Deep “Dish” Data Dive

Kevin Gilmore – M08331816

**First Reader:** Professor Peng Wang

**Second Reader:** Professor Liwei Chen

A slice of pizza sitting on top of a wooden cutting board

Description automatically generated

**Abstract**

This project contains three datasets in which I am addressing average rating, location, and pricing information of various pizza establishments around the US. The goal is to assist the average American consumer in making informed decisions about which pizza restaurants they would like to patronize. The data is first introduced, a data dictionary is created for the reader to further understand the data, the data is then cleaned, and then some initial exploratory analysis is done. After EA, some statistical modeling techniques were used to see the relationship and significance of different factors in relationship to price level and the owner of the barstool data (Dave). More mathematical numbers are looked at to break down the model summaries. A view of buying pizza at pizza establishments versus buying pizza at a place like a sandwich shop or bar/pub are also taken into consideration. I hope this project is informative, quantitative and enjoyable!

**Table of Contents**

1. **Introduction**
   1. **Data Introduction - Information about each dataset and all packages downloaded**
   2. **Data Dictionary –** Contains description of each variable
2. **Data Problem Descriptions:**
   1. **Jared and Barstool Data Issues**
   2. **Datafiniti**
3. **Solution Narrative with Exploratory Analysis**
4. **Solutions with Modeling and Mathematical Values**

**4.1. Linear Modeling**

**4.2. Non-Linear Modeling**

1. **Summary and Conclusions**
2. **Appendix**

**1.0 Introduction**

**1.1 - Data Introduction:** For my capstone project I am working on a pizza project. This project was found through R-Studios “Tidy Tuesday.” The trend of Tidy Tuesday is to provide a safe and supportive forum for people to practice both data wrangling and visualization, specifically the ‘tidyverse’ package. Most datasets are imperfect and require work to clean before. The “Pizza Party” analysis is based primarily on 3 datasets with different types of information/data about pizza restaurants in the US. The pizza data was originally compiled on September 30th, 2019. For those who are interested, there is a 5 minute podcast reviewing the top entries and different visualization techniques for Week 7’s project called “Pizza Party” - <https://www.tidytuesday.com/7> . The first dataset, “Barstool” contains multiple ratings. The ratings include critics, public voting, and the company Barstool. Additionally, information about pricing, location, and geolocation of the different pizza places is included. The dataset called “Jared” contains ratings restaurant names, review answers, the poll question, and total number of votes. The “Datafiniti” dataset contains restaurant names, addresses, city, country, province, latitude, longitude, restaurant category, max price, and minimum price. The “region” dataset contains state by region from the US Census Bureau. My approach was to divide the analysis to focus on two groups of consumers. First, those who are price conscious. Secondly, those who are interested in finding the pizza place near them with the highest ratings. With further analysis, GLM models were built to see the relationship of how provider rating correlates to other review scores. My model was then built and then a prediction was used on the testing dataset. Nonlinear methods of random forest and boosting for regression trees were examined as well. The regions dataset was created based on the definitions from the US Census Bureau - <https://en.wikipedia.org/wiki/File:Census_Regions_and_Division_of_the_United_States.svg>

Next, there are 9 packages which must be installed:

**Library(tidyverse)** – data manipulation and cleaning

**Library(knitr)** – data dictionary tables

**Library(kableExtra)** – table formatting

**Library(readxl)** – importing dataset descriptions

**Library(leaflet)** – interactive map & geolocation

**Library(broom)** – help create concise tables from R to Word

**Library(glmnet)** – lasso creation

**Library(gbm)** – generalized boosted regression models

**Library(randomForest)** – randomForest model creation

**1.2 – Data Dictionary:** Due to the nature of this project, I have created a data dictionary with tables to describe and explain each variable in each dataset. Having 4 datasets in a project can be confusing to readers without an explanation. I believe the data dictionary will help to better understand the nature of the datasets and the variables used to built different models.

|  |  |  |
| --- | --- | --- |
| **Jared Dataset** | | |
| **Variable** | **Class** | **Description** |
| "polla\_qid" | "integer" | "Quiz ID" |
| "answer" | "character" | "Answer (likert scale)" |
| "votes" | "integer" | "Number of votes for that question/answer combo" |
| "pollq\_id" | "integer" | "Poll Question ID" |
| "question" | "character" | "Question" |
| "place" | "character" | "Pizza Place" |
| "time" | "integer" | "Time of quiz" |
| "total\_votes" | "integer" | "Total number of votes for that pizza place" |
| "percent" | "double" | "Vote percent of total for that pizza place" |

