Syracuse University

IST 707

Professor: Dr. Ami Gates

**Project – Craft Beer and Brewery Analysis**

Samantha McNeilly

Thulasiram Ruppa Krishnan

June 15, 2019

Table of Contents

[Introduction 3](#_Toc11491138)

[Analysis and Model(s) 5](#_Toc11491139)

[About the Data 5](#_Toc11491140)

[State Analysis and Visualizations 6](#_Toc11491141)

[City Analysis and Visualizations 8](#_Toc11491142)

[Brewery Analysis and Visualizations 10](#_Toc11491143)

[Style Analysis and Visualizations 10](#_Toc11491144)

[Association Rules Pre-Processing 16](#_Toc11491145)

[Clustering Pre-Processing 16](#_Toc11491146)

[Decision Trees Pre-Processing 16](#_Toc11491147)

[Naïve Bayes, Support Vector Machines, K-Nearest Neighbor and Random Forest Pre-Processing 17](#_Toc11491148)

[Model(s) 18](#_Toc11491149)

[Association Rule Mining 18](#_Toc11491150)

[Clustering – K-means 21](#_Toc11491151)

[Clustering – Hierarchical 25](#_Toc11491152)

[Decision Trees 28](#_Toc11491153)

[Naïve Bayes 32](#_Toc11491154)

[Support Vector Machine 33](#_Toc11491155)

[K-Nearest Neighbor Pre-Processing 34](#_Toc11491156)

[Random Forest Pre-Processing 36](#_Toc11491157)

[Result(s) 38](#_Toc11491158)

[Association Rule Mining 38](#_Toc11491159)

[Clustering 38](#_Toc11491160)

[Decision Trees 38](#_Toc11491161)

[Naïve Bayes 39](#_Toc11491162)

[Support Vector Machines 39](#_Toc11491163)

[K-Nearest Neighbor 40](#_Toc11491164)

[Random Forest 41](#_Toc11491165)

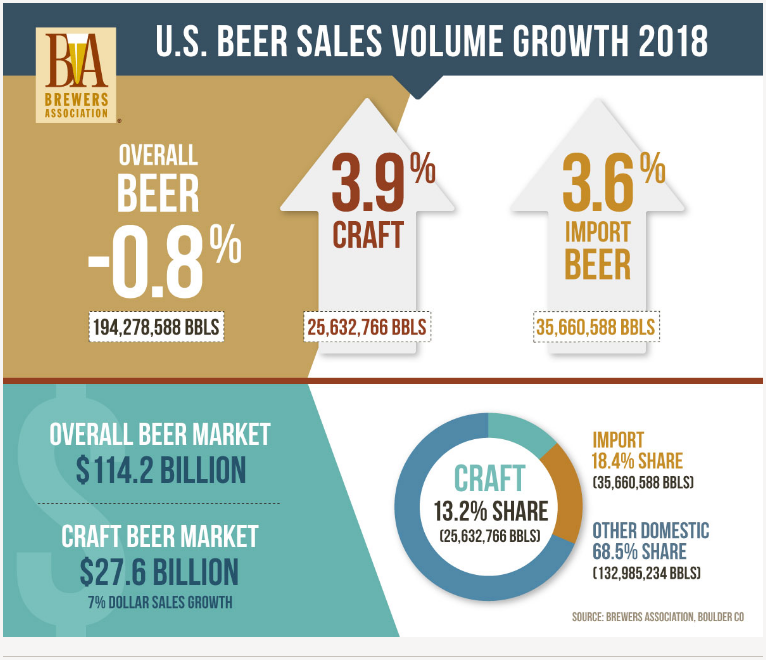
[Supervised Learned Model Comparison 42](#_Toc11491166)

[Conclusion 44](#_Toc11491167)

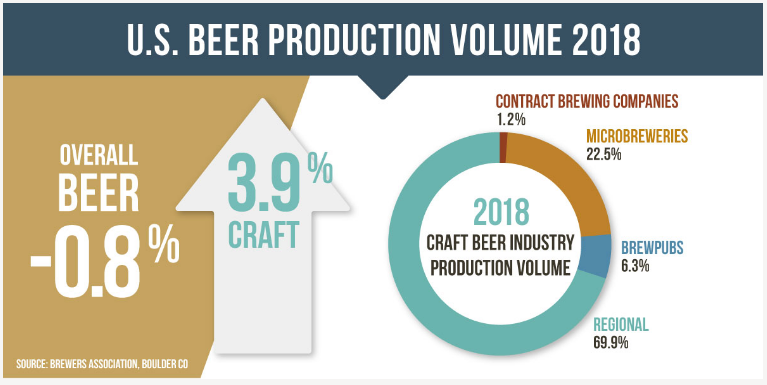
# Introduction

Overall U.S. beer volume sales were down 1% in 2018, whereas craft brewer sales continued to grow at a rate of 4% by volume, reaching 13.2% of the U.S. beer market by volume. Retail dollar sales of craft beer increased 7%, up to $27.6 billion, and now account for more than 24% of the $114.2 billion U.S. beer market. [[1]](#footnote-1)

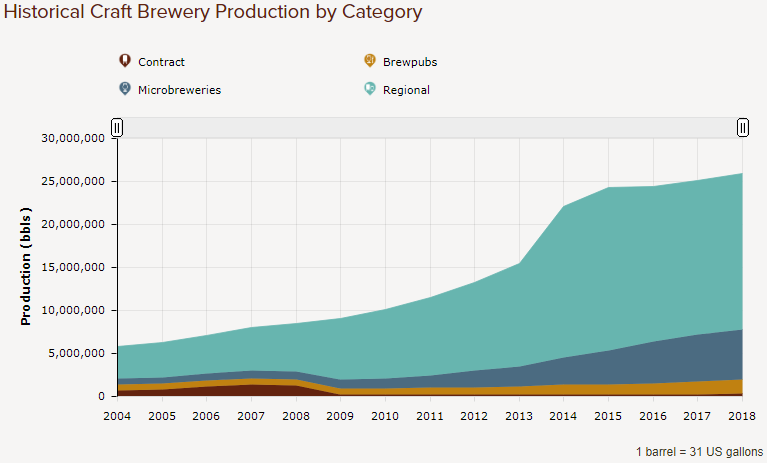
**Exhibit 1: U.S. Beer Sales Volume Growth 2018**



**Exhibit 2: U.S. Beer Production Volume 2018**



**Exhibit 3: Historical Craft Brewery Production by Category**



To understand these figures we first need to review how the Brewers Association defines the term craft brewer (note, used interchangeably with the term micro-brewery). The term craft brewer can be broken down into two key concepts: small (annual production of 6 million or less barrels of beer) and independent (less than 25% owned by a beverage alcohol industry member who is not considered craft). Some other concepts which help identify a brewer as craft include:

* Innovative: Taking a twist on a historical style to develop something new and innovative
* Daring: Use of non-traditional ingredients in the beer making process (e.g., yuzu, guava, passion fruit, coriander, salt, etc.)
* Philanthropic: Craft brewers tend to be very involved in the community
* Consumer-centric: Use of distinctive, individualistic approaches to connect with the consumer

With the growth in craft brewers there has also been an increase in the number of beer styles offered. To differentiate between beer styles the following elements of a beer are considered:

* Standard Reference Method (SRM) refers to a beers color
* Alcohol by Volume (ABV) refers to amount of alcohol in the beer (U.S. average is 5.9%)
* International Bitterness Units (IBU) refers to the measure of hops’ contribution to a beer’s bitterness (ranges from 0 to >100)
* Servicing Size refers to the size of the glasses that beer is served in (U.S. is generally a pint, or 16 ounces)
* Gravity refers to the amount of residual sugar in a beer (also known as mouthfeel)

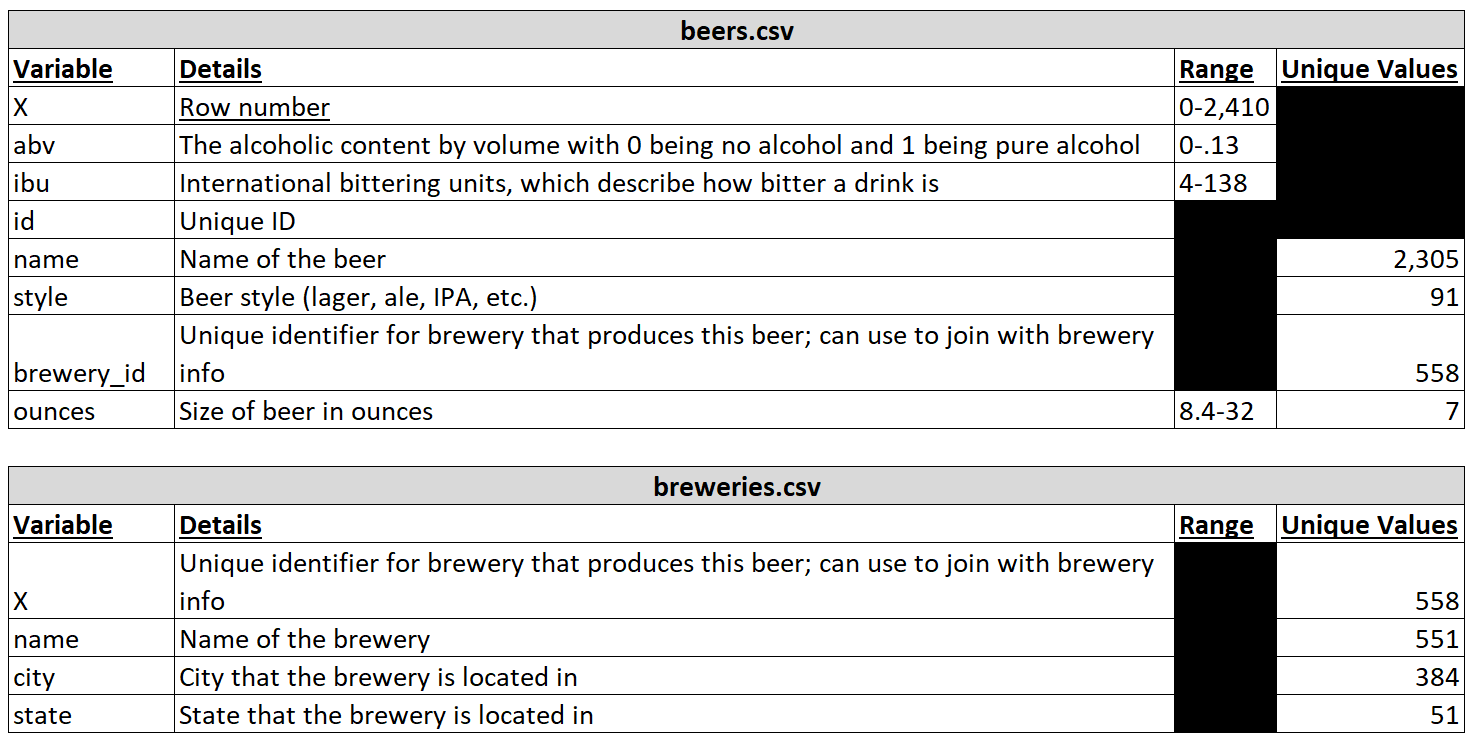
The remainder of this report examines how well ABV, IBU and serving size predict modern craft beer styles.

