Predicting Beer Reviews Using Data from BeerAdvocates

Ajay Iyer & Nicolai Jensen

Arizona State University

11/20/2022

Table of Contents

Abstract/Executive Summary

The Beer Advocates community is one of the largest beer communities on the internet. Many people love trying new beers and some go on a conquest to find the best beer they have ever tasted. Reviewers turn into connoisseurs by rating beer and communicating it to the whole community. This study was conducted to see if there are factors to find correlations and if they have the ability to predict the beer and beer rating.

There are many factors that go into reviewing a beer to find its rating so this study will try and predict a beer and its rating. Some of the factors include aroma, appearance, and taste ratings. This exploratory data analysis will help us understand the data which will help us create and locate the best model for predicting beer and its rating.

Using various machine learning algorithms and statistical techniques in Python, we are concluding that the Random Forest Regression models will be able to predict a beer and its rating with different aspects that are needed to review beer. The rest of the report will show an analysis of understanding the data that will help locate the best model needed that is able to predict beer and its rating.

Project Plan

Primary Company Details:

Founded - August 23, 1996

Founders - Todd Alström and Jason Alström

Headquarters - Boston, Massachusetts

Categories - Craft Beer, Reviews and Recommendations, Search

In February 2020, BeerAdvocate was acquired by Next Glass

Company Communication:

Website: beeradvocate.com

E-Mail: mail@beeradvocate.com

Twitter: @BeerAdvocate

Business Description:

BeerAdvocates is one of the biggest and oldest online beer communities. The community was founded in 1996 by the Alström Brothers. It is the go-to source for beer information for over millions of users, the host of top-notch beer festivals, and holds the industry standard for beer evaluations. Users can also search to find local breweries and establishments, rate and review them, start or join conversations, trade beers, connect with people, partake in online tastings, and discover more about the multiple live events that take place. Beer Advocates was the first online community and it sparked a competition as more communities were created and how similar they were. Communities like RateBeer, Probrewer, TalkBeer, and Untappd all have overlapping similarities with Beer Advocates but they have one or two functions. Beer Advocates has found

massive success due to the wide range of forums/discussions for users to discuss anything about beer from rating to trading with locals in the community.

The site sees a heavy amount of online traffic with the Breakfast Stout from Founders Brewing Company seeing 17,854 separate reviews over the span of the site's history, which is the most of any one beer. The website also includes discussion boards for people living all over the world to discuss beers from their region. These discussions include ones for the North, South, East, and West of the United States as well as other countries such as Australia, Belgium, Canada, Germany, and so many more. BeerAdvocate can keep track of which beers do well in the eyes of the consumer as well as ones that do poorly like Miller Genuine Draft 64 which has the lowest average overall score with just 1.6 out of a possible 5 points.

Business Analysis/Opportunity:

The dataset for this analysis was sourced from Datadoume on Kaggle. This dataset was created as a result of discussions at the O'Reilly Media Conference on hiring and testing for data skills. The dataset consists of 1.5 million beer reviews with 13 different attributes including 28,000 unique brewery ids, 5,742 brewery names, various review times, overall review score/aroma/appearance/palate/taste ranging from 0 to 5, 104 unique beer styles, 56,857 beers, various ranges of alcohol by volume in a beer, and various unique beer IDs. We are prepared to evaluate a few research questions and provide analytical regression models. These models will have capabilities based on the variables involved for predicting the best beer.

Research Questions:

With the world of beer ever expanding the need for solid advice is more necessary than ever. Many consumers will look at a review website like BeerAdvocate to get an idea of what

new beers to try and what they may not like. With the beer industry being flooded with such variability in products in recent years it will be more important than ever to identify patterns in consumer tendencies. This information will prove invaluable to both the consumer and distributor. The following questions will be researched in this project.

Research Question #1: **What will the beer rating be based on the aspects of beer reviewing (aroma, appearance, palate, and taste) as well as beer style and alcohol by volume?**

The attempt of this research question is to identify how an overall beer rating is influenced by each factor and to identify tendencies between rating and style of each beer. It is important to identify the tendencies of the reviews in order to determine how each beer within a specific genre or brewery can cater to the likes of the customer. This type of information is critical to beer companies since it can help to determine the market value of each beer based on the demographic that it calls to.

Research Question #2: **What kind of beer can we expect a reviewer to review next?**

Knowing what a consumer will buy is a powerful piece of information to any company. By looking at previous reviews and tendencies of major reviewers we will be able to identify characteristics of said reviewer and determine a probable next item that the reviewer will choose. By identifying what a consumer will like and want to buy next, the beer companies will be able to identify ways to sell to similar customers and likely raise sales based on probability.

Research Question #3: **How much variability exists within each beer style?**

By looking at the specific characteristics of beer styles it will be possible to get an idea of what constitutes a specific type of beer and how those aspects lean into customer appeal. It is important to consumers to be able to identify which beers they may also like depending on what aspects of the review they think are the most important in a beer. By identifying how each beer is different within a specific style it will give new understanding to the consumer on how each beer can be the right one for them.

Hypotheses:

Hypothesis #1: **The aspects of (aroma, appearance, palate, and taste) will have the greatest impact on the overall score but will not be able to tell the whole picture.**

Since these aspects of review feed heavily into the overall score they will be the determining factor but will not give the total picture of the beer. Two beers for example could very reasonably have the same scores in the categories but have wildly different styles and flavors that give off a different experience.

Hypothesis #2: **The reviewer will next pick a beer with a large amount of reviews from a similar beer type.**

It would make sense for the reviewer to lean into the majority and review a popular beer next with a similar category since they will be interested in how it stacks up to the others they have tried. It would also make sense for these reviewers to consistently input into the website to try to determine which beer is best within a specific style and approach their reviews from that angle.

Hypothesis #3: **Style like IPA's will have higher variability due to the wide range of ABV in the category.**

Since IPA's have a wide range of interpretation between breweries it would make sense to see a wide range of scores due to the variability in the process between beer brands. By not having a consistent formula for this style the reviews will express a lack of continuity. It will be important to identify if this an isolated problem or if the variability between brands is a more widespread issue in terms of prediction.

Data:

The data is combined with the reviews that were sourced from BeerAdvocates and uploaded on Kaggle. The dataset consists of various breweries that have reviews for their respective beer based on what the reviewers look for. The data only consists of 13 structured variables and that includes the brewery id which is the unique id for each of the breweries in the data. Brewery name shows the various brewery names and review time is how long it took the reviewer to score the beer. Review overall is the overall score of the beer by the reviewer and the review aroma is the score for the aroma of the beer. Review appearance is a score for the appearance of the beer while review profile name is the online name of the reviewer. Beer style tells us the style of the beer. The review palate is another feature that is scored along with taste. Beer name organizes the name of the beer with their respective review based on who reviewed the beer along with beer abbreviation and unique beer ID for the respective beer. All of the review variables are scored from a range of 0 to 5 with 5 being the best score while 0 is the worst score to have for the beer.

brewery_name: The names of the breweries that produced beers reviewed.

review_overall: Overall score/rating of the beer from a scale of 1 to 5.

review_aroma: Rating of the beer's aroma from a scale of 1 to 5.

review_appearence: Rating of the beer's appearance from a scale of 1 to 5.

review_profilename: Unique profile name of each reviewer in the BeerAdvocates Community.

beer_style: Names of the unique style the beer classifies as.

review_palate: Rating of the beer when it is on the roof of the mouth from a scale of 1 to 5.

review_taste: Rating of the beer's taste from a scale of 1 to 5.

beer_name: Name of the beer that was reviewed.

beer_abv: Beer's alcohol by volume, in percentages, is the standard measurement for the strength of the beer. The higher the abv, the stronger the beer.

Measurements:

It is vastly important to identify the need for this type of information and how the use of beer reviews can help the consumer base. When working with this data it is vital to remember that these reviews come from a consumer to consumer relationship and are of the sole purpose of helping others and sharing opinions. Therefore it is important to recall that the website is dictated by the consumer and that all they want to spread understanding and get others to learn about new products. So it is important for us to realize the significance of each review and how it determines the engagement of the consumer base. We will need to measure the engagement of the consumers alongside the performance of the distributors.

Methodology:

For research question 1, we are trying to identify the difference between beers and how that difference plays into the overall rating of each beer. From this information it should be

possible to determine the best and worst of what is to be offered on the market. This information could prove vital to the casual beer drinker as well as the avid fan. By providing a streamlined test of quality it could make the job of the consumer that much easier to see if what they are buying is of value. For this model we should use our structured data to perform a regression analysis of the data to identify trends.

For research question 2, we are trying to find what beer the consumer will review next. We will want to use our structured data to perform tests of prediction. By identifying specific users from their usernames we will be able to see how each individual reviews and determine patterns in their decision making. This will also allow us to develop ideas of how well a user will rate a beer and see how harsh they are in their grading.

For research question 3, we are trying to find how much variability exists within each beer style. By looking at the specific styles that are in the dataset it will become possible to see how each specific beer within each style has consistent data. The scores of the beer attributes will prove to be a valuable piece of information when determining the variability in styles.

After making all corrections and cleaning the data, regression analysis modeling will be done for making predictions on this problem. Regression models that will be incorporated to test/compare are Multiple Linear Regression, Decision Tree Regressor, Random Forest Regressor, and Support Vector Machine Regression.

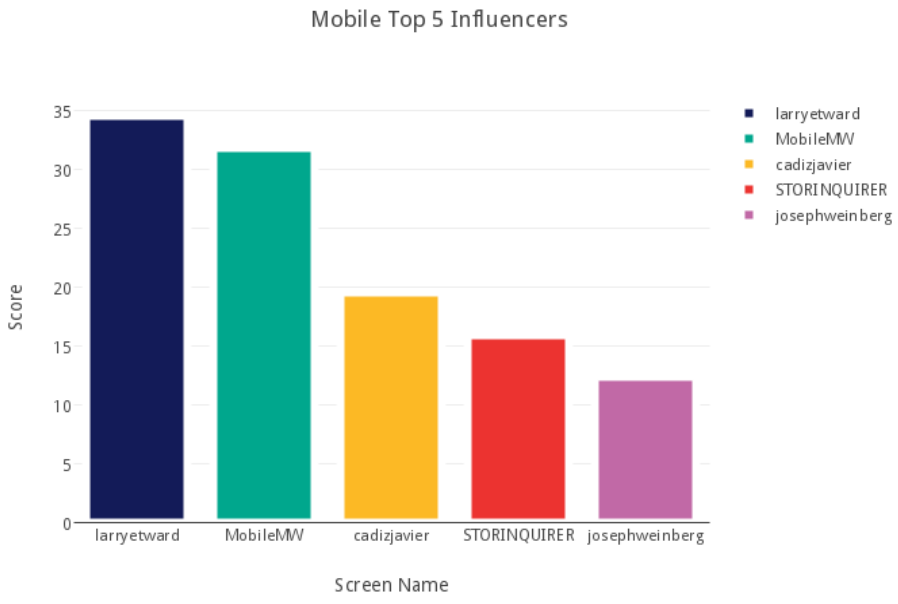
Computational Methods & Outputs:

The method that is best is using regression for our research questions. With regression, methods that can be used are adjusted R squared measuring the goodness of fit, root mean squared error along with root mean squared log error to validate which model performs better.

With this type of modeling, we can create accurate predictions from determining the next possible beer that can be reviewed based on the reviewer to predicting the beer score based on the important features.

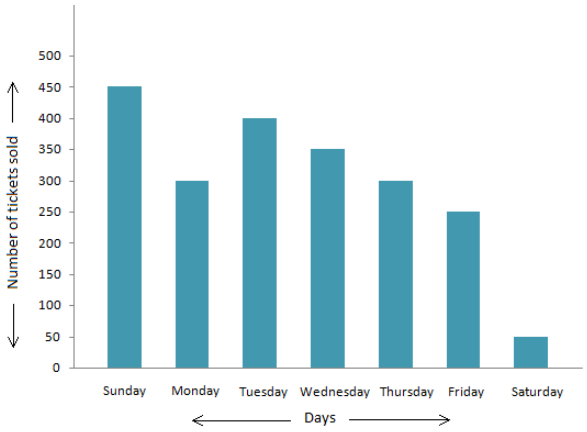
Output Summaries:

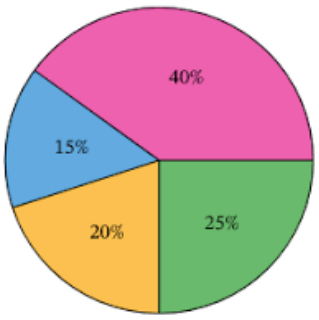
Research Question #1, "What will the beer rating be based on the aspects of beer reviewing (aroma, appearance, palate, and taste) as well as beer style and alcohol by volume?" Breweries will be able to use this information to see what beers are more preferred and what can they improve with other styles of beer. The top 5 beers can be compared based on each of the categories/features and where the best beer tends to be produced. Customers will also be able to see what type of beer they prefer when there are a large number of choices.



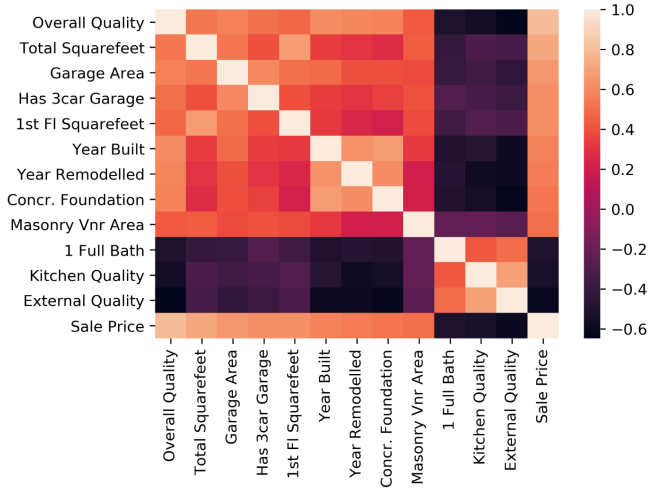
Research Question #2, "What kind of beer can we expect a reviewer to review next?" This analysis will show what types of trends happen when a customer decides to review the next beer. Along with beer, the features can include beer styles, alcohol by volume levels, and even the

specific brewery that is likely to be reviewed next. In this model, the reviewer username will also be included along with the features that are needed for the first research question. Distribution visualizations, count plots and pie charts, will be created to show the various prediction outputs of beer that are likely to come up.





Research Question #3, "How much variability exists within each beer style?" This analysis will show what correlation goes into the beer style. The features needed for the first two questions will be included in this analysis. A heat map will be included to show the correlation in beer styles along with all the other features. This type of visualization can be used to improve in specific categories depending on the beer. The heat map will display correlations of each variable and the higher the correlation, the stronger the relationship is between two variables.



Implementation:

There are so many beers that are being produced these days. With more and more, it can be confusing to pick which one is suitable for their experience. With these insights, we can help businesses improve their products for more cash-flow. For research question 1, “What will the beer rating be based on the aspects of beer reviewing (aroma, appearance, palate, and taste) as well as beer style and alcohol by volume?”, the user can predict what the beer rating could be based on those features' importance and the correlation with the final rating of the beer. This can help customers pick the beer when they would like to try new ones that they have not before. For research question 2, “What kind of beer can we expect a reviewer to review next?”, we can implement the same regression models and use it to predict the beer or even the beer style that will most likely be reviewed next. Customers will want to compare beers that are similar to each other in some standard and knowing this information will help. For research question 3, “How much variability exists within each beer style?”, we will develop another regression model but this time it might be better to use the Tree Regressor. Using those types of models, we will be able to visualize how different the trees make decisions to show the small changes in the features that can lead to different beers.

Literature Review

As more companies use data to discover more insights, businesses are able to create a product using analytics to solve various business challenges. Teams strategize various machine learning models based on the issue at hand and compare based on performance to be used for production. Regression techniques are used to predict outcomes quickly to understand the model's trend. Before implementing models, thorough analysis is required to establish which model will be most effective in solving the issue. There is a benefit in examining multiple perspectives as they pertain to user submitted rating evaluations that consist of various beers that are produced for everyone to enjoy.

There are various techniques that are used for predictive analytics. The combination of statistics and modeling helps make predictions for future outcomes based on old data. Regression analysis is used for most of the problems since it is able to show the trends from the predictions that are likely to occur based on the inputs (Halton 2).

A similar research thesis on Terminology in Beer Reviews was conducted by Malin Norman. One of the goals for the project was to decide what features are important for identifying and describing beers using 27 online reviews from leading beer magazines with 3 reviews for each beer style. The features used were "appearance, aroma, flavour, mouthfeel, and overall impression" (Norman 2). Based on the features that were provided, Norman knew that everyone would react differently so there must have been another way to conclude what features are important for the rating. It was also important for a reviewer to know the correct words for describing the beer which will lead to more sales in the business. Norman found that some reviewers wrote well thought of reviews but sometimes it does not make sense to the average customer. The reviews will have complex phrases to describe it that not many understand. Beer

needs to be rated in a simpler way so that everyone can understand how it will be based on their tastes. The researcher was able to create a Beer Flavour Wheel to help everyone understand the review by using better phrases about the beer. The wheel showed various descriptor categories and each one had a sub category. Using the data from online reviews, researchers will be able to conclude that phrases and aspects will convey a better rating system for each beer style.

By looking at the second research question and identifying how online product reviews can determine outcomes in the future it is important to look at sources like that of Floyd et al. that dive into how online product reviews affect research sales. Floyd and team took a sample of 26 empirical studies and from that derived 443 sales elasticities in order to determine how sales were affected by the review put online (Floyd, 2014). In the procedure of this study, while a wide variety of companies were studied, reviews from a beer review website named Ratebeer.com were studied and the researchers found 5 elasticities in that data, based off of a study done by Clemons et al. that will be covered next. The analysis of the data related to the sales growth rate as well as Sales rank and gross revenues. Concluding the procedure it was found that with reviews there does exist a correlation between reviews and sales and that includes in the beer space. By looking at the data examined in the study conducted by Floyd and team, there is a relationship between online product reviews and the way consumers purchase.

Related to the work in the Floyd study, the work of Clemons et al. on the hyperdifferentiation and the relation to the craft beer industry reviews. Clemons research mainly focuses on hyperdifferentiation and resonance marketing which relate to how firms can produce almost anything that appeals to a consumer and market toward an informed consumer that will purchase products if they truly want the product respectfully (Clemons, 2014). In Clemons' research it was found that companies that diversified and produced products that covered a wide

consumer base that they would perform better than companies stuck in their own niche. This relates to research question #2 since it can provide insight into how consumers will rate specific breweries based on the diversity of their product base and how well they market to those consumers that are interested.

When it comes to beer variability, the study conducted by Mosher and Trantham examines the SRM or Standard Reference Method to demonstrate differences between color and visibility in different styles of beer. The research provides data into the characteristic difference between beer styles such as Porter, Amber Ale, American Lager, and Hefeweizen. Research on such distinctions develops understanding of what characteristics determine beer style and how reviewers can determine which beer is similar to that of one previously reviewed.

Research Questions

This is a dataset that looks at around 1.5 million reviews from BeerAdvocates that rank specific beers on their aroma, appearance, palate, and taste. Along with those specific ranking categories, the data provides an overall score out of five for each beer. The dataset also contains information such as brewery identification, brewery name, review profile name, beer style, beer name, beer abv (Alcohol by Volume), and beer identification that will prove to help find trends within the dataset.

QUESTIONS

1. Which brewery produces the most beer?
    a. Looking at which brewery is most popular when it comes to producing a beer.

2. Which beer is more preferred based on each feature (aroma, appearance, taste, palate, etc.) and overall?
    a. This will take a look into the question of what beer is best and will do so by looking into how certain beers perform across the board and determine how each of the traits examined in the reviews play into overall score.

3. How does the aspect of the review relate to the abv present in the beer?
    a. Look to see how each aspect of the beer profile results in the abv of the beer and to see if any trends become apparent with certain types of beers and at specific abv levels.

4. Which of the aspects of beer reviewing (aroma, appearance, palate, and taste) have the largest impact on the success of the beer?
    a. Looking at the correlation between each feature to see what determines the score most of the time.

5. How much variability exists within each beer style and how do the other aspects of the beer review affect the overall rating of each beer style?
    a. Look at how beer styles are similar and different within the aspects of the rating system (aroma, appearance, palate, and taste) as well as looking at how specific reviewers of said styles can impact the trend of the style. How much impact does each specific reviewer have on the overall score of a beer and how accurately can the reviewer get to the average?

Exploratory Data Analysis

For this project, we gathered data from an online Data Science and Machine Learning Community called Kaggle which originally consisted of over 1.5 million observations of structured data. The dataset was downloaded from the website and it listed the beers that were reviewed by various beer connoisseurs from the BeerAdvocates community. When looking at our data it became obvious that we did not need every variable in order to fulfill the tasks we wished to complete so we cleaned up the dataset. We removed the values of "brewery_id", "review_time", and "beer_beerid" since they were not helpful in this particular analysis. After this step we dropped any duplicate entries as a precaution to not sway the analysis. Finally we ran a check for null instances and found a total of 68148 null values. After finding the total amount of null values we removed every row from the dataset with at least one null value. After making appropriate changes, the data was then condensed to 151648 instances and 12 variables for the our research and it is as follows:

brewery_name: The names of the breweries that produced beers reviewed.

review_overall: Overall score/rating of the beer from a scale of 1 to 5.

review_aroma: Rating of the beer's aroma from a scale of 1 to 5.

review_appearence: Rating of the beer's appearance from a scale of 1 to 5.

review_profilename: Unique profile name of each reviewer in the BeerAdvocates Community.

beer_style: Names of the unique style the beer classifies as.

review_palate: Rating of the beer when it is on the roof of the mouth from a scale of 1 to 5.

review_taste: Rating of the beer's taste from a scale of 1 to 5.

beer_name: Name of the beer that was reviewed.

beer_abv: Beer's alcohol by volume, in percentages, is the standard measurement for the strength of the beer. The higher the abv, the stronger the beer.

The first thing we did was to gain a grasp of the amplitude of the reviewers and not only the reviewer base as a whole but rather which individual reviewers outperformed the rest of the pack. We set out to accomplish this by sorting the individual reviewers by their profile name and then sorting the data in order to create a count of the reviews done by each profile. This data was then converted into a bar graph that demonstrates the top 10 reviewers by profile as seen in appendix A. Reviewers like northyorksammy, mikesgroove, and BuckeyeNation define a clear gap between them and the rest of the top 10 demonstrating the variation at the top end of the data. By identifying the top reviewers it is good to look at their overall tendencies to determine patterns in determining good beers from bad.

We wanted to accomplish a way to identify the trends of the specific components that went into the overall review score and how the 151648 instances compared to each other from the data. This output can be seen in appendix B where there are bar graphs presented for each of the four components of the review (review_aroma, review_appearence, review_palate, review_taste) as well as the overall scores (review_overall) and alcohol by volume levels (beer_abv). From this data it is possible to see the trends of the total reviews with the majority of reviews coming around 4 for all of the components and overall score. This combined with the fact that all of the bar graphs display a distinct skew to the left of the data shows that you can expect reviews to consist more of high scores rather than low scores meaning that there should be less variability among the top reviewed items. When it comes to ABV however, this data is not determined by the reviewer and has more of a say on the beer industry as a whole and demonstrates a clear bias toward lower ABV levels below 10%.

The next question we asked was that of which breweries were reviewed the most and how they compared to the rest of the top choices among reviewers. To help demonstrate the impact of the top breweries on the market we used a bar graph of the top 100 breweries reviewed and sorted high to low in order to get a good visualization of the difference between the top breweries as seen in appendix C. What we found was that not only was Samuel Adams the most reviewed brewery, it had around 5000 more reviews than the next best and over 13000 more than the fifth best. While there does exist a clearly high amount of breweries with a significant amount of data, Samuel Adams displays a clear hold on the market of brewery reviews and that information could prove vital to the success of their beers come later in this investigation.

After looking at the numbers of the breweries an obvious next step for us was to perform a similar analysis in terms of approach but instead with the top 100 beers reviewed rather than breweries as seen in appendix D. By looking at the data in the similar format it was easy to see a comparison between the spread of the top end of the data for both components of the product side of the equation. By looking at the bar graph for the top beers the distribution appears to be more uniform with a less steep decrease from the very top end with beer like 90 minute IPA and Old Rasputin Russian Imperial Stout to the rest of the pack. While those beers stand out, others like Samuel Adams Boston Lager, Octoberfest, and Summer Ale all making the top makes sense due to the success of the brewery as a whole.

Next we wanted to find the popular beer styles by once again using a similar strategy to the bar graphs of the breweries and beers to show a consistent pattern presented within the presentation of the data. This graph can be seen in appendix E, and shows how there exists a clear bias among reviewers toward the American IPA and American Pale Ale. With these

specific types of beers being reviewed so much more than the other top choices it shows a clear saturation of those specific beers and therefore a good base to pull consistent data from.

Following the understanding of the dense portions of our dataset we set out to explore the extremities by focusing on the beers with the very high ABV values and how they compared to each other. As seen in appendix F, the highest of ABVs extends all the way out to 57% alcohol by volume and while this value demonstrates a clear outlier compared to the rest of the data set, the bottom end of the graph depicts beers at around 20% ABV showing that the variance between the top and the bottom is not necessarily as big as it seems with the inclusion of the Schorschbräu Schorschbock 57%.

Next we thought it important to finally compare the total of our variables against each other in the form of a heat map as seen in appendix G. In this heat map we can determine how each variable interacts with each other and determine where the highest amount of correlation exists. While this data does not prove any causation between each variable and the success of the overall score, it does give us a good starting point. By looking at the values side by side we can determine the best fit of the model and see how the values trend together. When examining the output you can see that there is a correlation between taste and palate in regards to the overall rating with their values of 0.79 and 0.7 respectively. It was also intriguing to see the connection between the values of taste and palate as well as taste and aroma at 0.73 and 0.71 respectively. Lastly based on these correlations there appears to be no real connection between ABV and the scores of each category of beer reviewing on BeerAdvocate. This data can help to prove the connection between the sense when it comes to beer reviewing and how creating an overall sensory experience could lead to a better perception of the beer.

Next we thought it would be really important to see how scatterplots of the data could help to determine trends within the dataset and we accomplished that goal with the graph in appendix H. While the data presented in the graph can lead to little information regarding trends within specific aspects of beers it can show that there is not one distinct way to easily predict each beer based on one aspect alone. One thing that was of note was the categories that were correlated in the heat map did not show any definitive signs of difference which may show that while they can be correlated most of the time that is not always the case.

From this information no conclusion can truly be drawn between beer review aspects and the overall score of the beers as well as the tendencies of each reviewer. By identifying key connections and determining good starting places the data is more organized and easier to pull information from in the rest of the data exploration. While there is still no definitive conclusion it can be assumed that taste will have a higher impact on overall score than appearance and that ABV does not correlate with the scores of a review.

Methodology

RQ 1: **What will the beer rating be based on the aspects of beer reviewing (aroma, appearance, palate, and taste) as well as beer style and alcohol by volume?**

Document Classification

To predict the rating of the beer, the data needed heavily revolves around the reviewers from the BeerAdvocates community to rate different beers based on different aspects. For this type of analysis, regression techniques are needed to predict the rating of the beer from 1 to 5. To prepare the data, we use the variables that are associated with reviews as the independent variables and the dependent variable will be the overall review rating along with beer style and alcohol by volume. These variables are then fitted using the different modeling techniques listed below.

Other Preparation

Our dataset has approximately 5155 unique breweries reviewed, 44075 different beers reviewed, and 104 various beer styles reviewed. After fitting the data with each technique, we decided to compare the root mean square error (RMSE), root mean squared logarithmic error (RMSLE), and accuracy scores or R^2 for the best model. The RMSE is the measure of how spread out the residuals are from the regression line data points and usually lower values correlate to a better model. RMSLE is using the RMSE log-transformed predictions and actual values, and the lower the value is the better the model fits with the data. The accuracy score or the R^2 measures how well the model fits and the closer the score is to 100; the better the model fits with the data. Mean Absolute Error (MAE) is a measure of errors between paired observations expressing the same phenomenon, and the lower value corresponds to the model

having a better fit. After using various techniques for the best model, we will be able to compare these three measures to find the best modeling technique with the data.

RQ 2: **What kind of beer can we expect a reviewer to review next?**

Document Classification

The first step of this analysis was to analyze the information we saw in the dataset. The dataset contained various features that have the ability to predict the beer that would most likely be reviewed. The variables needed for this analysis were all the review variables that contained ratings from 1 to 5, name of the brewery, beer style, beer alcohol by volume, and name of the beer.

Other Preparation

During the Exploratory Data Analysis stage, we created various visualizations that show which beer was reviewed more frequently than the others. There were less than 45000 unique beers that were reviewed making it a little complicated to know what kind of beer was likely to be reviewed next. Next, we decided that it would be best to use the variables that rank the beers using the review scores to predict the beer that will be reviewed next. Then, we would convert the categorical variables, brewery name and beer style, into numbers which would be incorporated in the model. We chose to focus on review aroma, appearance, taste, palate, and overall, brewery name, beer style, beer alcohol by volume, and beer name for the modeling stage. We will then use the same techniques to find the most accurate model.

Modeling Techniques for Both:

- Multiple Linear Regression

    - Calculates the coefficients of predictor variables to estimate and predict based on the relationship between the dependent variables and 2 or more independent variables.
- Decision Tree Regressor
    - Builds a regression model in the form of a tree structure. It divides a dataset into progressively smaller sections by decision nodes while simultaneously developing a linked decision tree with a leaf node as the end result or prediction.
- Random Forest Regressor
    - Builds multiple regression tree models using the ensemble learning method. It combines the predictions of various machine learning algorithms to make an accurate prediction.

Data Visualizations & Analysis

Research Question #1:

For our first research question, we chose to look at the trends and how the model fits when predicting the overall review score of the beer. We selected to use a review of the aroma, appearance, palate, and taste as well as the style of the beer and the percentage of beer alcohol by volume after choosing all the variables required for the analysis. Using these features, we were able to fit the data into three machine learning models and compare how well it is able to predict using various metrics shown below.

Multiple Linear Regression Analysis

R^2	Adjusted R^2	MSE	RMSE	RMSLE	MAE
0.6689138828589304	0.669959809476504	0.1705700471704114	0.4130012677588428	0.0946847596142316	0.3099343490966088

Equation:

Review_overall = 0.52993541 + 0.0773937931(review_aroma) +

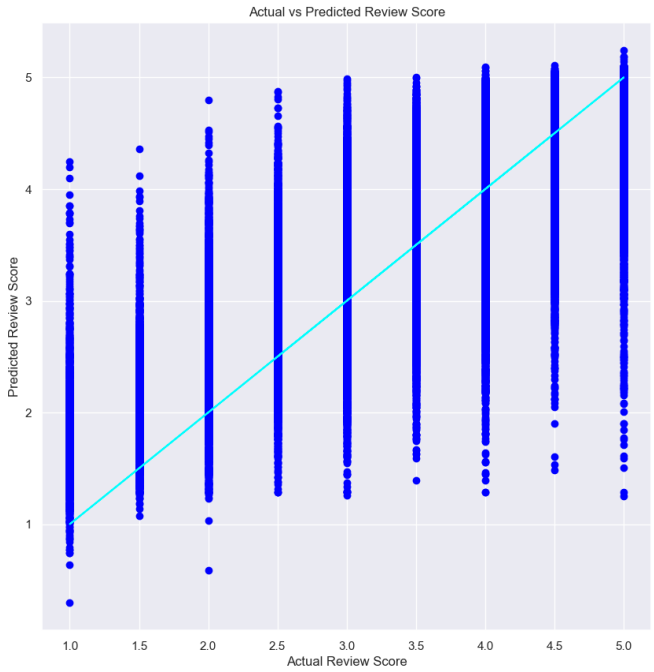
0.0480283932(review_appearance) - 0.000225142889(beer_style) +

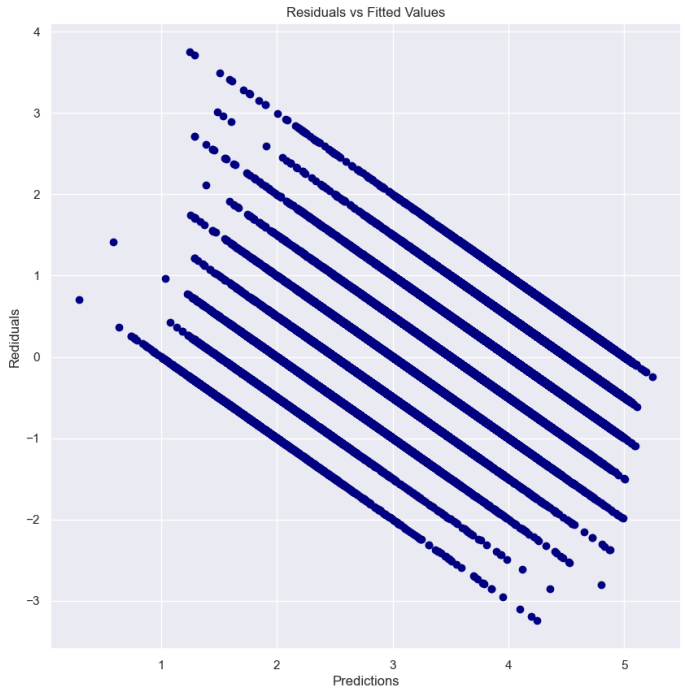
0.270770670(review_palate) + 0.553787327(review_taste) - 0.0418301216(beer_abv)

Intercept	When all variables are held constant then the overall will increase by 0.52993541
review_aroma	When review_aroma increases by one unit and all other variables are held constant, then the overall will increase by 0.0773937931
review_appearance	When review_appearance increases by one unit and all other variables are held constant,

	then the overall will increase by 0.0480283932
beer_style	When beer_style increases by one unit and all other variables are held constant, then the overall will decrease by 0.000225142889
review_palate	When review_palate increases by one unit and all other variables are held constant, then the overall will increase by 0.270770670
review_taste	When review_taste increases by one unit and all other variables are held constant, then the overall will increase by 0.553787327
beer_abv	When beer_abv increases by one unit and all other variables are held constant, then the overall will decrease by 0.0418301216

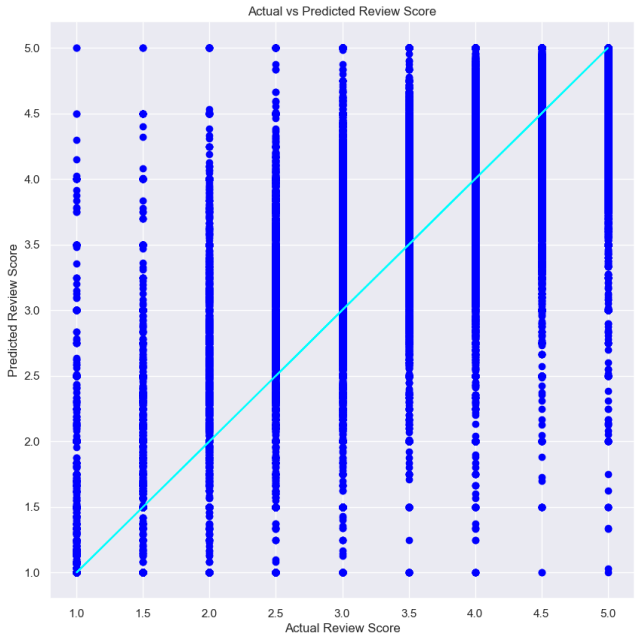
As seen through the results of the multiple linear regression analysis, an equation can be created to identify the overall value of the review given the aroma, appearance, style, palate, taste, and ABV reviews. By diving into the values of the equation you can begin to see the effect of each variable on the overall outcome of the review and therefore determine which values have the greatest influence on the outcome. It appears that the review for palate and taste have the most impact on the overall review since the multipliers are the largest out of the other variables. By contrast, beer aroma along with appearance still have a positive effect on the overall review but that impact is far less than that of palate and taste. Beer style and ABV actually have a negative effect on the overall score and therefore the higher the rating for ABV and style the lower the overall score will be in general.