|  |  |  |
| --- | --- | --- |
| **Barstool Dataset** | | |
| **Variable** | **Class** | **Description** |
| "name" | "character" | "Pizza place name" |
| "address1" | "character" | "Pizza place address" |
| "city" | "character" | "City" |
| "zip" | "double" | "Zip" |
| "country" | "character" | "Country" |
| "latitude" | "double" | "Latitude" |
| "longitude" | "double" | "Longitude" |
| "price\_level" | "double" | "Price rating (smaller = cheaper)" |
| "provider\_rating" | "double" | "Provider review score" |
| "provider\_review\_count" | "double" | "Provider review count" |
| "review\_stats\_all\_average\_score" | "double" | "Average Score" |
| "review\_stats\_all\_count" | "double" | "Count of all reviews" |
| "review\_stats\_all\_total\_score" | "double" | "Review total score" |
| "review\_stats\_community\_average\_score" | "double" | "Community average score" |
| "review\_stats\_community\_count" | "double" | "community review count" |
| "review\_stats\_community\_total\_score" | "double" | "community review total score" |
| "review\_stats\_critic\_average\_score" | "double" | "Critic average score" |
| "review\_stats\_critic\_count" | "double" | "Critic review count" |
| "review\_stats\_critic\_total\_score" | "double" | "Critic total score" |
| "review\_stats\_dave\_average\_score" | "double" | "Dave (Barstool) average score" |
| "review\_stats\_dave\_count" | "double" | "Dave review count" |
| "review\_stats\_dave\_total\_score" | "double" | "Dave total score" |

|  |  |  |
| --- | --- | --- |
| **Datafiniti Dataset** | | |
| **Variable** | **Class** | **Description** |
| "polla\_qid" | "integer" | "Quiz ID" |
| "answer" | "character" | "Answer (likert scale)" |
| "votes" | "integer" | "Number of votes for that question/answer combo" |
| "pollq\_id" | "integer" | "Poll Question ID" |
| "question" | "character" | "Question" |
| "place" | "character" | "Pizza Place" |
| "time" | "integer" | "Time of quiz" |
| "total\_votes" | "integer" | "Total number of votes for that pizza place" |
| "percent" | "double" | "Vote percent of total for that pizza place" |

**2.0 Data Problem Description:**

**2.1 Jared and Barstool Data Issues -** The first major problem I had was different pizza rating scales. The ‘Jared’ dataset contained written answers: “Never Again,” “Poor,” “Fair,” “Average,” “Good,” and “Excellent.” The ‘Barstool’ dataset was based on a 1-10 scale. I used a mutate function to convert the 1-6 ‘Jared’ likert scale to a 1-10 scale. After the scales were made similar, a true comparison could be done. Once the answers are converted to integers, a true calculation is done to find the average pizza rating for each pizza place:

|  |  |
| --- | --- |
| **Average Rating** | |
| **Place** | **Average Score** |
| "5 Boroughs Pizza" | 7.333 |
| "Artichoke Basille's Pizza" | 8 |
| "Arturo's" | 7.428 |
| "Bella Napoli" | 7.066 |
| "Ben's of SoHo 14th Street" | 4.8 |
| "Ben's of SoHo Spring Street" | 6.444 |
| "Big Slice Pizza" | 6.266 |
| "Bleecker Street Pizza" | 8.285 |
| "Bravo Pizza" | NA |
| "Cavallo's Pizza" | 7.27 |
| "Champion Pizza" | 5.666 |
| "Dona Bella" | 8 |
| "Dough Boys" | 7.230 |
| "Famous Original Ray's" | 5.857 |
| "Fiore's" | 7.821 |

An average rating of 6.8 was given to all restaurants. Two columns are dropped for being insignificant, which are the *jared$polla\_qid* and *jared$time*. When I first tried to get the mean score, there were some restaurants which contained NA values of no vote. I took the mean of average scores omitting the 0 values.

Next, I wanted to see exactly which restaurants were the same in both datasets. I used an intersect function and then outputted a table representing the restaurants that are in both the ‘barstool’ and ‘Jared’ datasets.

|  |  |
| --- | --- |
| **Intersecting Restaurants in Datasets** | |
| "Little Italy Pizza" | "Williamsburg Pizza" |
| "Girello" | "Steve's Pizza" |
| "Saluggi's" | "5 Boroughs Pizza" |
| "Bleecker Street Pizza" | "Artichoke Basille's Pizza" |
| "Champion Pizza" | "Joe's Pizza" |
| "Kiss My Slice" | "Prince Street Pizza" |
| "Vinny Vincenz" | "Arturo's" |
| "Gotham Pizza" | "Stella's Pizza" |
| "Pizza Italia" | "Highline Pizza" |
| "NY Pizza Suprema" | "Rivoli Pizza" |
| "Rocco's Pizza Joint" | "Previti Pizza" |