# Analysis and Model(s)

## About the Data

The original dataset was split between two .csv files; one containing information on 2,410 U.S. craft beers and the other containing information on 510 U.S. craft breweries. The beers and breweries datasets are linked together by the “brewery\_id” and according to Kaggle.com the data was collected in January 2017 from CraftCans.com. **Exhibit 4** provides details related to each of variables in the datasets.

**Exhibit 4: Beers and Breweries Dataset Pre-Processing**



The following pre-processing steps were taken to prepare the datasets for analysis:

* Removed of variables “X” and “id” from the beers dataset
* Joined of the beers and breweries dataset based on beers$brewery\_id and breweries$X
* With the two tables joined there are two name variables (name.x and name.y) representing beer\_name (name.x) and brewery\_name (name.y), these column names were updated to beer\_name and brewery\_name, respectively
* The joined datasets were added to two new dataframes
  + Full.beers.breweries: Used for preliminary analysis (includes NA’s)
  + Beers.breweries: Used for modeling (NA’s removed)

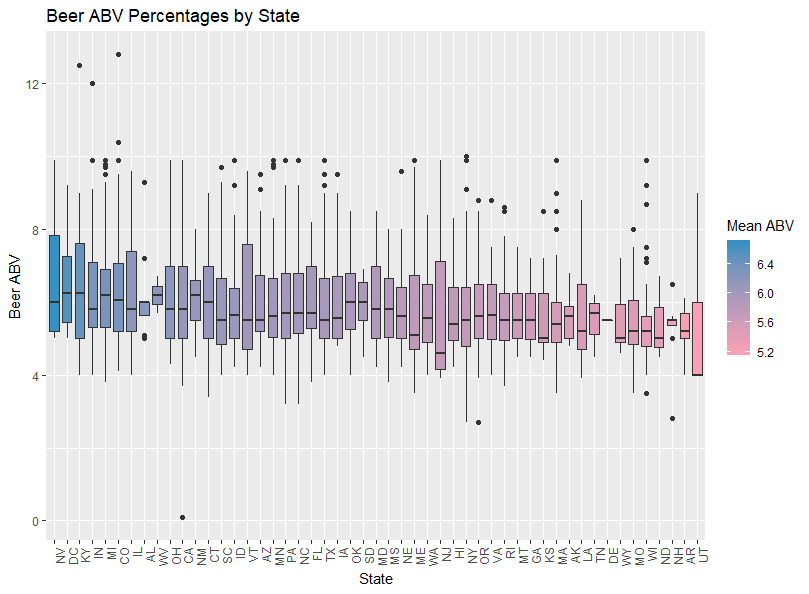
Before preliminary analysis of the full.beers.breweries dataset was performed the following additional steps were taken:

* An NA factor in style was updated to the appropriate style (i.e., beer 867 was updated to "MÃ¤rzen / Oktoberfest" and beer 854 was updated to "Scottish Ale")
* ABV was converted from a decimal to a percentage with one decimal place (i.e., abv\*100, digits = 1)
* The factor in city and state were removed
* City and state were combined into one variable, named “location” (e.g., Bend, OR)

### State Analysis and Visualizations

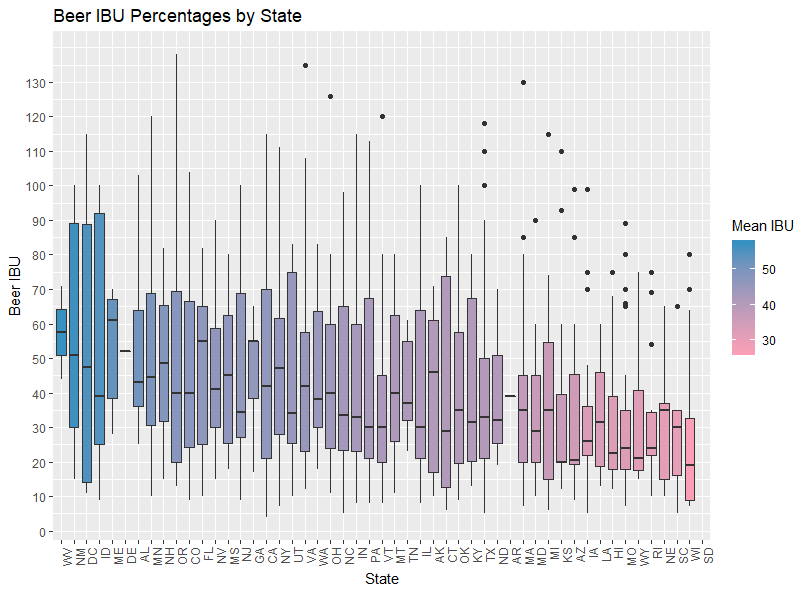
Then the following visualizations leveraged the full.beers.breweries dataframe to analyze the variable “state”:

**Exhibit 5: Beer ABV Percentage by State**

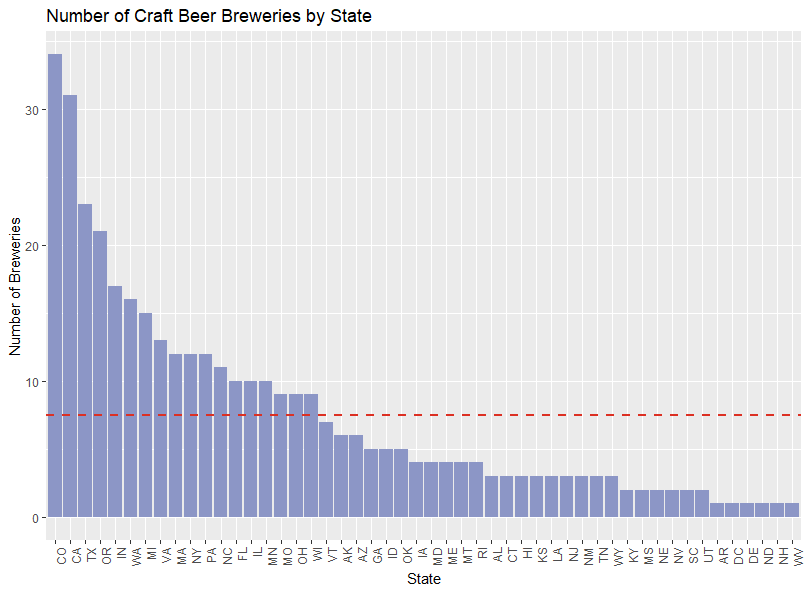


**Exhibit 5** identifies Utah (“UT”) as have the lowest mean ABV of all 51 states (50 plus DC). This is consistent with expectations. A large population of Utah residents belong to The Church of Jesus Christ of Latter-Day Saints and the church advises against consumption of alcohol for its members. Additionally, Utah legislation limits beer to 4.0 ABV, which can be sold by grocery stores, convenience stores and in other establishments with “beer only” type licenses. In order to sell beer with >4.0 ABV in Utah you must be a state liquor store, package agency or an establishment with a license to sell liquor. Laws around the sale of alcohol even restrict the preparation of alcoholic beverages in restaurants to be separate from consumers, this is commonly referred to as the “Zion Curtain”. Interestingly, new legislation has passed which increases the ABV limit in beers to 5.0 on November 1, 2019.

**Exhibit 6: Beer IBU Percentages by State**



**Exhibit 7: Number of Craft Beer Breweries by State**

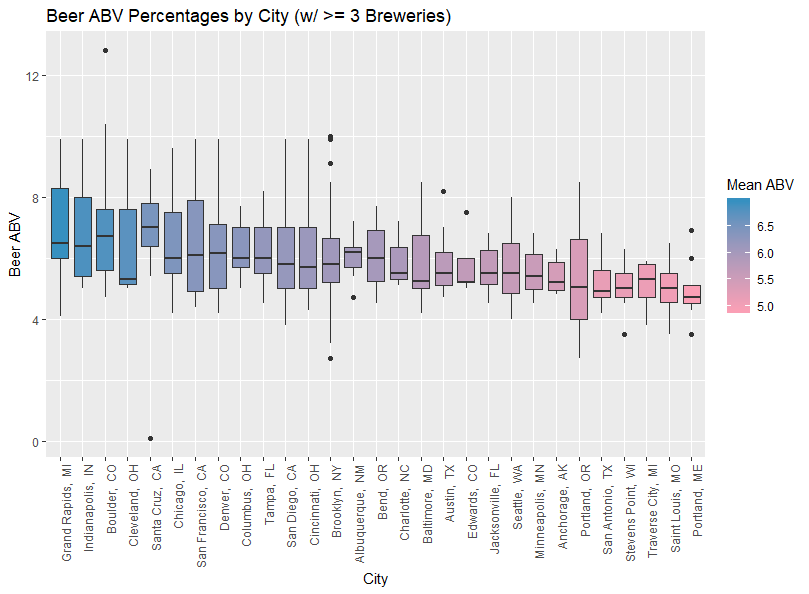


Note, **Exhibit 7** shows in descending order the number of craft beer breweries by state. The dotted red line represents the average number of craft beer breweries across all states, which is equal to 7.46.

### City Analysis and Visualizations

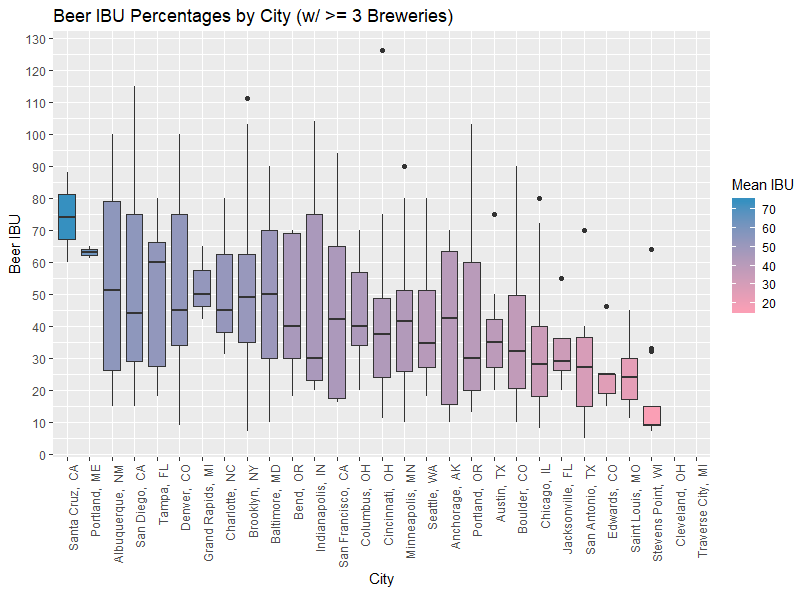
Similar to Exhibits 5, 6 and 7 the following exhibits breakdown ABV, IBU and number of breweries by the variable “city”.

