoiseLevel_'very_loud'<8.37670851
yes
no, missing
leaf=-0.0625411943
leaf=-0.287507504
Pizza<2.49266124
yes
no, missing
leaf=-0.28014338
leaf=-0.0789213181
Sandwiches<2.12100983
yes
no, missing
average_hours_open<0.95404166
yes
no, missing
leaf=-0.270493895
leaf=-0.0592960157
RestaurantsPriceRange2<0.644143403
yes
no, missing
leaf=-0.359950066
leaf=-0.125536621
average_hours_open<0.527318299
yes
no, missing
Coffee & Tea<3.14067411
yes
no, missing
review_count<-0.341220766
yes
no, missing
Chicken Wings<3.67277837
yes
no, missing
RestaurantsPriceRange2<0.644143403
yes
no, missing
leaf=-0.219073102
leaf=-0.117684394
Italian<3.19592953
yes
no, missing
leaf=-0.284566045
leaf=-0.469368398
Mexican<3.09722638
yes
no, missing
average_hours_open<-1.89078069
yes
no, missing
leaf=-0.287681341
leaf=-0.0156520288
average_hours_open<0.0828148127
yes
no, missing
leaf=-0.0955771878
leaf=-0.289142549
Burgers<2.59095621
yes
no, missing
average_hours_open<-1.53517783
yes
no, missing
BusinessParking_lot<0.69616878
yes
no, missing
leaf=-0.196360096
leaf=-0.0103367688
GoodForMeal_breakfast<3.57067585
yes
no, missing
leaf=-0.150033737
leaf=-0.0394507647
average_hours_open<0.447307676
yes
no, missing
BusinessParking_garage<4.71585941
yes
no, missing
leaf=-0.524197161
leaf=-0.0809067264
leaf=-0.157875627
Sandwiches<2.12100983
yes
no, missing
review_count<-0.0244815294
yes
no, missing
RestaurantsPriceRange2<0.644143403
yes
no, missing
average_hours_open<0.669559419
yes
no, missing
leaf=-0.336449832
leaf=-0.426780909
BusinessParking_street<1.81376016
yes
no, missing
leaf=-0.315541655
leaf=-0.0061658225
American (Traditional)<2.24068475
yes
no, missing
Alcohol_'none'<0.760314882
yes
no, missing
leaf=-0.121360093
leaf=-0.129549891
average_hours_open<1.22074378
yes
no, missing
leaf=-0.360500842
leaf=-0.0421243683
review_count<-0.24576509
yes
no, missing
Burgers<2.59095621
yes
no, missing
Ambience_classy<2.42653966
yes
no, missing
leaf=-0.254739851
leaf=0.0421243683
average_hours_open<1.06072247
yes
no, missing
leaf=-0.160632104
leaf=-0.561250925
Pizza<2.49266124
yes
no, missing
NoiseLevel_'average'<0.455777049
yes
no, missing
leaf=-0.307875663
leaf=-0.0640597194
leaf=-0.140548242