**2.2 – Datafiniti Problems and Cleaning –** Since there weren’t any NA values in the ‘datafiniti’ dataset, I simply removed duplicate values and changed the name of “province” to “State Code” to align with the column name in the region dataset. My next task was to make sense of the categories column, which originally contained values in one column:

|  |
| --- |
| "Pizza,Restaurant,American restaurants,Pizza Place,Restaurants" |
| "Pizza,Pizza Place,Restaurants" |
| "Restaurant,Pizza Place,Restaurants" |
| "Pizza,Carry-out food,Pizza Place,Restaurants" |
| "Pizza,American restaurants,Pizza Place,Pizza equipment and supplies,Restaurants" |
| "Pizza Place" |

I accomplished this in a few ways. First, I created a unique id for each pizza place. I then separated the different categories into individual columns and then finally created a new dataframe with only the unique category ID and separated categories.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| State Code | Cat1 | Cat2 | Cat3 | Cat4 | Cat5 | Cat6 | Cat7 | Cat8 | Cat9 | Cat10 |
| "AK" | "Pizza Place" | NA | NA | NA | NA | NA | NA | NA | NA | NA |
| "AK" | "Pizza Place" | "Take Out Restaurants" | "Restaurants" | "Pizza" | NA | NA | NA | NA | NA | NA |
| "AK" | "Pizza" | "Take Out Restaurants" | "Pizza Place" | "Restaurants" | NA | NA | NA | NA | NA | NA |
| "AL" | "Pizza Place" | NA | NA | NA | NA | NA | NA | NA | NA | NA |
| "AL" | "Pizza Place" | NA | NA | NA | NA | NA | NA | NA | NA | NA |
| "AL" | "Pizza Place" | NA | NA | NA | NA | NA | NA | NA | NA | NA |

As seen above, many NA values were given, and the categories need to be gathered into one column.

The list is then shrunk into unique values. Afterwards, repeating the steps above on the word “and” allowed me to further separate categories such as “Italian Restaurant and Pizza Place” into their own unique values. I then looked at a count of the categories and consolidated them into 10 main categories and an “Other” category containing descriptions with less than 10 records each. Finally, merging with the original dataset to end up with a row for each pizza place and category for analysis by category.

**3.0 Solution Narrative with Exploratory Analysis –** Once the datasets were clean, I was able to analyze the data and find solutions to my issues. I first looked at a brief overview of price by category to understand which type of restaurant has the most expensive and cheapest pizza. Sandwich places have the lowest average price minimum while Bar/pubs have the highest average minimum price. The minimum price of the bar/pub is five times as much as sandwich places:

|  |  |  |  |
| --- | --- | --- | --- |
| **Price by Category** | | | |
| **New\_cat** | **Avg\_Min\_Price** | **Avg\_Max\_Price** | **Range** |
| "Sandwich Place" | 2 | 26.2 | 24.2 |
| "Pizza Restaurant" | 2.52 | 26.60 | 24.08 |
| "Caterer" | 4.3 | 27.55 | 23.25 |
| "Restaurant" | 4.40 | 27.64 | 23.23 |
| "Delivery / Carry-out" | 4.54 | 27.72 | 23.18 |
| "Pizza Place" | 4.67 | 27.80 | 23.13 |
| "Karaoke" | 5 | 28 | 23 |
| "Other" | 6.16 | 28.59 | 22.43 |
| "American" | 7.57 | 29.38 | 21.80 |
| "Italian" | 9.73 | 30.89 | 21.16 |
| "Bar / Pub" | 10.84 | 31.50 | 20.66 |

Afterwards, I took a summary of the minimum and maximum price range. The minimum max was $50 while the maximum max was $55. No matter how nice of a restaurant, you should never pay more than $50-55 for pizza. To study this further, two boxplots were created to review the range for each type of restaurant.

A screenshot of a cell phone

Description automatically generatedA picture containing photo, white

Description automatically generated

After reviewing the two plots, it is interesting to see the category of Pizza place to have the largest range seemingly in both minimum and maximum. On the contrary, places like sandwich shops and karaoke do not seem to ever get too expensive. I think category is an important factor, as I typically would not go get pizza at a sandwich shop, but I do think location is more of a factor. Due to this, I then looked at price by region/state in terms of minimum and maximum. I found the West to be the most expensive and the state of Connecticut to have the average highest max price. There is a large drop off between the most expensive state even down to the 10th most expensive state. Below is a summary table:

|  |  |  |
| --- | --- | --- |
| **Price by Region** | | |
| **Region** | **Avg\_min\_price** | **Avg\_max\_price** |
| “Midwest” | 4.35 | 27.61 |
| “Northeast” | 3.52 | 27.04 |
| “South” | 5.13 | 28.09 |
| “West” | 5.74 | 28.50 |

|  |  |
| --- | --- |
| **Price by State** | |
| **State** | **Avg\_max\_price** |
| "Connecticut" | 40 |
| "South Dakota" | 32.5 |
| "Iowa" | 31 |
| "Alabama" | 30 |
| "Nebraska" | 30 |
| "Arizona" | 29.92 |
| "West Virginia" | 29.28 |
| "Maryland" | 29.21 |
| "Louisiana" | 29.09 |
| "Oregon" | 29.09 |
| "Idaho" | 28.75 |