**Exhibit 8: Beer ABV Percentage by City**

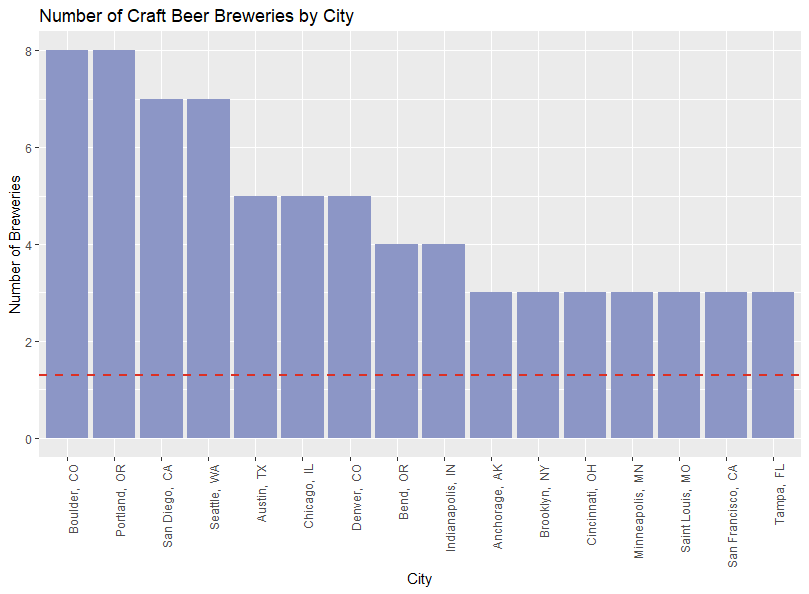


Note, the dataset contains 384 unique locations (i.e., city, state combinations). In order to represent a legible visualization **Exhibit 8** (above) and **Exhibits 9 and 10** (below) only represents those cities with three or more breweries.

**Exhibit 9: Beer IBU Percentages by City**



**Exhibit 10: Number of Craft Beer Breweries by City**

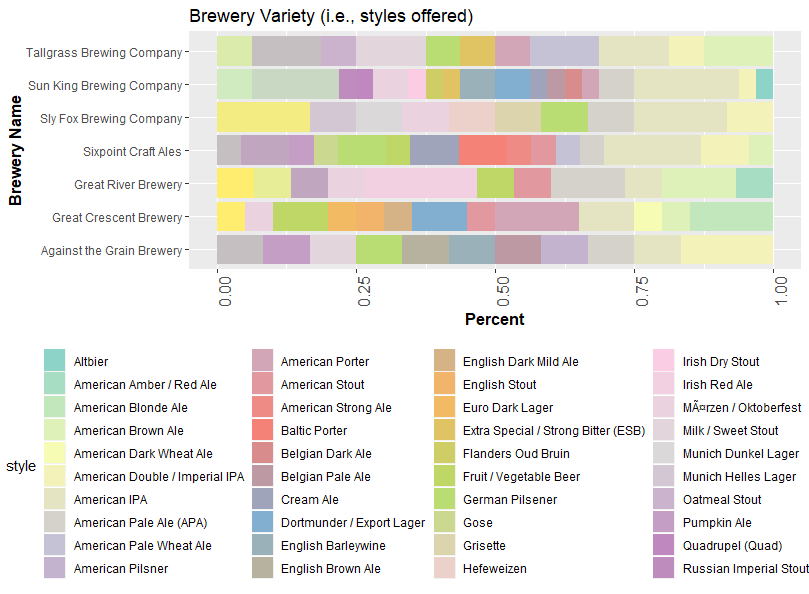


Note, **Exhibit 10** shows in descending order the number of craft beer breweries by location. The dotted red line represents the average number of craft beer breweries across all locations, which is equal to 1.3.

### Brewery Analysis and Visualizations

**Exhibit 11** depicts a stacked bar plot with breweries that offer more than 10 unique styles of beer (i.e., the breweries with the largest variety of beer). Seven breweries offer the largest selection.

**Exhibit 11: Brewery Variety**



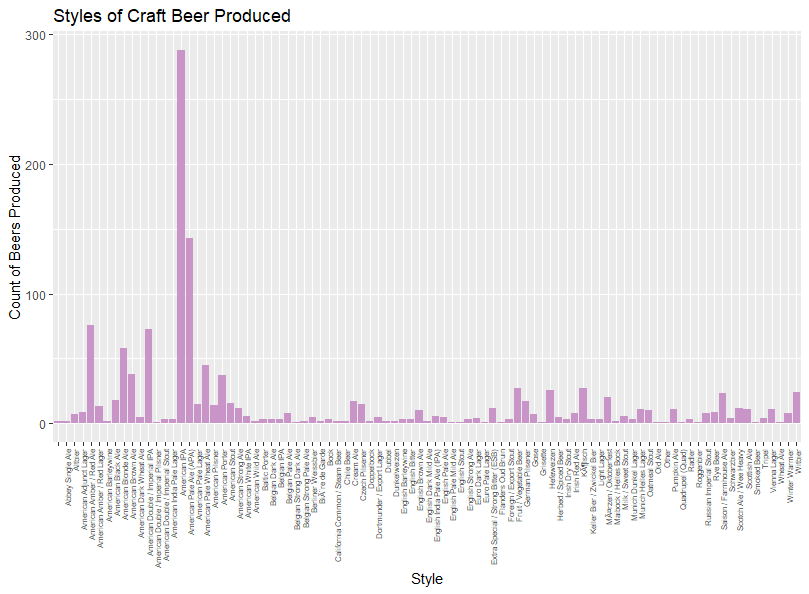
### Style Analysis and Visualizations

The variable style was used for all of the models defined in this report. Therefore, additional preprocessing steps were necessary to the beers.breweries dataframe in order to prepare the data for analysis. Those additional steps are as follows:

* Removed any NA’s across the dataframe (note, ABV contained 62 NA’s and IBU 1,005). The updated dataframe resulted in 1,405 observations

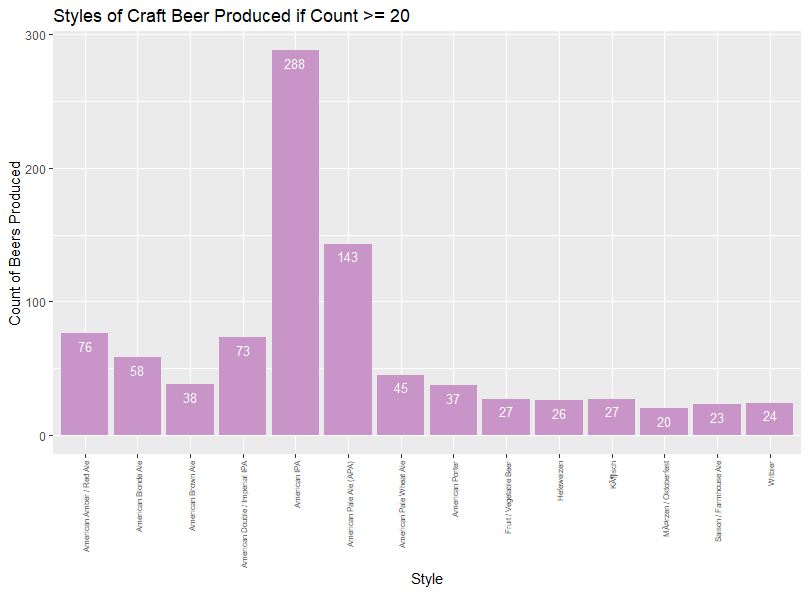
After preprocessing there are now 90 unique styles of craft beer. Exhibit 12 shows the count of observations for all 90 styles, whereas, Exhibit 13 shows only those where the count of observations is greater than 20 for each individual style. In order to improve the accuracy of the models contained in this report it is important to note all models are based on those styles that have 20 or more observations in the dataset.

**Exhibit 12: Styles of Craft Beer Produced**



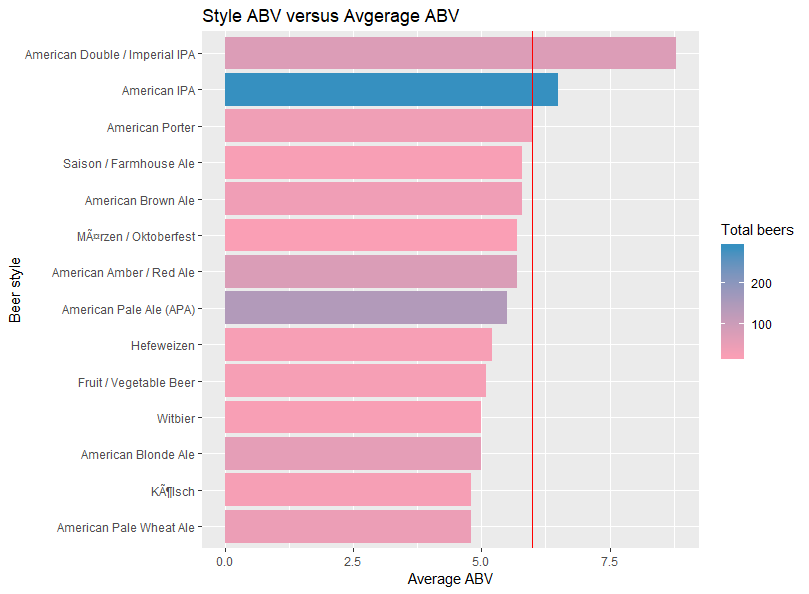
**Exhibit 13** (below) identifies American IPA as the most popular style of craft beer. Craftbeer.com reports that American IPA is the most entered category at the Great American Beer Festival for over a decade and is the top-selling style in supermarkets and liquor stores in the U.S.