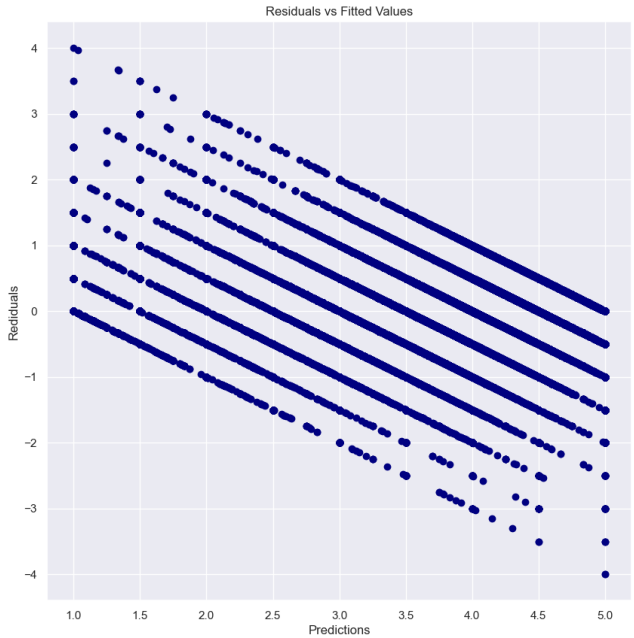

Actual vs Predicted Review Score
Predicted Review Score
Actual Review Score
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
1
2
3
4
5


Residuals vs Fitted Values
Rediduals
Predictions
1
2
3
4
5
−3
−2
−1
0
1
2
3
4

Decision Tree Regression Analysis

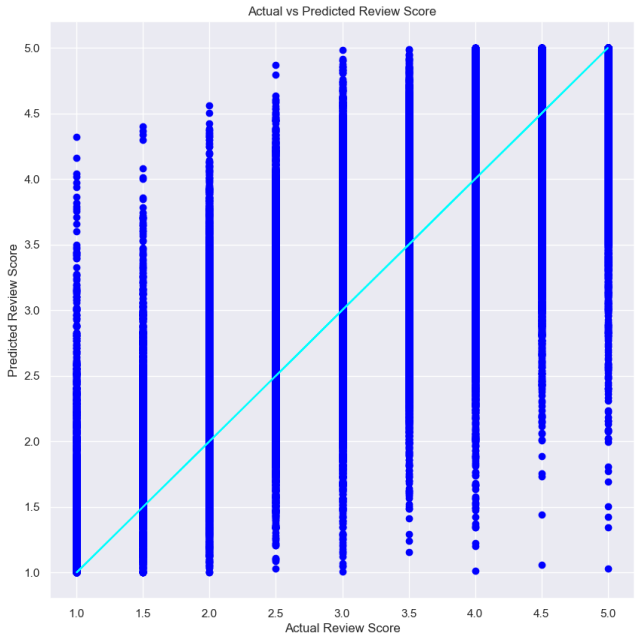
R^2	Adjusted R^2	MSE	RMSE	RMSLE	MAE
0.5461928103235448	0.7569875294107073	0.23379389754479973	0.4835223857742263	0.1123013936954261 6	0.3488029241110173

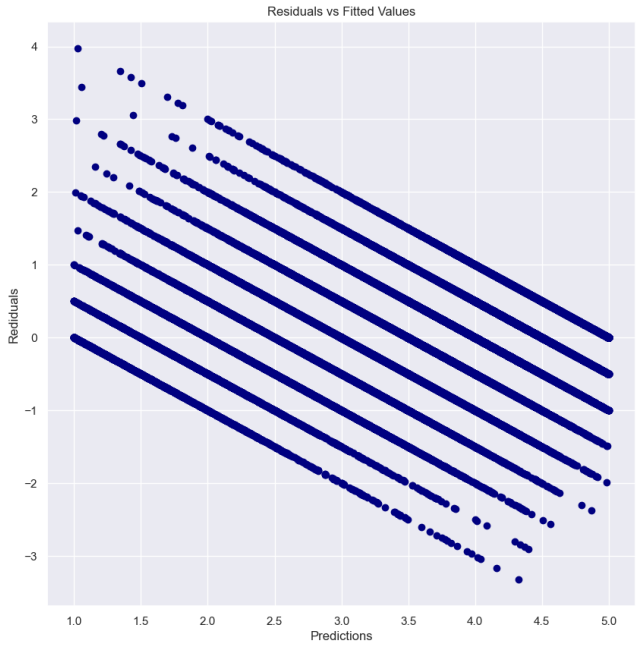




Random Forest Regression Analysis

R^2	Adjusted R^2	MSE	RMSE	RMSLE	MAE
0.6330788484838392	0.7629945711062868	0.18903165938324917	0.43477771261099524	0.09903692923422806	0.3271555506764976





Following the completion of the Multiple Linear regression analysis, Decision Tree regression analysis, and Random Forest regression analysis, it can be seen that the highest adjusted R^2 value came from the Random Forest output while the remaining statistics were all optimized in the Multiple Linear regression analysis. However, despite this there does not exist a strong correlation between any of these outputs and the determination of the beer rating based on these variables. The value of 0.6689 for the optimal R^2 value from the multiple linear regression represents the proportion of variance between the independent and dependent variables. The value of 0.6689 does not represent a very strong relationship between the independent and dependent variables. With an adjusted R^2 value of 0.7630 there is a level of correlation between the variables and the accuracy of the random forest regression model. The mean squared error value of 0.1706 from the multiple linear regression is the average square of the errors and the lower the value the better. The root mean squared error of 0.4130 from the multiple linear regression is another value that is in its optimal range closer to zero. The root mean square log error of 0.0947 from the multiple linear regression is another value that determines the error values of the function but also takes into account the log of the root of the squared errors. Finally, the mean absolute error of 0.3099 from once again from the multiple linear regression measures the errors between paired observations. Overall, the values derived from the multiple linear regression led to the most accurate answers and therefore the outputs from that action give the greatest prediction of the overall scores. Two Residual Plots were also created to evaluate each model. The first plot shows the Actual vs the Predicted Review Score while the second plot shows the Residuals Vs Fitted Values. The first plot shows us that the values are varied throughout. The Residual vs Fitted Values plot displays a downward trend for all of the models

tested. Therefore, we can conclude that the all 3 models for the first research question are not statistically significant.

Research Question #2:

For our second research question, we chose to look at the trends and see if the model is able to predict the next beer that will be reviewed. We decided to select the same variables from the previous analysis along with the brewery name to predict the beer name. Using these features, we were able to fit the data into the same three machine learning models and compare how well it is able to predict the beer that could possibly be reviewed next using the same metrics shown below.

Multiple Linear Regression Analysis

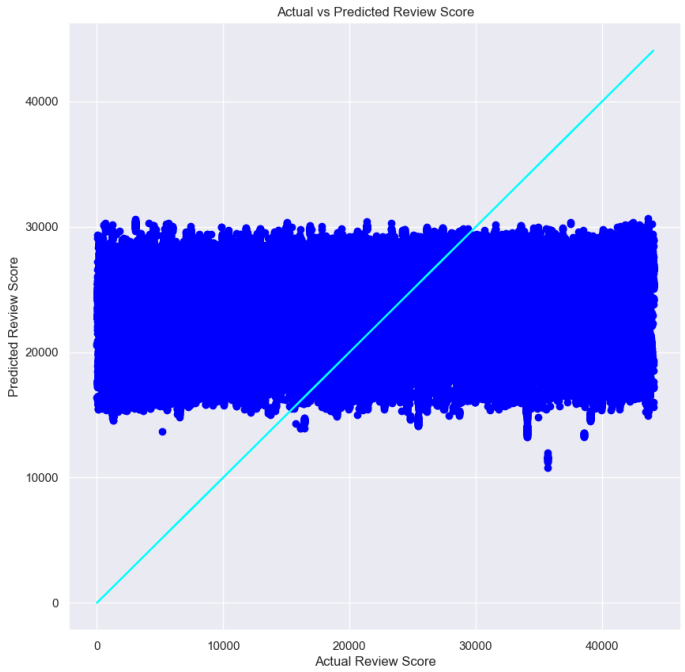
R^2	Adjusted R^2	MSE	RMSE	RMSLE	MAE
0.074733245434403440085	0.075171236640684459	151143827.5767823	12294.056595639304	1.0249548221783888	10614.10508849166

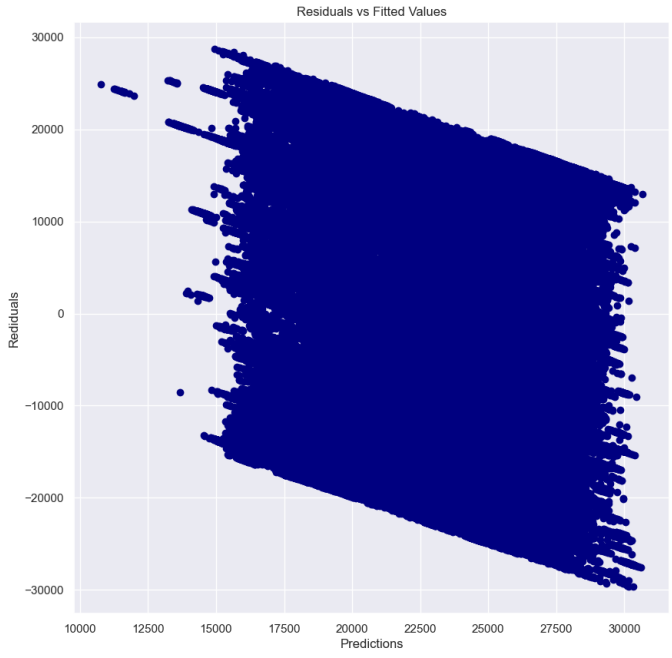
Equation:

Beer_name = 16706.63626683 + 2.16250136(brewery_name) + 61.08995313(review_overall) - 237.5875561(review_aroma) + 257.8412274(review_appearance) + 30.31623794(beer_style) + 16.42168507(review_palate) - 12.67344141(review_taste) - 199.51191461(beer_abv)

Intercept	When all variables are held constant then the beer_name will increase by 16706.63626683
brewery_name	When brewery_name increases by one unit

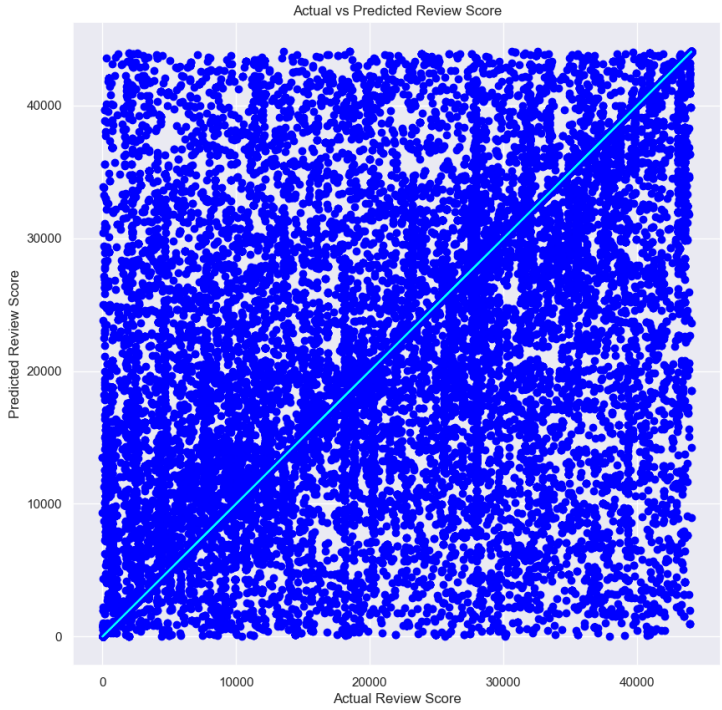
	and all other variables are held constant, then the overall will increase by 2.16250136
review_overall	When review_overall increases by one unit and all other variables are held constant, then the overall will increase by 61.08995313
review_aroma	When review_aroma increases by one unit and all other variables are held constant, then the overall will decrease by 237.5875561
review_appearance	When review_appearance increases by one unit and all other variables are held constant, then the overall will increase by 257.8412274
beer_style	When beer_style increases by one unit and all other variables are held constant, then the overall will increase by 30.31623794
review_palate	When review_palate increases by one unit and all other variables are held constant, then the overall will increase by 16.42168507
review_taste	When review_taste increases by one unit and all other variables are held constant, then the overall will decrease by 12.67344141
beer_abv	When beer_abv increases by one unit and all other variables are held constant, then the overall will decrease by 199.51191461

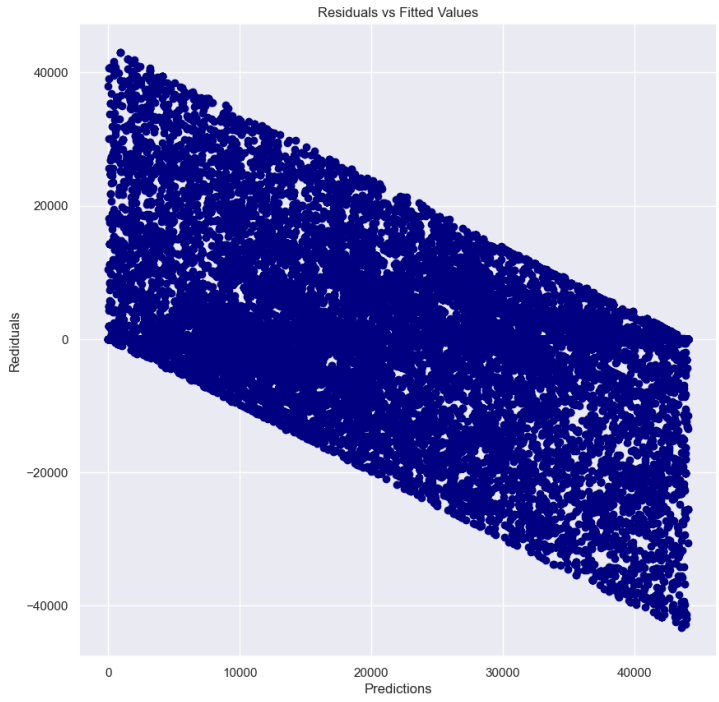




Decision Tree Regression Analysis

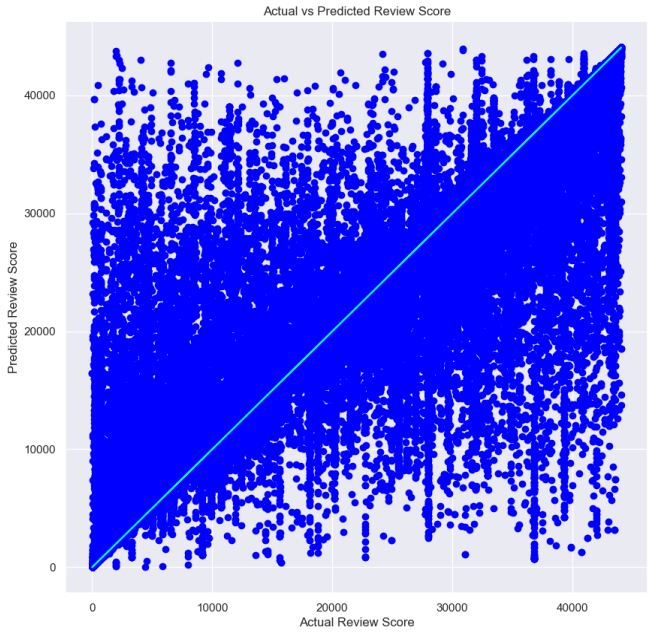
R^2	Adjusted R^2	MSE	RMSE	RMSLE	MAE
0.910619549059959	0.9677841560969072	14600441.871446088	3821.0524559924765691	0.3280434559465691	782.0841951842308

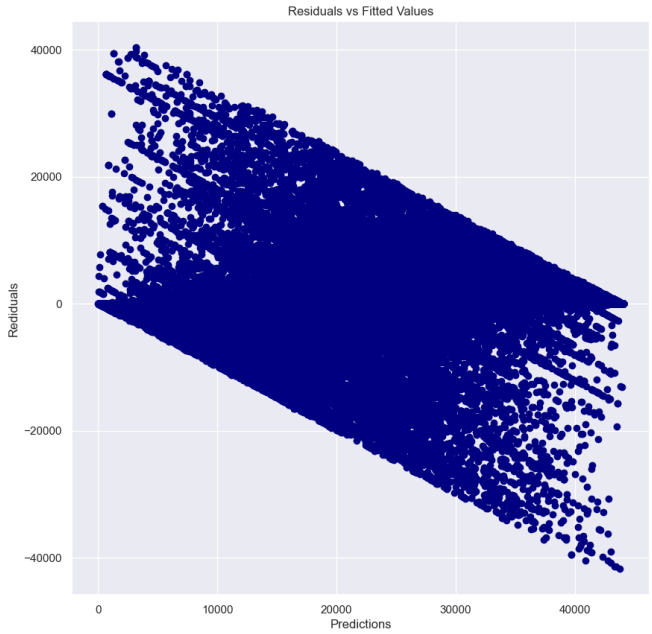

Actual vs Predicted Review Score
Predicted Review Score
Actual Review Score
0
10000
20000
30000
40000


Residuals vs Fitted Values
Rediduals
Predictions
40000
20000
0
−20000
−40000
0
10000
20000
30000
40000

Random Forest Regression Analysis

R^2	Adjusted R^2	MSE	RMSE	RMSLE	MAE
0.9356833813427162	0.9684650890606377	10494061.27479847	3239.453854401768	0.3037829792006356	822.0198831762157





The value of 0.9357 for the optimal R^2 value from the random forest regression represents the proportion of variance between the independent and dependent variables. With an adjusted R^2 value of 0.9685 there is a level of correlation between the variables and the accuracy score of the random forest regression model. The mean squared error value of 10494061.2748 from the random forest regression is the average square of the errors and the lower the value the better. The root mean squared error of 3239.4539 from the random forest regression is another value that is in its optimal range closer to zero. The root mean square log error of 0.3038 from the random forest regression is another value that determines the error values of the function but also takes into account the log of the root of the squared errors. Finally, the mean absolute error of 782.0842 from the decision tree regression measures the errors between paired observations. The same two Residual Plots were created to evaluate each model once again to see if they are statistically significant. The first plot shows us that the values are varied throughout. The Residual vs Fitted Values plot displays a downward trend for all of the models tested. Therefore, we can conclude that the all 3 models for the second research question are not statistically significant.

Analysis

The CSV file contained over 1 million ratings of beer that was reviewed from BeerAdvocates community and it was loaded into Python for data analysis for more insights. We decided to eliminate brewery_id, review_time, and beer_beerid since they were needed for the analysis and model. Next, we printed the information from the dataset that listed the count of non-null values and data types of each variable for more understanding. We found out that there were 68,148 rows that had missing values and chose to remove them. Next we created multiple histogram plots for all numerical variables to show the distribution of all the values recorded. We

wanted to see the breweries that were reviewed the most so we chose to make a bar graph that displays the top 100 breweries reviewed and another bar graph displaying the top 10 breweries reviewed to visually see the distribution better. Same type of bar graph was created to show the top 100 beers and top 10 beers reviewed. Another variable we wanted to analyze was the reviewers. A horizontal bar graph was created to show the top 10 reviewers and the most beers someone reviewed was more than 5000. We also decided to review the beer styles and to find out the most popular ones. Another bar graph was created to display the most popular beer styles that were reviewed. Out of the 104 different beer styles, the most popular one was the American IPA with more than 100k reviews. After looking at the beer_abv variable, we would find the highest beer alcohol by volume. Most of the beers in the data had less than 20% alcohol by volume so we created a horizontal bar plot to show the different beers. The highest alcohol by volume was the Schorschbrau Schorschbock which was at 57%. The last thing a part of the analysis was creating a heatmap to show the correlation between the all numerical variables. We specifically wanted to see if the review_overall variable was correlating well with the rest of the variables. We found that beer_abv had the lowest correlation with less than 0.33 with the rest of variables while the correlation of the other variables was 0.5 or higher.

For the first research question, we gathered the variables to perform data preprocessing to prepare the data for the modeling stage. All the variables were float values except for beer_style which was categorical. Categorical variables are not recognized in a linear regression model so the idea was to associate the categorical values to a respective number. Using the LabelEncoder package, the inputs in the beer_style variable were converted to a respective number also known as One Hot Encoding so dummy variables would not have to be created. Now, we were able to fit the data using multiple linear regression, decision-tree regression, and random forest regression

to find the best model that is able to predict the overall review score. In order to evaluate the predictive accuracy of the models, we wanted to split the data into a training set and testing set. We decided to use 80% of the data as the training set and the remaining 20% as the testing set after shuffling the dataset 3 times for randomization. The independent variables that were included for all the models were as follows:

Variable	**Definition**
review_aroma	Rating the beer's aroma on a scale from 1 to 5.
review_appearance	Rating the beer's appearance on a scale from 1 to 5.
beer_style	Style of the beer that was reviewed.
review_palate	Rating beer's palate on a scale from 1 to 5.
review_taste	Rating the beer's taste on a scale from 1 to 5.
beer_abv	Alcohol by Volume percentage in the beer that was reviewed.

The training set was made to fit trends in the dataset with each machine learning model and test data was used to predict so we can compare the actual values versus the predictions. After running all 3 algorithms, we compared the R^2, RMSE, MSE, MAE, and RMSLE values to find the best model. The multiple linear regression model had the best values of coefficient of determination (R^2), mean squared error, root mean squared error, root mean squared logarithmic error, and mean absolute error. The decision tree regression model had the best adjusted R^2 which is also the accuracy of the model while the random forest turned out to be the worst out of the 3. We also chose to create multiple residual plots to compare the predicted and actual values of the overall review score. After comparing the plots, we noticed that there is

no correlation between the predicted and actual review score. Therefore, we can conclude that the models are not useful.

For the second research question, we gathered the same variables from the first question and added one more independent variable to perform data preprocessing to prepare the data for the modeling stage. The independent variable added is listed below:

Variable	**Definition**
brewery_name	Name of the brewery where the beer was manufactured

Now, there were 2 new categorical variables, brewery_name, and beer_name, in the dataset and we needed to change these to numerical values in order to work for linear regression. The LabelEncoder procedure was done once again to convert those inputs to a respective number. After preparing the data, we started to build the models. We used beer_name as our dependent variable while the rest were independent variables. We split the data into a train set which was 80% of the data and a test set at 20%. The same models and metrics from the first research question were used to determine the best model. After running the models, we found out that the multiple linear regression model had the poorest metric values when compared to the rest of the models in predicting the next beer that will be reviewed. The Decision Tree Model had the best Mean Absolute Error while the Random Forest Model had positive scores for the rest of the metrics. We also decided to take a look at a residual plot by creating a scatter plot. The plots are referenced in their respective sections of the appendix (H-M). After interpreting and comparing the residual plots, we arrived at the conclusion that all three models for the second research question are not statistically significant.

Ethical Recommendations

The exploration of beer review data has led to the understanding that the use of the data and the result that we collected could have large effects on the outcome of both the consumer and producers. With the introduction of new data on the outcome of review we can reasonably see a decrease in the variability seen in the beer market. If a particular formula can be found that leads to more traffic and attention it would make sense for many different brands to convert to a particular singular model. We want to protect individuality that leads to such a diverse selection of customers, the results of a prediction can possibly cause companies to change their formulas to try and conform to a norm. This can lead to a high number of similar beers that can flood out the market and create a bad customer experience.

The world of beers has a rich and continued history in the cultures of many countries around the world. By introducing a system that can predict reviewers choices it can cause for much of the history behind beer making to vanish if a sustainable pattern can exist between consumer tendencies and brewing technique. If, per say, that a particular style of brewing that is only done in small parts of the world were to receive less attention from the prediction model to others it could cause the consumer base to completely ignore that rare style and therefore run it out of the market for good. In every case, the thing that will help the longevity of the market is to keep the individuality that has resulted in the market for multiple online beer reviews. The whole idea of an online space for beer reviews is to promote the variability in the market and to celebrate what makes each beer special to each drinker.

The potential risks for not releasing this information include a severe loss in revenue for breweries and the potential for not being able to find genres and styles that consumers could be very drawn to. The release of the information has the potential to help breweries in revenue due

to not only help promote their successful beers but it can also help with the introduction of the market to their other products as well. There is importance for certain beers to fail for some of these companies to help determine how they want to move forward so if they are forced into consistent mold there will be no way for smaller brands to stand out and stand up to the bigger breweries. There is a distinct need for the reviews for smaller breweries in order to get the word out on their products but a string of bad reviews can do the opposite and effectively tank their product. At the end of the day, if companies are able to learn and grow from the information studied then there is a need for the information to be studied and used.

Challenges

When conducting this research there always exists the challenge of finding significance within the model implemented to test the data. With as little quantitative data as we are working with in this data it is difficult to have lenience when measuring regression due to the small number of variables. Without a large number of variables to work with when attempting to find a regression relationship it can be difficult to find enough significant variables in connection to the dependent variables.

Another issue that appeared quite frequently was getting the data to be shown in the way we wanted it to be displayed. In many instances it was hard to realize our thoughts on how the data should be presented in order to give the best visualization of the information. When plotting the information it can be hard to determine the best possible graph or plot to demonstrate the differences that we want the data to show. With that being said it could take a long time to develop proper graphs in many cases which could also not demonstrate the data as well as we would still want.

Another challenge that we faced was trying to find new metrics to compare in the models that would help to show the effectiveness of the data. While we had a good understanding of how the data needed to be connected in order to show our theories, it became difficult to find which variables would work together and prove the connectedness of the model.

Recommendations and Next Steps

Overall, we ended up deciding that the models for predicting beer rating based on the features of beer reviewing were not statistically significant due to a wide number of errors. The metrics and residual plots were not strong enough and therefore, we are able to conclude these models are not fit for predicting beer rating and the beer that will be reviewed next. After a thorough research, the regression algorithms used to predict the next reviewed beer were incorrect. Classification algorithms need to be used since we are predicting a categorical variable.

Analysis can always be performed in different ways and to find more insights and this one is no different. If we were to do this analysis again, we would try to find and access more data that would cover sales and create a map of brewery locations to see which brewery receives the most sales. There are also more models that can be tested to find a more accurate outcome. Models we would like to try are Regression with Ridge and Lasso, Support Vector Machine, and Gradient-Boosted Tree Regressors for predicting a beer's rating. We would also like to add more statistical plots to help us properly decide if the models are significant. Quantile-Quantile Plot will help us understand if the residuals are normally distributed. Using different hypothesis testing techniques will also help us understand if our model is significant by comparing the p-values. For predicting the next reviewed beer, we would test by using multiclass - classification algorithms since there are more than 1000 different classes or beers reviewed. Algorithms that could be tested are Gaussian Naive Bayes, K neighbors Classifier, Decision Tree Classifier, and Random Forest Classifier. The algorithms listed are inherently multiclass and can be performed for this type of analysis.

References

Beeradvocate. BeerAdvocate. (1996). Retrieved September 4, 2022, from https://www.beeradvocate.com/

Clemons, Eric, et al. "When Online Reviews Meet Hyperdifferentiation: A Study of the Craft Beer Industry." *Taylor & Francis*, 8 Dec. 2014, https://www.tandfonline.com/doi/abs/10.2753/MIS0742-1222230207

Datadoume. "Beer Reviews." *Kaggle*, Datadoume, 2 Oct. 2018, https://www.kaggle.com/datasets/rdoume/beerreviews

Floyd, Kristopher, et al. "How Online Product Reviews Affect Retail Sales: A Meta-Analysis." *Journal of Retailing*, JAI, 29 May 2014, https://www.sciencedirect.com/science/article/abs/pii/S0022435914000293

Halton, Clay. "Predictive Analytics Definition." *Investopedia*, Investopedia, 11 May 2022, https://www.investopedia.com/terms/p/predictive-analytics.asp

"Mobile Top 5 Influencers." Plotly, 7 February 2021 https://chart-studio.plotly.com/~bpm/253.embed

Mosher, M., Trantham, K. (2021). Beer Styles. In: Brewing Science: A Multidisciplinary Approach. Springer, Cham. https://doi.org/10.1007/978-3-030-73419-0_2

Norman, Malin. “Terminology in Beer Reviews.” *DIVA*, 11 Mar. 2019, https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1294624&dswid=-4377

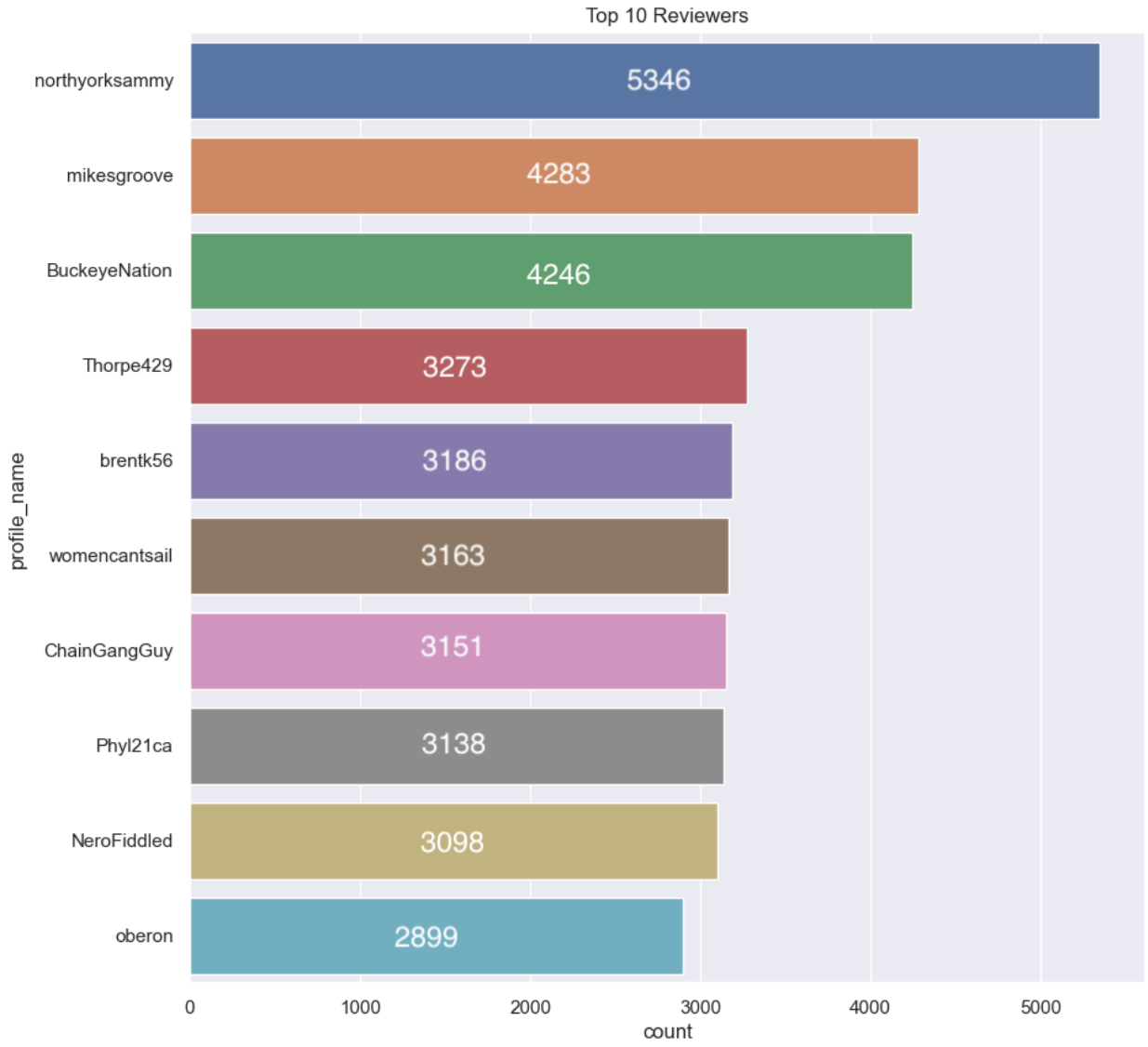
“Pie Chart with Percentages.” Nagwa. 2022, https://www.nagwa.com/en/explainers/245194820905/

Szabo, Bibor. “Basic Seaborn Heatmap”. Medium. 25 May 2020, https://medium.com/@szabo.bibor/how-to-create-a-seaborn-correlation-heatmap-in-python-834c0686b88e

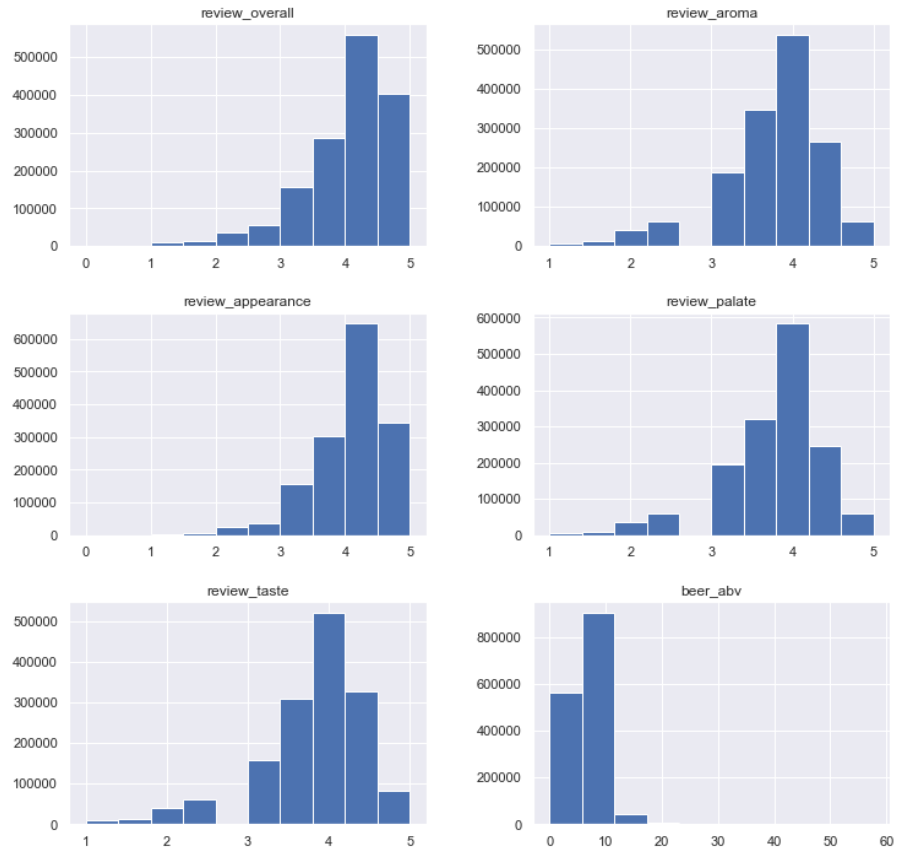
“Ticket Sales Bar Graph.” Math Only Math. 2022, https://www.math-only-math.com/worksheet-on-representing-data-on-bar-graph.html