Tables are great, but a more general view can be easier to understand the relationship of all prices. I made a point plot with minimum price range on my Y axis and maximum on my X axis. Most pizza prices are between $10 and $30. Some outliers are seen between $40 and $55.

A picture containing group, white, light, water

Description automatically generated

While the pizza party dataset contains a lot of valuable information in terms of pricing and ratings, it lacks data from all states. After further research, I discovered there is no large database or set for pizza restaurants in the US. Yelp contains many reviews, but these are all singular and not put into a database. Below is a map of where all the review ratings are from:

A close up of a map

Description automatically generated

It is easy to see many reviews are coming from New York, Florida, and the eastern side of the map. There are dots on the West, but primarily from California. As stated earlier, the west contains the highest price.

Between the datasets there are many types of reviewers. These were Dave from Barstool, the community, pizza critics, and the Jared dataset replies. It is interesting to see the critics contained the highest reviews with a 7.256 average rating. Dave from Barstool is the toughest reviewer with an average of 6.622. These are drastic once you realize the thousands of pizza restaurants reviewed.

**4.0 Solutions with Modeling and Mathematical Values**

**4.1. Linear Modeling –** Now that all the datasets are cleaned, locations are given, and prices are examined; I want to see the relationship of different models and variables. Scatter plots are a great way of examining two numerical variables in their relationship to each other. The first plot I examined was the barstool dataset which included price level and the provider ratings. As price increases, not all the ratings do. However, all ratings are in between are a 3 or 4 out of a 1-5 scale. More expensive pizza eliminates any one- or two-star ratings but does not guarantee the best pizza.

A screenshot of a cell phone

Description automatically generated

To futher understand provider rating, I am going to build a Linear Regression model on the barstool dataset. This model will help to understand the variables driving the provider rating. Additionally, I reduced the barstool dataset by eliminating the character values of name, address, and country. This dataset is now called barstool2.

I built a training model on 70% of the data selecting all my variables and had provider rating as my independent variable. Below is the output of my model:

|  |  |  |  |
| --- | --- | --- | --- |
| **70% Training Model** | | | |
| **Variable** | **Estimate** | **Std. Error** | **Pr(>|t|)** |
| Latitude | -1.310e-03 | 6.133e-03 | 0.830964 |
| Longitude | -2.608e-03 | 2.304e-03 | 0.258567 |
| Price\_level | -1.425e-01 | 4.633e-02 | 0.002291 \*\* |
| Provider\_review\_count | 1.502e-04 | 4.705e-05 | 0.001558 \*\* |
| Review\_stats\_all\_average\_score | 1.421e-01 | 4.103e-02 | 0.000612 \*\*\* |
| Review\_stats\_all\_count | -8.160e-02 | 3.222e-01 | 0.800230 |
| Review\_stats\_all\_total\_score | -8.190e-03 | 2.595e-02 | 0.752520 |
| Review\_stats\_community\_average\_score | 5.546e-03 | 1.753e-02 | 0.751992 |
| Review\_stats\_community\_count | 6.465e-02 | 3.215e-01 | 0.840760 |
| Review\_stats\_community\_total\_score | 1.008e-02 | 2.571e-02 | 0.695191 |
| Review\_stats\_critic\_average\_score | -3.442e-02 | 2.787e-02 | 0.217714 |
| Review\_stats\_critic\_total\_score | 3.638e-02 | 6.514e-02 | 0.576922 |

The variables price level, provider review count, and review stats all average score are seen as significant. Next I am going to test MSE, R squared, adjusted R squared, AIC and BIC on the initial training model with all variables.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **All training variables testing statistics** | | | | |
| **MSE** | **R-Squared** | **Adj. R-Squared** | **AIC** | **BIC** |
| 0.213 | .220 | 0.190 | 434.027 | 486.958 |