**Exhibit 13: Styles of Craft Beer (observations >=20)**



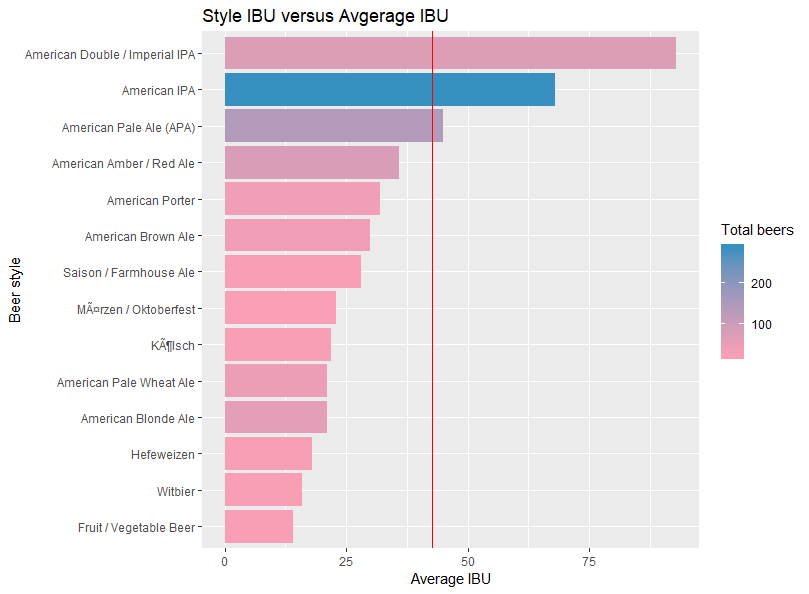
**Exhibit 14** depictsa bar plot with style abv versus average abv, filled with the count of observations related to that particular style. The red line represents the average abv across all beer styles, which is equal to 5.99.

**Exhibit 14: Style ABV versus Average ABV**



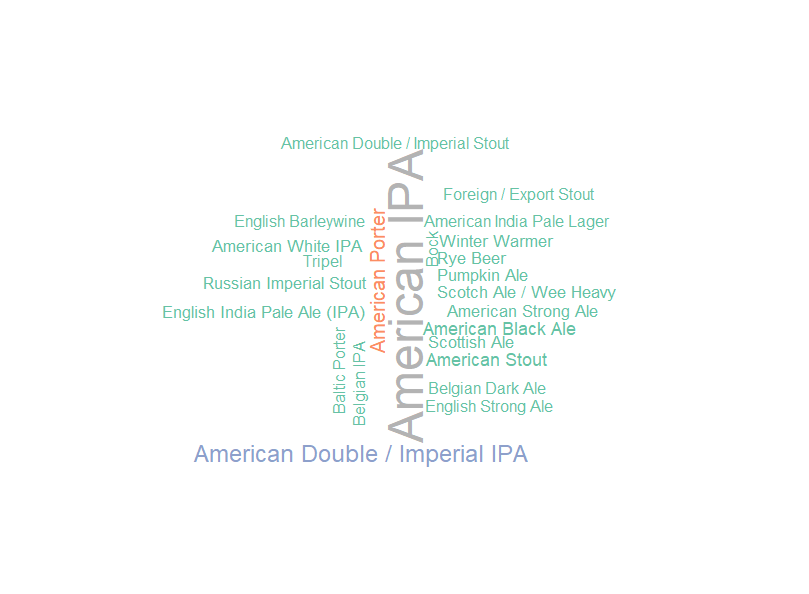
**Exhibit 15** depictsa bar plot with style ibu versus average ibu, filled with the count of observations related to that particular style. The red line represents the average ibu across all beer styles, which is equal to 43.

**Exhibit 15: Style IBU versus Average IBU**



**Exhibits 16, 17 and 18** are word clouds of the beer style broken into three distinct categories: High ABV (>= 6.0 ABV), Medium ABV (>= 4.0 & <= 5.9 ABV) and Low ABV (<= 3.9 ABV). Note, size and color of the style are based on the number of observations in the dataset.

**Exhibit 16: Beer Styles with High ABV**



**Exhibit 17: Beer Styles with Medium ABV**

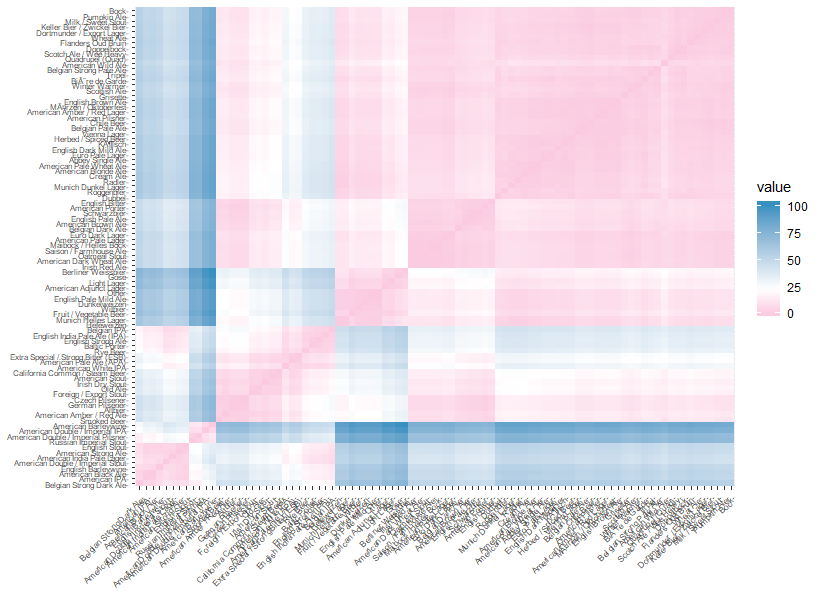


**Exhibit 18: Beer Styles with Low ABV**



The heatmap in **Exhibit 19** on the next page identifies the differences and similarities between the various styles of beer. Note, pink represents the most similar styles and blue those that are the most dissimilar.

**Exhibit 19: Heatmap of Beer Styles**



Now that the preliminary analysis and visualizations have been performed there is additional pre-processing that needs to completed in order to prepare the data for use in the models. Those steps include:

### Association Rules Pre-Processing

* Ounces was discretized into a custom bin:
  + If ounces were 12 or less the factor "12 ounces or less" was created
  + If ounces were 16 or more the factor "16 ounces or more" was created
  + Note, there were no measurements between 12 and 16 ounces
* IBU was discretized into a custom bin:
  + For IBU between 0-20 the factor "0-20" was created
  + For IBU between 21-40 the factor "21-40" was created
  + For IBU between 41-60 the factor "41-60" was created
  + For IBU between 61-80 the factor "61-80" was created
  + For IBU between 81-100 the factor "81-100" was created
  + For IBU of 101 or greater the factor "101+" was created
* ABV was discretized into a custom bin:
  + For ABV 0-3.9 the factor “Light” was created
  + For ABV 4.0 – 5.9 the factor “Normal” was created
  + For ABV 6.0 or greater the factor “High” was created

### Clustering Pre-Processing

* Created a table grouped by style which returns: count of beers, mean abv, mean ibu and mean ounces for a particular style
* Converted the created table to a dataframe
* Converted count of beers within that style to numeric
* Converted style to row names
* Removed the style variable
* Scale the entire dataframe (i.e., count, mean abv, mean ibu and mean ounces)

### Decision Trees Pre-Processing

* Subset only those beer styles with greater than 20 observations (based on count returned in the table created for clustering). The represented beer styles are as follows:
  + American IPA
  + American Pale Ale (APA)
  + American Amber / Red Ale
  + American Double / Imperial IPA
  + American Blonde Ale
  + American Pale Wheat Ale
  + American Brown Ale
  + American Porter
  + Fruit / Vegetable Beer
  + KÃ¶lsch
  + Hefeweizen
  + Witbier
  + Saison / Farmhouse Ale
  + MÃ¤rzen / Oktoberfest
* Dropped unnecessary factors in style
* Used the initial\_split() function to create a train (70%) and test (30%) dataset

### Naïve Bayes, Support Vector Machines, K-Nearest Neighbor and Random Forest Pre-Processing

* Subset only those beer styles with greater than 20 observations (based on count returned in the table created for clustering). The represented beer styles are as follows:
  + American IPA
  + American Pale Ale (APA)
  + American Amber / Red Ale
  + American Double / Imperial IPA
  + American Blonde Ale
  + American Pale Wheat Ale
  + American Brown Ale
  + American Porter
  + Fruit / Vegetable Beer
  + KÃ¶lsch
  + Hefeweizen
  + Witbier
  + Saison / Farmhouse Ale
  + MÃ¤rzen / Oktoberfest
* Dropped unnecessary factors in style
* Dataset is split into 3 block and 2 blocks (67%) are used for training and one block (33%) for testing

## Model(s)

### Association Rule Mining

Association Rule Mining is used to find an association between different objects in a set, find frequent patterns in a transaction database, relational databases or any other information repository. In simpler terms, association rule mining provides you with an output of rules in the form of “if this”, “then that”.

To perform association rule mining in R the apriori function from the arules library is used. The apriori algorithm employs level-wise search for frequent itemsets. The following arguments can be used with the apriori algorithm:

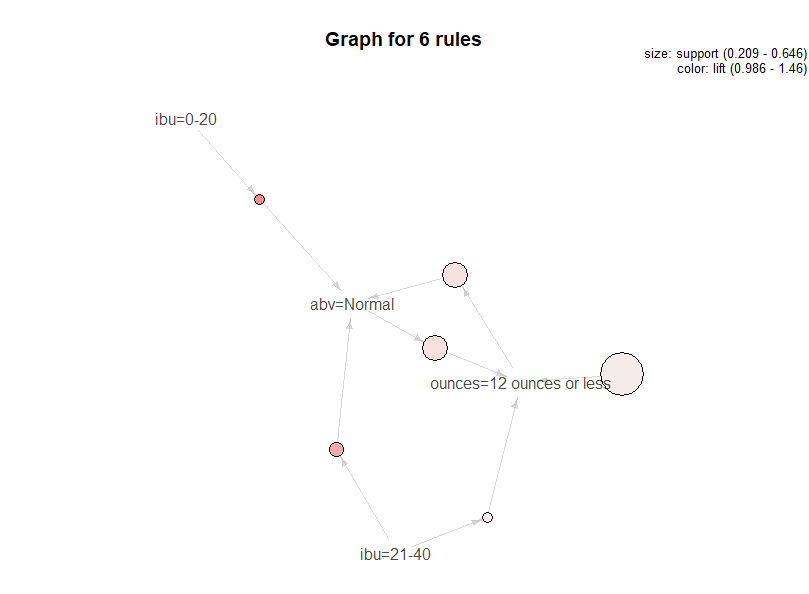
**Table 6: Apriori Arguments**

|  |  |
| --- | --- |
| **Argument** | **Explanation** |
| Data | object of class '>transactions or any data structure which can be coerced into '>transactions (e.g., a binary matrix or data.frame). |
| Parameter | object of class '>APparameter or named list. The default behavior is to mine rules with minimum support of 0.1, minimum confidence of 0.8, maximum of 10 items (maxlen), and a maximal time for subset checking of 5 seconds (maxtime).  Note: Common setting for support is 20-40% of the transactions. Strong confidence rules generally have >= .9, however, .6 to .8 range might be okay. |
| Appearance | object of class '>APappearance or named list. With this argument item appearance can be restricted (implements rule templates). By default all items can appear unrestricted. |
| Control | object of class '>APcontrol or named list. Controls the algorithmic performance of the mining algorithm (item sorting, report progress (verbose), etc.) |