Appendix

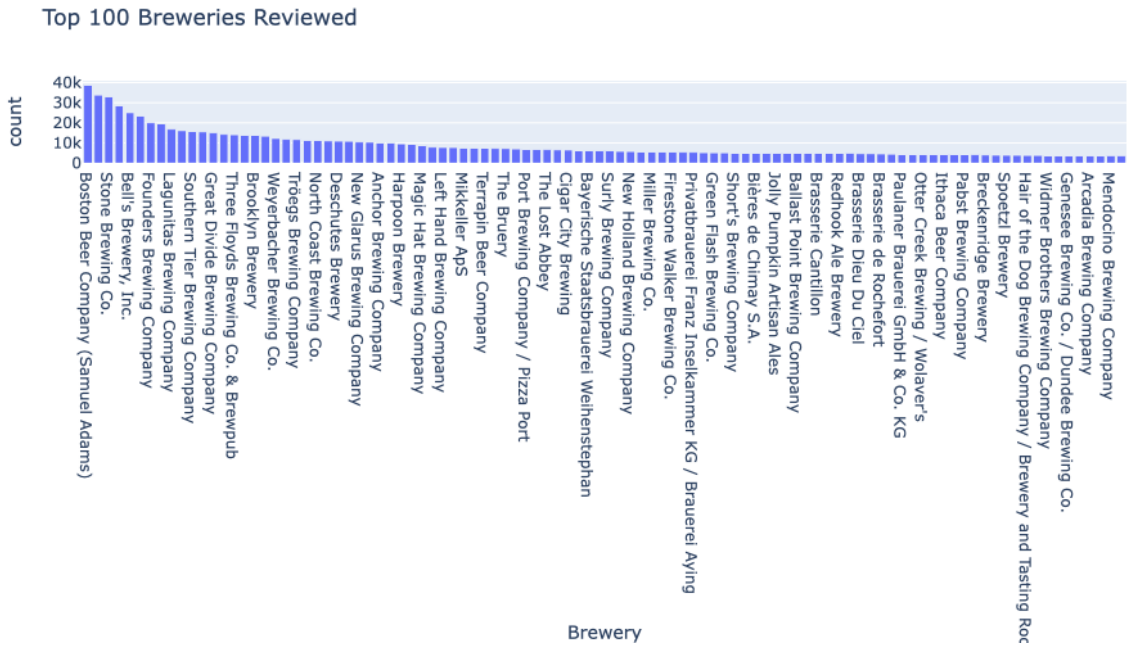
A. Top 10 Reviewers

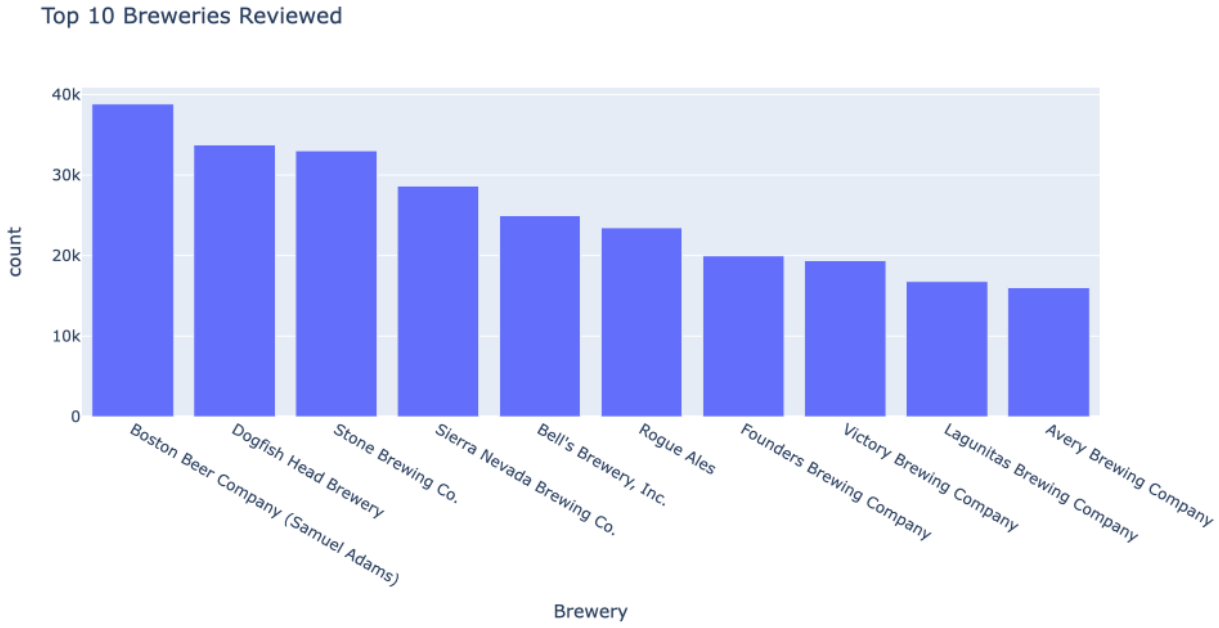


B. Distribution of Numerical Columns

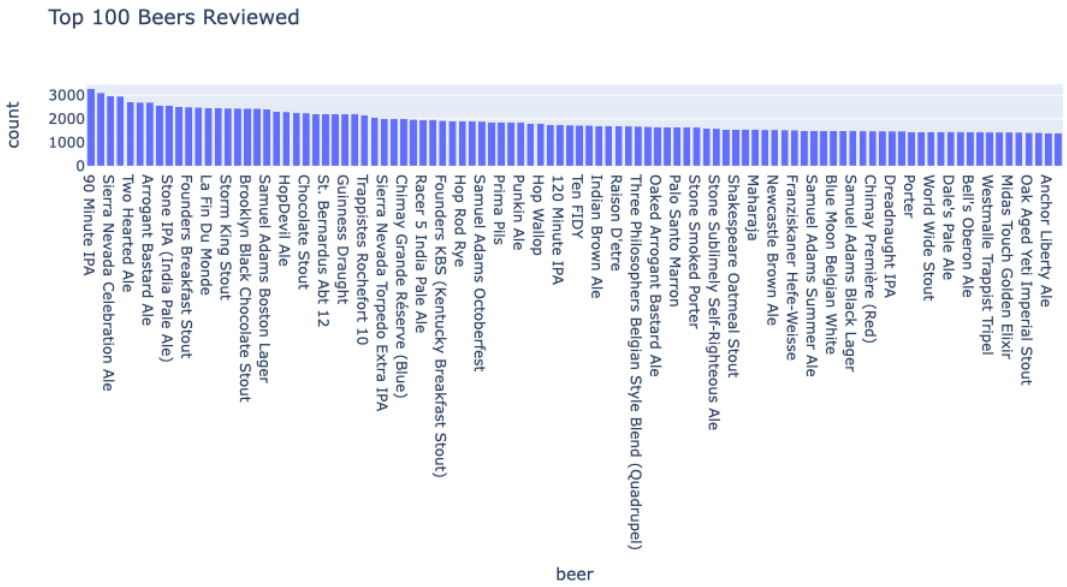


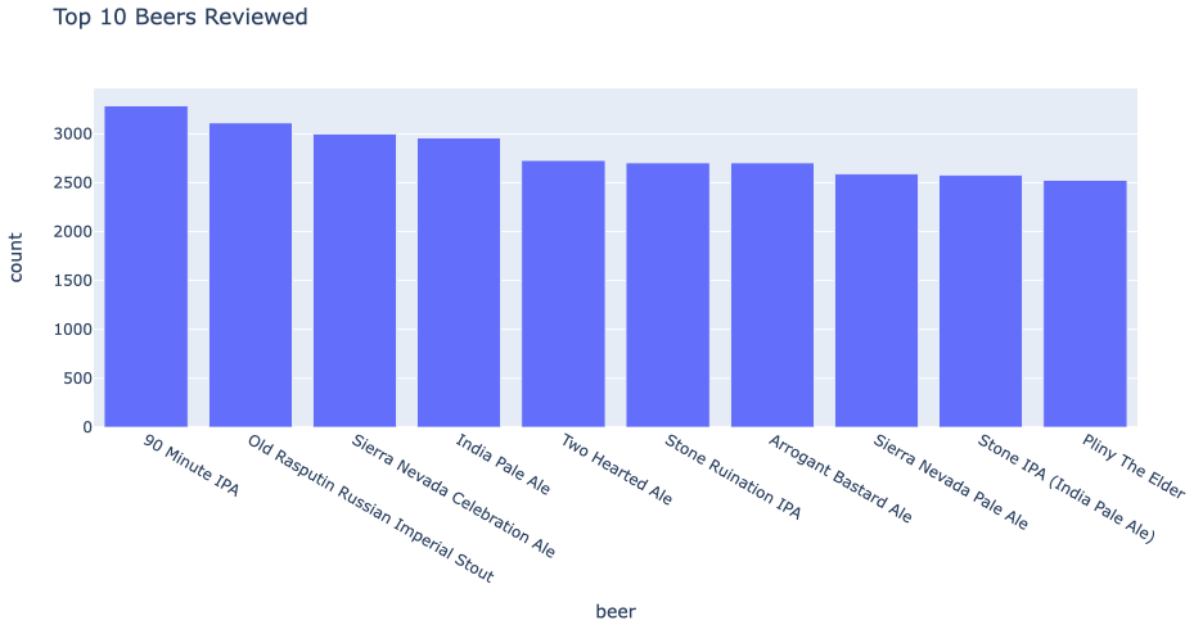
C. Top 100 Breweries Reviewed & Top 10 Breweries Reviewed



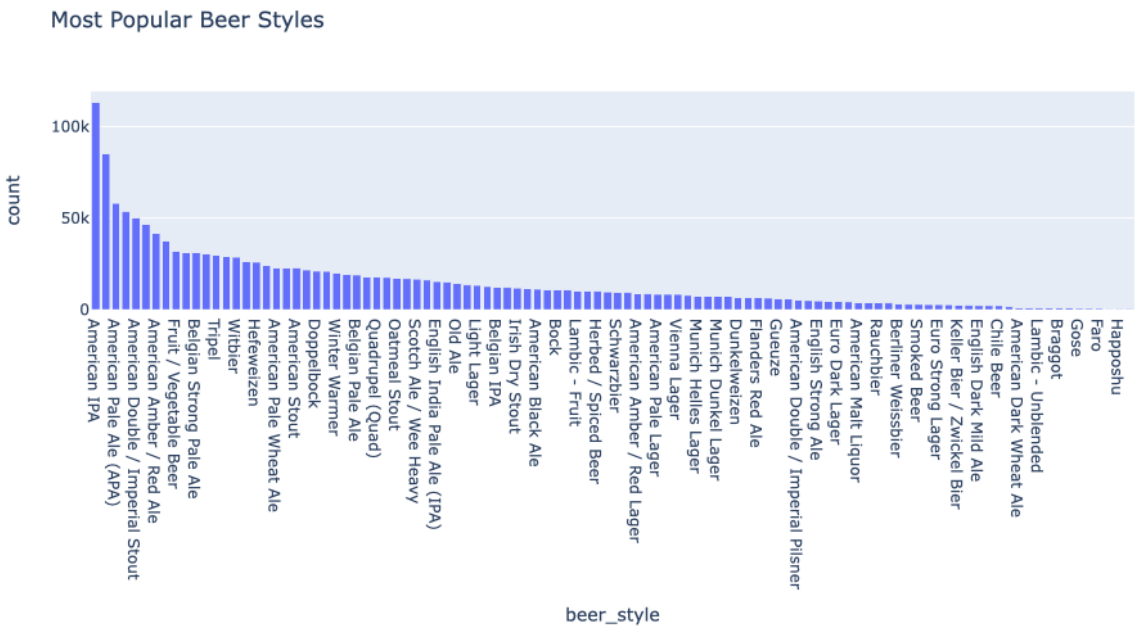


D. Top 100 Beers Reviewed & Top 10 Beers Reviewed

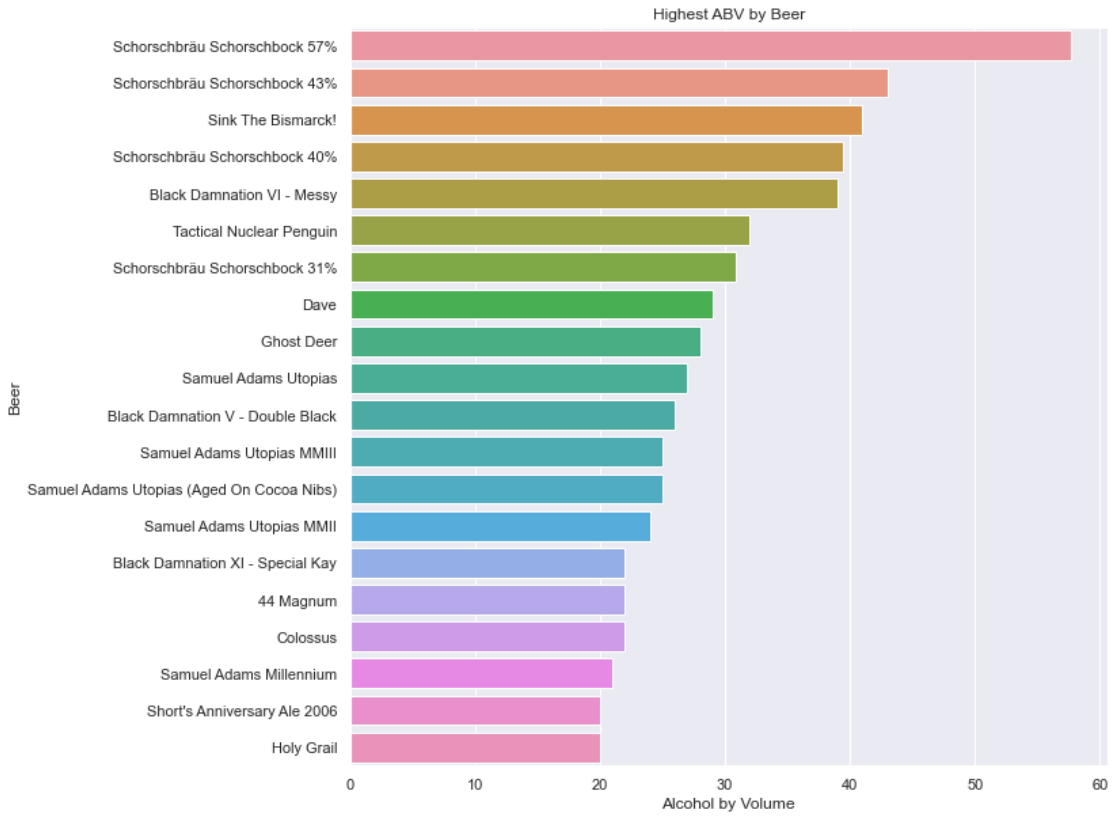




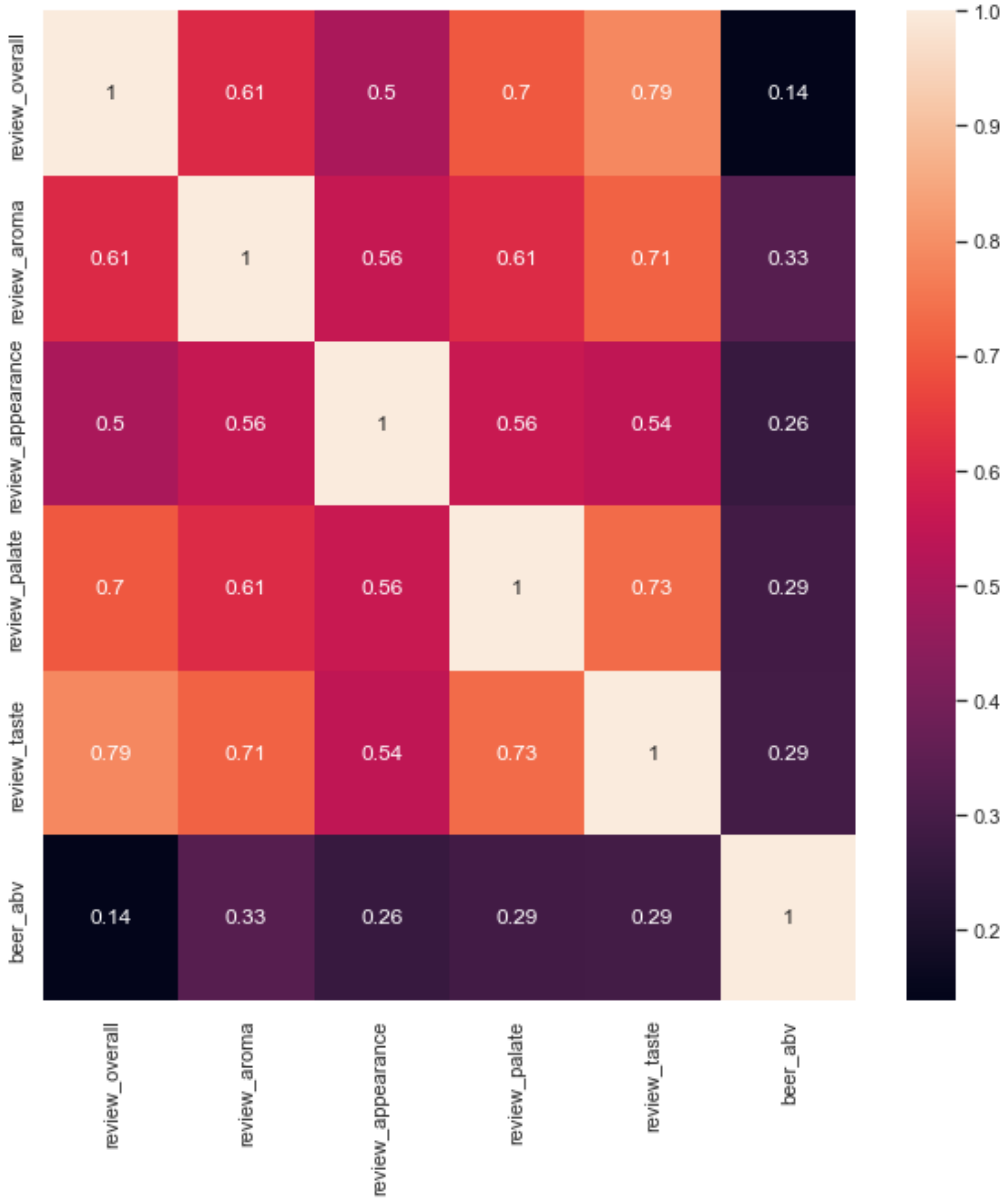
E. Most Popular Beer Styles Reviewed



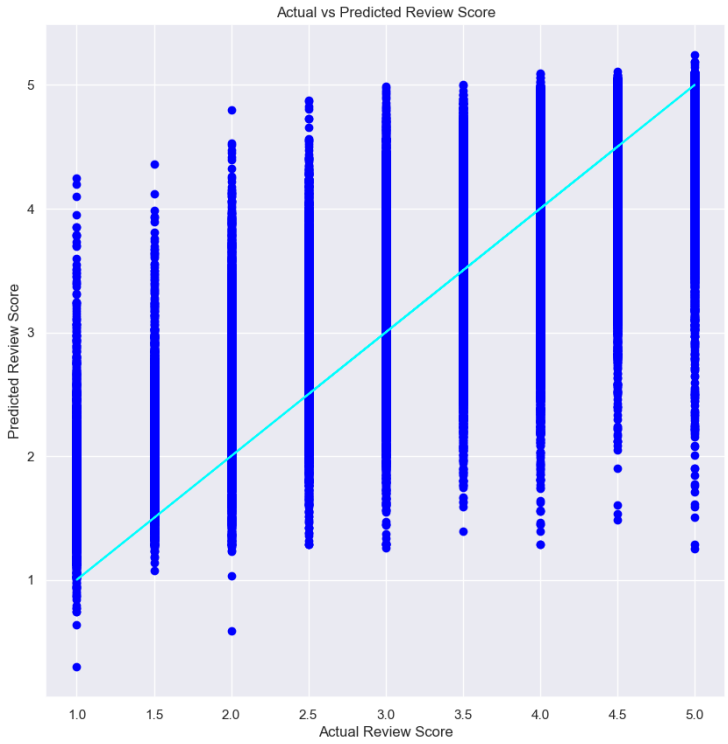
F. Highest ABV by Beer

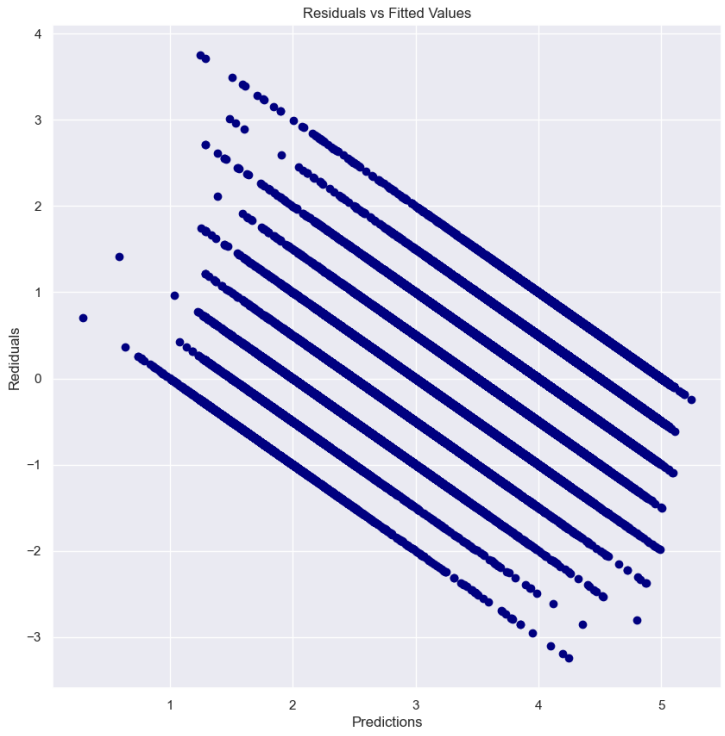


G. Heat Map comparing review aspects

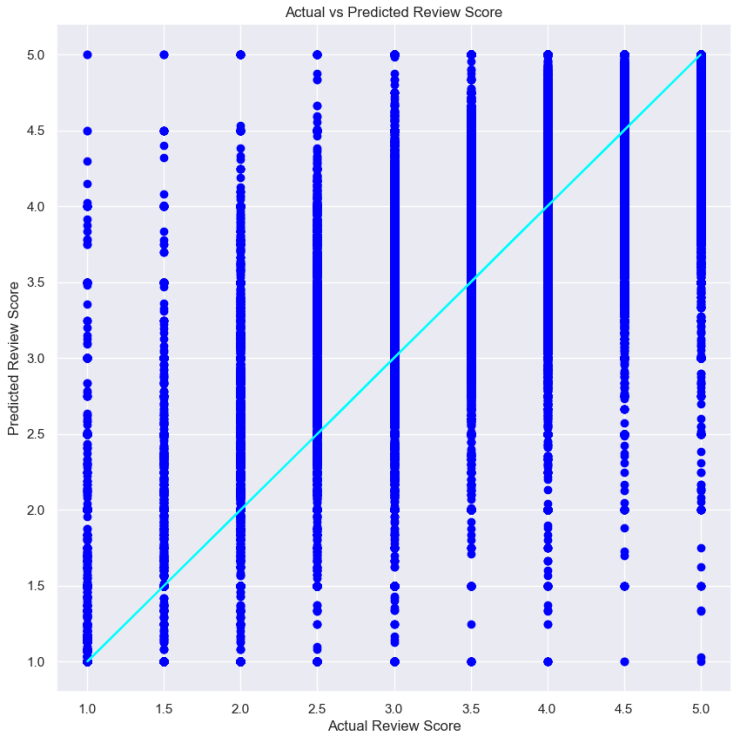


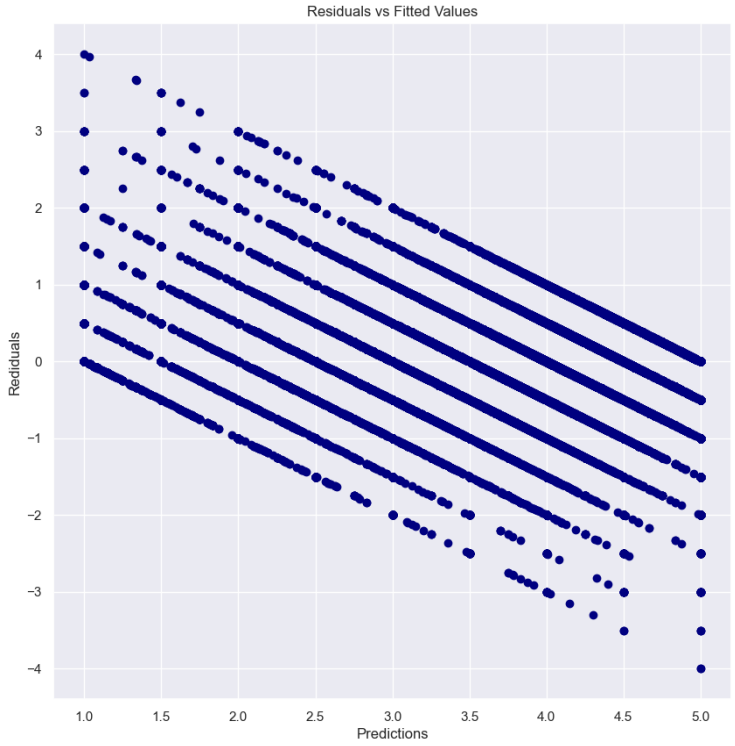
H. RQ1 Residual Analysis Plots for Multiple Linear Regression



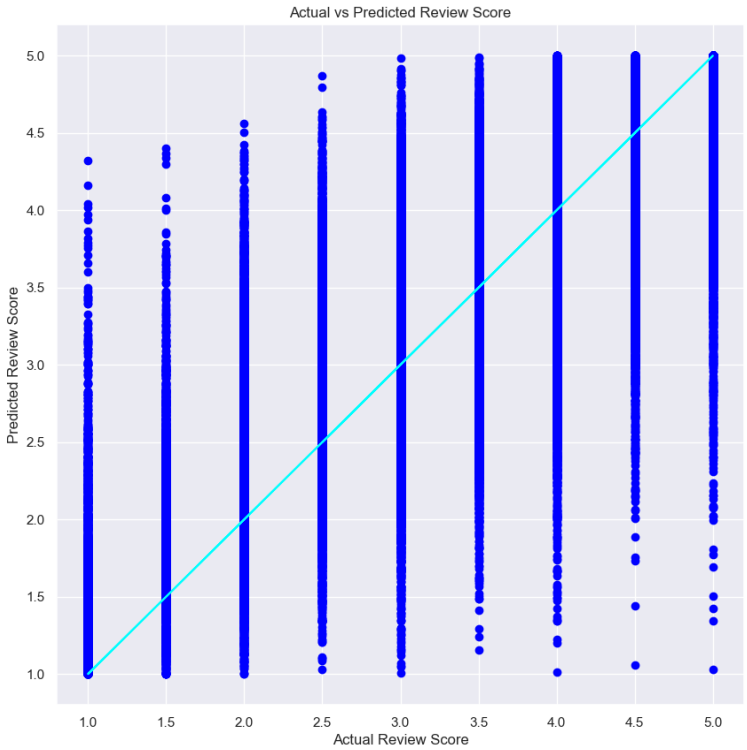


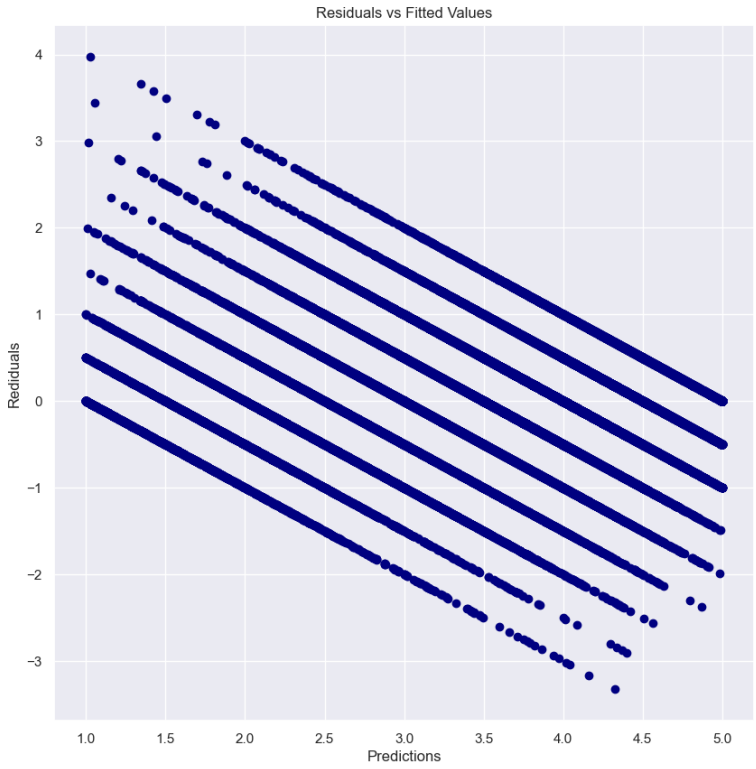
I. RQ1 Residual Analysis Plots for Decision Tree Regression



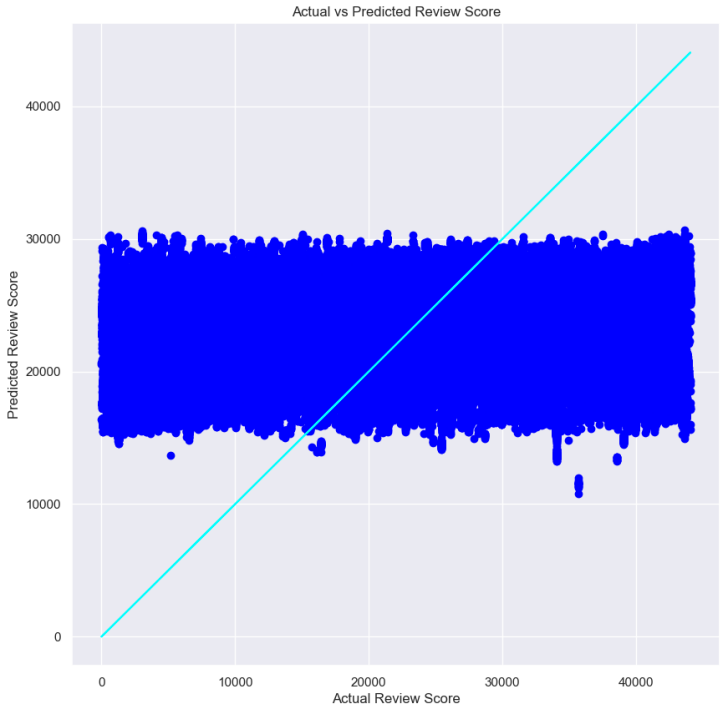


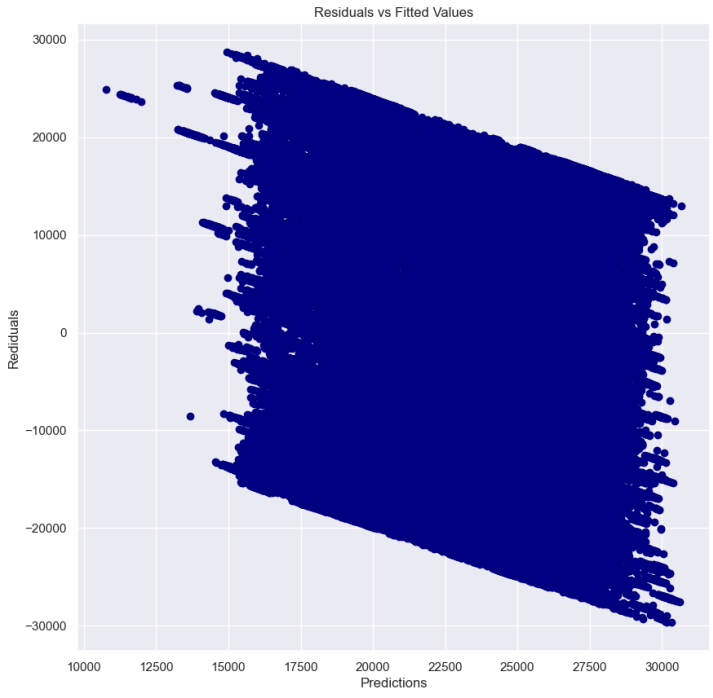
J. RQ1 Residual Analysis Plots for Random Forest Regression



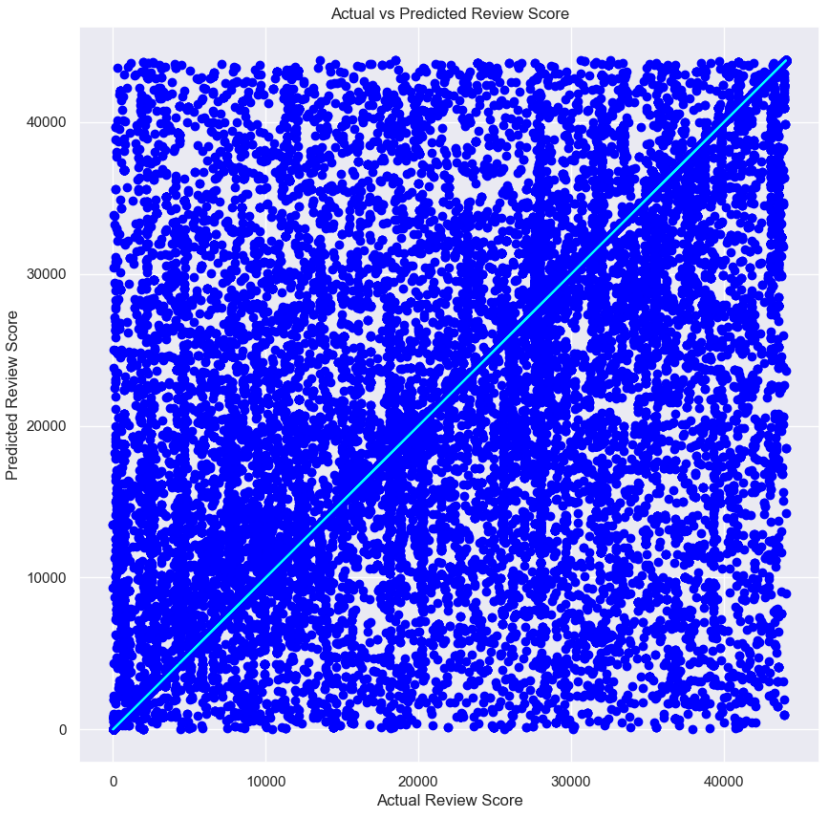


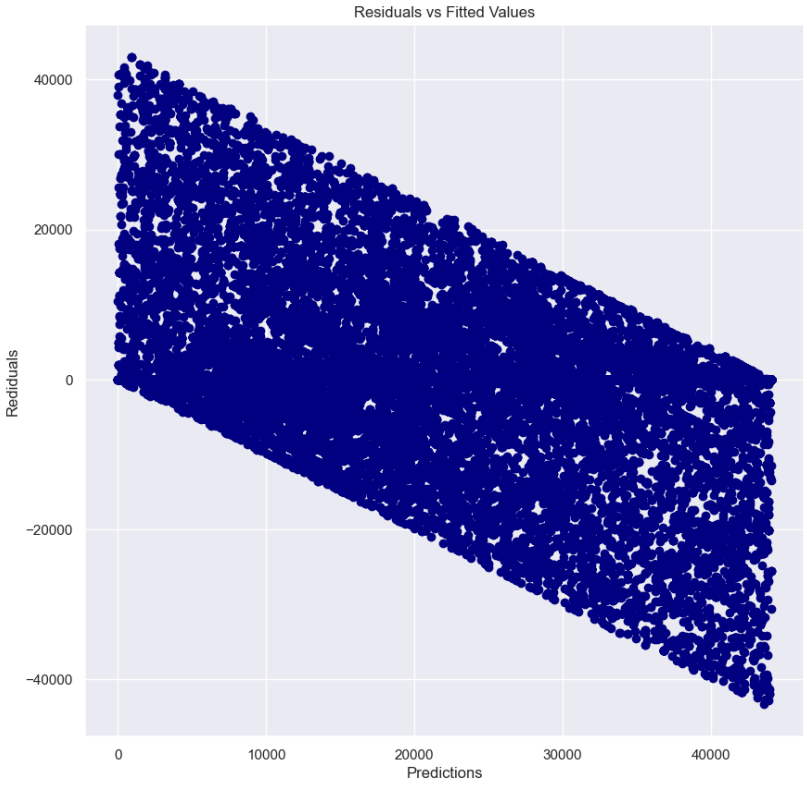
K. RQ2 Residual Analysis Plots for Multiple Linear Regression