In the model above, the variables price level, provider review count and review stats all average score are seen as significant. I am now going to build a model with these variables selected. The new model output:

|  |  |  |  |
| --- | --- | --- | --- |
| **My 70% Training Model** | | | |
| **Variable** | **Estimate** | **Std. Error** | **Pr(>|t|)** |
| Price\_level | -1.237e-01 | 4.422e-02 | 0.005452 \*\* |
| Provider\_review\_count | 1.471e-04 | 4.263e-05 | 0.000633 \*\*\* |
| Review\_stats\_all\_average\_score | 1.322e-01 | 1.735e-02 | 2.88e-13 \*\*\* |

I am now going to perform the same performance statistics as the initial training model to see how the new model compares.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **My model testing statistics** | | | | |
| **MSE** | **R-Squared** | **Adj. R-Squared** | **AIC** | **BIC** |
| 0.211 | 0.205 | 0.197 | 422.457 | 441.360 |

After reviewing the two models, my model performs better compared to the first. The AIC and BIC values becoming smaller indicate a better fit. The MSE value is slightly lower as well.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model Comparison** | | | | | |
| **Model** | **MSE** | **R-Squared** | **Adj. R-Squared** | **AIC** | **BIC** |
| All variables – training set | 0.213 | .220 | 0.1907 | 434.0278 | 486.958 |
| My model – training set | 0.211 | 0.205 | 0.1977 | 422.4572 | 441.360 |

Next, I am using the testing data to evaluate the performance of the model I built using the training data. I am using the predict function to get the predicted values from the test set. I am then taking the MSE, which is the average of the squared differences between the predicted and actual values. My testing error is larger than my training error.

|  |  |
| --- | --- |
| **MSE Comparison** | |
| **Model** | **MSE** |
| My Model – training set | 0.211 |
| My Model – testing set | 0.239 |

I then used the LASSO method on my **training dataset**. For LASSO I used the glmnet package. This package does not take data frame as input, so I first specified the x matrix and y vector. My alpha value was set equal to 1. I then used a 5-fold cross validation on my dataset to pick lambda. Below is a plot of the LASSO fit:

A screenshot of a cell phone

Description automatically generated

Using my plot and the lambda.min function, I found the best lambda to be 0.01281311.

I then used LASSO on my **testing dataset**. Below is a plot of the LASSO fit:

A close up of a logo

Description automatically generated

Using my plot and the lambda.min function, I found the best lambda to be 0.05227127.

If you would like to ensure both set of quantiles come from the same distribution, you can examine a Q-Q plot. A Q-Q plot is a scatterplot created by plotting two sets of quantiles against one another. If both sets of quantiles came from the same distribution, we should see the points forming a line that is roughly straight.

A screenshot of a map

Description automatically generated

According to the Q-Q plot, the quantiles come from the same distribution.

**4.2. Non-Linear Modeling**

Next, I would like to examine a few nonlinear methods. The first nonlinear method I would like the examine is Random Forest. The idea of random forest is to randomly select m out of p predictors as candidate variables for each split in each tree. Commonly, m = the square root of p. The reason of doing this is that it can decorrelates the trees such that it reduces variances when we aggregate the trees. I built a model with provider rating as my independent variable. I had a mean of squared residuals of 0.2127433 and 19.28% of variance explained. Below is a table of random forest variable importance.

|  |  |  |
| --- | --- | --- |
| **Random Forest** | | |
| **Variables** | **%IncMSE** | **IncNodePurity** |
| Latitude | 0.001 | 9.140 |
| Longitude | 0.002 | 8.939 |
| Price\_level | 0.005 | 3.505 |
| Provider\_review\_count | 0.041 | 19.08 |
| Review\_stats\_all\_average\_score | 0.044 | 15.32 |
| Review\_stats\_all\_count | 0.015 | 4.135 |
| Review\_stats\_all\_total\_score | 0.028 | 8.788 |
| Review\_stats\_community\_average\_score | 0.056 | 16.174 |
| Review\_stats\_community\_count | 0.017 | 4.077 |
| Review\_stats\_community\_total\_score | 0.032 | 7.508 |
| Review\_stats\_critic\_average\_score | 0.00063 | 1.072 |
| Review\_stats\_critic\_count | 7.156 e-05 | 0.334 |
| Review\_stats\_critic\_total\_score | 0.00025 | 0.936 |
| Review\_stats\_dave\_average\_score | 0.012 | 7.022 |
| Review\_stats\_dave\_count | 0 | 0 |
| Review\_stats\_dave\_total\_score | 0.013 | 7.468 |

The Review\_stats\_critic\_count category has the largest %IncMSE and the Provider\_review\_count and Review\_stats\_all\_average\_score have the highest IncNodePurity. IncNodePurity relates to the loss function which by best splits are chosen. More useful variables achieve higher increases in node purities. Provider review count, review stats all average score and review stats community average score are the highest, which stays consistent with the previous models.