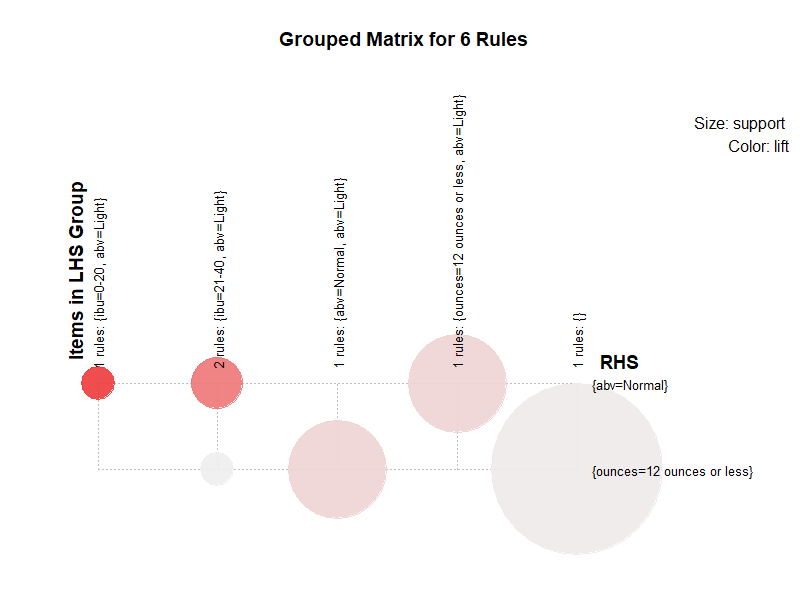
The left-hand side (lhs) of a rule (anything before =>) is known as X and the right-hand side (rhs) of a rule (anything after =>) is known as Y. The fraction of transactions (or observations in this case) that contain both X and Y is referred to as the “support” for a specific rule. Further, how frequently Y appears in transactions (or observations) that contain X is known as the confidence. Lift is the measure of dependent or correlated events. Lift should be above 1.0 and the higher the lift the better.

**Model 1** (both the graph-based and grouped matrix visualization) below depicts an apriori algorithm using the processed beer.breweries dataset. The following parameters were used: support = 0.2 and confidence = 0.6. **Table 1** lists the 6 rules that were created in **Model 1** in descending order based on confidence.

**Model 1: Graph-based Visualization: Apriori algorithm, support = 0.2 and confidence = 0.6**

****

**Model 1: Grouped Matrix Visualization: Apriori algorithm, support = 0.2 and confidence = 0.6**

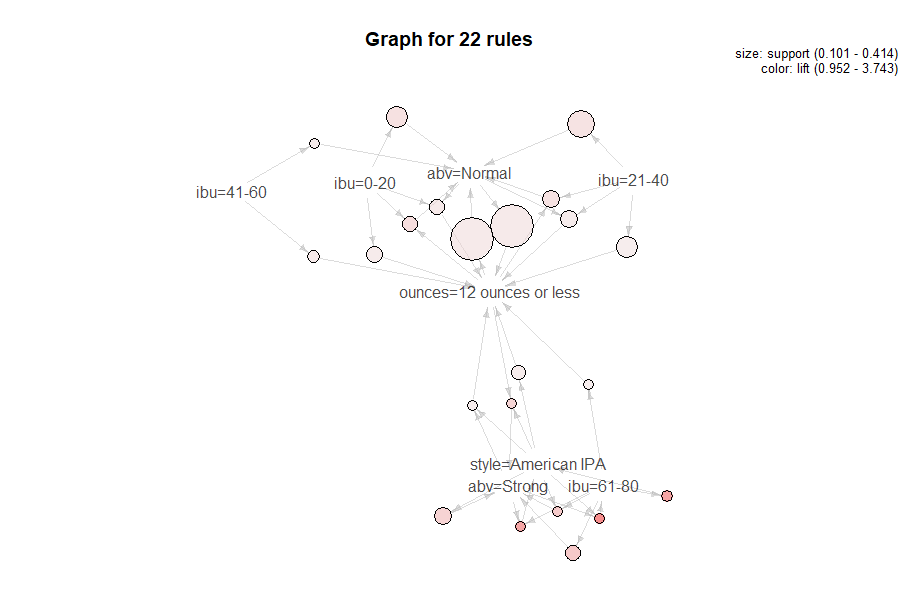
****

**Table 1: Sorted rules from Model 1**

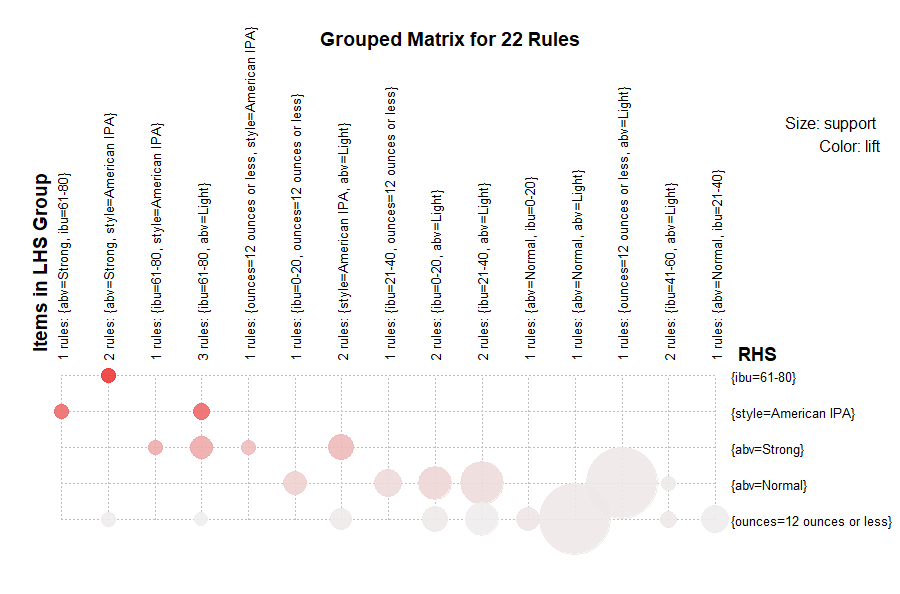
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **LHS** |  | **RHS** | **Support** | **Confidence** | **Lift** | **Count** |
| {ibu=0-20}} | => | {abv=Normal} | 0.2092527 | 0.8698225 | 1.4600963 | 294 |
| {ibu=21-40} | => | {abv=Normal} | 0.2647687 | 0.8051948 | 1.3516114 | 372 |
| {abv=Normal} | => | {ounces=12 ounces or less} | 0.4142349 | 0.6953405 | 1.0771261 | 582 |
| {} | => | {ounces=12 ounces or less} | 0.6455516 | 0.6455516 | 1.0000000 | 907 |
| {ounces=12 ounces or less} | => | {abv=Normal} | 0.4142349 | 0.6416759 | 1.0771261 | 582 |
| {ibu=21-40} | => | {ounces=12 ounces or less} | 0.2092527 | 0.6363636 | 0.9857673 | 294 |

**Model 2** (both the graph-based and grouped matrix visualization) below depicts an apriori algorithm using the processed beer.breweries dataset. The following parameters were used: support = 0.1 and confidence = 0.6. **Table 2** lists the 6 rules that were created in **Model 2** in descending order based on confidence.

**Model 2: Graph-based Visualization: Apriori algorithm, support = 0.1 and confidence = 0.6**

****

**Model 2: Grouped Matrix Visualization: Apriori algorithm, support = 0.1 and confidence = 0.6**

****

**Table 2: Sorted rules from Model 2**

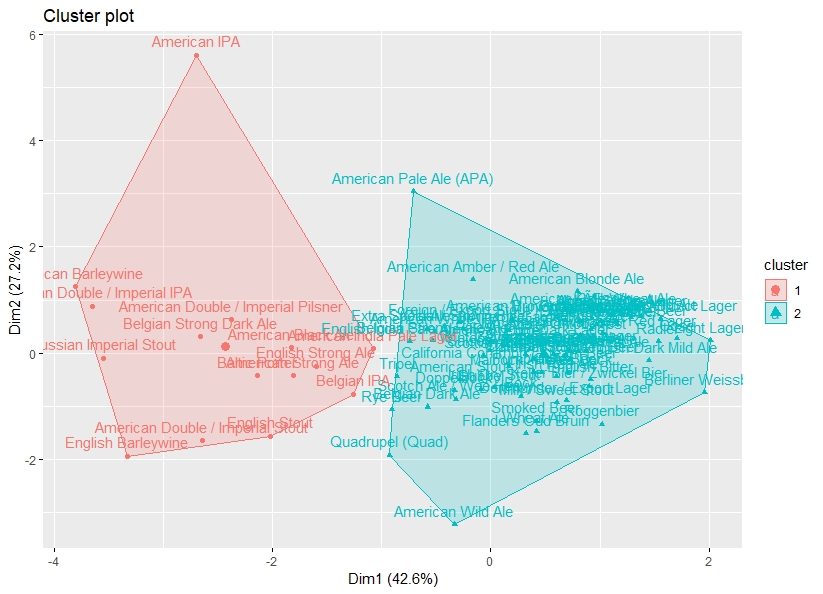
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **LHS** |  | **RHS** | **Support** | **Confidence** | **Lift** | **Count** |
| {ibu=0-20,ounces=12 ounces or less} | => | {abv=Normal} | 0.1523132 | 0.9344978 | 1.5686612 | 214 |
| {ibu=61-80} | => | {abv=Strong} | 0.1501779 | 0.9134199 | 2.3765833 | 211 |
| {ibu=61-80,style=American IPA} | => | {abv=Strong} | 0.1024911 | 0.9 | 2.3416667 | 144 |
| {ibu=0-20} | => | {abv=Normal} | 0.2092527 | 0.8698225 | 1.4600963 | 294 |
| {ibu=21-40,ounces=12 ounces or less} | => | {abv=Normal} | 0.1736655 | 0.829932 | 1.3931355 | 244 |
| {ibu=21-40} | => | {abv=Normal} | 0.2647687 | 0.8051948 | 1.3516114 | 372 |
| {style=American IPA} | => | {abv=Strong} | 0.166548 | 0.7774086 | 2.0227021 | 234 |
| {style=American IPA,ounces=12 ounces or less} | => | {abv=Strong} | 0.1067616 | 0.7537688 | 1.9611949 | 150 |
| {abv=Normal,ibu=0-20} | => | {ounces=12 ounces or less} | 0.1523132 | 0.7278912 | 1.1275491 | 214 |
| {abv=Normal} | => | {ounces=12 ounces or less} | 0.4142349 | 0.6953405 | 1.0771261 | 582 |
| {ibu=61-80} | => | {style=American IPA} | 0.113879 | 0.6926407 | 3.2330903 | 160 |
| {ibu=41-60} | => | {ounces=12 ounces or less} | 0.116726 | 0.6861925 | 1.0629553 | 164 |
| {abv=Strong,ibu=61-80} | => | {style=American IPA} | 0.1024911 | 0.6824645 | 3.1855899 | 144 |
| {ibu=0-20} | => | {ounces=12 ounces or less} | 0.1629893 | 0.6775148 | 1.049513 | 229 |
| {style=American IPA} | => | {ounces=12 ounces or less} | 0.141637 | 0.6611296 | 1.0241312 | 199 |
| {abv=Normal,ibu=21-40} | => | {ounces=12 ounces or less} | 0.1736655 | 0.655914 | 1.016052 | 244 |
| {ounces=12 ounces or less} | => | {abv=Normal} | 0.4142349 | 0.6416759 | 1.0771261 | 582 |
| {abv=Strong,style=American IPA} | => | {ounces=12 ounces or less} | 0.1067616 | 0.6410256 | 0.992989 | 150 |
| {ibu=21-40} | => | {ounces=12 ounces or less} | 0.2092527 | 0.6363636 | 0.9857673 | 294 |
| {ibu=41-60} | => | {abv=Normal} | 0.1053381 | 0.6192469 | 1.0394765 | 148 |
| {abv=Strong,style=American IPA} | => | {ibu=61-80} | 0.1024911 | 0.6153846 | 3.7429237 | 144 |
| {ibu=61-80} | => | {ounces=12 ounces or less} | 0.1010676 | 0.6147186 | 0.9522378 | 142 |