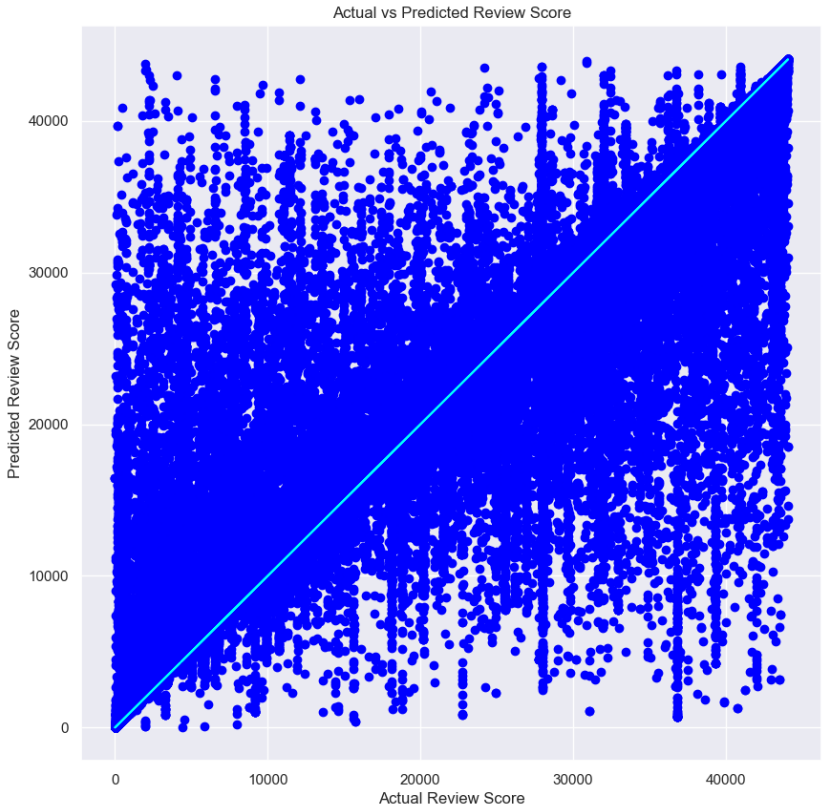


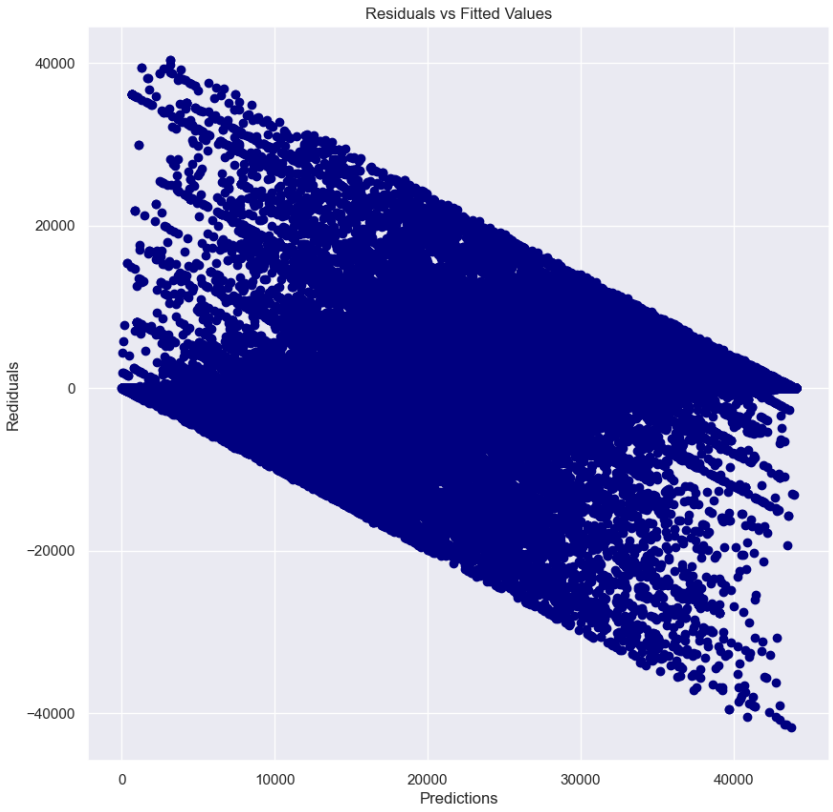
L. RQ2 Residual Analysis Plots for Decision Tree Regression


Actual vs Predicted Review Score
Predicted Review Score
Actual Review Score
0
10000
20000
30000
40000

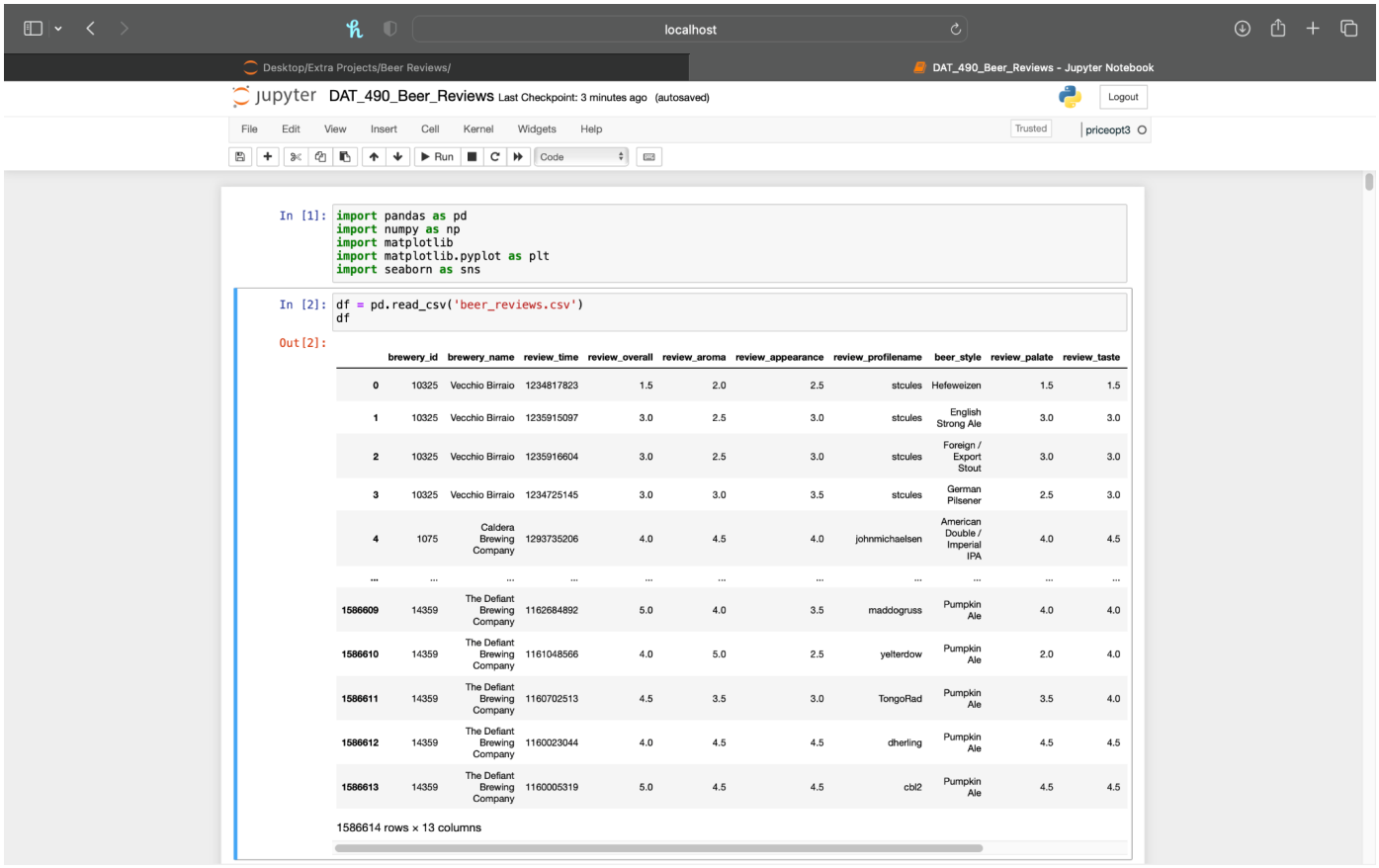

Residuals vs Fitted Values
Rediduals
Predictions
40000
20000
0
−20000
−40000
0
10000
20000
30000
40000

M. RQ2 Residual Analysis Plots for Random Forest Regression



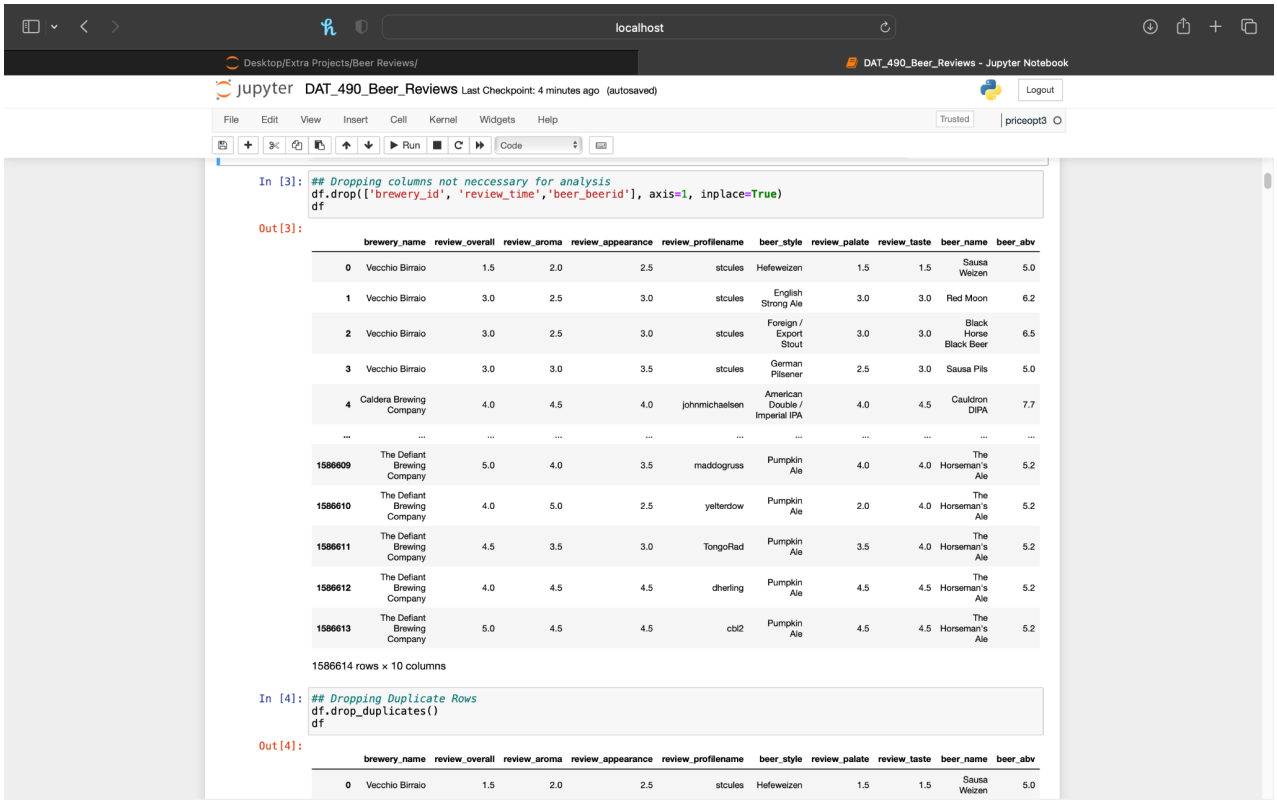

Residuals vs Fitted Values
40000
20000
0
−20000
−40000
Rediduals
0
10000
20000
30000
40000
Predictions

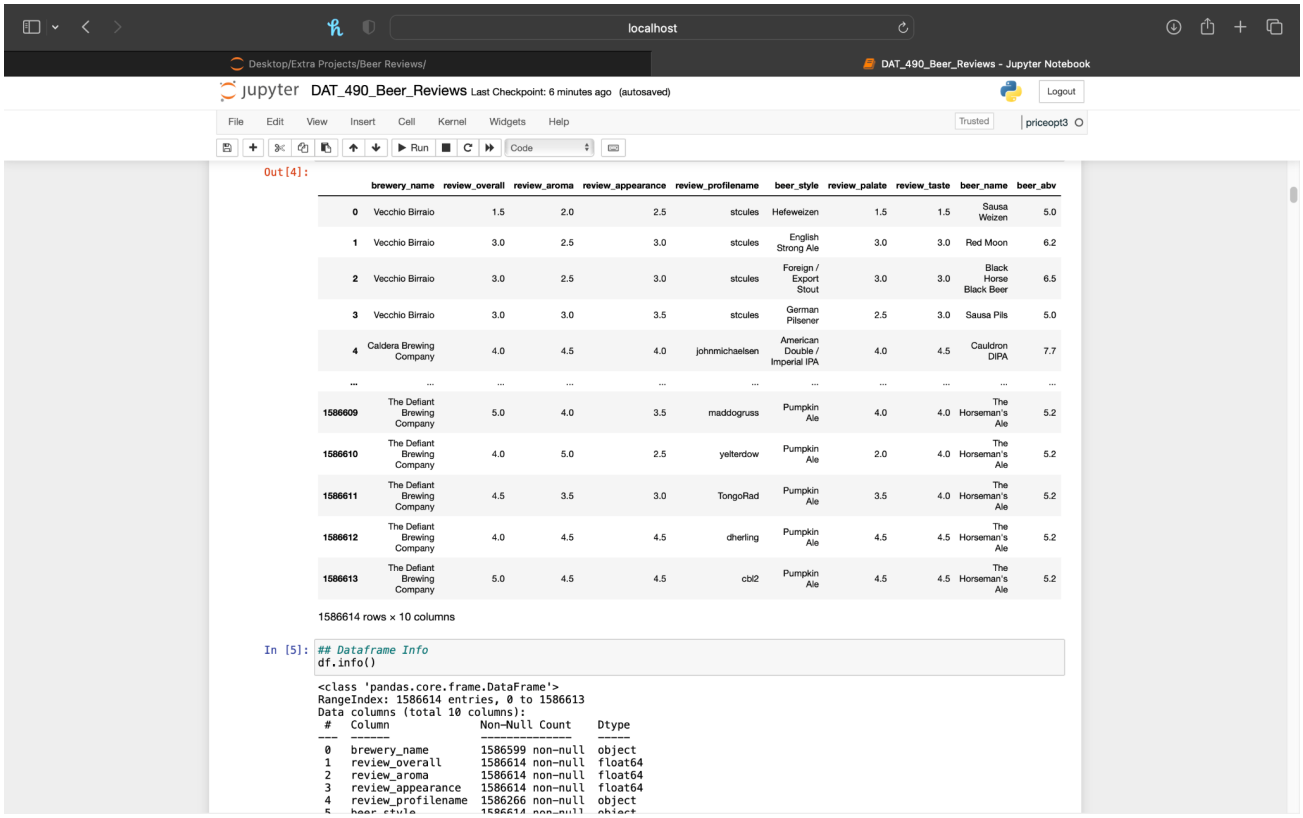
Code

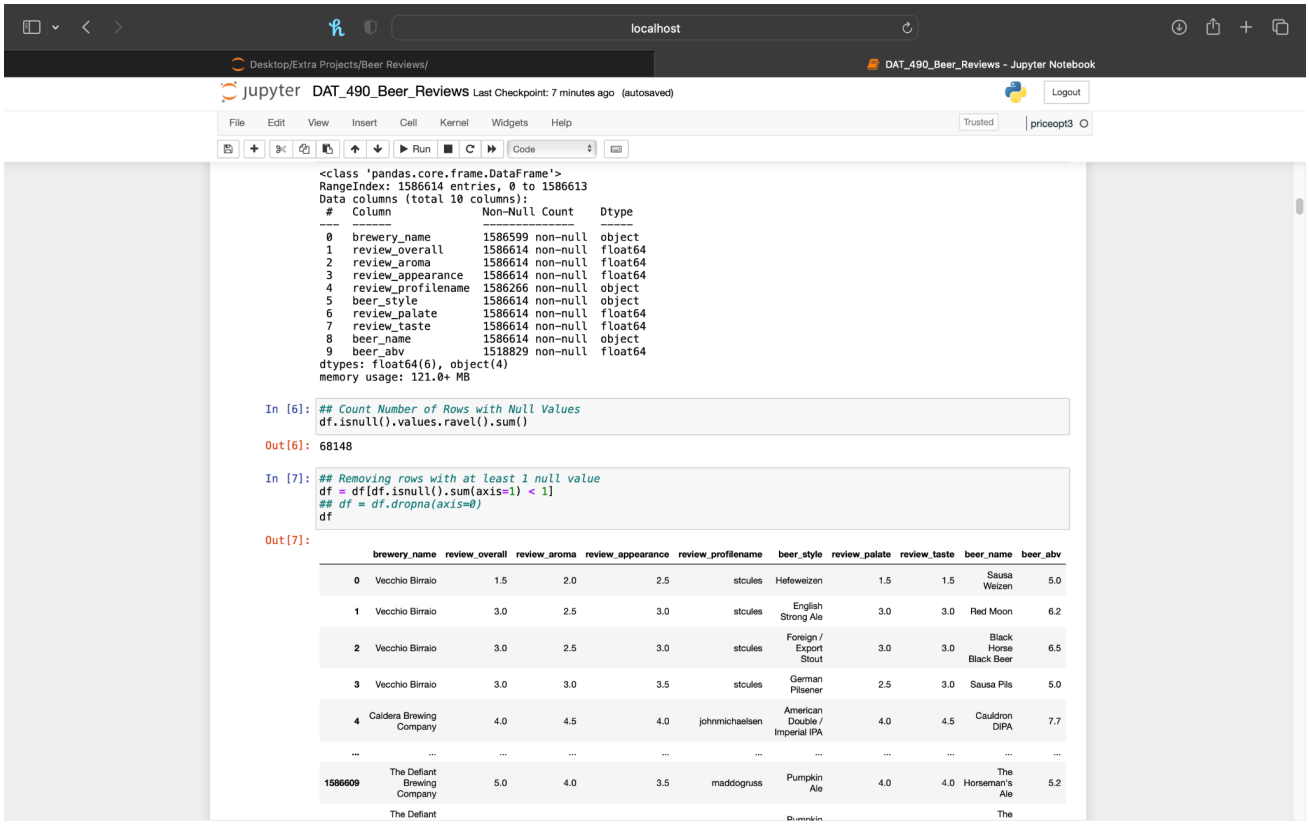


	brewery_id	brewery_name	review_time	review_overall	review_aroma	review_appearance	review_profilename	beer_style	review_palate	review_taste
0	10325	Vecchio Birraio	1234817823	1.5	2.0	2.5	stcules	Hefeweizen	1.5	1.5
1	10325	Vecchio Birraio	1235915097	3.0	2.5	3.0	stcules	English Strong Ale	3.0	3.0
2	10325	Vecchio Birraio	1235916604	3.0	2.5	3.0	stcules	Foreign / Export Stout	3.0	3.0
3	10325	Vecchio Birraio	1234725145	3.0	3.0	3.5	stcules	German Pilsener	2.5	3.0
4	1075	Caldera Brewing Company	1293735206	4.0	4.5	4.0	johnmichaelsen	American Double / Imperial IPA	4.0	4.5
...	...	...	...	...	...	...	...	...	...	...
1586609	14359	The Defiant Brewing Company	1162684892	5.0	4.0	3.5	maddogruss	Pumpkin Ale	4.0	4.0
1586610	14359	The Defiant Brewing Company	1161048566	4.0	5.0	2.5	yelterdow	Pumpkin Ale	2.0	4.0
1586611	14359	The Defiant Brewing Company	1160702513	4.5	3.5	3.0	TongoRad	Pumpkin Ale	3.5	4.0
1586612	14359	The Defiant Brewing Company	1160023044	4.0	4.5	4.5	dherling	Pumpkin Ale	4.5	4.5
1586613	14359	The Defiant Brewing Company	1160005319	5.0	4.5	4.5	cbl2	Pumpkin Ale	4.5	4.5

1586614 rows × 13 columns


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 4 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
In [3]:
Dropping columns not neccessary for analysis
df.drop(['brewery_id', 'review_time','beer_beerid'], axis=1, inplace=True)
df
Out[3]:
brewery_name review_overall review_aroma review_appearance review_profilename beer_style review_palate review_taste beer_name beer_abv
0 Vecchio Birraio 1.5 2.0 2.5 stcules Hefeweizen 1.5 1.5 Sausa Weizen 5.0
1 Vecchio Birraio 3.0 2.5 3.0 stcules English Strong Ale 3.0 3.0 Red Moon 6.2
2 Vecchio Birraio 3.0 2.5 3.0 stcules Foreign / Export Stout 3.0 3.0 Black Horse Black Beer 6.5
3 Vecchio Birraio 3.0 3.0 3.5 stcules German Pilsener 2.5 3.0 Sausa Pils 5.0
4 Caldera Brewing Company 4.0 4.5 4.0 johnmichaelsen American Double / Imperial IPA 4.0 4.5 Cauldron DIPA 7.7
... ...
1586609 The Defiant Brewing Company 5.0 4.0 3.5 maddogruss Pumpkin Ale 4.0 4.0 The Horseman's Ale 5.2
1586610 The Defiant Brewing Company 4.0 5.0 2.5 yelterdow Pumpkin Ale 2.0 4.0 The Horseman's Ale 5.2
1586611 The Defiant Brewing Company 4.5 3.5 3.0 TongoRad Pumpkin Ale 3.5 4.0 The Horseman's Ale 5.2
1586612 The Defiant Brewing Company 4.0 4.5 4.5 dherling Pumpkin Ale 4.5 4.5 The Horseman's Ale 5.2
1586613 The Defiant Brewing Company 5.0 4.5 4.5 cbl2 Pumpkin Ale 4.5 4.5 The Horseman's Ale 5.2
1586614 rows × 10 columns
In [4]:
Dropping Duplicate Rows
df.drop_duplicates()
df
Out[4]:
brewery_name review_overall review_aroma review_appearance review_profilename beer_style review_palate review_taste beer_name beer_abv
0 Vecchio Birraio 1.5 2.0 2.5 stcules Hefeweizen 1.5 1.5 Sausa Weizen 5.0


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 6 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Out[4]:
brewery_name review_overall review_aroma review_appearance review_profilename beer_style review_palate review_taste beer_name beer_abv
0 Vecchio Birraio 1.5 2.0 2.5 stcules Hefeweizen 1.5 1.5 Sausa Weizen 5.0
1 Vecchio Birraio 3.0 2.5 3.0 stcules English Strong Ale 3.0 3.0 Red Moon 6.2
2 Vecchio Birraio 3.0 2.5 3.0 stcules Foreign / Export Stout 3.0 3.0 Black Horse Black Beer 6.5
3 Vecchio Birraio 3.0 3.0 3.5 stcules German Pilsener 2.5 3.0 Sausa Pils 5.0
4 Caldera Brewing Company 4.0 4.5 4.0 johnmichaelsen American Double / Imperial IPA 4.0 4.5 Cauldron DIPA 7.7
... ...
1586609 The Defiant Brewing Company 5.0 4.0 3.5 maddogruss Pumpkin Ale 4.0 4.0 The Horseman's Ale 5.2
1586610 The Defiant Brewing Company 4.0 5.0 2.5 yelterdow Pumpkin Ale 2.0 4.0 The Horseman's Ale 5.2
1586611 The Defiant Brewing Company 4.5 3.5 3.0 TongoRad Pumpkin Ale 3.5 4.0 The Horseman's Ale 5.2
1586612 The Defiant Brewing Company 4.0 4.5 4.5 dherling Pumpkin Ale 4.5 4.5 The Horseman's Ale 5.2
1586613 The Defiant Brewing Company 5.0 4.5 4.5 cbl2 Pumpkin Ale 4.5 4.5 The Horseman's Ale 5.2
1586614 rows × 10 columns
In [5]:
Dataframe Info
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1586614 entries, 0 to 1586613
Data columns (total 10 columns):
Column Non-Null Count Dtype
0 brewery_name 1586599 non-null object
1 review_overall 1586614 non-null float64
2 review_aroma 1586614 non-null float64
3 review_appearance 1586614 non-null float64
4 review_profilename 1586266 non-null object


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 7 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1586614 entries, 0 to 1586613
Data columns (total 10 columns):
Column Non-Null Count Dtype
0 brewery_name 1586599 non-null object
1 review_overall 1586614 non-null float64
2 review_aroma 1586614 non-null float64
3 review_appearance 1586614 non-null float64
4 review_profilename 1586266 non-null object
5 beer_style 1586614 non-null object
6 review_palate 1586614 non-null float64
7 review_taste 1586614 non-null float64
8 beer_name 1586614 non-null object
9 beer_abv 1518829 non-null float64
dtypes: float64(6), object(4)
memory usage: 121.0+ MB
In [6]:
Count Number of Rows with Null Values
df.isnull().values.ravel().sum()
Out[6]: 68148
In [7]:
Removing rows with at least 1 null value
df = df[df.isnull().sum(axis=1) < 1]
df = df.dropna(axis=0)
df
Out[7]:
brewery_name review_overall review_aroma review_appearance review_profilename beer_style review_palate review_taste beer_name beer_abv
0 Vecchio Birraio 1.5 2.0 2.5 stcules Hefeweizen 1.5 1.5 Sausa Weizen 5.0
1 Vecchio Birraio 3.0 2.5 3.0 stcules English Strong Ale 3.0 3.0 Red Moon 6.2
2 Vecchio Birraio 3.0 2.5 3.0 stcules Foreign / Export Stout 3.0 3.0 Black Horse Black Beer 6.5
3 Vecchio Birraio 3.0 3.0 3.5 stcules German Pilsener 2.5 3.0 Sausa Pils 5.0
4 Caldera Brewing Company 4.0 4.5 4.0 johnmichaelsen American Double / Imperial IPA 4.0 4.5 Cauldron DIPA 7.7
... ...
1586609 The Defiant Brewing Company 5.0 4.0 3.5 maddogruss Pumpkin Ale 4.0 4.0 The Horseman's Ale 5.2

localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

jupyter DAT_490_Beer_Reviews Last Checkpoint: 7 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help — Trusted — priceopt3

Run — Code

Out[7]:

	brewery_name	review_overall	review_aroma	review_appearance	review_profilename	beer_style	review_palate	review_taste	beer_name	beer_abv
0	Vecchio Birraio	1.5	2.0	2.5	stcules	Hefeweizen	1.5	1.5	Sausa Weizen	5.0
1	Vecchio Birraio	3.0	2.5	3.0	stcules	English Strong Ale	3.0	3.0	Red Moon	6.2
2	Vecchio Birraio	3.0	2.5	3.0	stcules	Foreign / Export Stout	3.0	3.0	Black Horse Black Beer	6.5
3	Vecchio Birraio	3.0	3.0	3.5	stcules	German Pilsener	2.5	3.0	Sausa Pils	5.0
4	Caldera Brewing Company	4.0	4.5	4.0	johnmichaelsen	American Double / Imperial IPA	4.0	4.5	Cauldron DIPA	7.7
...	...	...	...	...	...	...	...	...	...	...
1586609	The Defiant Brewing Company	5.0	4.0	3.5	maddogruss	Pumpkin Ale	4.0	4.0	The Horseman's Ale	5.2
1586610	The Defiant Brewing Company	4.0	5.0	2.5	yelterdow	Pumpkin Ale	2.0	4.0	The Horseman's Ale	5.2
1586611	The Defiant Brewing Company	4.5	3.5	3.0	TongoRad	Pumpkin Ale	3.5	4.0	The Horseman's Ale	5.2
1586612	The Defiant Brewing Company	4.0	4.5	4.5	dherling	Pumpkin Ale	4.5	4.5	The Horseman's Ale	5.2
1586613	The Defiant Brewing Company	5.0	4.5	4.5	cbl2	Pumpkin Ale	4.5	4.5	The Horseman's Ale	5.2

1518478 rows × 10 columns

In [8]:
```
df.describe()
```

Out[8]:

	review_overall	review_aroma	review_appearance	review_palate	review_taste	beer_abv
count	1.518478e+06	1.518478e+06	1.518478e+06	1.518478e+06	1.518478e+06	1.518478e+06
mean	3.823938e+00	3.746218e+00	3.850383e+00	3.753735e+00	3.804082e+00	7.042488e+00
std	7.172663e-01	6.953440e-01	6.143106e-01	6.793350e-01	7.286079e-01	2.322568e+00
min	0.000000e+00	1.000000e+00	0.000000e+00	1.000000e+00	1.000000e+00	1.000000e-02
25%	3.500000e+00	3.500000e+00	3.500000e+00	3.500000e+00	3.500000e+00	5.200000e+00
50%	4.000000e+00	4.000000e+00	4.000000e+00	4.000000e+00	4.000000e+00	6.500000e+00

localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

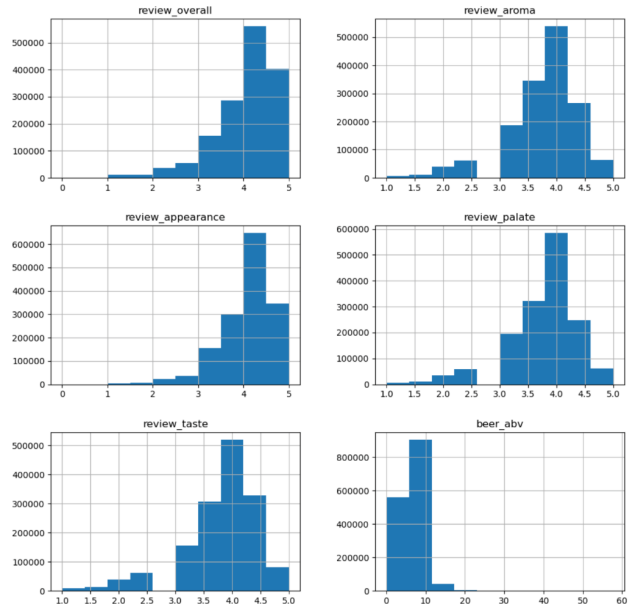
jupyter DAT_490_Beer_Reviews Last Checkpoint: 9 minutes ago (autosaved)

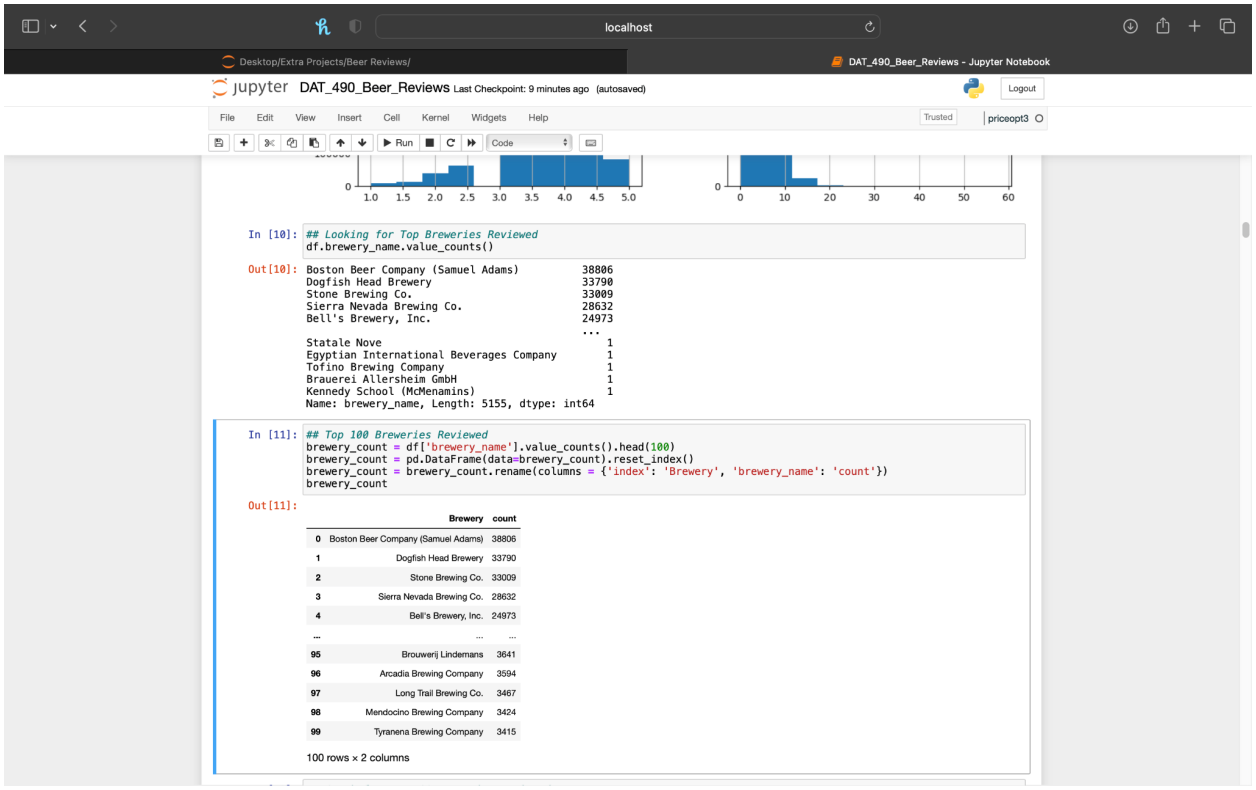
Logout

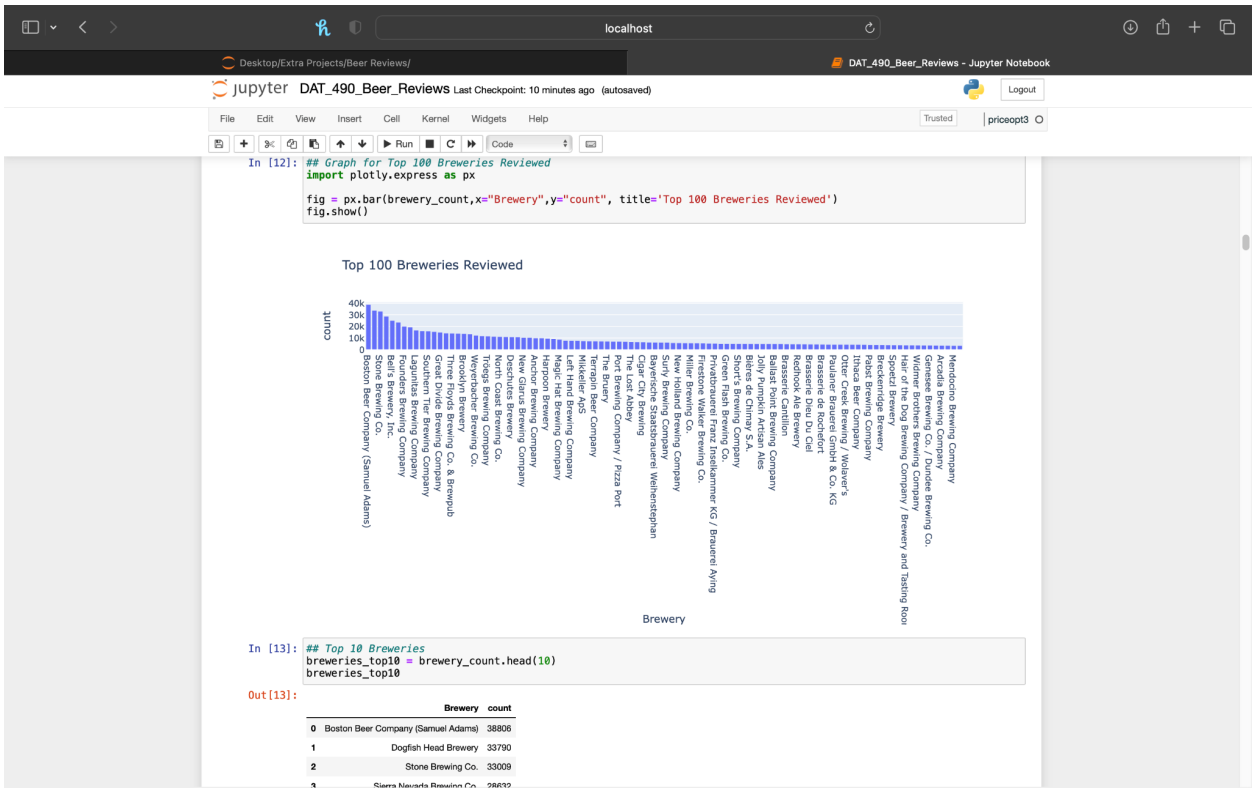
File Edit View Insert Cell Kernel Widgets Help — Trusted — priceopt3

Run — Code

In [9]:
```
df.hist(figsize=(12,12))
plt.show()
```




localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 9 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
0
1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
0
0 10 20 30 40 50 60
In [10]:
Looking for Top Breweries Reviewed
df.brewery_name.value_counts()
Out[10]:
Boston Beer Company (Samuel Adams) 38806
Dogfish Head Brewery 33790
Stone Brewing Co. 33009
Sierra Nevada Brewing Co. 28632
Bell's Brewery, Inc. 24973
...
Statale Nove 1
Egyptian International Beverages Company 1
Tofino Brewing Company 1
Brauerei Allersheim GmbH 1
Kennedy School (McMenamins) 1
Name: brewery_name, Length: 5155, dtype: int64
In [11]:
Top 100 Breweries Reviewed
brewery_count = df['brewery_name'].value_counts().head(100)
brewery_count = pd.DataFrame(data=brewery_count).reset_index()
brewery_count = brewery_count.rename(columns = {'index': 'Brewery', 'brewery_name': 'count'})
brewery_count
Out[11]:
Brewery count
0 Boston Beer Company (Samuel Adams) 38806
1 Dogfish Head Brewery 33790
2 Stone Brewing Co. 33009
3 Sierra Nevada Brewing Co. 28632
4 Bell's Brewery, Inc. 24973
... ...
95 Brouwerij Lindemans 3641
96 Arcadia Brewing Company 3594
97 Long Trail Brewing Co. 3467
98 Mendocino Brewing Company 3424
99 Tyranena Brewing Company 3415
100 rows × 2 columns