Next, I perform Random Forest on my **testing dataset**. I had a mean of squared residuals of 0.2292534 and and 13.35% of variance explained. Below is a table of random forest variable importance:

|  |  |  |
| --- | --- | --- |
| **Random Forest - Testing** | | |
| **Variables** | **%IncMSE** | **IncNodePurity** |
| Latitude | -0.002 | 3.0967 |
| Longitude | 0.001 | 3.7648 |
| Price\_level | 0.0008 | 0.4831 |
| Provider\_review\_count | 0.0348 | 6.0633 |
| Review\_stats\_all\_average\_score | 0.0275 | 3.0308 |
| Review\_stats\_all\_count | 0.0172 | 1.7290 |
| Review\_stats\_all\_total\_score | 0.0181 | 2.6151 |
| Review\_stats\_community\_average\_score | 0.0341 | 4.8426 |
| Review\_stats\_community\_count | 0.0181 | 1.7074 |
| Review\_stats\_community\_total\_score | 0.0240 | 2.4932 |
| Review\_stats\_critic\_average\_score | 0.0014 | 0.3209 |
| Review\_stats\_critic\_count | 0.0012 | 0.1821 |
| Review\_stats\_critic\_total\_score | 0.0024 | 0.3104 |
| Review\_stats\_dave\_average\_score | 0.0095 | 1.8442 |
| Review\_stats\_dave\_count | 0 | 0 |
| Review\_stats\_dave\_total\_score | 0.0080 | 1.9140 |

In this table, my results stay consistent as provider review count, review stats all average score, and review stats community average score have the highest %IncMSE. These also contain the highest IncNodePurity values, along with location.

The next nonlinear method I am going to test is boosting for regression trees. I have set the number of small trees to 10000 and the shrinkage to 0.01. The shrinkage is another tuning parameter which controls how much contribution each tree makes. Based off my variables, I thought the interaction depth of 8 would be sufficient. It is important to set the parameters correctly to avoid overfitting.

I built the model based on training data with provider rating as my independent variable, below is the output with relative influence:

|  |  |
| --- | --- |
| **Boosting for Regression Trees** | |
| **Variable** | **Relative Influence** |
| Provider\_review\_count | 18.023 |
| Review\_stats\_all\_average\_score | 14.040 |
| Review\_stats\_community\_average\_score | 12.849 |
| Longitude | 12.189 |
| Latitude | 12.157 |
| Review\_stats\_dave\_average\_score | 9.080 |
| Review\_stats\_all\_total\_score | 8.660 |
| Review\_stats\_community\_total\_score | 4.475 |
| Price\_level | 3.416 |
| Review\_stats\_all\_count | 3.220 |

After running the model, the output shows provider review count and review stats all average score as the highest relative influence. This also stays consistent with previous models built. I then went ahead and plotted these two variables:

A screenshot of a map

Description automatically generated

The provider review count graph shows the y increasing from 3.4 to 4.0 while the the x-axis is increasing from 0 to roughly 2000. Once the x-axis hits 2000, the line starts to flatten out.

The next plot shows the review stats all average score:

A close up of a map

Description automatically generated

This plotted line is flat until the review stats all average score hits roughly 3.6. There is more randomness in the increase, however it does keep increasing rather linearly.

I then did prediction on my **testing sample** with the number of trees being 100. The MSE value is 0.2289288

Next I investigate how the testing error changes with the different number of trees. The plot is below:

A close up of a map

Description automatically generated

**5.0 Summary and Conclusions -** First, I hope you enjoyed the overview of the data I was given. We have learned that if you are looking for a cheap pizza, simply go to a pizza restaurant! Bars/pubs contain the highest minimum and maximum prices!

In terms of location, the west side of the USA contains the highest average minimum and maximum price. The Northeast part of the USA contains both the cheapest minimum and maximum price.

If you are looking to travel anywhere across the USA and want to pay top dollar, visit Connecticut. If you prefer to save some money, I recommend visiting South Carolina, Arkansas or Kansas.

A higher price does not guarantee a higher rating, however it does eliminate a low star rating. The higher prices tended to vary in 3-4 stars out of 5. In linear modeling, two models were built. The second model performance is better than the first. This model stays consistent with the testing dataset. This model contains variables of price level, provider review count, and review stats all average score. LASSO is performed with the best lambda minimum being 0.01281311 on the training set and 0.05227127 in the testing set. The nonlinear methods of randomForest and boosting stay consistent with the models previously built, both in the complete and testing models. Provider review count and Review stats all average score have the highest relative influence in both models. RandomForest shows these same variables having the highest IncNodePurity in both the complete model and testing model as well.