### Clustering – K-means

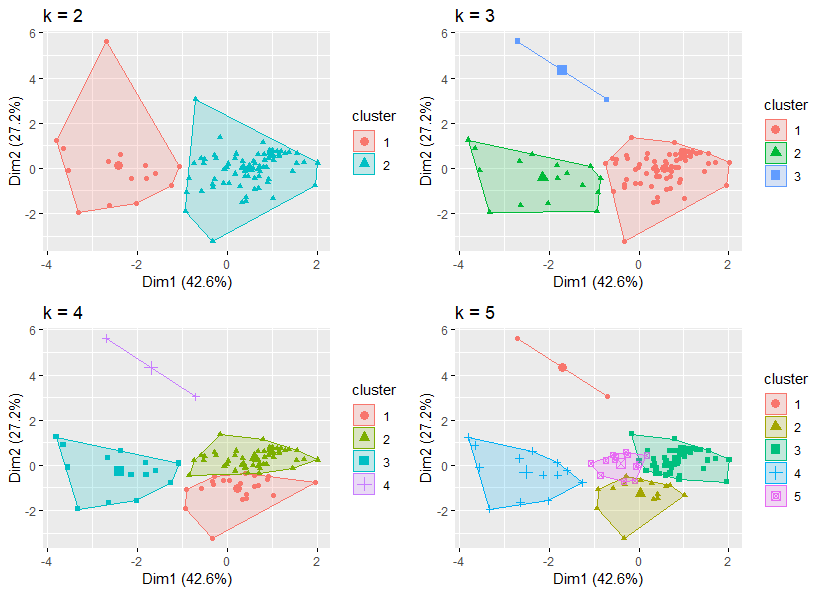
Unsupervised Machine Learning (ML), also known as clustering, is an exploratory data analysis technique used for identifying groups (i.e., clusters) in the data set of interest. Each group contains observations with similar profile according to a specific criteria. Similarity between observations is defined using some inter-observation distance measures including Euclidean and correlation-based distance measures. **Exhibit 19: Heatmap of Beer Styles** (in the “About the Data” section) depicts the euclidean distance distance between each style of beer. This starts to illustrate which styles have large dissimilarities (blue) versus those that appear to be fairly similar (pink).

**Model 3** first explores partitioning algorithms (or K-means clusters). Partitioning algorithms are clustering approaches that split the data sets, containing n observations, into a set of k groups (i.e. clusters). The algorithms require the analyst to specify the number of clusters to be generated. The kmeans function from the cluster library is used to compute the clusters. **Model 3** (below) visualizes what K=2, K=3, K=4 and K=5 clusters looks like.

**Model 3: K-means with K=2 (enlarged to see detail)**

****

**Model 3: K-means with K=2, 3, 4 and 5**



**Table 3: Clusters Using K-means with K = 3**

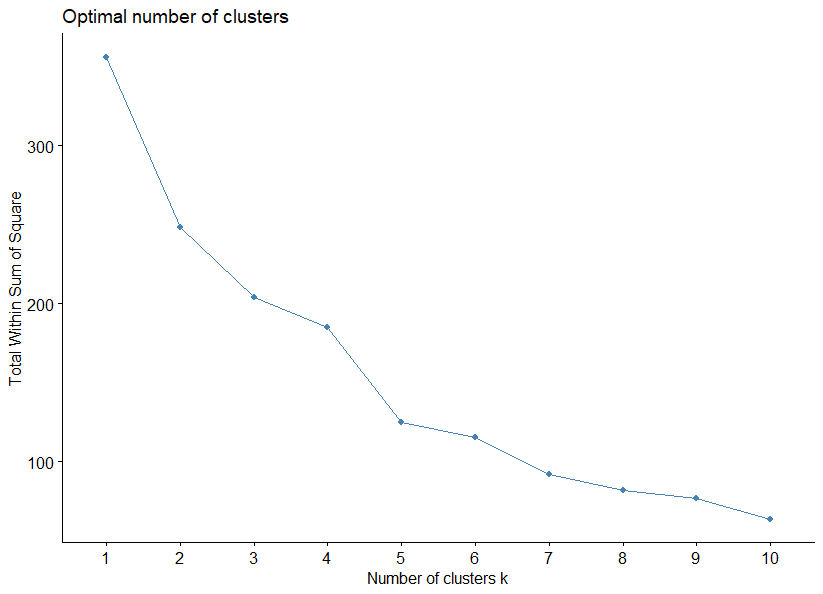
|  |  |  |
| --- | --- | --- |
| **Cluster 1** | **Cluster 2** | **Cluster 3** |
| American Amber / Red Ale, American Blonde Ale, American Pale Wheat Ale, Fruit / Vegetable Beer, American Porter, American Brown Ale, Witbier, Hefeweizen, KÃ¶lsch, MÃ¤rzen / Oktoberfest, Saison / Farmhouse Ale, Cream Ale, German Pilsener, American Stout, American Pale Lager, Czech Pilsener, American Pilsner, American Amber / Red Lager, Scotch Ale / Wee Heavy, Extra Special / Strong Bitter (ESB), Munich Helles Lager, Pumpkin Ale, English Brown Ale, Vienna Lager, Scottish Ale, Oatmeal Stout, American Adjunct Lager, Irish Red Ale, Belgian Pale Ale, Winter Warmer, Altbier, Gose, Milk / Sweet Stout, English India Pale Ale (IPA), American White IPA, Dortmunder / Export Lager, Berliner Weissbier, American Dark Wheat Ale, Euro Dark Lager, Herbed / Spiced Beer, English Pale Ale, Schwarzbier, Belgian Dark Ale, Bock, English Bitter, Irish Dry Stout, Munich Dunkel Lager, Foreign / Export Stout, Light Lager, Abbey Single Ale, Belgian Strong Pale Ale, Chile Beer, Dunkelweizen, Grisette, Wheat Ale, Keller Bier / Zwickel Bier, Radler, American Wild Ale, California Common / Steam Beer, Dubbel, Maibock / Helles Bock, English Pale Mild Ale, Flanders Oud Bruin, Other, Smoked Beer, BiÃ¨re de Garde, Doppelbock, English Dark Mild Ale, Euro Pale Lager, Old Ale, Roggenbier | American Black Ale, American Double / Imperial IPA, American Strong Ale, Rye Beer, Russian Imperial Stout, American Double / Imperial Stout, Tripel, Baltic Porter, American India Pale Lager, Belgian IPA, English Barleywine, English Strong Ale, American Barleywine, Belgian Strong Dark Ale, American Double / Imperial Pilsner, English Stout, Quadrupel (Quad) | American IPA, American Pale Ale (APA) |

As the analyst specifies the number of clusters to use; preferably the analyst would like to use the optimal number of clusters. To aid the analyst, the following are the three most popular methods for determining the optimal clusters:

* Elbow method
* Silhouette method
* Gap statistic

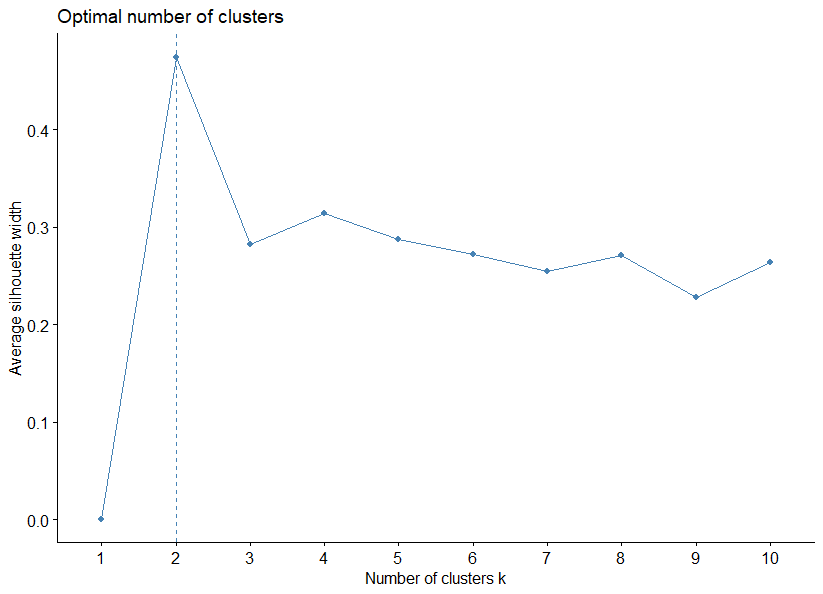
The basic idea behind cluster partitioning methods, such as k-means clustering, is to define clusters such that the total intra-cluster variation (known as total within-cluster variation or total within-cluster sum of square) is minimized (Elbow Method).