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 10 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
In [12]:
Graph for Top 100 Breweries Reviewed
import plotly.express as px
fig = px.bar(brewery_count,x="Brewery",y="count", title='Top 100 Breweries Reviewed')
fig.show()
Top 100 Breweries Reviewed
count
40k
30k
20k
10k
0
Boston Beer Company (Samuel Adams)
Stone Brewing Co.
Bell's Brewery, Inc.
Founders Brewing Company
Lagunitas Brewing Company
Southern Tier Brewing Company
Three Floyds Brewing Co. & Brewpub
Brooklyn Brewery
Weyerbacher Brewing Co.
North Coast Brewing Co.
Tröegs Brewing Company
Deschutes Brewery
New Belgium Brewing Company
Anchor Brewing Company
Harpoon Brewery
Magic Hat Brewing Company
Left Hand Brewing Company
Mikkeller ApS
Terrain Beer Company
The Bruery
Port Brewing Company / Pizza Port
The Lost Abbey
Cigar City Brewing
Bayerische Staatsbrauerei Weihenstephan
Surly Brewing Company
New Holland Brewing Company
Rogue Ales
Firestone Walker Brewing Co.
Privatbrauerei Franz Inselkammer KG / Brauerei Aying
Green Flash Brewing Co.
Short's Brewing Company
Bells Brewery Inc.
Jolly Pumpkin Artisan Ales
Bières de Chimay S.A.
Russian River Brewing Company
Brasserie Cantillon
Redhook Ale Brewery
Brasserie Dieu Du Ciel
Brasserie de Rochefort
Paulaner Brauerei GmbH & Co. KG
Ommegang Brewery
Otter Creek Brewing / Wolaver's
Pabst Brewing Company
Breckenridge Brewery
Spoetzl Brewery
Hair of the Dog Brewing Company / Brewery and Tasting Roo
Widmer Brothers Brewing Company
Goose Island Beer Co.
Schneider Weisse Brewing Co.
Arcadia Brewing Company
Mendocino Brewing Company
Brewery
In [13]:
Top 10 Breweries
breweries_top10 = brewery_count.head(10)
breweries_top10
Out[13]:
Brewery count
0 Boston Beer Company (Samuel Adams) 38806
1 Dogfish Head Brewery 33790
2 Stone Brewing Co. 33009
3 Sierra Nevada Brewing Co. 28632

localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

jupyter DAT_490_Beer_Reviews Last Checkpoint: 10 minutes ago (autosaved) Logout

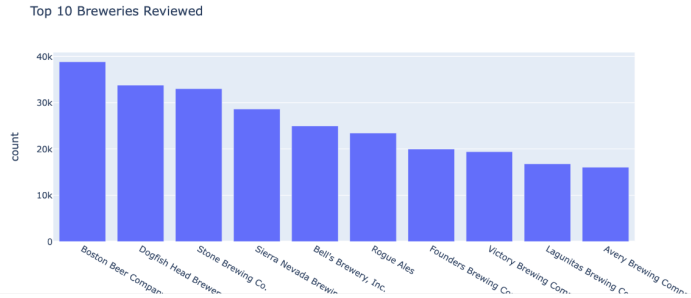
File Edit View Insert Cell Kernel Widgets Help Trusted priceopt3

Run Code

Out[13]:

	Brewery	count
0	Boston Beer Company (Samuel Adams)	38806
1	Dogfish Head Brewery	33790
2	Stone Brewing Co.	33009
3	Sierra Nevada Brewing Co.	28632
4	Bell's Brewery, Inc.	24973
5	Rogue Ales	23446
6	Founders Brewing Company	19951
7	Victory Brewing Company	19401
8	Lagunitas Brewing Company	16766
9	Avery Brewing Company	16042

In [14]:
```
## Graph for Top 10 Breweries Reviewed
import plotly.express as px

fig = px.bar(breweries_top10,x="Brewery",y="count", title='Top 10 Breweries Reviewed')
fig.show()
```



localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

jupyter DAT_490_Beer_Reviews Last Checkpoint: 11 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted priceopt3

Run Code

Brewery

In [15]:
```
df['beer_name'].value_counts()
```

Out[15]:
```
90 Minute IPA                          3289
Old Rasputin Russian Imperial Stout    3110
Sierra Nevada Celebration Ale          2999
India Pale Ale                         2960
Two Hearted Ale                        2727
                                       ...
Caves Right Pilsner                       1
Public House                              1
Fish In A Barrel                          1
Kenny The IPA                             1
Moska Negra                               1
Name: beer_name, Length: 44075, dtype: int64
```

In [16]:
```
## Top 100 Beers Reviewed
beers = df['beer_name'].value_counts().head(100)
beers = pd.DataFrame(data=beers).reset_index()
beers = beers.rename(columns = {'index': 'beer', 'beer_name': 'count'})
beers
```

Out[16]:

	beer	count
0	90 Minute IPA	3289
1	Old Rasputin Russian Imperial Stout	3110
2	Sierra Nevada Celebration Ale	2999
3	India Pale Ale	2960
4	Two Hearted Ale	2727
...	...	...
95	Yeti Imperial Stout	1429
96	Oak Aged Yeti Imperial Stout	1422
97	Delirium Tremens	1420
98	Anchor Liberty Ale	1417
99	The Abyss	1412

100 rows × 2 columns

In [17]: ## Graph for Top 100 Beers Reviewed

localhost

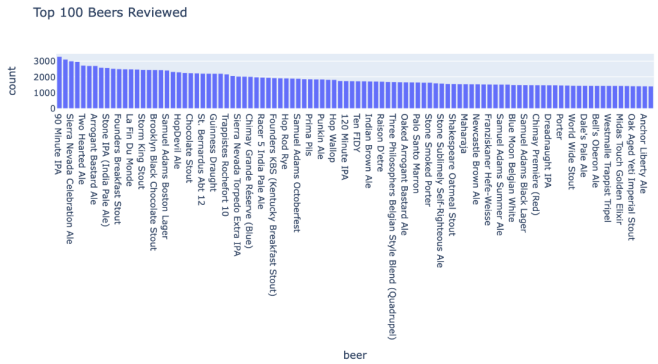
Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

jupyter DAT_490_Beer_Reviews Last Checkpoint: 11 minutes ago (autosaved) — Logout

File Edit View Insert Cell Kernel Widgets Help — Trusted — priceopt3

Run Code

In [17]:
```
## Graph for Top 100 Beers Reviewed
import plotly.express as px

fig = px.bar(beers,x="beer",y="count", title='Top 100 Beers Reviewed')
fig.show()
```



In [18]:
```
## Top 10 Beers
beers_top10 = beers.head(10)
beers_top10
```

Out[18]:

	beer	count
0	90 Minute IPA	3289
1	Old Rasputin Russian Imperial Stout	3110
2	Sierra Nevada Celebration Ale	2999
3	India Pale Ale	2960

localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

jupyter DAT_490_Beer_Reviews Last Checkpoint: 12 minutes ago (autosaved) — Logout

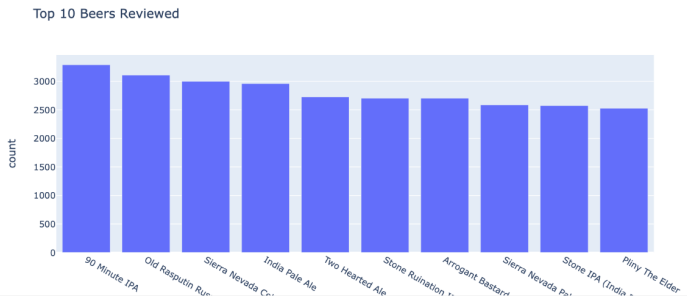
File Edit View Insert Cell Kernel Widgets Help — Trusted — priceopt3

Run Code

Out[18]:

	beer	count
0	90 Minute IPA	3289
1	Old Rasputin Russian Imperial Stout	3110
2	Sierra Nevada Celebration Ale	2999
3	India Pale Ale	2960
4	Two Hearted Ale	2727
5	Stone Ruination IPA	2702
6	Arrogant Bastard Ale	2702
7	Sierra Nevada Pale Ale	2587
8	Stone IPA (India Pale Ale)	2574
9	Pliny The Elder	2527

In [19]:
```
## Graph for Top 10 Beers Reviewed
import plotly.express as px

fig = px.bar(beers_top10,x="beer",y="count", title='Top 10 Beers Reviewed')
fig.show()
```



localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

jupyter DAT_490_Beer_Reviews Last Checkpoint: 12 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help — Trusted — priceopt3

Run — Code

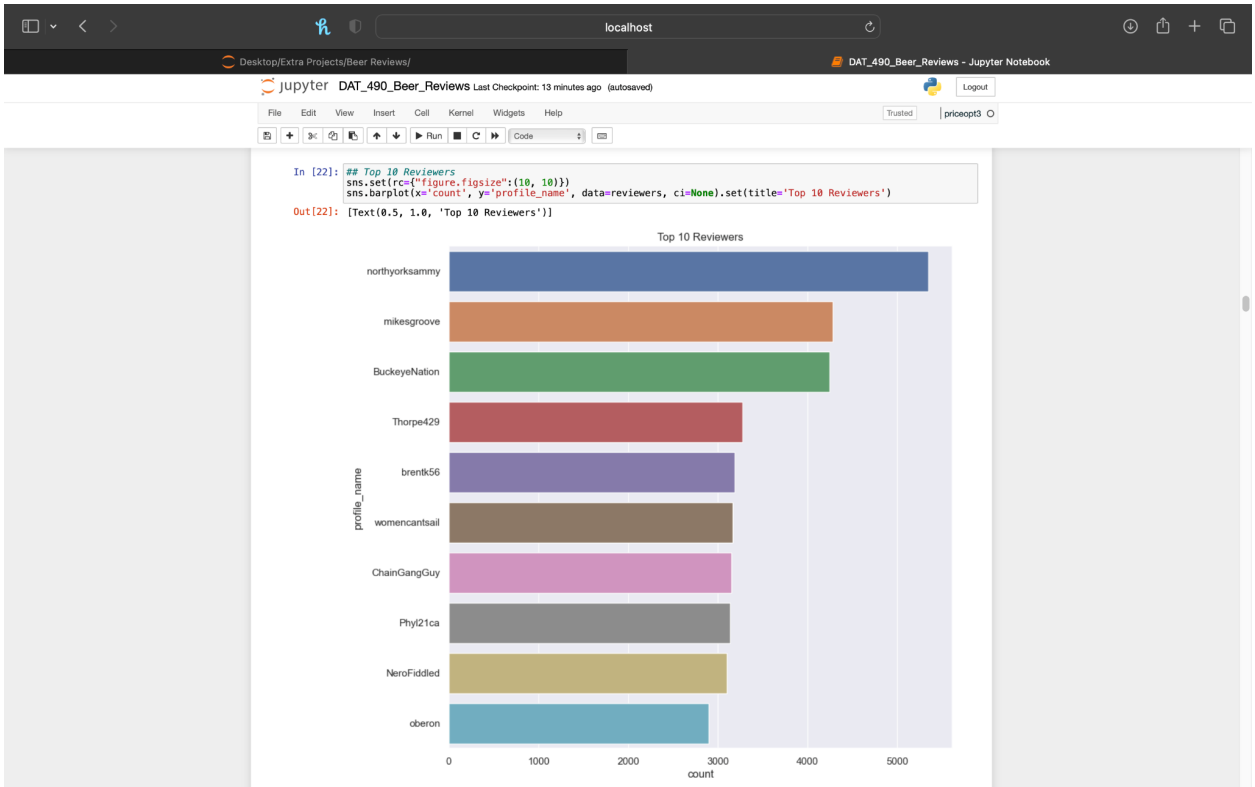
In [20]:
```
## Top Reviewers and Total Count
df.review_profilename.value_counts()
```

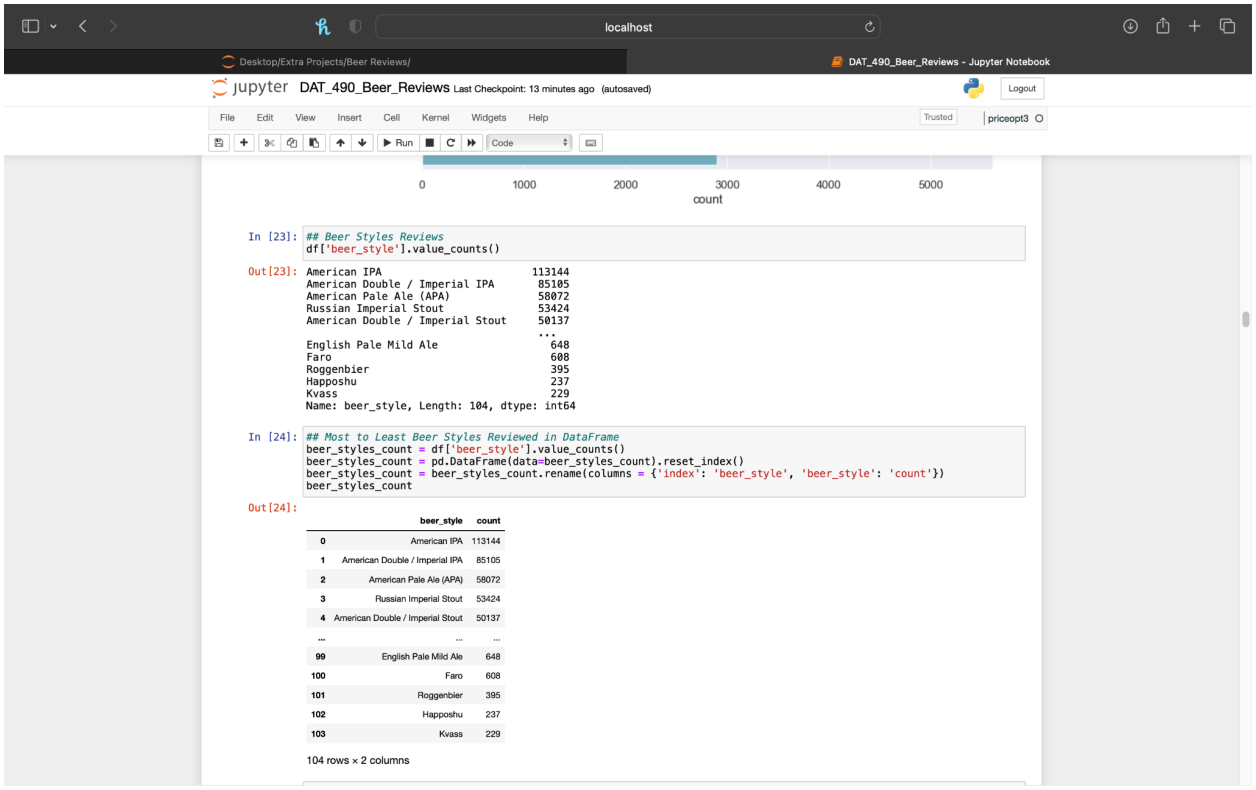
Out[20]:
```
northyorksammy    5346
mikesgroove       4283
BuckeyeNation     4246
Thorpe429         3273
brentk56          3186
                  ...
highTempo            1
thgrteins            1
ani                  1
endgames             1
Scottnr88            1
Name: review_profilename, Length: 32908, dtype: int64
```

In [21]:
```
## Reviewers Count in DataFrame
reviewers = df['review_profilename'].value_counts().head(10)
reviewers = pd.DataFrame(data=reviewers).reset_index()
reviewers = reviewers.rename(columns = {'index': 'profile_name', 'review_profilename': 'count'})
reviewers
```

Out[21]:

	profile_name	count
0	northyorksammy	5346
1	mikesgroove	4283
2	BuckeyeNation	4246
3	Thorpe429	3273
4	brentk56	3186
5	womencantsail	3163
6	ChainGangGuy	3151
7	Phyl21ca	3138
8	NeroFiddled	3098
9	oberon	2899





In [23]:
```
## Beer Styles Reviews
df['beer_style'].value_counts()
```

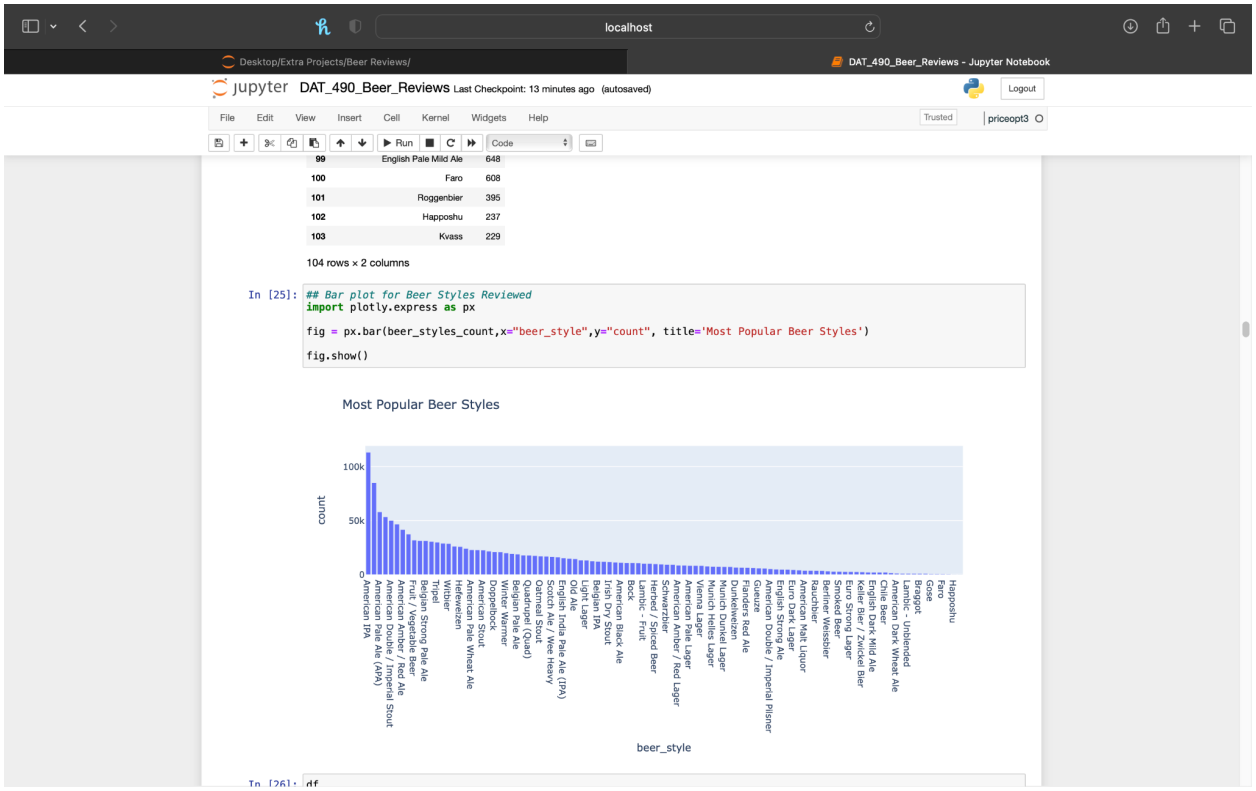
Out[23]:
```
American IPA                        113144
American Double / Imperial IPA       85105
American Pale Ale (APA)              58072
Russian Imperial Stout               53424
American Double / Imperial Stout     50137
                                      ...
English Pale Mild Ale                  648
Faro                                   608
Roggenbier                             395
Happoshu                               237
Kvass                                  229
Name: beer_style, Length: 104, dtype: int64
```

In [24]:
```
## Most to Least Beer Styles Reviewed in DataFrame
beer_styles_count = df['beer_style'].value_counts()
beer_styles_count = pd.DataFrame(data=beer_styles_count).reset_index()
beer_styles_count = beer_styles_count.rename(columns = {'index': 'beer_style', 'beer_style': 'count'})
beer_styles_count
```

Out[24]:

	beer_style	count
0	American IPA	113144
1	American Double / Imperial IPA	85105
2	American Pale Ale (APA)	58072
3	Russian Imperial Stout	53424
4	American Double / Imperial Stout	50137
...	...	...
99	English Pale Mild Ale	648
100	Faro	608
101	Roggenbier	395
102	Happoshu	237
103	Kvass	229

104 rows × 2 columns



99	English Pale Mild Ale	648
100	Faro	608
101	Roggenbier	395
102	Happoshu	237
103	Kvass	229

104 rows × 2 columns

In [25]:
```
## Bar plot for Beer Styles Reviewed
import plotly.express as px

fig = px.bar(beer_styles_count,x="beer_style",y="count", title='Most Popular Beer Styles')

fig.show()
```


In [26]:
```
df
```

localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

jupyter DAT_490_Beer_Reviews Last Checkpoint: 14 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted priceopt3

Run Code

In [26]:
```
df
```

Out[26]:

	brewery_name	review_overall	review_aroma	review_appearance	review_profilename	beer_style	review_palate	review_taste	beer_name	beer_abv
0	Vecchio Birraio	1.5	2.0	2.5	stcules	Hefeweizen	1.5	1.5	Sausa Weizen	5.0
1	Vecchio Birraio	3.0	2.5	3.0	stcules	English Strong Ale	3.0	3.0	Red Moon	6.2
2	Vecchio Birraio	3.0	2.5	3.0	stcules	Foreign / Export Stout	3.0	3.0	Black Horse Black Beer	6.5
3	Vecchio Birraio	3.0	3.0	3.5	stcules	German Pilsener	2.5	3.0	Sausa Pils	5.0
4	Caldera Brewing Company	4.0	4.5	4.0	johnmichaelsen	American Double / Imperial IPA	4.0	4.5	Cauldron DIPA	7.7
...	...	...	...	...	...	...	...	...	...	...

In [27]:
```
## Sorting Reviews by Beer Abv higher or equal to 20 in descending order
df_high_abv = df.loc[df['beer_abv'] >= 20, ('beer_name', 'beer_abv','review_aroma', 'review_appearance', 'review_pa
df_high_abv = df_high_abv.sort_values('beer_abv', ascending=False)
df_high_abv
```

Out[27]:

	beer_name	beer_abv	review_aroma	review_appearance	review_palate	review_taste	review_overall
12919	Schorschbräu Schorschbock 57%	57.7	4.0	4.0	4.0	3.5	4.0
12939	Schorschbräu Schorschbock 43%	43.0	4.0	3.5	4.0	4.0	4.0
12940	Schorschbräu Schorschbock 43%	43.0	4.0	4.0	4.0	4.5	3.5
746386	Sink The Bismarck!	41.0	3.0	3.5	3.5	3.5	2.5
746400	Sink The Bismarck!	41.0	5.0	4.0	4.5	5.0	4.5
...	...	...	...	...	...	...	...
1375144	Holy Grail	20.0	3.5	3.0	3.5	3.0	3.0
1375145	Holy Grail	20.0	4.5	3.5	3.5	4.0	4.0
1375146	Holy Grail	20.0	4.0	4.5	4.0	5.0	4.5
1375147	Holy Grail	20.0	4.5	3.5	4.0	4.0	4.0
1199889	Short's Anniversary Ale 2006	20.0	4.0	4.5	4.0	4.5	4.0

756 rows × 7 columns

localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

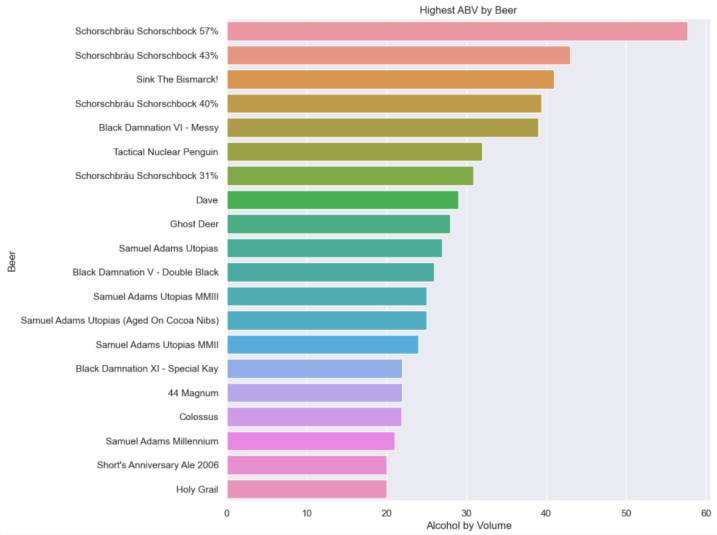
jupyter DAT_490_Beer_Reviews Last Checkpoint: 14 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted priceopt3

Run Code

In [28]:
```
## Highest ABV by Beer
sns.barplot(y='beer_name', x='beer_abv', data=df_high_abv, ci=None).set(title='Highest ABV by Beer', ylabel='Beer',
```

Out[28]:
```
[Text(0.5, 1.0, 'Highest ABV by Beer'),
 Text(0, 0.5, 'Beer'),
 Text(0.5, 0, 'Alcohol by Volume')]
```



localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

jupyter DAT_490_Beer_Reviews Last Checkpoint: 15 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help — Trusted — priceopt3

Run Code

In [29]:
```
df_high_abv.describe()
```

Out[29]:

	beer_abv	review_aroma	review_appearance	review_palate	review_taste	review_overall
count	756.000000	756.000000	756.000000	756.000000	756.000000	756.000000
mean	28.437513	4.222222	4.106481	4.059524	4.206349	3.878968
std	5.331012	0.745997	0.599866	0.833743	0.849082	0.963008
min	20.000000	1.000000	2.000000	1.000000	1.000000	1.000000
25%	26.000000	4.000000	4.000000	4.000000	4.000000	3.500000
50%	27.000000	4.500000	4.000000	4.000000	4.500000	4.000000
75%	29.000000	5.000000	4.500000	4.500000	5.000000	4.500000
max	57.700000	5.000000	5.000000	5.000000	5.000000	5.000000

In [30]:
```
## Correlation of all numeric variables
plt.figure(figsize=(10, 10))
sns.heatmap(df.corr(), annot=True)
```

Out[30]: <AxesSubplot: >

localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

jupyter DAT_490_Beer_Reviews Last Checkpoint: 16 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help — Trusted — priceopt3

Run Code

review_ov review_ar review_appeara review_pa review_ta beer_

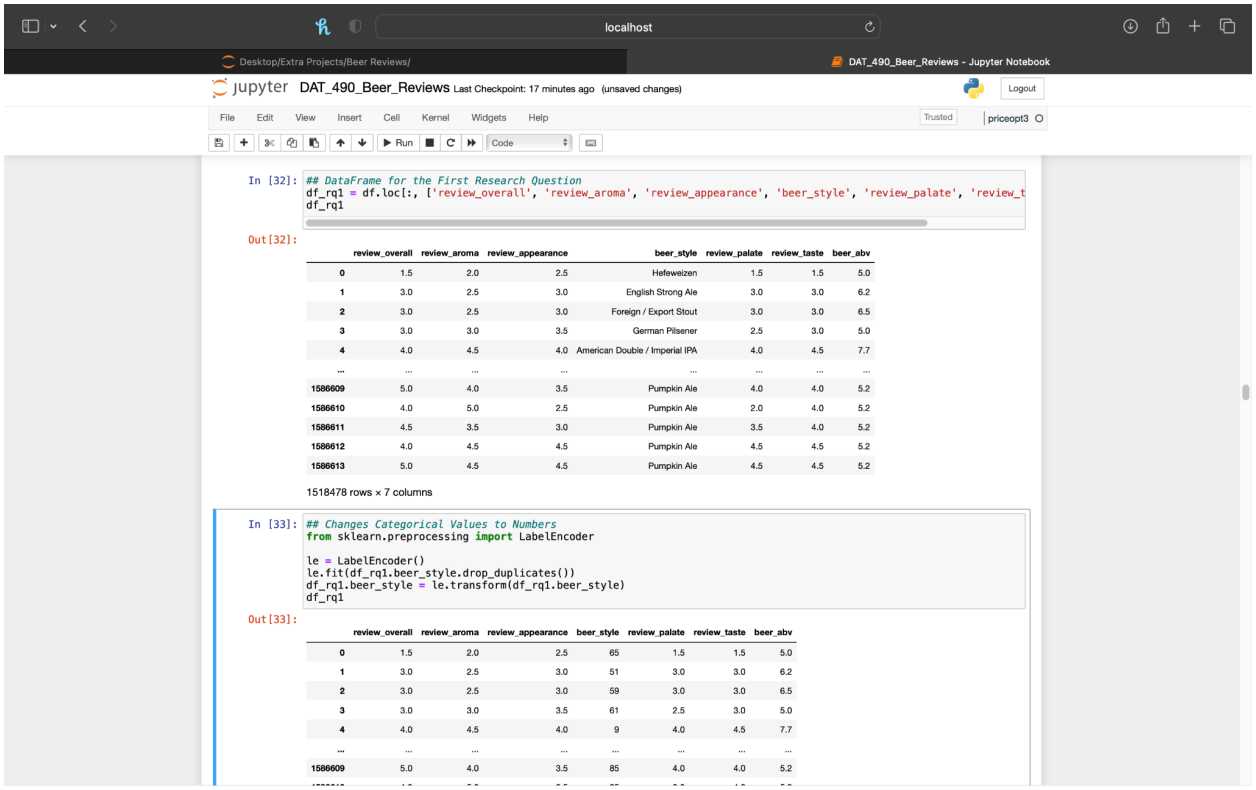
Data Preprocessing

In [31]:
```
df
```

Out[31]:

	brewery_name	review_overall	review_aroma	review_appearance	review_profilename	beer_style	review_palate	review_taste	beer_name	beer_abv
0	Vecchio Birraio	1.5	2.0	2.5	stcules	Hefeweizen	1.5	1.5	Sausa Weizen	5.0
1	Vecchio Birraio	3.0	2.5	3.0	stcules	English Strong Ale	3.0	3.0	Red Moon	6.2
2	Vecchio Birraio	3.0	2.5	3.0	stcules	Foreign / Export Stout	3.0	3.0	Black Horse Black Beer	6.5
3	Vecchio Birraio	3.0	3.0	3.5	stcules	German Pilsener	2.5	3.0	Sausa Pils	5.0
4	Caldera Brewing Company	4.0	4.5	4.0	johnmichaelsen	American Double / Imperial IPA	4.0	4.5	Cauldron DIPA	7.7
...	...	...	...	...	...	...	...	...	...	...
1586609	The Defiant Brewing Company	5.0	4.0	3.5	maddogruss	Pumpkin Ale	4.0	4.0	The Horseman's Ale	5.2
1586610	The Defiant Brewing Company	4.0	5.0	2.5	yelterdow	Pumpkin Ale	2.0	4.0	The Horseman's Ale	5.2
1586611	The Defiant Brewing Company	4.5	3.5	3.0	TongoRad	Pumpkin Ale	3.5	4.0	The Horseman's Ale	5.2
1586612	The Defiant Brewing Company	4.0	4.5	4.5	dherling	Pumpkin Ale	4.5	4.5	The Horseman's Ale	5.2
1586613	The Defiant Brewing Company	5.0	4.5	4.5	cbl2	Pumpkin Ale	4.5	4.5	The Horseman's Ale	5.2

1518478 rows × 10 columns



In [32]:
```
## DataFrame for the First Research Question
df_rq1 = df.loc[:, ['review_overall', 'review_aroma', 'review_appearance', 'beer_style', 'review_palate', 'review_t
df_rq1
```

Out[32]:

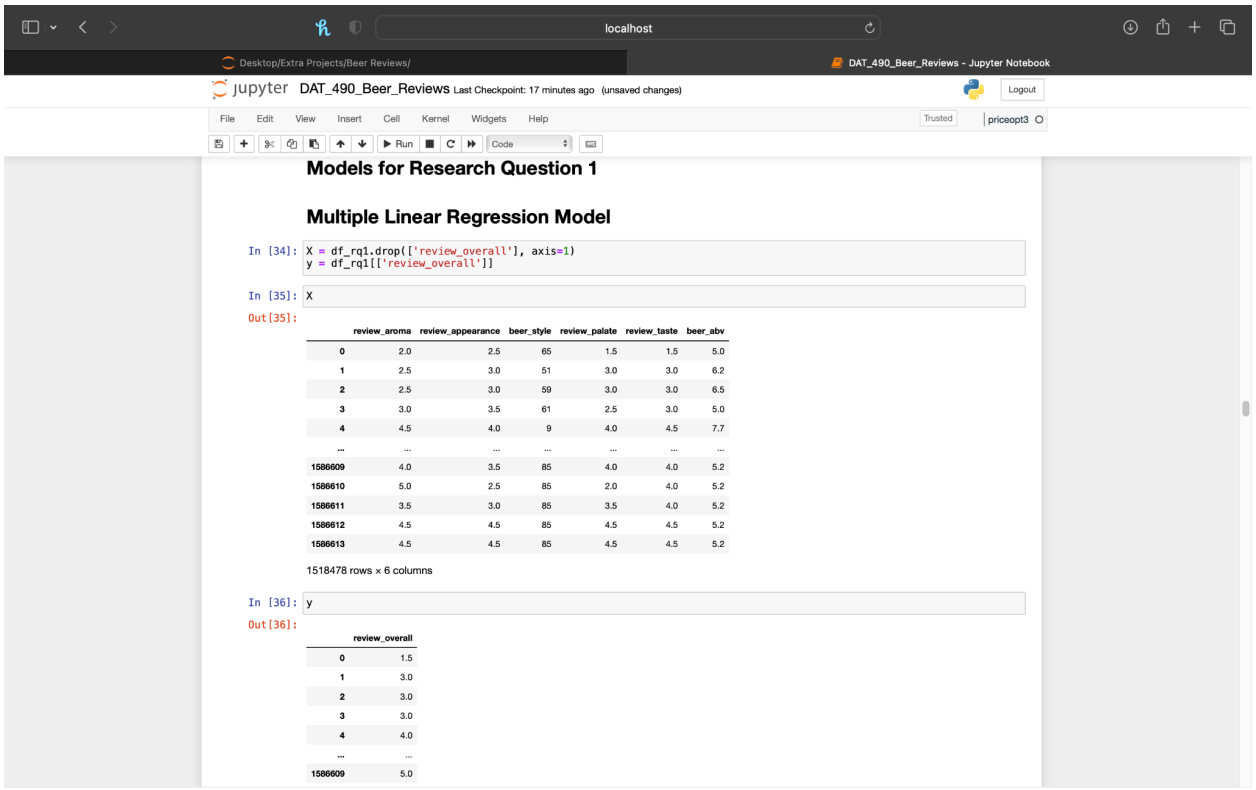
	review_overall	review_aroma	review_appearance	beer_style	review_palate	review_taste	beer_abv
0	1.5	2.0	2.5	Hefeweizen	1.5	1.5	5.0
1	3.0	2.5	3.0	English Strong Ale	3.0	3.0	6.2
2	3.0	2.5	3.0	Foreign / Export Stout	3.0	3.0	6.5
3	3.0	3.0	3.5	German Pilsener	2.5	3.0	5.0
4	4.0	4.5	4.0	American Double / Imperial IPA	4.0	4.5	7.7
...	...	...	...	...	...	...	...
1586609	5.0	4.0	3.5	Pumpkin Ale	4.0	4.0	5.2
1586610	4.0	5.0	2.5	Pumpkin Ale	2.0	4.0	5.2
1586611	4.5	3.5	3.0	Pumpkin Ale	3.5	4.0	5.2
1586612	4.0	4.5	4.5	Pumpkin Ale	4.5	4.5	5.2
1586613	5.0	4.5	4.5	Pumpkin Ale	4.5	4.5	5.2

1518478 rows × 7 columns

In [33]:
```
## Changes Categorical Values to Numbers
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
le.fit(df_rq1.beer_style.drop_duplicates())
df_rq1.beer_style = le.transform(df_rq1.beer_style)
df_rq1
```