**6.0 Appendix**

setwd("C:/Users/14408/Desktop/Data Wrangling")

datafiniti <- read\_csv("pizza\_datafiniti.csv")

barstool <- read\_csv("pizza\_barstool.csv")

jared <- read\_csv("pizza\_jared.csv")

region <- read\_csv("us census bureau regions and divisions.csv")

jared\_desc <- read\_excel("dataset\_desc.xlsx", sheet = "Jared")

barstool\_desc <- read\_excel("dataset\_desc.xlsx", sheet = "Barstool")

datafiniti\_desc <- read\_excel("dataset\_desc.xlsx", sheet = "Jared")

kable(jared\_desc) %>%

kable\_styling(bootstrap\_options = "striped", full\_width = F, position = "left")

colSums(is.na(barstool))

colSums(is.na(jared))

colSums(is.na(jared))

colSums(is.na(region))

jared <- jared %>%

mutate(answer = case\_when(

.$answer=="Never Again" ~ 0,

.$answer=="Poor" ~ 2,

.$answer=="Fair" ~ 4,

.$answer=="Average" ~ 6,

.$answer=="Good"~ 8,

.$answer=="Excellent" ~ 10))

jared$answer <- as.integer(jared$answer)

str(jared$answer)

jared <- mutate(jared,Weighted\_Rating = answer\*votes)

(Jared\_Average <- jared %>%

group\_by(place) %>%

summarise(avg\_score = sum(Weighted\_Rating)/sum(votes)))

(Jared\_Mean <- mean(Jared\_Average$avg\_score, na.rm = T))

jared$polla\_qid = NULL

jared$time = NULL

head(jared)

Critic\_average <- mean(NA^(barstool$review\_stats\_critic\_average\_score == 0)\*barstool$review\_stats\_critic\_average\_score, na.rm = TRUE)

Daves\_average <- mean(NA^(barstool$review\_stats\_dave\_average\_score == 0)\*barstool$review\_stats\_dave\_average\_score, na.rm = TRUE)

Community\_average <- mean(NA^(barstool$review\_stats\_community\_average\_score == 0)\*barstool$review\_stats\_community\_average\_score, na.rm = TRUE)

datafiniti\_unique <- unique(datafiniti)

datafiniti\_unique *# 2,285 unique observations*

names(datafiniti\_unique)[names(datafiniti\_unique) == 'province'] <- 'State Code'

datafiniti\_cat <- datafiniti\_unique %>% separate(categories, c("Cat1", "Cat2", "Cat3", "Cat4","Cat5",

"Cat6", "Cat7", "Cat8", "Cat9", "Cat10",

"Cat11"), sep = ",")

datafiniti\_cat %>%

group\_by(Category) %>%

summarize(count = n()) %>%

arrange(desc(count))

datafiniti\_new <- merge(datafiniti\_unique, datafiniti\_cat, by = 'loc\_id')

datafiniti\_new <- datafiniti\_new %>%

select(-c(categories, Category))

datafiniti\_new <- unique(datafiniti\_new)

kable(head(select(datafiniti\_new, c(loc\_id, name, new\_cat)), 10)) %>%

kable\_styling(bootstrap\_options = "striped", full\_width = F, position = "left")

datafiniti\_new %>%

group\_by(new\_cat) %>%

summarize(Avg\_Min\_Price = mean(price\_range\_min), Avg\_Max\_Price = mean(price\_range\_max)) %>%

mutate(Range = Avg\_Max\_Price - Avg\_Min\_Price) %>%

arrange(Avg\_Min\_Price)

summary(datafiniti\_unique$price\_range\_min) *# 3rd quartile is still 0*

datafiniti\_unique <- datafiniti\_unique %>%

mutate(Range = price\_range\_max - price\_range\_min)

datafiniti\_new <- datafiniti\_new %>%

mutate(Range = price\_range\_max - price\_range\_min)

summary(datafiniti\_unique$Range)

summary(datafiniti\_new$Range)

datafiniti\_unique %>%

ggplot(aes(x = price\_range\_min)) +

geom\_histogram(binwidth = 2) +

coord\_cartesian(xlim = c(0,40))

datafiniti\_unique %>%

ggplot(aes(x = price\_range\_max)) +

geom\_histogram(binwidth = 2) +

coord\_cartesian(xlim = c(20,55))

datafiniti\_unique %>%

ggplot(aes(x = Range)) +

geom\_histogram(binwidth = 2) +

coord\_cartesian(xlim = c(10,30))

datafiniti\_new %>%

ggplot(aes(x = new\_cat, y = price\_range\_min)) +

geom\_boxplot() +

coord\_flip()

datafiniti\_new %>%

ggplot(aes(x = new\_cat, y = price\_range\_max)) +

geom\_boxplot() +

coord\_flip()

datafiniti\_unique <-

datafiniti\_unique %>%

merge(region, by = "State Code") %>%

select(-(Division))