**Model 3: Elbow Method to Determine Appropriate Number of Clusters (K-means)**



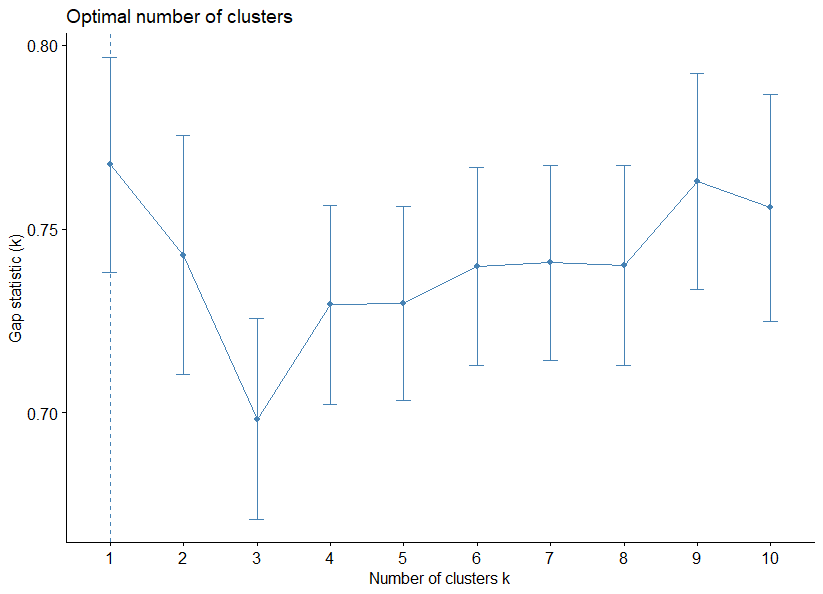
The average silhouette approach measures the quality of a clustering. That is, it determines how well each object lies within its cluster. A high average silhouette width indicates a good clustering. The average silhouette method computes the average silhouette of observations for different values of k. The optimal number of clusters k is the one that maximizes the average silhouette over a range of possible values for k.

**Model 3: Average Silhouette to Determine Appropriate Number of Clusters (K-means)**



The gap statistic compares the total intracluster variation for different values of k with their expected values under null reference distribution of the data (i.e. a distribution with no obvious clustering). The reference dataset is generated using Monte Carlo simulations of the sampling process.

**Model 3: Gap Statistic to Determine Appropriate Number of Clusters (K-means)**



### Clustering – Hierarchical

Hierarchical clustering is an alternative approach to k-means clustering for identifying groups in the dataset. It does not require the analyst to pre-specify the number of clusters to be generated as is required by the k-means approach. Furthermore, hierarchical clustering has an added advantage over K-means clustering in that it results in an attractive tree-based representation of the observations, called a dendrogram.

There are different functions available in R for computing hierarchical clustering. The commonly used functions are:

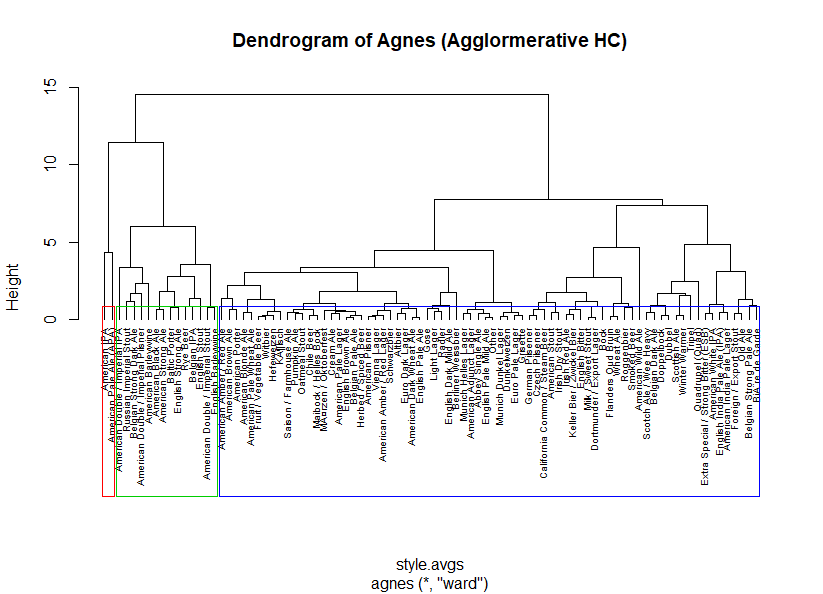
* hclust [in stats package] and agnes [in cluster package] for agglomerative hierarchical clustering (HC)
* diana [in cluster package] for divisive HC

The agnes function calculates the agglomerative coefficient, which measures the amount of clustering structure found (values closer to 1 suggest strong clustering structure). Table 3 below shows the calculated agglomerative coefficient (AC) for the following clustering methods: average, single, complete and ward.

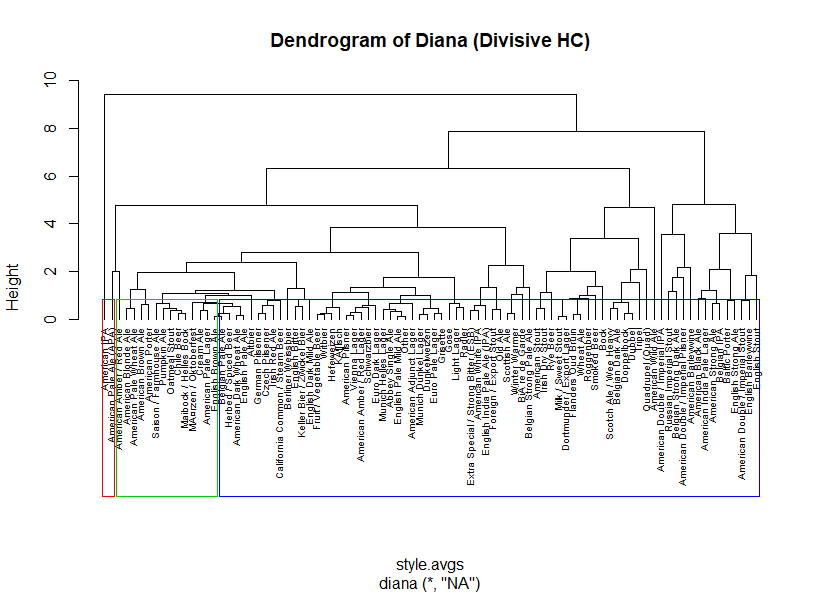
**Table 4:** **Agglormerative Coefficient**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | Average | Single | Complete | Ward |
| **AC** | 0.9128545 | 0.8685785 | 0.9267182 | 0.9439754 |

**Model 4: Dendrogram of Agnes (Hierarchial Clustering), Method = Ward**

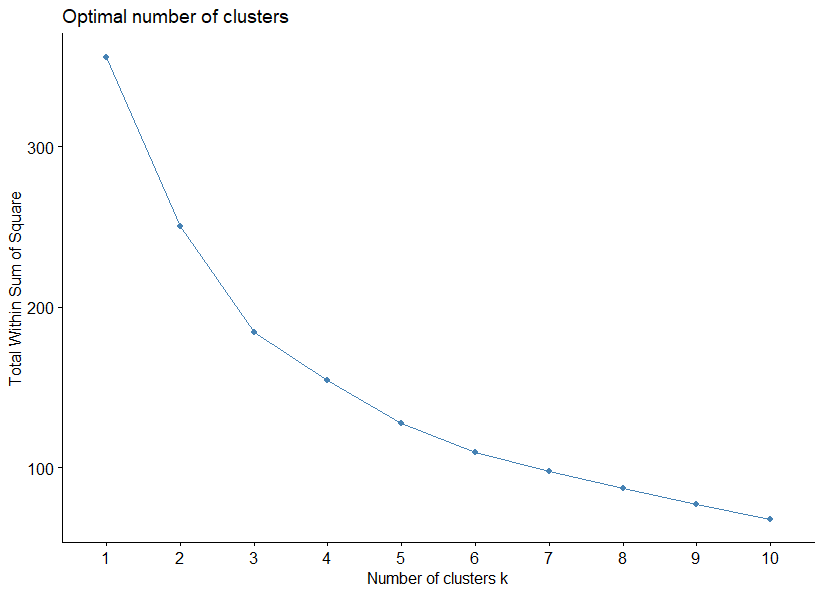


The R function diana provided by the cluster package allows us to perform divisive hierarchical clustering. Diana works similar to agnes; however, there is no method to provide.

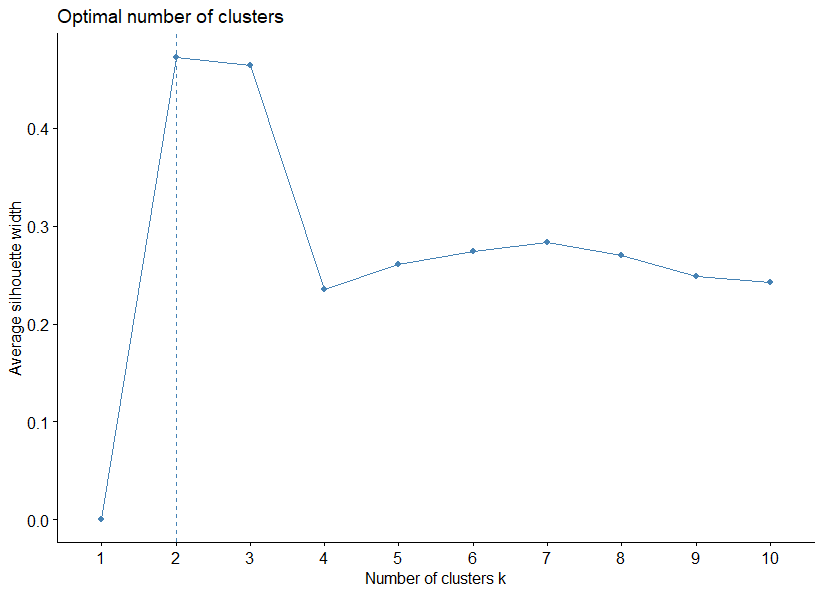
**Model 4: Dendrogram of Diana (Divisive HC)**

Similar to K-means clustering the optimal number of clusters can be calculated using the elbow, average silhouette and gap statistic methods.