Out[33]:

	review_overall	review_aroma	review_appearance	beer_style	review_palate	review_taste	beer_abv
0	1.5	2.0	2.5	65	1.5	1.5	5.0
1	3.0	2.5	3.0	51	3.0	3.0	6.2
2	3.0	2.5	3.0	59	3.0	3.0	6.5
3	3.0	3.0	3.5	61	2.5	3.0	5.0
4	4.0	4.5	4.0	9	4.0	4.5	7.7
...	...	...	...	...	...	...	...
1586609	5.0	4.0	3.5	85	4.0	4.0	5.2



Models for Research Question 1

Multiple Linear Regression Model

In [34]:
```
X = df_rq1.drop(['review_overall'], axis=1)
y = df_rq1[['review_overall']]
```

In [35]:
```
X
```

Out[35]:

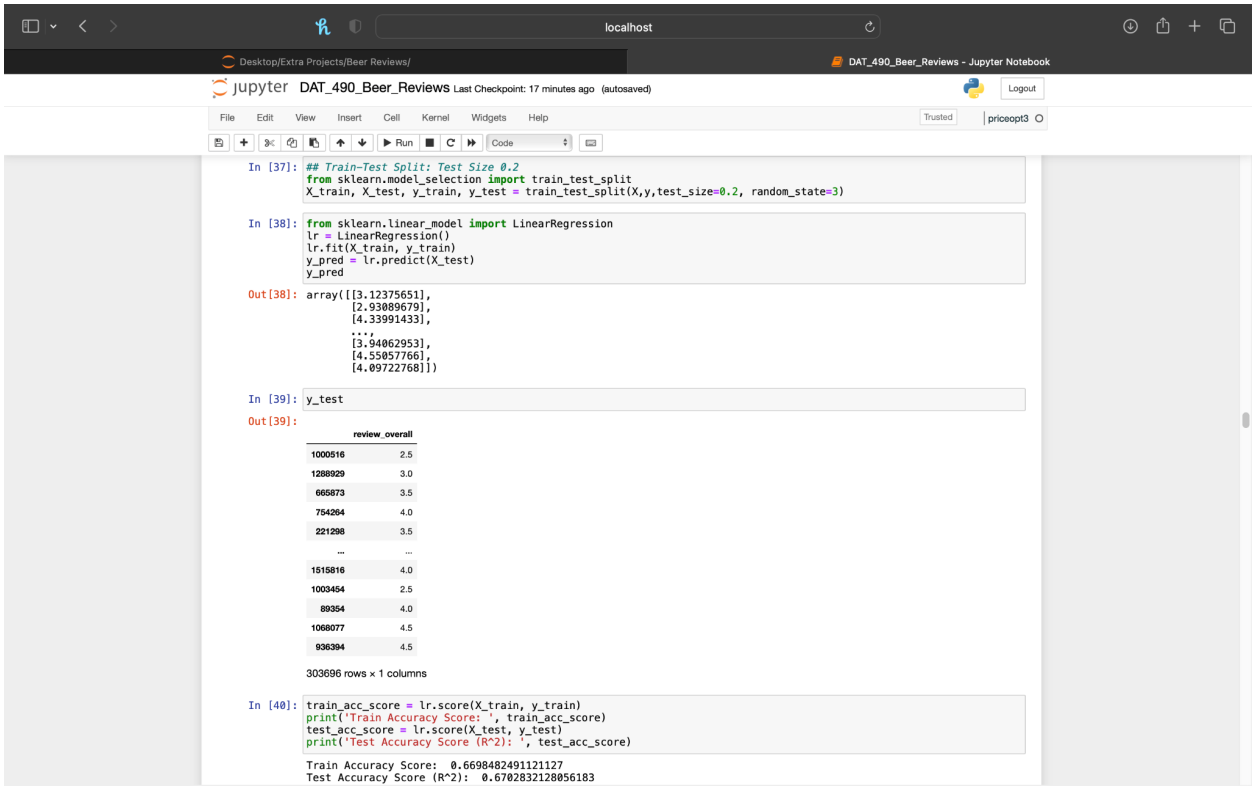
	review_aroma	review_appearance	beer_style	review_palate	review_taste	beer_abv
0	2.0	2.5	65	1.5	1.5	5.0
1	2.5	3.0	51	3.0	3.0	6.2
2	2.5	3.0	59	3.0	3.0	6.5
3	3.0	3.5	61	2.5	3.0	5.0
4	4.5	4.0	9	4.0	4.5	7.7
...	...	...	...	...	...	...
1586609	4.0	3.5	85	4.0	4.0	5.2
1586610	5.0	2.5	85	2.0	4.0	5.2
1586611	3.5	3.0	85	3.5	4.0	5.2
1586612	4.5	4.5	85	4.5	4.5	5.2
1586613	4.5	4.5	85	4.5	4.5	5.2

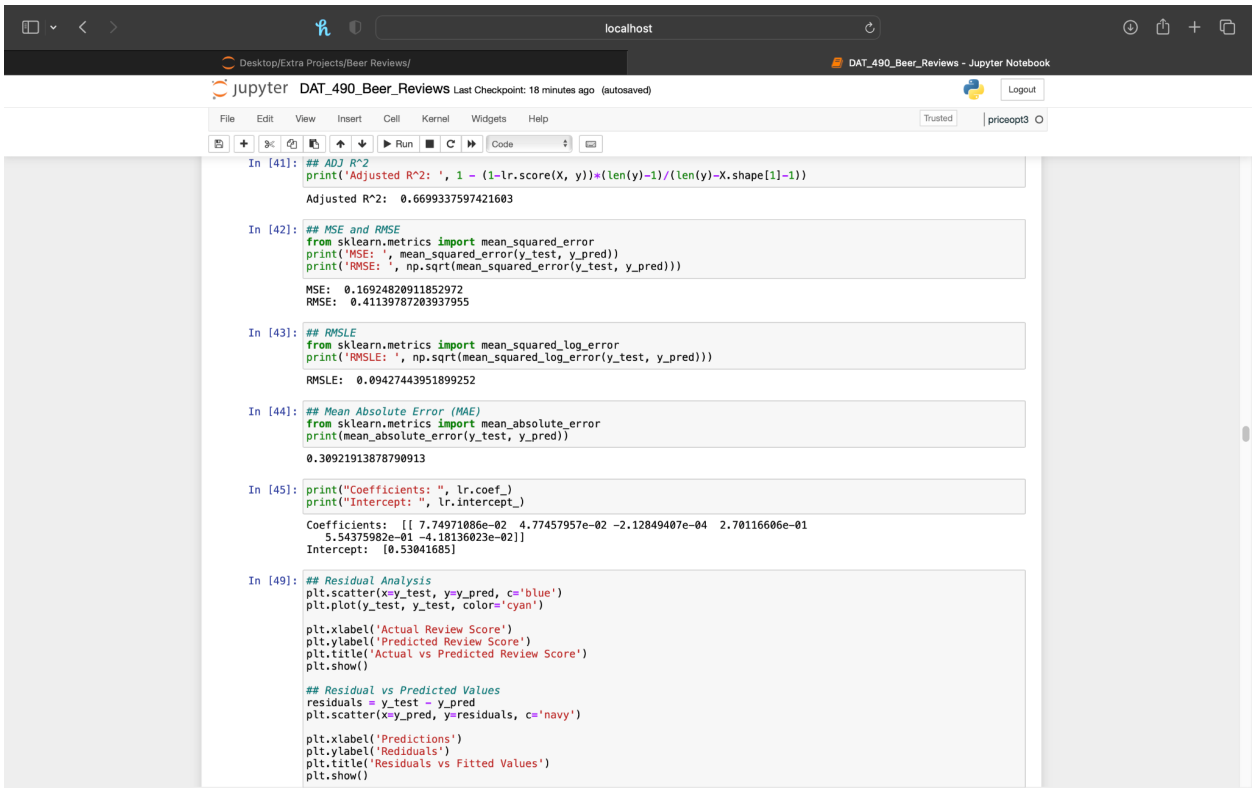
1518478 rows × 6 columns

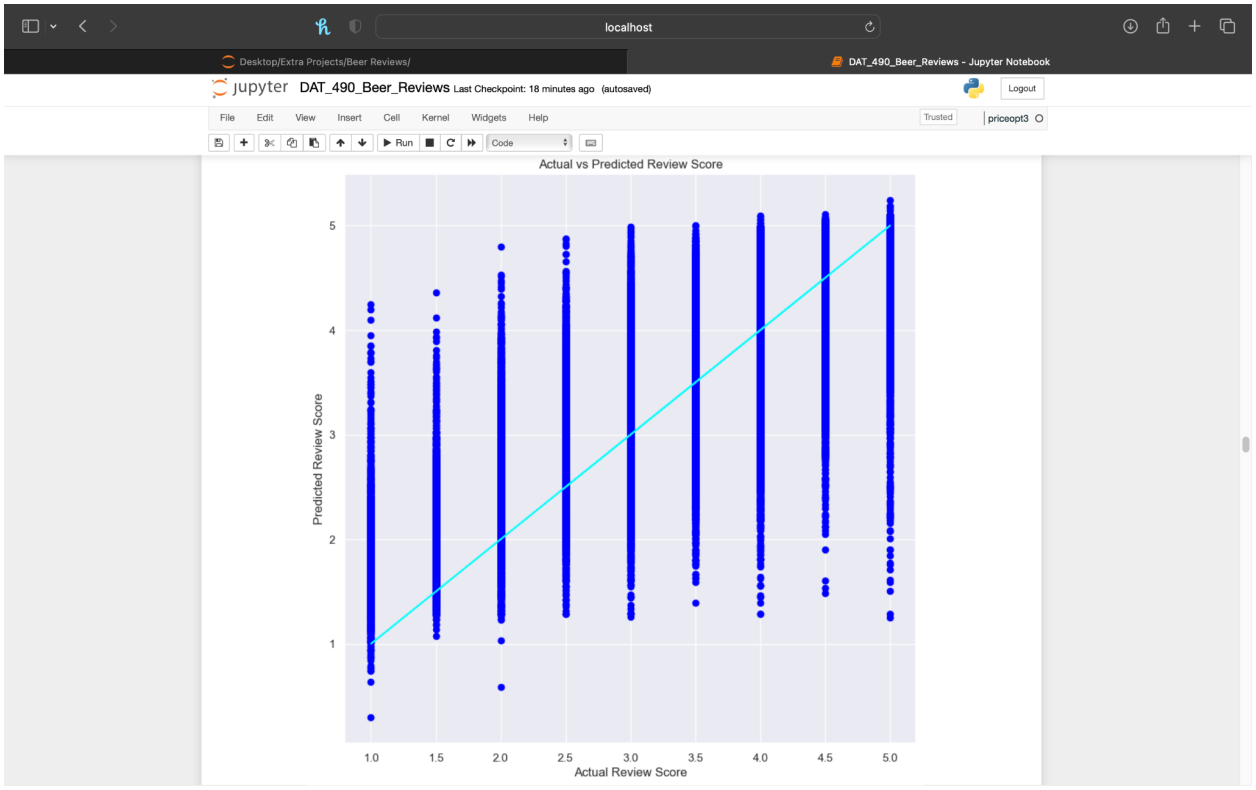
In [36]:
```
y
```

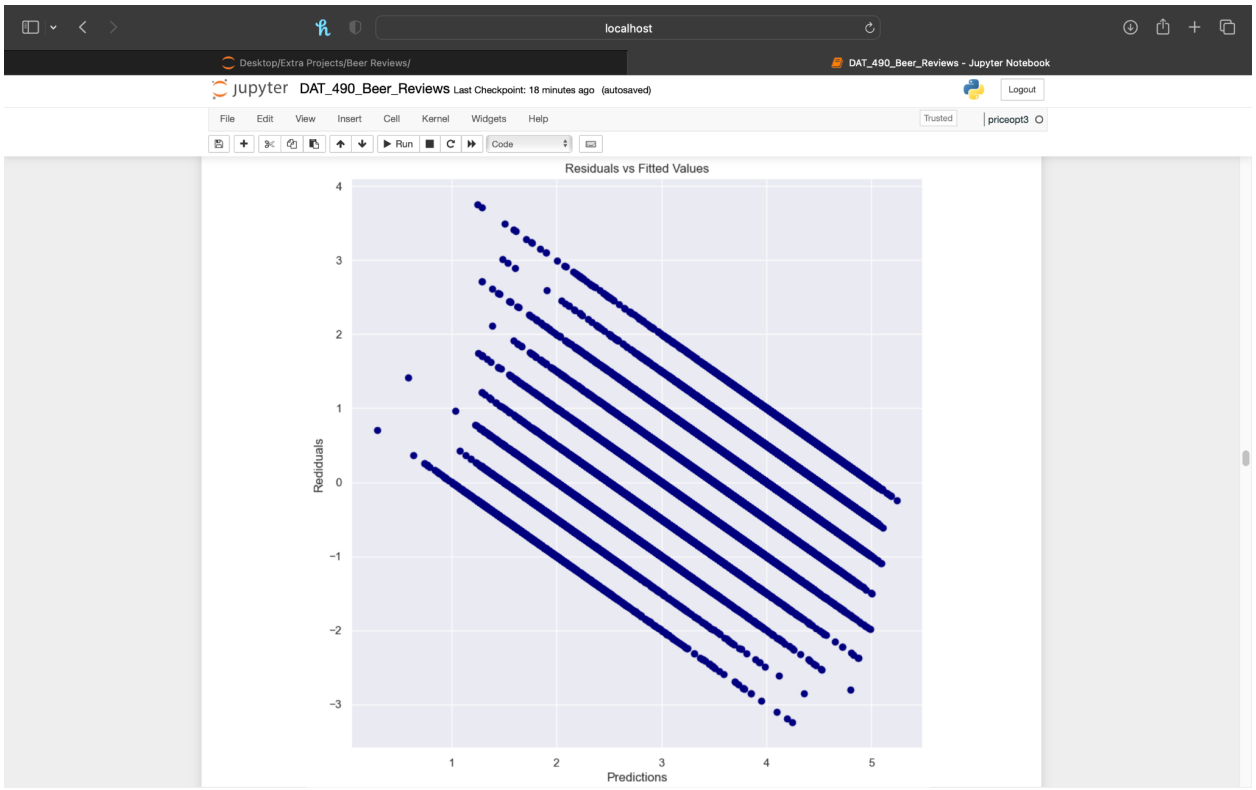
Out[36]:

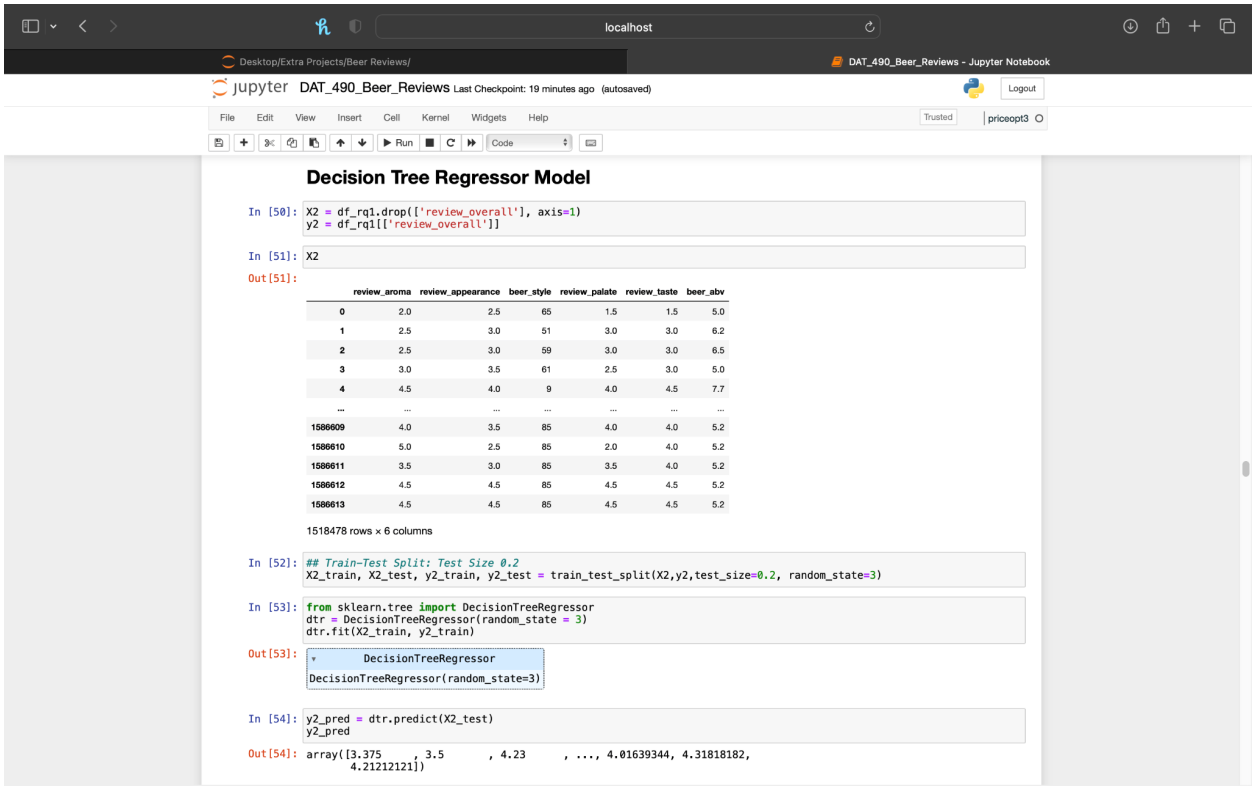
	review_overall
0	1.5
1	3.0
2	3.0
3	3.0
4	4.0
...	...
1586609	5.0

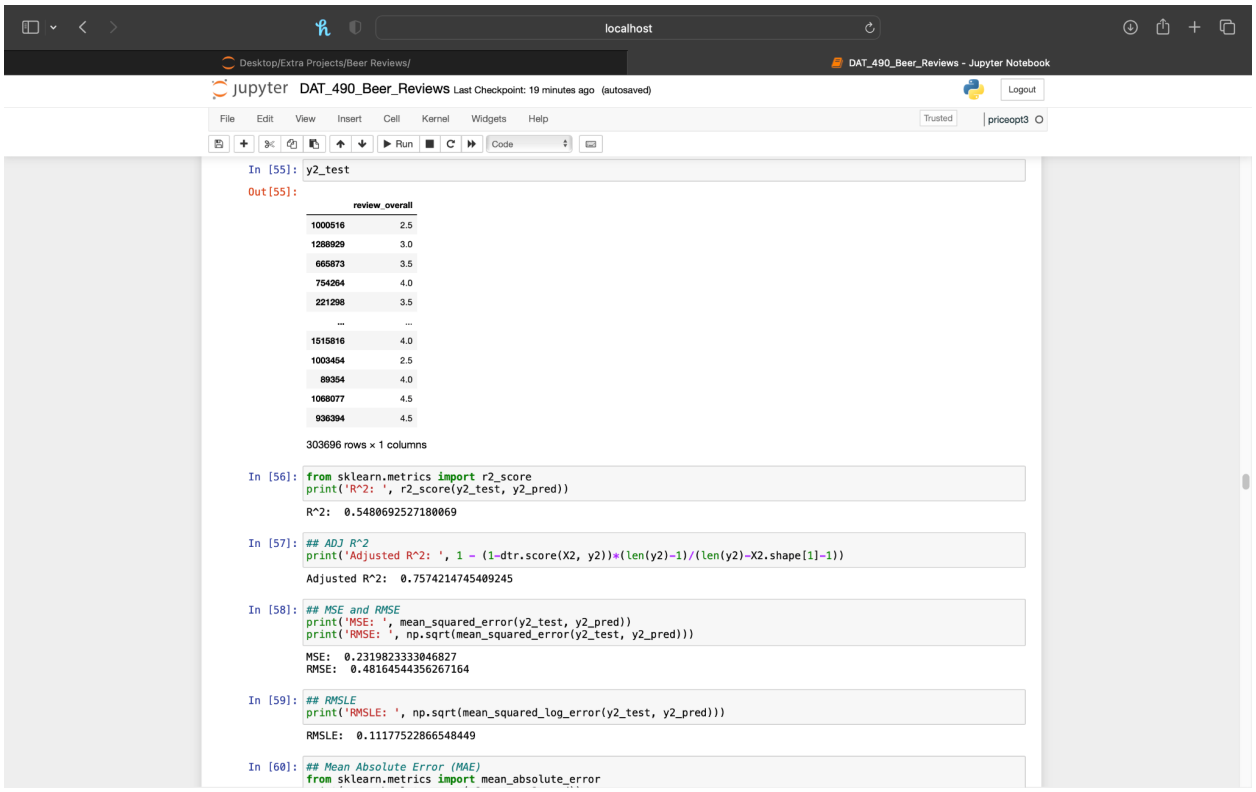

localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 17 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
In [37]:
Train-Test Split: Test Size 0.2
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2, random_state=3)
In [38]:
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)
y_pred
Out[38]:
array([[3.12375651],
[2.93089679],
[4.33991433],
...,
[3.94062953],
[4.55057766],
[4.09722768]])
In [39]:
y_test
Out[39]:
review_overall
1000516 2.5
1288929 3.0
665873 3.5
754264 4.0
221298 3.5
... ...
1515816 4.0
1003454 2.5
89354 4.0
1068077 4.5
936394 4.5
303696 rows × 1 columns
In [40]:
train_acc_score = lr.score(X_train, y_train)
print('Train Accuracy Score: ', train_acc_score)
test_acc_score = lr.score(X_test, y_test)
print('Test Accuracy Score (R^2): ', test_acc_score)
Train Accuracy Score: 0.669848249112 1127
Test Accuracy Score (R^2): 0.6702832128056183


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 18 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
In [41]:
ADJ R^2
print('Adjusted R^2: ', 1 - (1-lr.score(X, y))*(len(y)-1)/(len(y)-X.shape[1]-1))
Adjusted R^2: 0.6699337597421603
In [42]:
MSE and RMSE
from sklearn.metrics import mean_squared_error
print('MSE: ', mean_squared_error(y_test, y_pred))
print('RMSE: ', np.sqrt(mean_squared_error(y_test, y_pred)))
MSE: 0.16924820911852972
RMSE: 0.41139787203937955
In [43]:
RMSLE
from sklearn.metrics import mean_squared_log_error
print('RMSLE: ', np.sqrt(mean_squared_log_error(y_test, y_pred)))
RMSLE: 0.09427443951899252
In [44]:
Mean Absolute Error (MAE)
from sklearn.metrics import mean_absolute_error
print(mean_absolute_error(y_test, y_pred))
0.30921913878790913
In [45]:
print("Coefficients: ", lr.coef_)
print("Intercept: ", lr.intercept_)
Coefficients: [[7.74971086e-02 4.77457957e-02 -2.12849407e-04 2.70116606e-01
5.54375982e-01 -4.18136023e-02]]
Intercept: [0.53041685]
In [49]:
Residual Analysis
plt.scatter(x=y_test, y=y_pred, c='blue')
plt.plot(y_test, y_test, color='cyan')
plt.xlabel('Actual Review Score')
plt.ylabel('Predicted Review Score')
plt.title('Actual vs Predicted Review Score')
plt.show()
Residual vs Predicted Values
residuals = y_test - y_pred
plt.scatter(x=y_pred, y=residuals, c='navy')
plt.xlabel('Predictions')
plt.ylabel('Rediduals')
plt.title('Residuals vs Fitted Values')
plt.show()


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 18 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Actual vs Predicted Review Score
Predicted Review Score
Actual Review Score
1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
1 2 3 4 5


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 18 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Residuals vs Fitted Values
Rediduals
Predictions
1 2 3 4 5
−3 −2 −1 0 1 2 3 4


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 19 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Decision Tree Regressor Model
In [50]:
X2 = df_rq1.drop(['review_overall'], axis=1)
y2 = df_rq1[['review_overall']]
In [51]:
X2
Out[51]:
review_aroma review_appearance beer_style review_palate review_taste beer_abv
0 2.0 2.5 65 1.5 1.5 5.0
1 2.5 3.0 51 3.0 3.0 6.2
2 2.5 3.0 59 3.0 3.0 6.5
3 3.0 3.5 61 2.5 3.0 5.0
4 4.5 4.0 9 4.0 4.5 7.7
... ...
1586609 4.0 3.5 85 4.0 4.0 5.2
1586610 5.0 2.5 85 2.0 4.0 5.2
1586611 3.5 3.0 85 3.5 4.0 5.2
1586612 4.5 4.5 85 4.5 4.5 5.2
1586613 4.5 4.5 85 4.5 4.5 5.2
1518478 rows × 6 columns
In [52]:
Train-Test Split: Test Size 0.2
X2_train, X2_test, y2_train, y2_test = train_test_split(X2,y2,test_size=0.2, random_state=3)
In [53]:
from sklearn.tree import DecisionTreeRegressor
dtr = DecisionTreeRegressor(random_state = 3)
dtr.fit(X2_train, y2_train)
Out[53]:
DecisionTreeRegressor
DecisionTreeRegressor(random_state=3)
In [54]:
y2_pred = dtr.predict(X2_test)
y2_pred
Out[54]:
array([3.375 , 3.5 , 4.23 , ..., 4.01639344, 4.31818182,
4.21212121])


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 19 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
In [55]:
y2_test
Out[55]:
review_overall
1000516 2.5
1288929 3.0
665873 3.5
754264 4.0
221298 3.5
... ...
1515816 4.0
1003454 2.5
89354 4.0
1068077 4.5
936394 4.5
303696 rows × 1 columns
In [56]:
from sklearn.metrics import r2_score
print('R^2: ', r2_score(y2_test, y2_pred))
R^2: 0.5480692527180069
In [57]:
ADJ R^2
print('Adjusted R^2: ', 1 - (1-dtr.score(X2, y2))*(len(y2)-1)/(len(y2)-X2.shape[1]-1))
Adjusted R^2: 0.7574214745409245
In [58]:
MSE and RMSE
print('MSE: ', mean_squared_error(y2_test, y2_pred))
print('RMSE: ', np.sqrt(mean_squared_error(y2_test, y2_pred)))
MSE: 0.2319823333046827
RMSE: 0.48164544356267164
In [59]:
RMSLE
print('RMSLE: ', np.sqrt(mean_squared_log_error(y2_test, y2_pred)))
RMSLE: 0.11177522866548449
In [60]:
Mean Absolute Error (MAE)
from sklearn.metrics import mean_absolute_error

localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

jupyter DAT_490_Beer_Reviews Last Checkpoint: 19 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted priceopt3

Run Code

In [60]:
```
## Mean Absolute Error (MAE)
from sklearn.metrics import mean_absolute_error
print(mean_absolute_error(y2_test, y2_pred))
```

```
0.34773093847201736
```

In [67]:
```
## Residual Analysis
plt.scatter(x=y2_test, y=y2_pred, c='blue')
plt.plot(y2_test, y2_test, color='cyan')

plt.xlabel('Actual Review Score')
plt.ylabel('Predicted Review Score')
plt.title('Actual vs Predicted Review Score')
plt.show()

## Residual vs Predicted Values
y2_pred = y2_pred.reshape(303696,1)
residuals = y2_test - y2_pred
plt.scatter(x=y2_pred, y=residuals, c='navy')

plt.xlabel('Predictions')
plt.ylabel('Rediduals')
plt.title('Residuals vs Fitted Values')
plt.show()
```

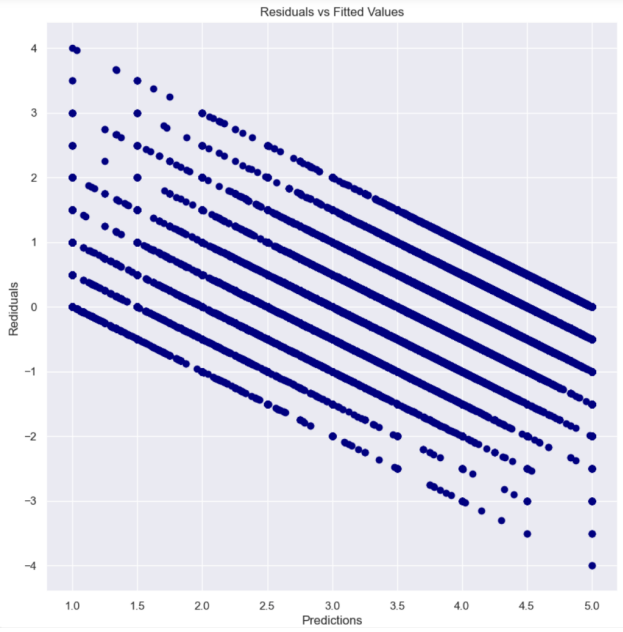
localhost

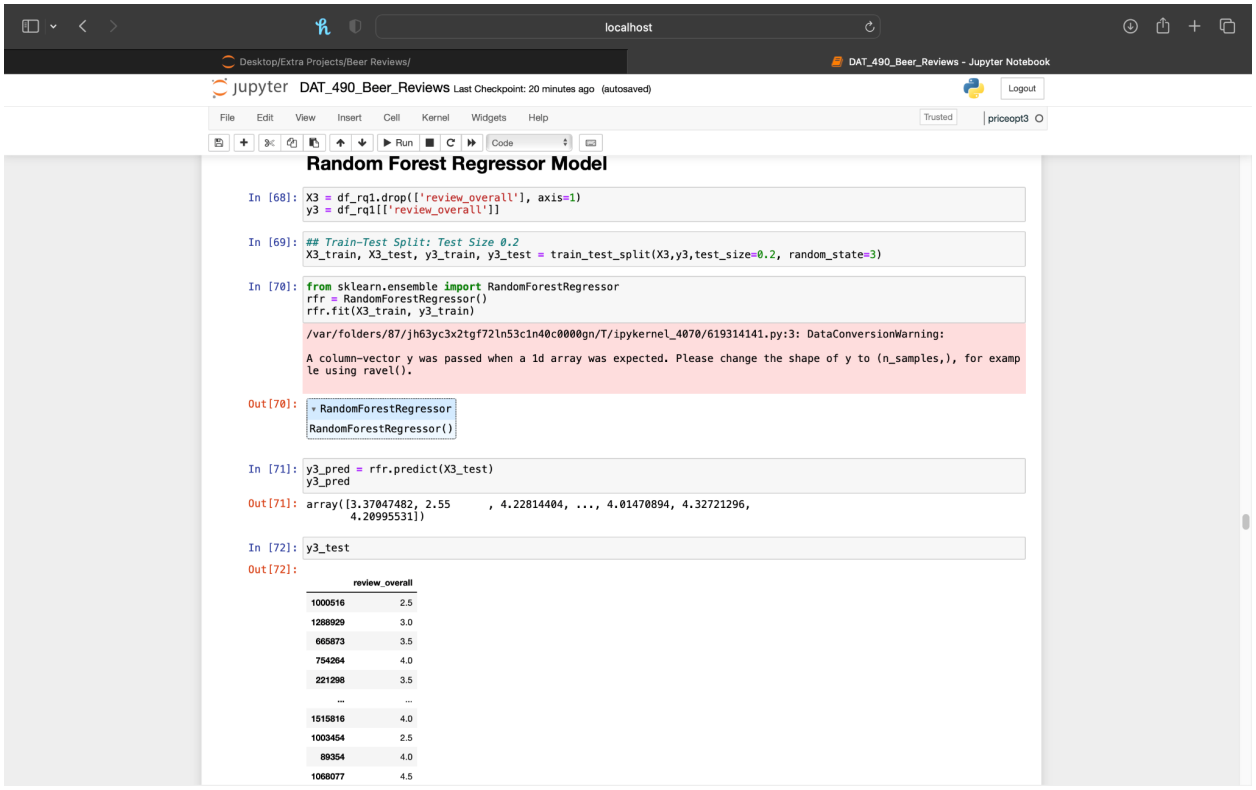
Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

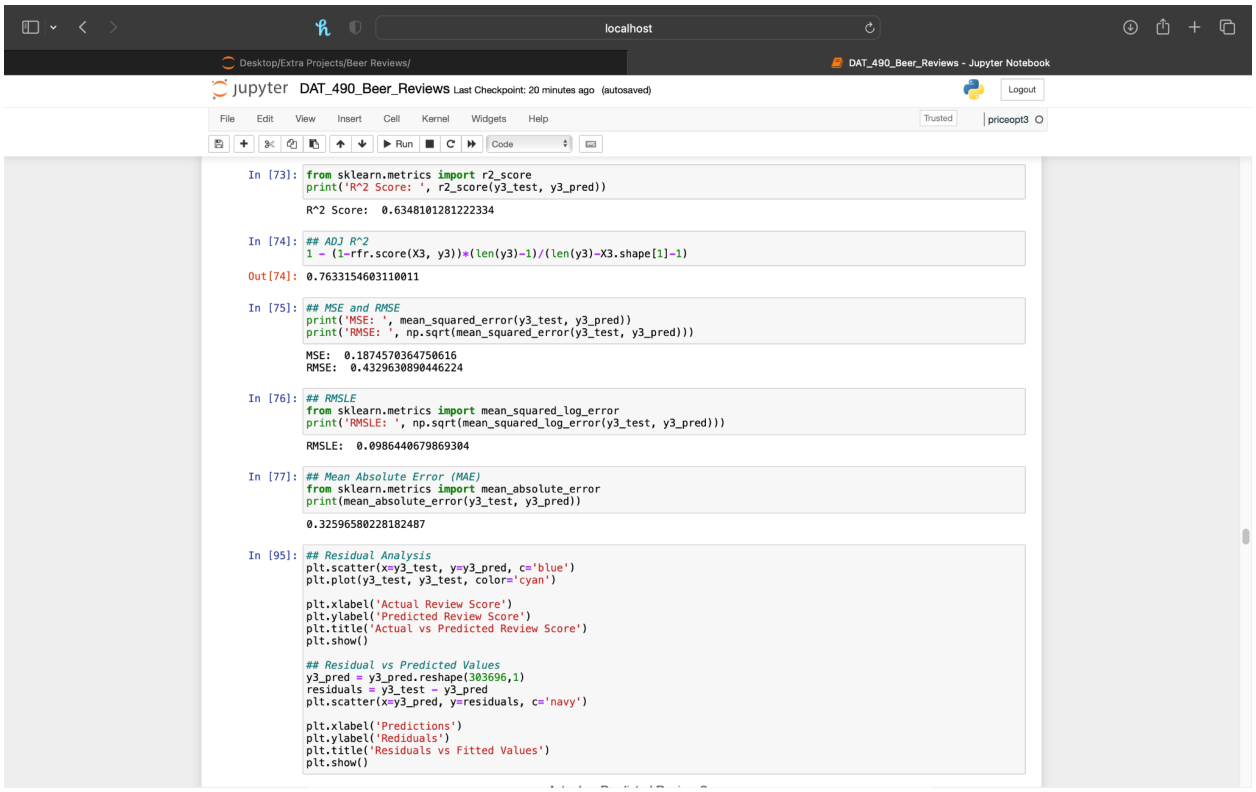
jupyter DAT_490_Beer_Reviews Last Checkpoint: 20 minutes ago (autosaved) Logout