*# Min / max by region*

datafiniti\_unique %>%

group\_by(Region) %>%

summarize(Avg\_min\_price = mean(price\_range\_min),

Avg\_max\_price = mean(price\_range\_max))

datafiniti\_unique %>%

group\_by(State) %>%

summarize(Avg\_max\_price = mean(price\_range\_max)) %>%

arrange(desc(Avg\_max\_price))

datafiniti\_unique %>%

group\_by(State) %>%

summarize(Avg\_min\_price = mean(price\_range\_min)) %>%

arrange(Avg\_min\_price)

m <- leaflet(barstool) %>% addTiles()

m %>% addCircles(lng = ~ barstool$longitude, lat = ~ barstool$latitude, popup = barstool$nameL, weight = 8, color = "#fb3004", stroke = TRUE)

ggplot(data = datafiniti, aes(x=price\_range\_max,y=price\_range\_min)) +

geom\_point(color = "blue", size=2,shape=17)

plot(barstool$price\_level, barstool$provider\_rating, xlab = "Price", ylab = "Rating")

barstool2<- read\_csv("pizza\_barstool2.csv")

set.seed(08331816)

index <- sample(nrow(barstool2),nrow(barstool2)\*0.70)

barstool.train <- barstool2[index,]

barstool.test <- barstool2[-index,]

barstool.glm0 <- lm(provider\_rating~., data = barstool.train)

summary(barstool.glm0)

model\_summary <- summary(barstool.glm0)

(model\_summary$sigma)^2

model\_summary$r.squared

model\_summary$adj.r.squared

AIC(barstool.glm0)

BIC(barstool.glm0)

barstool.glm1 <- lm(provider\_rating~ price\_level+provider\_review\_count+review\_stats\_all\_average\_score, data = barstool.train)

summary(barstool.glm1)

model\_summary2 <- summary(barstool.glm1)

(model\_summary2$sigma)^2

model\_summary2$adj.r.squared

model\_summary2$r.squared

AIC(barstool.glm1)

BIC(barstool.glm1)

pi <- predict(object = barstool.glm0, newdata = barstool.test)

predict(barstool.glm0)

mean((pi-barstool.test$provider\_rating)^2)

library(glmnet)

set.seed(08331816)

lasso\_fit <- glmnet(x=as.matrix(barstool.train[, -c (which(colnames(barstool.train)=='provider\_rating'))]),y=barstool.train$provider\_rating, alpha=1)

lasso\_fit

cv\_lasso\_fit$lambda.min

cv\_lasso\_fit = cv.glmnet(x = as.matrix(barstool.train[, -c(which(colnames(barstool.train)=='provider\_rating'))]), y = barstool.train$provider\_rating, alpha = 1, nfolds = 5)

plot(cv\_lasso\_fit)

cv\_lasso\_fit$lambda.min

cv\_lasso\_fit\_test = cv.glmnet(x = as.matrix(barstool.test[, -c(which(colnames(barstool.test)=='provider\_rating'))]), y = barstool.test$provider\_rating, alpha = 1, nfolds = 5)

plot(cv\_lasso\_fit\_test)

cv\_lasso\_fit\_test$lambda.min

Barstool\_prediction = predict(lasso\_fit, as.matrix(barstool.test[, -c(which(colnames(barstool.test)=='provider\_rating'))]), s =

cv\_lasso\_fit$lambda.min)

Barstool\_prediction

#plot(barstool.glm1)

library(randomForest)

barstool\_rf <- randomForest(provider\_rating~ ., data = barstool2, importance = TRUE)

barstool\_rf

barstool\_rf$importance

library(gbm)

barstool\_boost <- gbm(provider\_rating~., data = barstool.train, distribution = "gaussian", n.trees = 10000, shrinkage = 0.01, interaction.depth = 8)

summary(barstool\_boost)

barstool.boost.pred.test <- predict(barstool\_boost, barstool.test, n.trees = 100)

mean((barstool.test$provider\_rating-barstool.boost.pred.test)^2)

# Now investigating how the testing error changes with different trees

ntree <- seq(100,10000,100)

predmat <- predict(barstool\_boost,newdata = barstool.test, n.trees = ntree)

err <- apply((predmat-barstool.test$provider\_rating)^2,2,mean)

plot(ntree,err,col=2,lwd=2, xlab = "n.trees", ylab = "Test MSE")

plot(barstool\_boost, i="provider\_review\_count")

plot(barstool\_boost, i="review\_stats\_all\_average\_score")