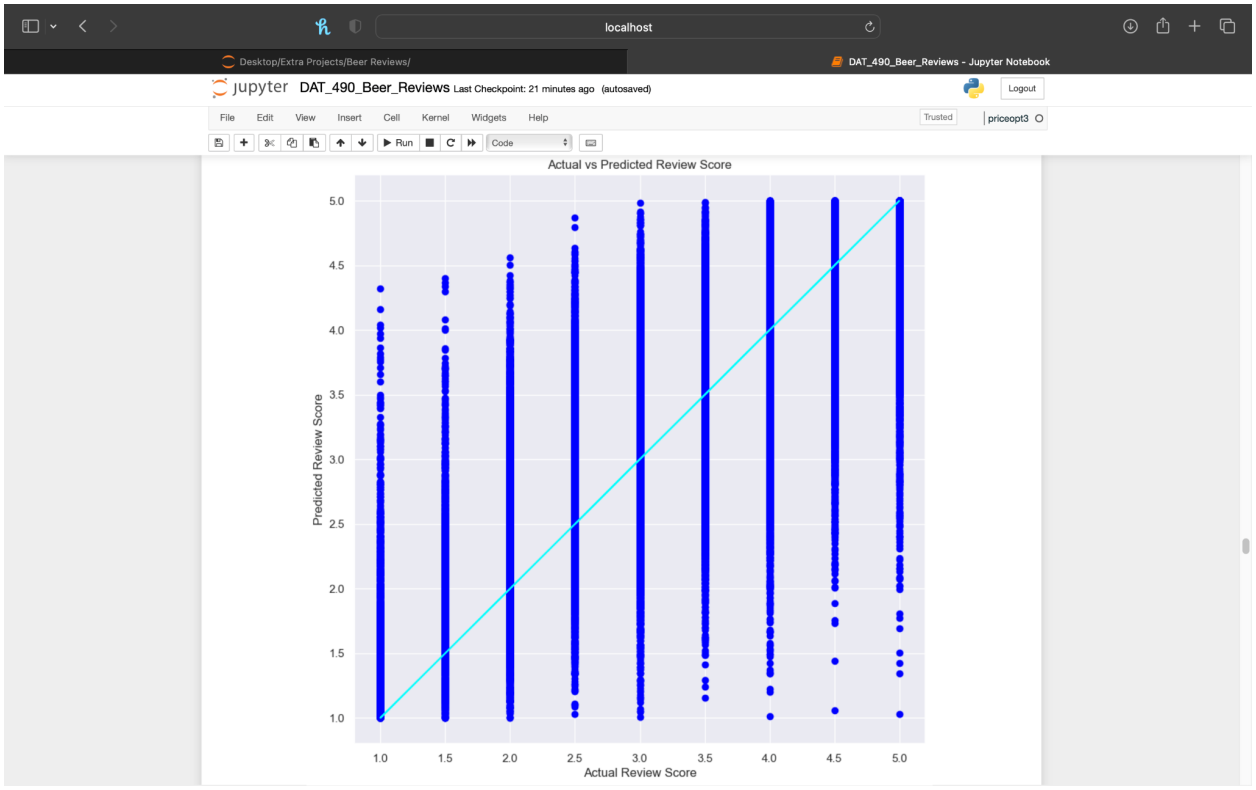
File Edit View Insert Cell Kernel Widgets Help Trusted priceopt3

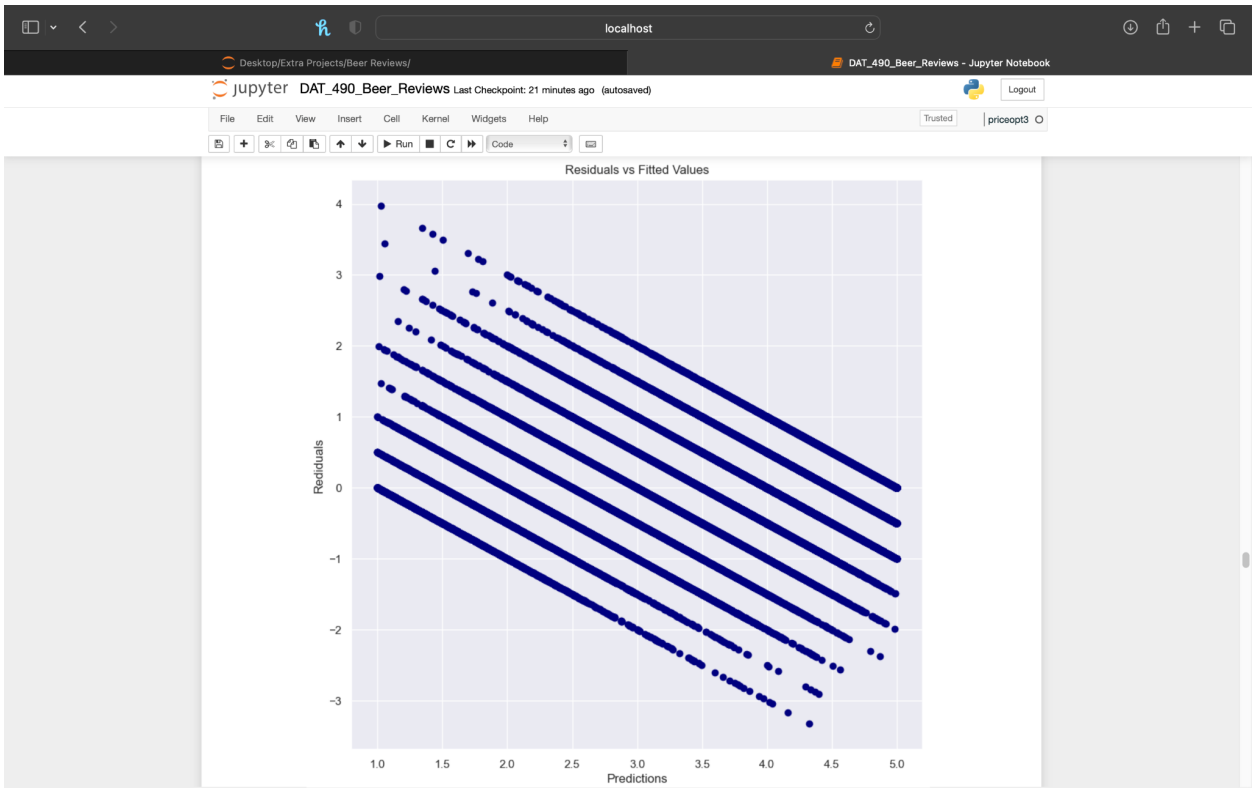
Run Code

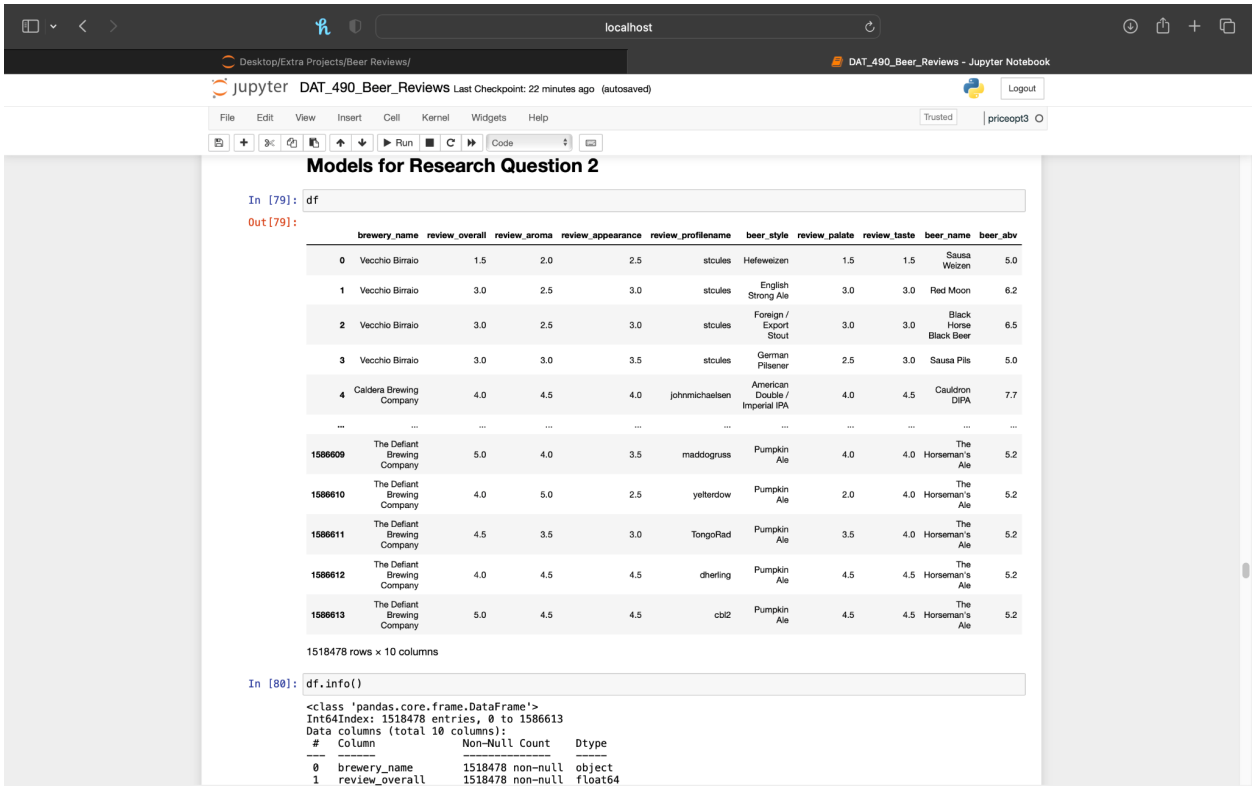


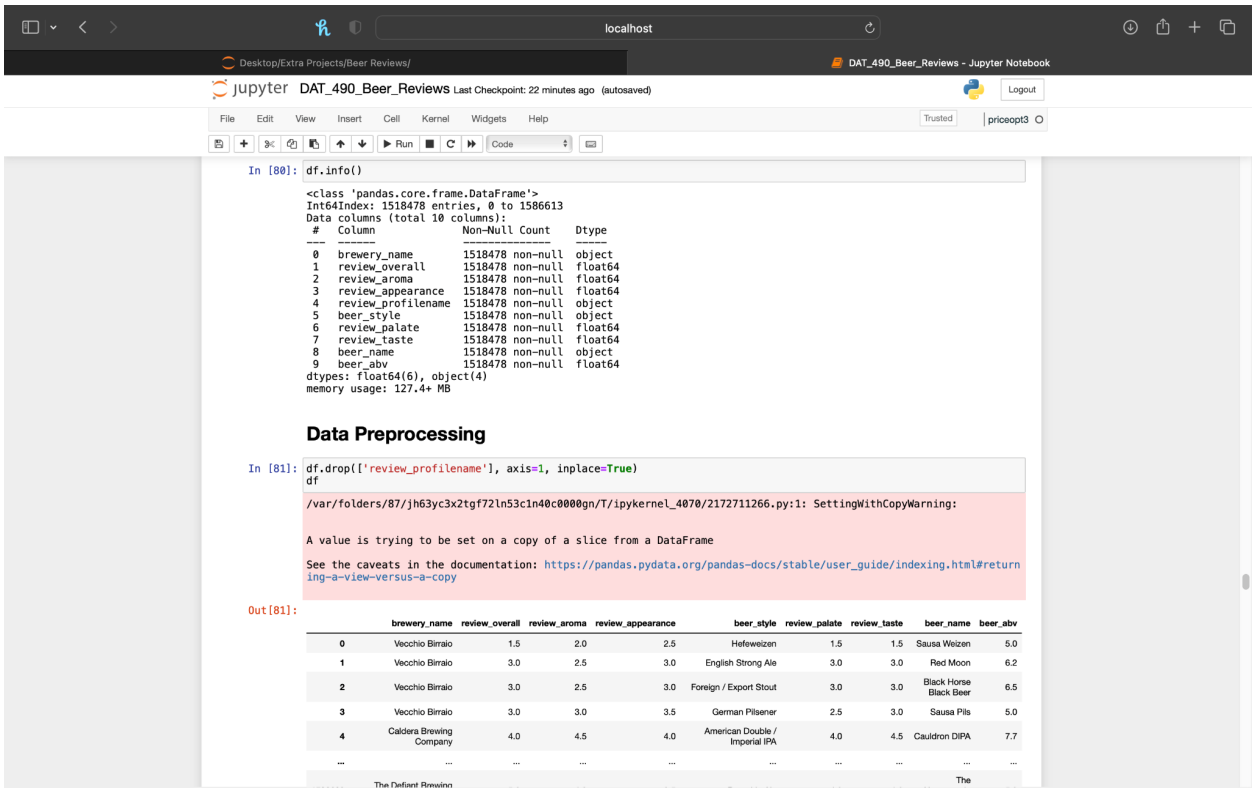

localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 20 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Random Forest Regressor Model
In [68]:
X3 = df_rq1.drop(['review_overall'], axis=1)
y3 = df_rq1[['review_overall']]
In [69]:
Train-Test Split: Test Size 0.2
X3_train, X3_test, y3_train, y3_test = train_test_split(X3,y3,test_size=0.2, random_state=3)
In [70]:
from sklearn.ensemble import RandomForestRegressor
rfr = RandomForestRegressor()
rfr.fit(X3_train, y3_train)
/var/folders/87/jh63yc3x2tgf72ln53c1n40c0000gn/T/ipykernel_4070/619314141.py:3: DataConversionWarning:
A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for examp
le using ravel().
Out[70]:
RandomForestRegressor
RandomForestRegressor()
In [71]:
y3_pred = rfr.predict(X3_test)
y3_pred
Out[71]:
array([3.37047482, 2.55 , 4.22814404, ..., 4.01470894, 4.32721296,
4.20995531])
In [72]:
y3_test
Out[72]:
review_overall
1000516 2.5
1288929 3.0
665873 3.5
754264 4.0
221298 3.5
... ...
1515816 4.0
1003454 2.5
89354 4.0
1068077 4.5

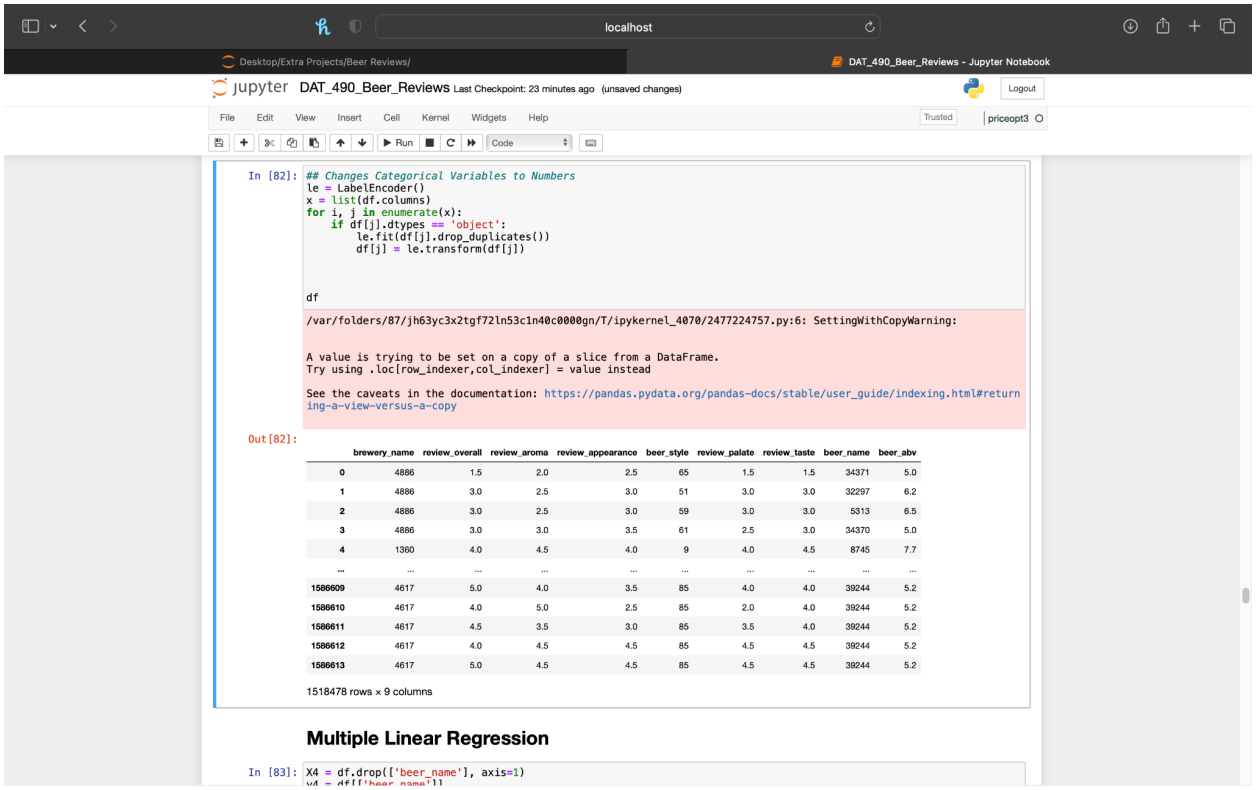

localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 20 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
In [73]:
from sklearn.metrics import r2_score
print('R^2 Score: ', r2_score(y3_test, y3_pred))
R^2 Score: 0.6348101281222334
In [74]:
ADJ R^2
1 - (1-rfr.score(X3, y3))*(len(y3)-1)/(len(y3)-X3.shape[1]-1)
Out[74]:
0.7633154603110011
In [75]:
MSE and RMSE
print('MSE: ', mean_squared_error(y3_test, y3_pred))
print('RMSE: ', np.sqrt(mean_squared_error(y3_test, y3_pred)))
MSE: 0.18745703647506 16
RMSE: 0.4329630890446224
In [76]:
RMSLE
from sklearn.metrics import mean_squared_log_error
print('RMSLE: ', np.sqrt(mean_squared_log_error(y3_test, y3_pred)))
RMSLE: 0.0986440679869304
In [77]:
Mean Absolute Error (MAE)
from sklearn.metrics import mean_absolute_error
print(mean_absolute_error(y3_test, y3_pred))
0.3259658022818248 7
In [95]:
Residual Analysis
plt.scatter(x=y3_test, y=y3_pred, c='blue')
plt.plot(y3_test, y3_test, color='cyan')
plt.xlabel('Actual Review Score')
plt.ylabel('Predicted Review Score')
plt.title('Actual vs Predicted Review Score')
plt.show()
Residual vs Predicted Values
y3_pred = y3_pred.reshape(303696,1)
residuals = y3_test - y3_pred
plt.scatter(x=y3_pred, y=residuals, c='navy')
plt.xlabel('Predictions')
plt.ylabel('Rediduals')
plt.title('Residuals vs Fitted Values')
plt.show()

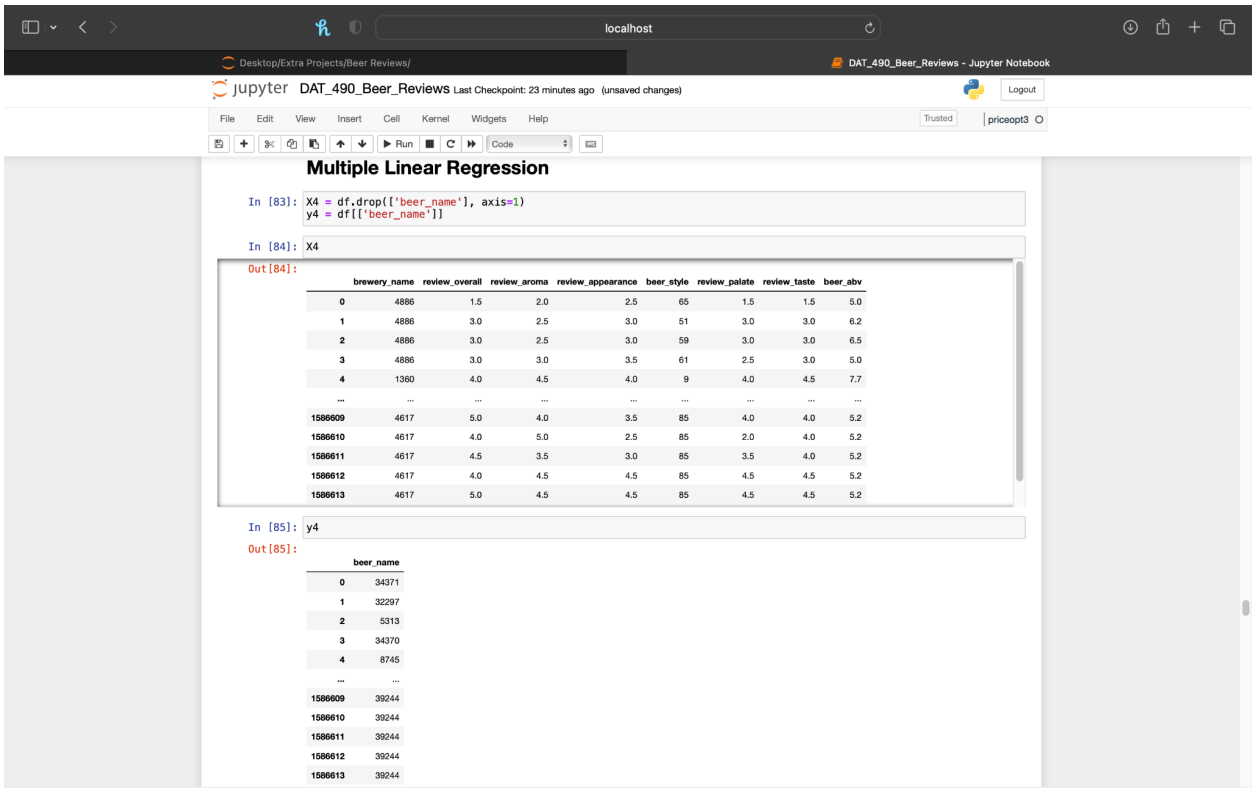

localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 21 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Actual vs Predicted Review Score
Predicted Review Score
Actual Review Score
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

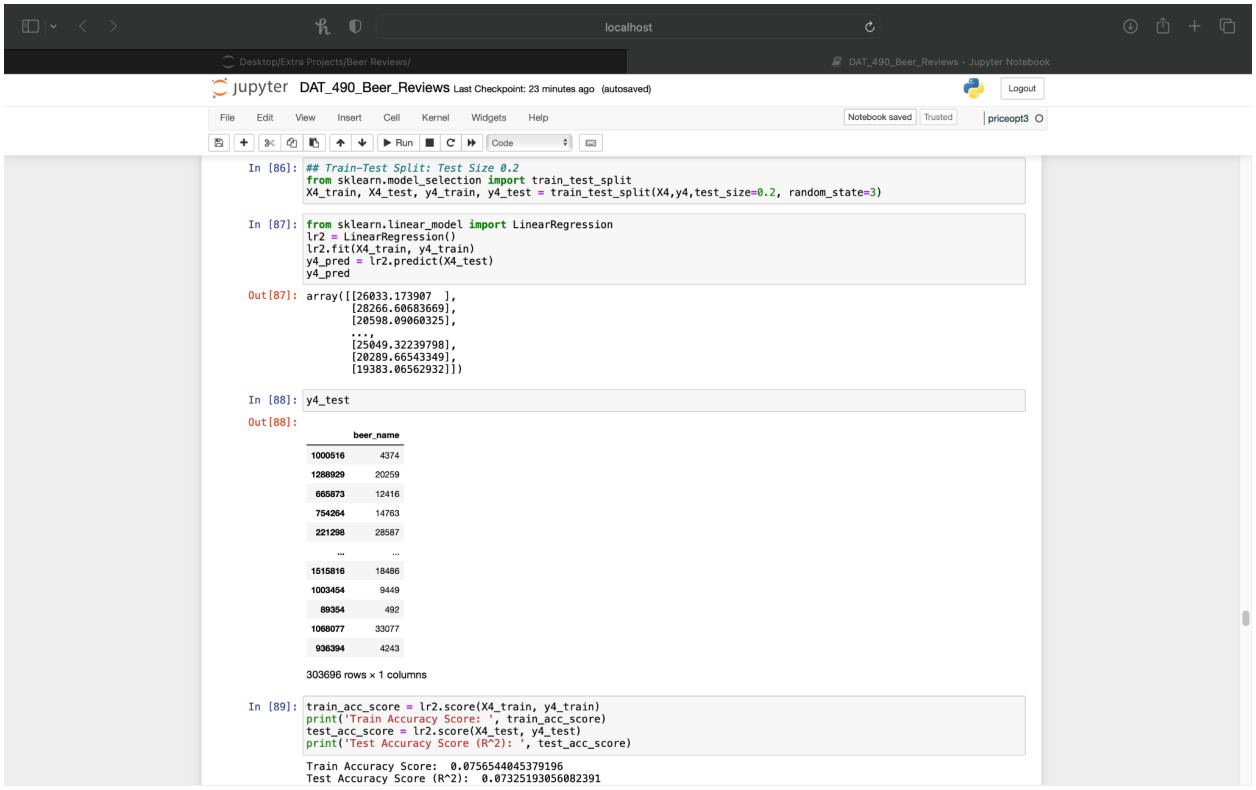

localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 21 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Residuals vs Fitted Values
Rediduals
Predictions
4
3
2
1
0
−1
−2
−3
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

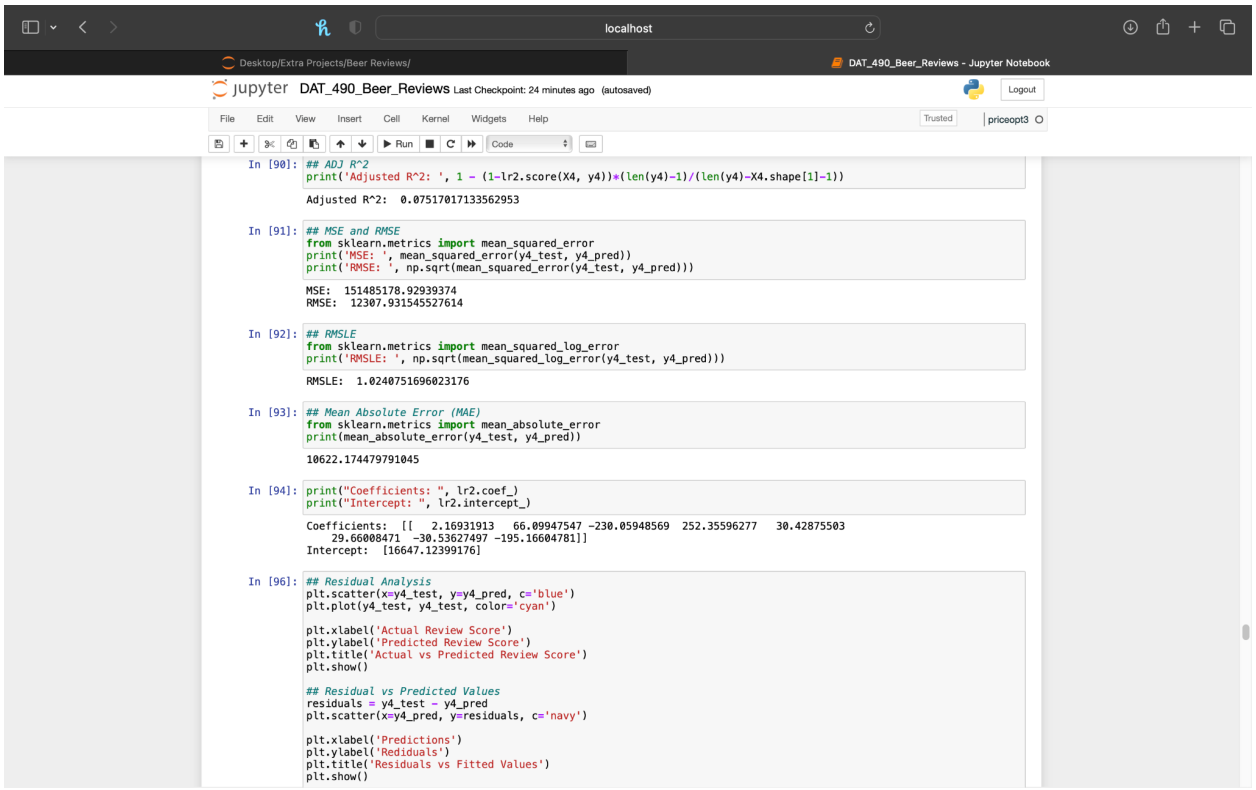

localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 22 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Models for Research Question 2
In [79]: df
Out[79]:
brewery_name review_overall review_aroma review_appearance review_profilename beer_style review_palate review_taste beer_name beer_abv
0 Vecchio Birraio 1.5 2.0 2.5 stcules Hefeweizen 1.5 1.5 Sausa Weizen 5.0
1 Vecchio Birraio 3.0 2.5 3.0 stcules English Strong Ale 3.0 3.0 Red Moon 6.2
2 Vecchio Birraio 3.0 2.5 3.0 stcules Foreign / Export Stout 3.0 3.0 Black Horse Black Beer 6.5
3 Vecchio Birraio 3.0 3.0 3.5 stcules German Pilsener 2.5 3.0 Sausa Pils 5.0
4 Caldera Brewing Company 4.0 4.5 4.0 johnmichaelsen American Double / Imperial IPA 4.0 4.5 Cauldron DIPA 7.7
... ...
1586609 The Defiant Brewing Company 5.0 4.0 3.5 maddogruss Pumpkin Ale 4.0 4.0 The Horseman's Ale 5.2
1586610 The Defiant Brewing Company 4.0 5.0 2.5 yelterdow Pumpkin Ale 2.0 4.0 The Horseman's Ale 5.2
1586611 The Defiant Brewing Company 4.5 3.5 3.0 TongoRad Pumpkin Ale 3.5 4.0 The Horseman's Ale 5.2
1586612 The Defiant Brewing Company 4.0 4.5 4.5 dherling Pumpkin Ale 4.5 4.5 The Horseman's Ale 5.2
1586613 The Defiant Brewing Company 5.0 4.5 4.5 cbl2 Pumpkin Ale 4.5 4.5 The Horseman's Ale 5.2
1518478 rows × 10 columns
In [80]: df.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1518478 entries, 0 to 1586613
Data columns (total 10 columns):
Column Non-Null Count Dtype
0 brewery_name 1518478 non-null object
1 review_overall 1518478 non-null float64

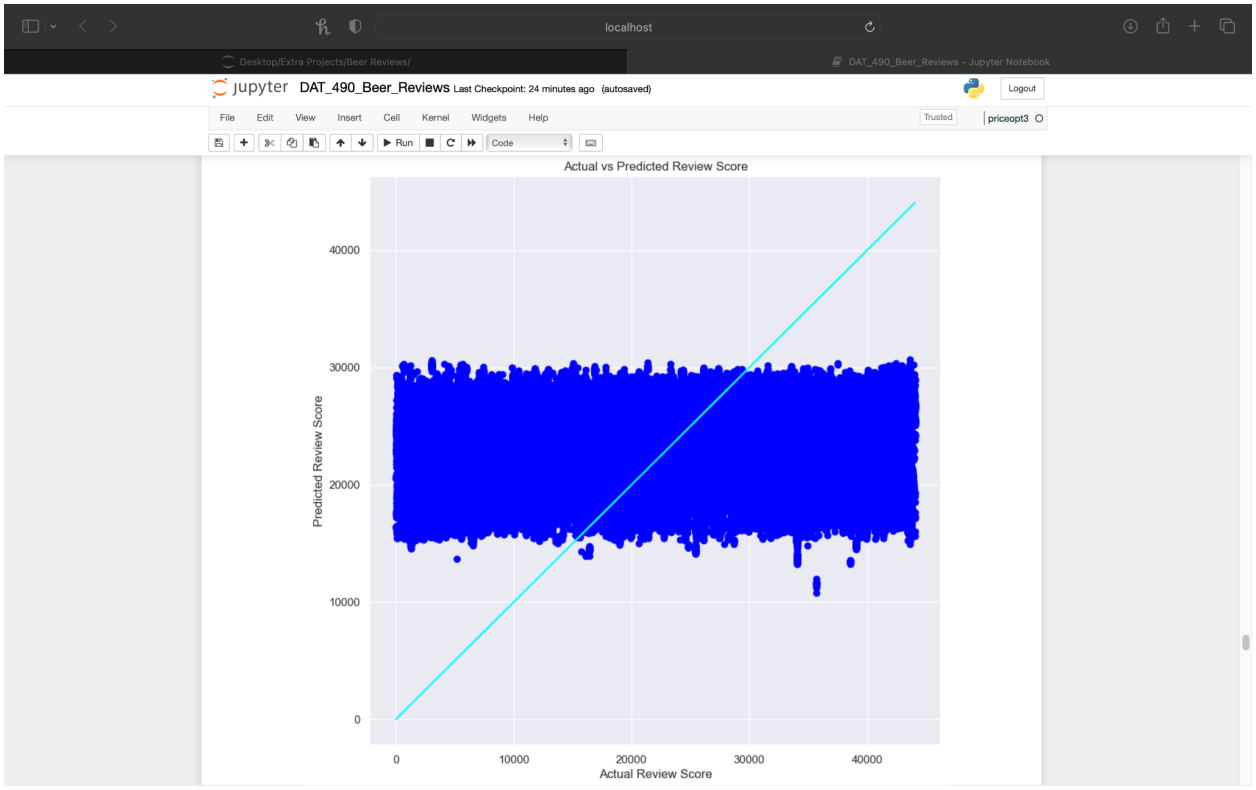

localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 22 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
In [80]: df.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1518478 entries, 0 to 1586613
Data columns (total 10 columns):
Column Non-Null Count Dtype
0 brewery_name 1518478 non-null object
1 review_overall 1518478 non-null float64
2 review_aroma 1518478 non-null float64
3 review_appearance 1518478 non-null float64
4 review_profilename 1518478 non-null object
5 beer_style 1518478 non-null object
6 review_palate 1518478 non-null float64
7 review_taste 1518478 non-null float64
8 beer_name 1518478 non-null object
9 beer_abv 1518478 non-null float64
dtypes: float64(6), object(4)
memory usage: 127.4+ MB
Data Preprocessing
In [81]: df.drop(['review_profilename'], axis=1, inplace=True)
df
/var/folders/87/jh63yc3x2tgf72ln53c1n40c0000gn/T/ipykernel_4070/2172711266.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
Out[81]:
brewery_name review_overall review_aroma review_appearance beer_style review_palate review_taste beer_name beer_abv
0 Vecchio Birraio 1.5 2.0 2.5 Hefeweizen 1.5 1.5 Sausa Weizen 5.0
1 Vecchio Birraio 3.0 2.5 3.0 English Strong Ale 3.0 3.0 Red Moon 6.2
2 Vecchio Birraio 3.0 2.5 3.0 Foreign / Export Stout 3.0 3.0 Black Horse Black Beer 6.5
3 Vecchio Birraio 3.0 3.0 3.5 German Pilsener 2.5 3.0 Sausa Pils 5.0
4 Caldera Brewing Company 4.0 4.5 4.0 American Double / Imperial IPA 4.0 4.5 Cauldron DIPA 7.7
... ...
The

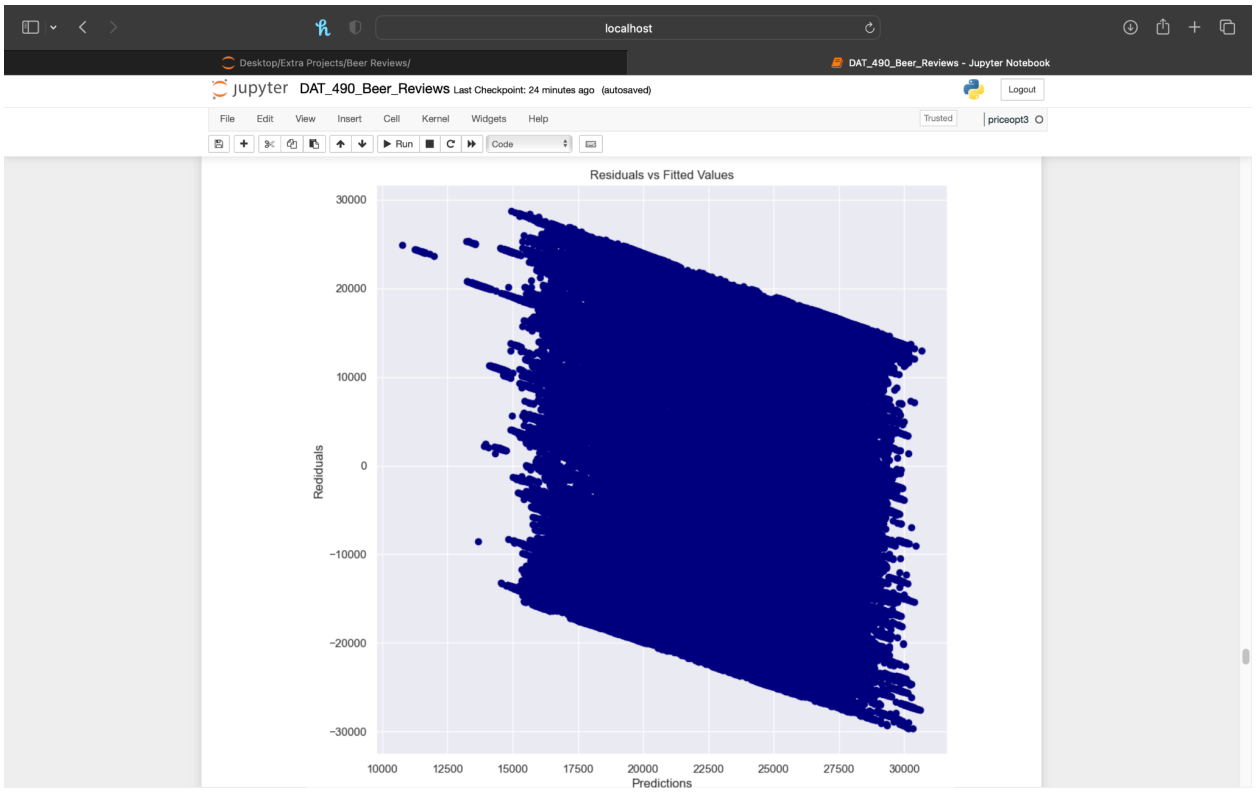

localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 23 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
In [82]:
Changes Categorical Variables to Numbers
le = LabelEncoder()
x = list(df.columns)
for i, j in enumerate(x):
if df[j].dtypes == 'object':
le.fit(df[j].drop_duplicates())
df[j] = le.transform(df[j])
df
/var/folders/87/jh63yc3x2tgf72ln53c1n40c0000gn/T/ipykernel_4070/2477224757.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
Out[82]:
brewery_name review_overall review_aroma review_appearance beer_style review_palate review_taste beer_name beer_abv
0 4886 1.5 2.0 2.5 65 1.5 1.5 34371 5.0
1 4886 3.0 2.5 3.0 51 3.0 3.0 32297 6.2
2 4886 3.0 2.5 3.0 59 3.0 3.0 5313 6.5
3 4886 3.0 3.0 3.5 61 2.5 3.0 34370 5.0
4 1360 4.0 4.5 4.0 9 4.0 4.5 8745 7.7
... ...
1586609 4617 5.0 4.0 3.5 85 4.0 4.0 39244 5.2
1586610 4617 4.0 5.0 2.5 85 2.0 4.0 39244 5.2
1586611 4617 4.5 3.5 3.0 85 3.5 4.0 39244 5.2
1586612 4617 4.0 4.5 4.5 85 4.5 4.5 39244 5.2
1586613 4617 5.0 4.5 4.5 85 4.5 4.5 39244 5.2
1518478 rows × 9 columns
Multiple Linear Regression
In [83]:
X4 = df.drop(['beer_name'], axis=1)

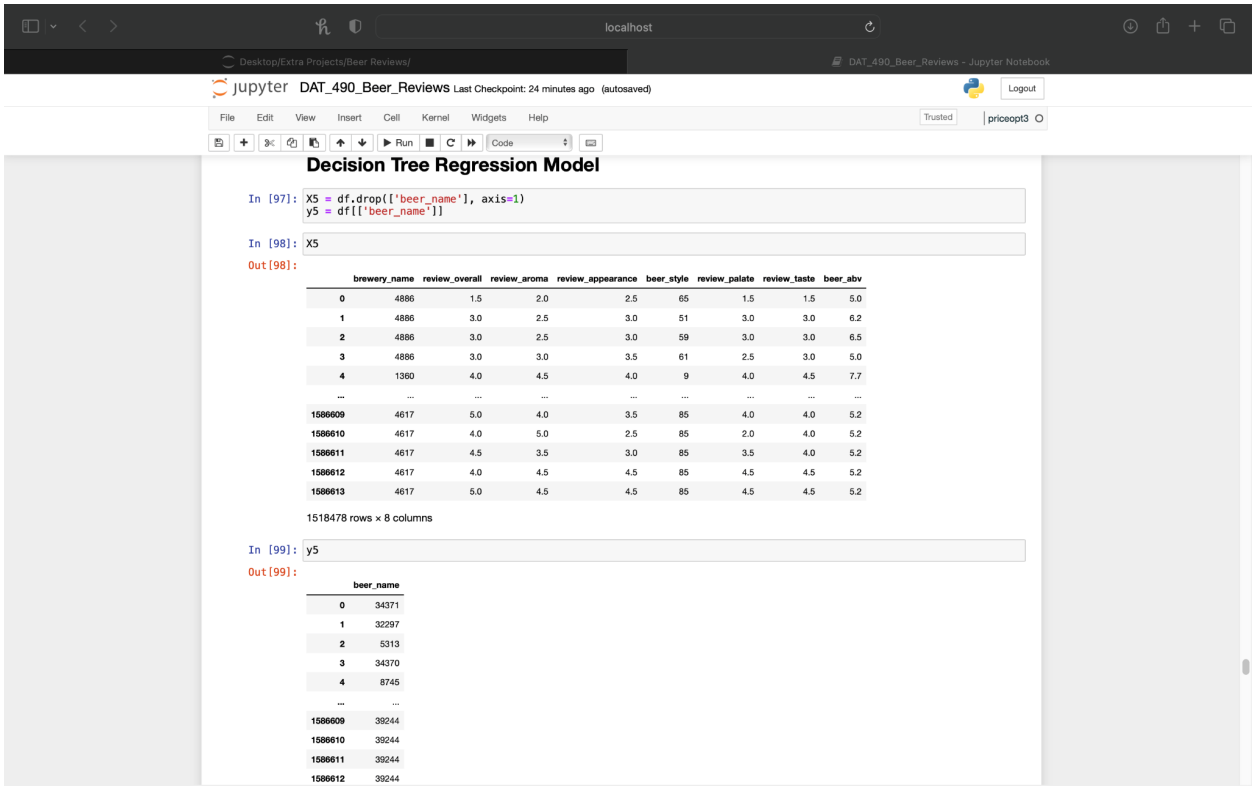

localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 23 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Multiple Linear Regression
In [83]:
X4 = df.drop(['beer_name'], axis=1)
y4 = df[['beer_name']]
In [84]:
X4
Out[84]:
brewery_name review_overall review_aroma review_appearance beer_style review_palate review_taste beer_abv
0 4886 1.5 2.0 2.5 65 1.5 1.5 5.0
1 4886 3.0 2.5 3.0 51 3.0 3.0 6.2
2 4886 3.0 2.5 3.0 59 3.0 3.0 6.5
3 4886 3.0 3.0 3.5 61 2.5 3.0 5.0
4 1360 4.0 4.5 4.0 9 4.0 4.5 7.7
... ...
1586609 4617 5.0 4.0 3.5 85 4.0 4.0 5.2
1586610 4617 4.0 5.0 2.5 85 2.0 4.0 5.2
1586611 4617 4.5 3.5 3.0 85 3.5 4.0 5.2
1586612 4617 4.0 4.5 4.5 85 4.5 4.5 5.2
1586613 4617 5.0 4.5 4.5 85 4.5 4.5 5.2
In [85]:
y4
Out[85]:
beer_name
0 34371
1 32297
2 5313
3 34370
4 8745
... ...
1586609 39244
1586610 39244
1586611 39244
1586612 39244
1586613 39244

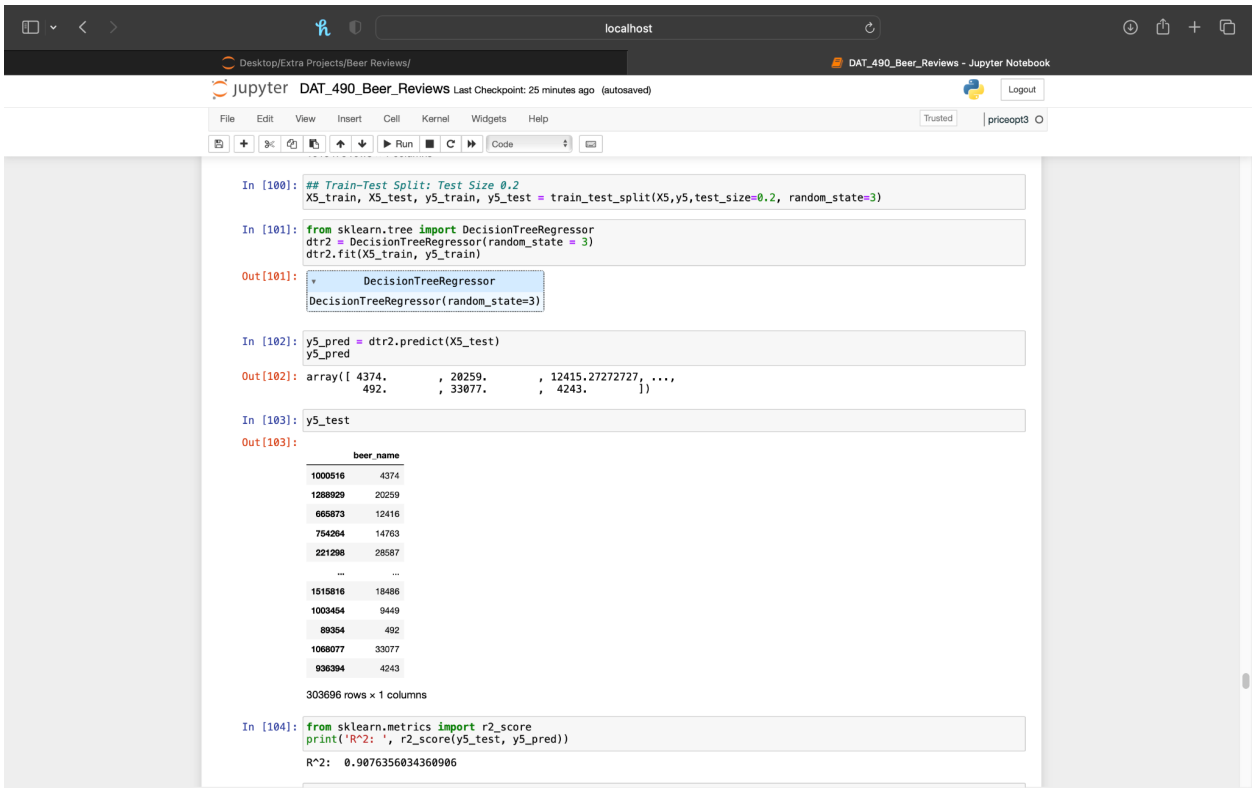

localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 23 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Notebook saved Trusted priceopt3
Run Code
In [86]:
Train-Test Split: Test Size 0.2
from sklearn.model_selection import train_test_split
X4_train, X4_test, y4_train, y4_test = train_test_split(X4,y4,test_size=0.2, random_state=3)
In [87]:
from sklearn.linear_model import LinearRegression
lr2 = LinearRegression()
lr2.fit(X4_train, y4_train)
y4_pred = lr2.predict(X4_test)
y4_pred
Out[87]:
array([[26033.173907],
[28266.60683669],
[20598.09060325],
...,
[25049.32239798],
[20289.66543349],
[19383.06562932]])
In [88]:
y4_test
Out[88]:
beer_name
1000516 4374
1288929 20259
665873 12416
754264 14763
221298 28587
... ...
1515816 18486
1003454 9449
89354 492
1068077 33077
936394 4243
303696 rows × 1 columns
In [89]:
train_acc_score = lr2.score(X4_train, y4_train)
print('Train Accuracy Score: ', train_acc_score)
test_acc_score = lr2.score(X4_test, y4_test)
print('Test Accuracy Score (R^2): ', test_acc_score)
Train Accuracy Score: 0.0756544045379196
Test Accuracy Score (R^2): 0.07325193056082391


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 24 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted priceopt3
Run Code
In [90]:
ADJ R^2
print('Adjusted R^2: ', 1 - (1-lr2.score(X4, y4))*(len(y4)-1)/(len(y4)-X4.shape[1]-1))
Adjusted R^2: 0.07517017133562953
In [91]:
MSE and RMSE
from sklearn.metrics import mean_squared_error
print('MSE: ', mean_squared_error(y4_test, y4_pred))
print('RMSE: ', np.sqrt(mean_squared_error(y4_test, y4_pred)))
MSE: 151485178.92939374
RMSE: 12307.931545527614
In [92]:
RMSLE
from sklearn.metrics import mean_squared_log_error
print('RMSLE: ', np.sqrt(mean_squared_log_error(y4_test, y4_pred)))
RMSLE: 1.0240751696023176
In [93]:
Mean Absolute Error (MAE)
from sklearn.metrics import mean_absolute_error
print(mean_absolute_error(y4_test, y4_pred))
10622.174479791045
In [94]:
print("Coefficients: ", lr2.coef_)
print("Intercept: ", lr2.intercept_)
Coefficients: [[2.16931913 66.09947547 -230.05948569 252.35596277 30.42875503
29.66008471 -30.53627497 -195.16604781]]
Intercept: [16647.12399176]
In [96]:
Residual Analysis
plt.scatter(x=y4_test, y=y4_pred, c='blue')
plt.plot(y4_test, y4_test, color='cyan')
plt.xlabel('Actual Review Score')
plt.ylabel('Predicted Review Score')
plt.title('Actual vs Predicted Review Score')
plt.show()
Residual vs Predicted Values
residuals = y4_test - y4_pred
plt.scatter(x=y4_pred, y=residuals, c='navy')
plt.xlabel('Predictions')
plt.ylabel('Rediduals')
plt.title('Residuals vs Fitted Values')
plt.show()


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 24 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Actual vs Predicted Review Score
Predicted Review Score
Actual Review Score
0
10000
20000
30000
40000


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 24 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Residuals vs Fitted Values
Rediduals
Predictions
30000
20000
10000
0
−10000
−20000
−30000
10000
12500
15000
17500
20000
22500
25000
27500
30000


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 24 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Decision Tree Regression Model
In [97]:
X5 = df.drop(['beer_name'], axis=1)
y5 = df[['beer_name']]
In [98]:
X5
Out[98]:
brewery_name review_overall review_aroma review_appearance beer_style review_palate review_taste beer_abv
0 4886 1.5 2.0 2.5 65 1.5 1.5 5.0
1 4886 3.0 2.5 3.0 51 3.0 3.0 6.2
2 4886 3.0 2.5 3.0 59 3.0 3.0 6.5
3 4886 3.0 3.0 3.5 61 2.5 3.0 5.0
4 1360 4.0 4.5 4.0 9 4.0 4.5 7.7
... ...
1586609 4617 5.0 4.0 3.5 85 4.0 4.0 5.2
1586610 4617 4.0 5.0 2.5 85 2.0 4.0 5.2
1586611 4617 4.5 3.5 3.0 85 3.5 4.0 5.2
1586612 4617 4.0 4.5 4.5 85 4.5 4.5 5.2
1586613 4617 5.0 4.5 4.5 85 4.5 4.5 5.2
1518478 rows × 8 columns
In [99]:
y5
Out[99]:
beer_name
0 34371
1 32297
2 5313
3 34370
4 8745
... ...
1586609 39244
1586610 39244
1586611 39244
1586612 39244


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 25 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
In [100]:
Train-Test Split: Test Size 0.2
X5_train, X5_test, y5_train, y5_test = train_test_split(X5,y5,test_size=0.2, random_state=3)
In [101]:
from sklearn.tree import DecisionTreeRegressor
dtr2 = DecisionTreeRegressor(random_state = 3)
dtr2.fit(X5_train, y5_train)
Out[101]:
DecisionTreeRegressor
DecisionTreeRegressor(random_state=3)
In [102]:
y5_pred = dtr2.predict(X5_test)
y5_pred
Out[102]:
array([4374. , 20259. , 12415.27272727, ...,
492. , 33077. , 4243.])
In [103]:
y5_test
Out[103]:
beer_name
1000516 4374
1288929 20259
665873 12416
754264 14763
221298 28587
... ...
1515816 18486
1003454 9449
89354 492
1068077 33077
936394 4243
303696 rows × 1 columns
In [104]:
from sklearn.metrics import r2_score
print('R^2: ', r2_score(y5_test, y5_pred))
R^2: 0.9076356034360906

localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

jupyter DAT_490_Beer_Reviews Last Checkpoint: 25 minutes ago (autosaved) — Logout

File Edit View Insert Cell Kernel Widgets Help — Trusted — priceopt3

Run Code

In [105]:
```
## ADJ R^2
print('Adjusted R^2: ', 1 - (1-dtr2.score(X5, y5))*(len(y5)-1)/(len(y5)-X5.shape[1]-1))
```

```
Adjusted R^2:  0.9672119543674422
```

In [106]:
```
## MSE and RMSE
print('MSE: ', mean_squared_error(y5_test, y5_pred))
print('RMSE: ', np.sqrt(mean_squared_error(y5_test, y5_pred)))
```

```
MSE:  15097778.567433638
RMSE:  3885.5860005195664
```

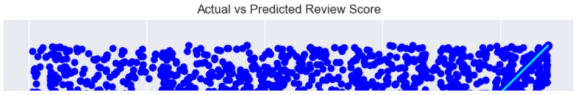
In [107]:
```
## RMSLE
print('RMSLE: ', np.sqrt(mean_squared_log_error(y5_test, y5_pred)))
```

```
RMSLE:  0.3315722574252012
```

In [108]:
```
## Mean Absolute Error (MAE)
from sklearn.metrics import mean_absolute_error
print(mean_absolute_error(y5_test, y5_pred))
```

```
798.1947704934809
```

In [110]:
```
## Residual Analysis
plt.scatter(x=y5_test, y=y5_pred, c='blue')
plt.plot(y5_test, y5_test, color='cyan')

plt.xlabel('Actual Review Score')
plt.ylabel('Predicted Review Score')
plt.title('Actual vs Predicted Review Score')
plt.show()

## Residual vs Predicted Values
y5_pred = y5_pred.reshape(303696,1)
residuals = y5_test - y5_pred
plt.scatter(x=y5_pred, y=residuals, c='navy')

plt.xlabel('Predictions')
plt.ylabel('Rediduals')
plt.title('Residuals vs Fitted Values')
plt.show()
```



localhost

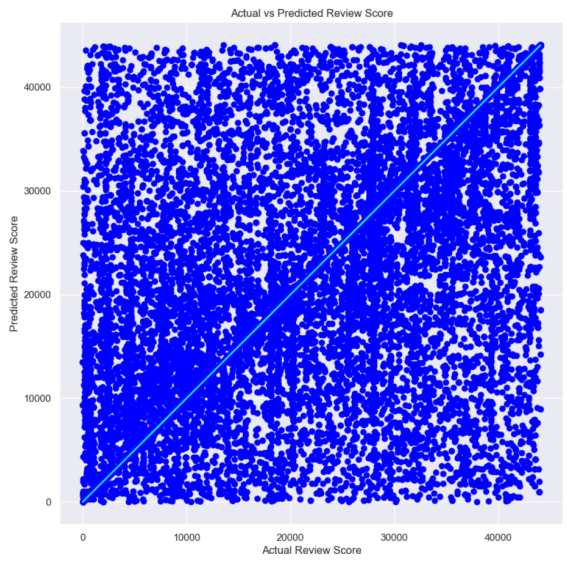
Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

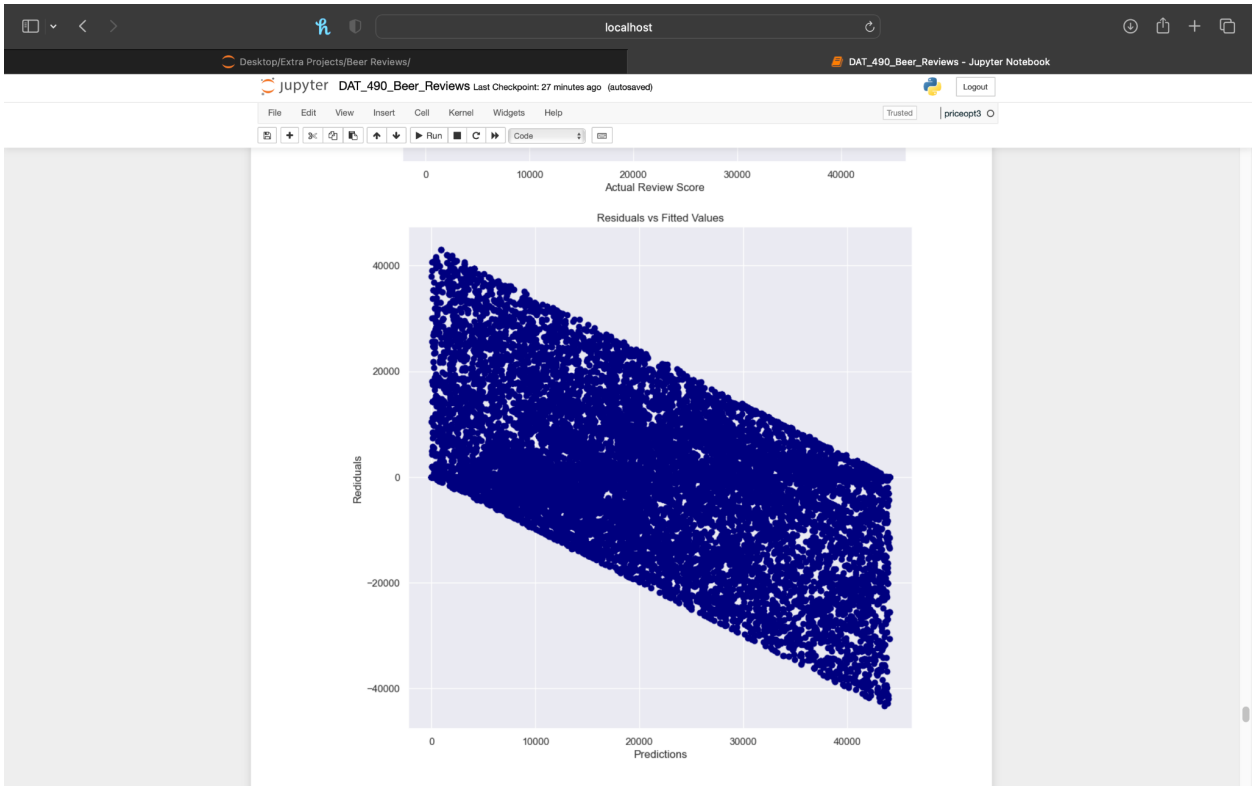
jupyter DAT_490_Beer_Reviews Last Checkpoint: 26 minutes ago (autosaved) — Logout

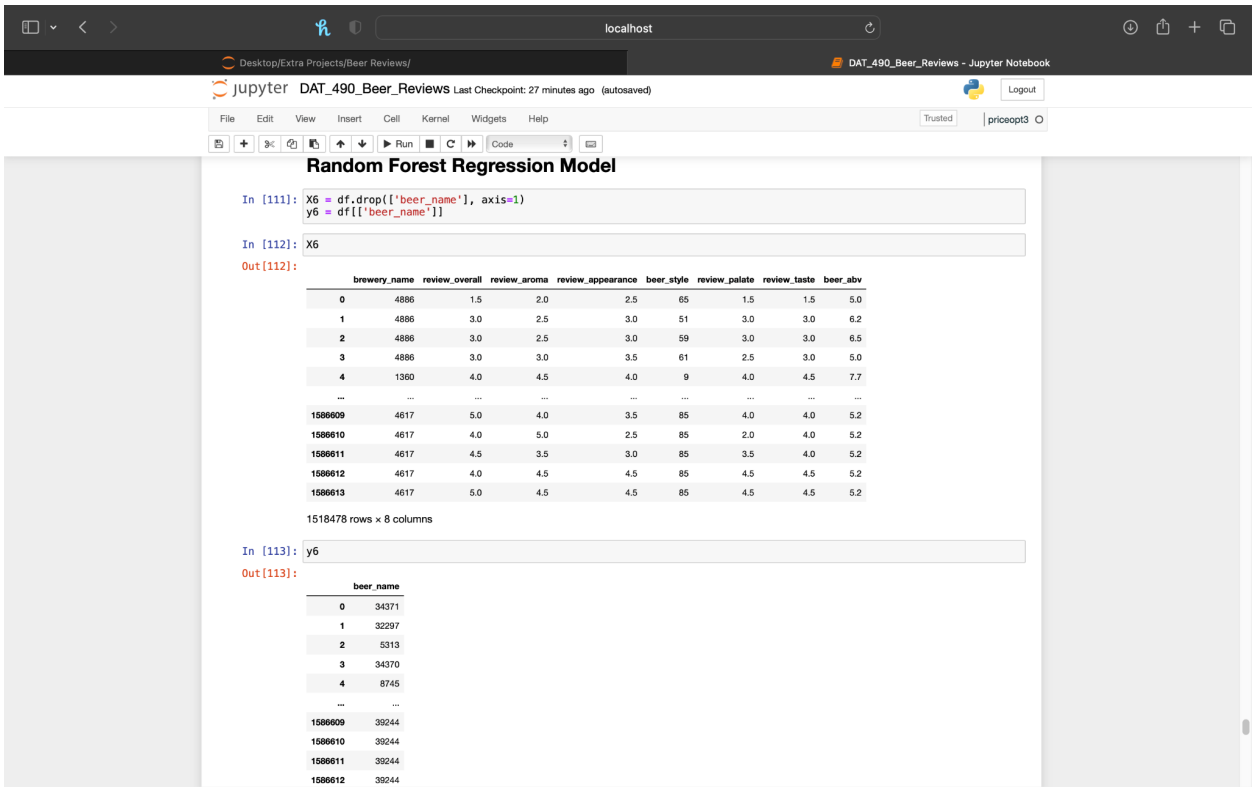
File Edit View Insert Cell Kernel Widgets Help — Trusted — priceopt3

Run Code

```
plt.scatter(x=y5_pred, y=residuals, c='navy')

plt.xlabel('Predictions')
plt.ylabel('Rediduals')
plt.title('Residuals vs Fitted Values')
plt.show()
```







Random Forest Regression Model

In [111]:
```
X6 = df.drop(['beer_name'], axis=1)
y6 = df[['beer_name']]
```

In [112]:
```
X6
```

Out[112]:

	brewery_name	review_overall	review_aroma	review_appearance	beer_style	review_palate	review_taste	beer_abv
0	4886	1.5	2.0	2.5	65	1.5	1.5	5.0
1	4886	3.0	2.5	3.0	51	3.0	3.0	6.2
2	4886	3.0	2.5	3.0	59	3.0	3.0	6.5
3	4886	3.0	3.0	3.5	61	2.5	3.0	5.0
4	1360	4.0	4.5	4.0	9	4.0	4.5	7.7
...	...	...	...	...	...	...	...	...
1586609	4617	5.0	4.0	3.5	85	4.0	4.0	5.2
1586610	4617	4.0	5.0	2.5	85	2.0	4.0	5.2
1586611	4617	4.5	3.5	3.0	85	3.5	4.0	5.2
1586612	4617	4.0	4.5	4.5	85	4.5	4.5	5.2
1586613	4617	5.0	4.5	4.5	85	4.5	4.5	5.2

1518478 rows × 8 columns

In [113]:
```
y6
```

Out[113]:

	beer_name
0	34371
1	32297
2	5313
3	34370
4	8745
...	...
1586609	39244
1586610	39244
1586611	39244
1586612	39244

localhost

Desktop/Extra Projects/Beer Reviews/ — DAT_490_Beer_Reviews - Jupyter Notebook

jupyter DAT_490_Beer_Reviews Last Checkpoint: 27 minutes ago (autosaved) — Logout

File Edit View Insert Cell Kernel Widgets Help — Trusted — priceopt3

Run Code

In [114]:
```
## Train-Test Split: Test Size 0.2
X6_train, X6_test, y6_train, y6_test = train_test_split(X6,y6,test_size=0.2, random_state=3)
```

In [115]:
```
from sklearn.ensemble import RandomForestRegressor
rfr2 = RandomForestRegressor()
rfr2.fit(X6_train, y6_train)
```

```
/var/folders/87/jh63yc3x2tgf72ln53c1n40c0000gn/T/ipykernel_4070/4116104146.py:3: DataConversionWarning:

A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for examp
le using ravel().
```

Out[115]:
```
RandomForestRegressor
RandomForestRegressor()
```

In [116]:
```
y6_pred = rfr2.predict(X6_test)
y6_pred
```

Out[116]:
```
array([ 4374.        , 20620.34      , 12414.63937726, ...,
         492.        , 33077.        ,  4243.        ])
```

In [117]:
```
y6_test
```

Out[117]:

	beer_name
1000516	4374
1288929	20259
665873	12416
754264	14763
221298	28587
...	...
1515816	18486
1003454	9449
89354	492
1068077	33077
936394	4243

303696 rows × 1 columns

In [118]:
```
from sklearn.metrics import r2_score
print('R^2 Score: ', r2_score(y6_test, y6_pred))
```

```
R^2 Score:  0.9345024110864564
```

In [119]:
```
## ADJ R^2
print('Adjusted R^2: ', 1 - (1-rfr2.score(X6, y6))*(len(y6)-1)/(len(y6)-X6.shape[1]-1))
```

```
Adjusted R^2:  0.9682366756041001
```

In [120]:
```
## MSE and RMSE
print('MSE: ', mean_squared_error(y6_test, y6_pred))
print('RMSE: ', np.sqrt(mean_squared_error(y6_test, y6_pred)))
```

```
MSE:  10706160.93326885
RMSE:  3272.0270373682506
```

In [121]:
```
## RMSLE
print('RMSLE: ', np.sqrt(mean_squared_log_error(y6_test, y6_pred)))
```

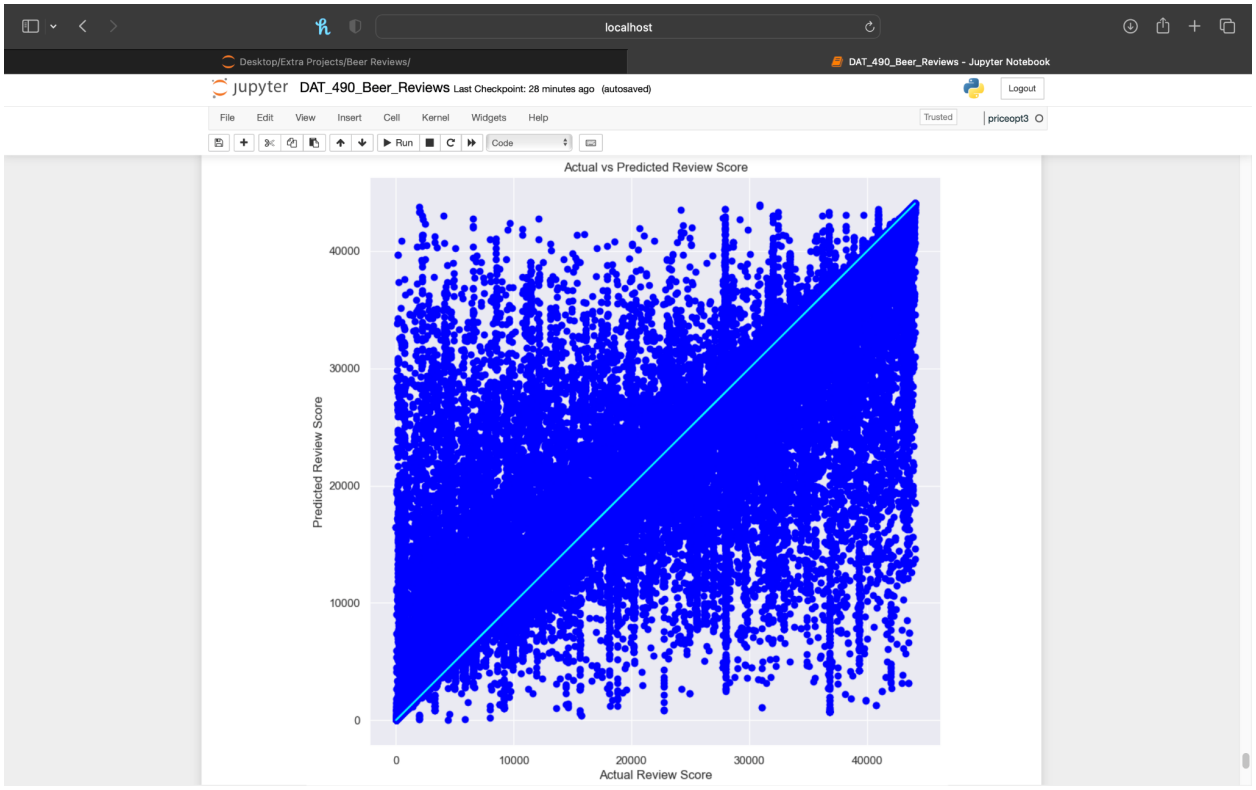
```
RMSLE:  0.3044376869862613
```

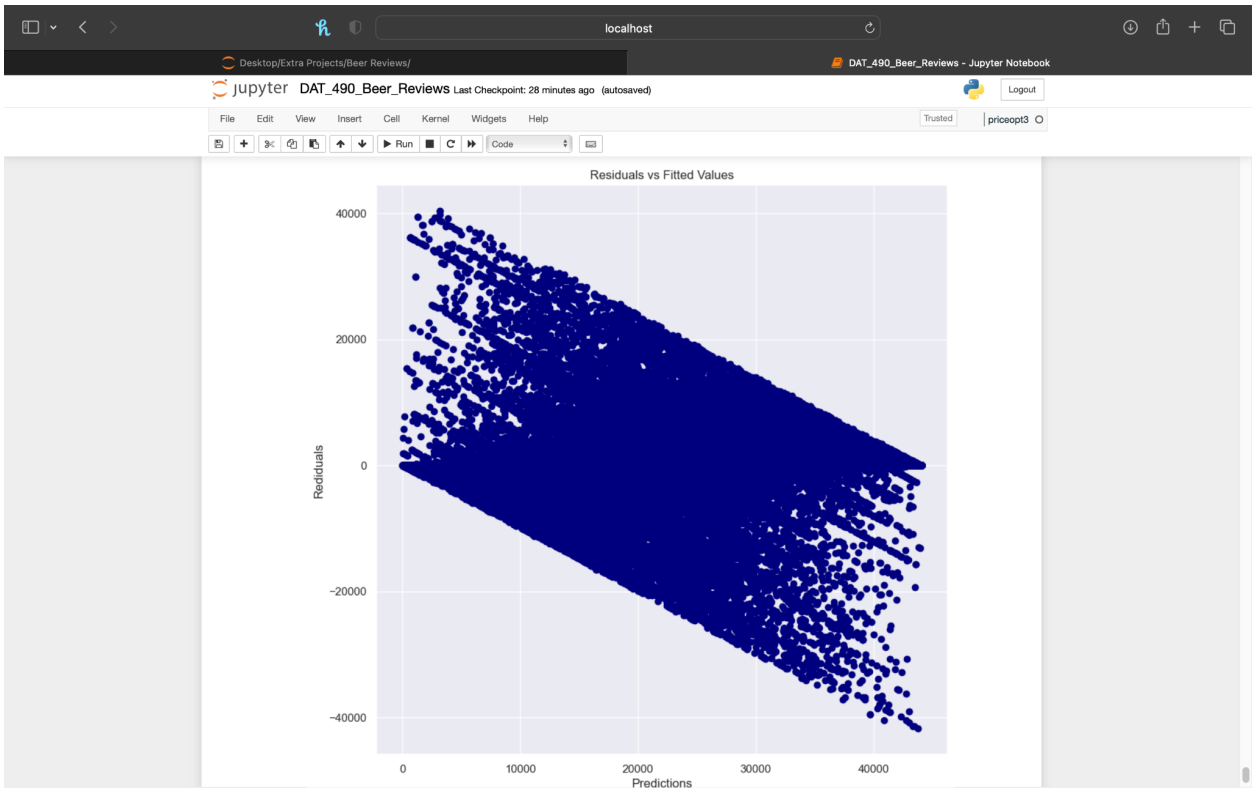
In [122]:
```
## Mean Absolute Error (MAE)
from sklearn.metrics import mean_absolute_error
print(mean_absolute_error(y6_test, y6_pred))
```

```
831.655141618741
```

In [123]:
```
## Residual Analysis
plt.scatter(x=y6_test, y=y6_pred, c='blue')
plt.plot(y6_test, y6_test, color='cyan')

plt.xlabel('Actual Review Score')
plt.ylabel('Predicted Review Score')
plt.title('Actual vs Predicted Review Score')
plt.show()

## Residual vs Predicted Values
y6_pred = y6_pred.reshape(303696,1)
residuals = y6_test - y6_pred
plt.scatter(x=y6_pred, y=residuals, c='navy')

plt.xlabel('Predictions')
plt.ylabel('Rediduals')
plt.title('Residuals vs Fitted Values')
plt.show()
```

Actual vs Predicted Review Score


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 28 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Actual vs Predicted Review Score
Predicted Review Score
Actual Review Score
0
10000
20000
30000
40000


localhost
Desktop/Extra Projects/Beer Reviews/
DAT_490_Beer_Reviews - Jupyter Notebook
jupyter DAT_490_Beer_Reviews Last Checkpoint: 28 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
priceopt3
Run
Code
Residuals vs Fitted Values
Rediduals
Predictions
40000
20000
0
−20000
−40000
0
10000
20000
30000
40000