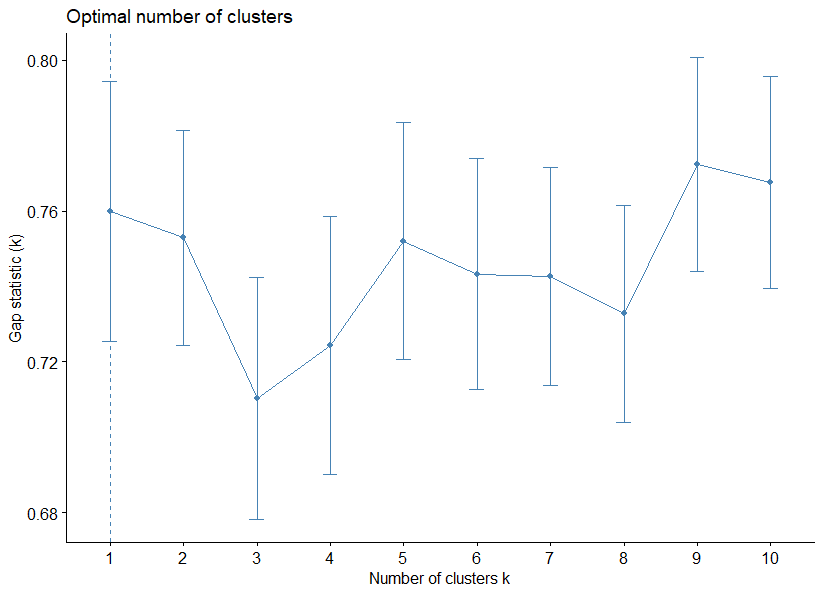
**Model 4: Elbow Method to Determine Appropriate Number of Clusters (HC)**

****

**Model 4: Average Silhouette Method to Determine Appropriate Number of Clusters (HC)**

****

**Model 4: Gap Statistic to Determine Appropriate Number of Clusters (HC)**

****

### Decision Trees

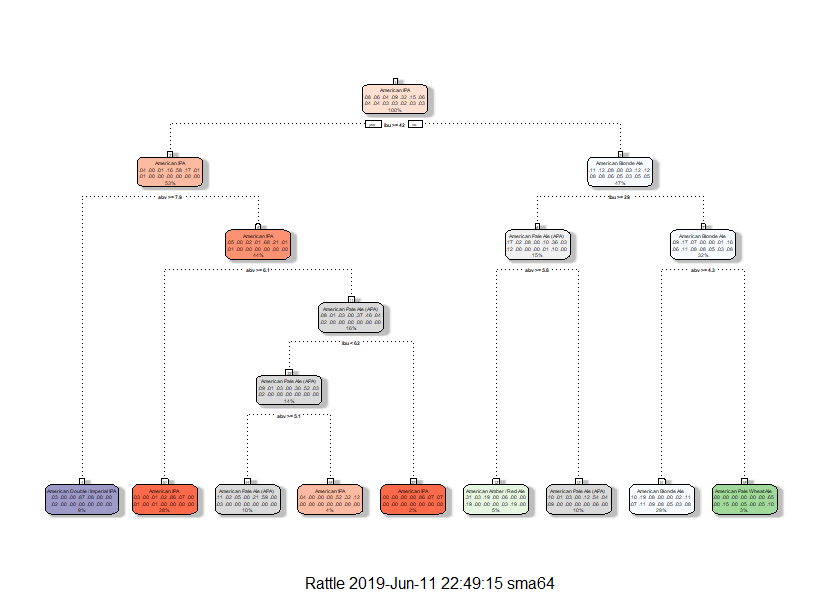
Recursive partitioning is a fundamental tool in data mining. It helps explore the structure of a set of data, while developing easy to visualize decision rules for predicting a categorical (classification tree) or continuous (regression tree) outcome. Classification and regression trees are generated through the rpart library. To grow a tree the following function is used:

rpart(formula, data=, method=,control=) where

* Formula is in the format “outcome ~ predictor1 + predictor2 + predictor 3… (note, this is read predict outcome using a function of predictor1, predictor 2, predictor3, etc.)
* Data= specifies the dataframe
* Method= “class” for classification tree or “anova” for a regression tree (note, the models defined below all use class)
* Control= optional parameters for controlling tree growth. For example, control=rpart.control(minsplit=30, cp=0.001) requires that the minimum number of observations in a node be 30 before attempting a split and that a split must decrease the overall lack of fit by a factor of 0.001 (cost complexity factor) before being attempted.

**Model 5.1** leverages style ~ abv+ibu+ounces and the default settings of rpart. The decision tree in **Model 5.1** depicts the results of the first fit model. **Table 5** lists the variable importance of the formula used in Model 5.1.

**Model 5.1: style ~ abv+ibu+ounces (default settings)**

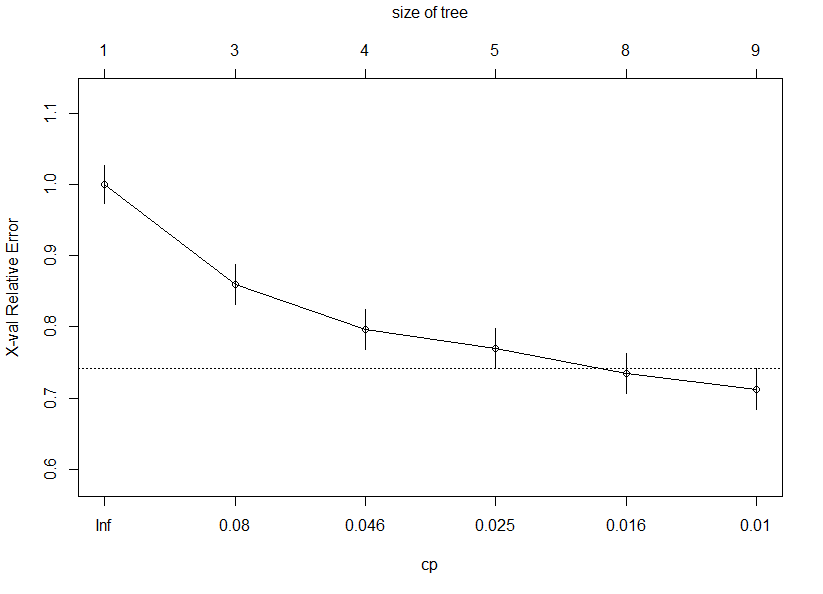
****

**Table 5: Variable Importance of Model 5.1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | Abv | Ibu | Ounces |
| **Importance** | 54 | 45 | 1 |

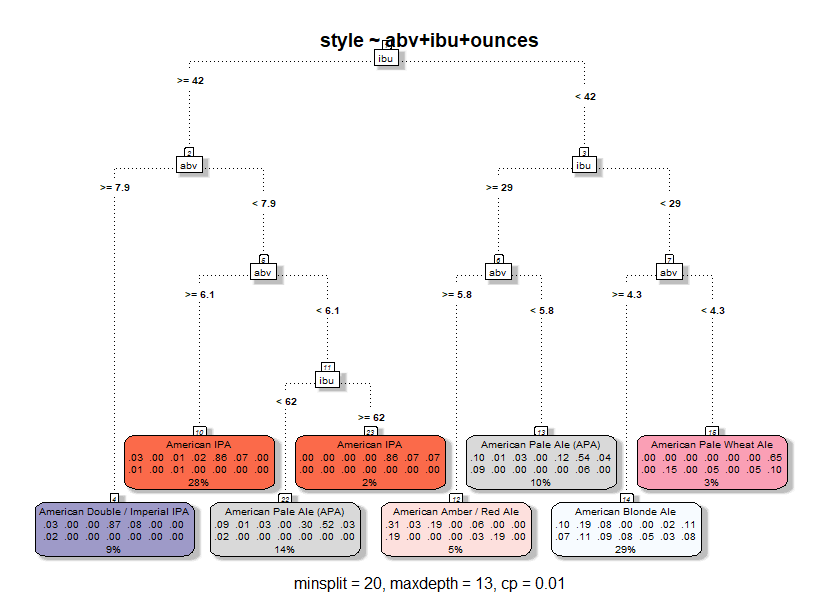
Behind the scenes rpart is automatically applying a range of cost complexity (α) values to prune the tree. In the plot below there are diminishing returns after 8 terminal nodes, the dashed line which goes through the point |T|~8. It’s common to instead use the smallest tree within one standard deviation of the minimum cross validation error, thus, a tree with six or seven terminal nodes would experience similar results within a small margin of error (see **Model 5.1: Cost Complexity**).

**Model 5.1: Cost Complexity for** **style ~ abv+ibu+ounces (default settings)**



**Model 5.2** leverages style ~ abv+ibu+ounces and the following parameters: minsplit = 19, maxdepth = 11, cp = 0.011. The decision tree in **Model 5.2** depicts the results of the first fit model.

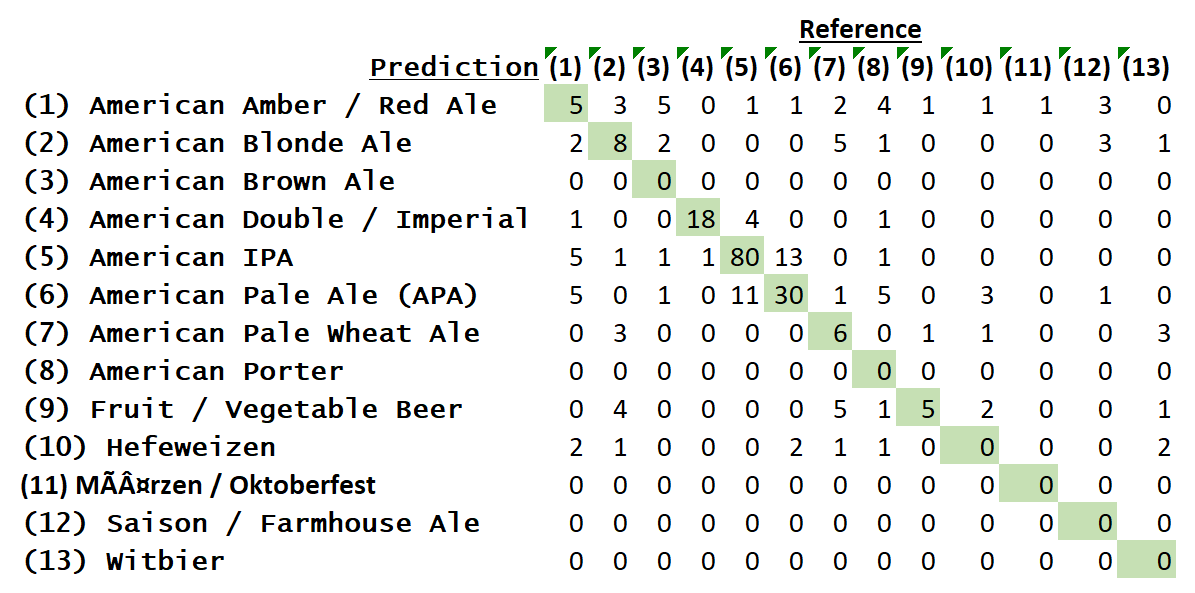
**Model 5.2: style ~ abv+ibu+ounces (minsplit = 19, maxdepth = 11, cp = 0.011)**

****

Single tree models suffer from high variance. Although pruning the tree helps reduce this variance, there are alternative methods that actually exploite the variability of single trees in a way that can significantly improve performance over and above that of single trees. Bootstrap aggregating (bagging) is one such approach. Bagging combines and averages multiple models. Averaging across multiple trees reduces the variability of any one tree and reduces overfitting, which improves predictive performance.

**Model 5.3** leverages style ~ abv+ibu+ounces and the following parameters: minsplit = 19, maxdepth = 11, cp = 0.011. What sets this model apart from Model 5.2 is that it’s used within the bagging function (bagging()). By default bagging performs 25 bootstrap samples and trees. The decision tree in **Model 5.3** depicts the results of the first fit model.

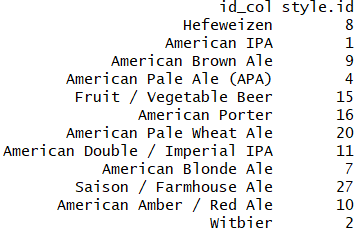
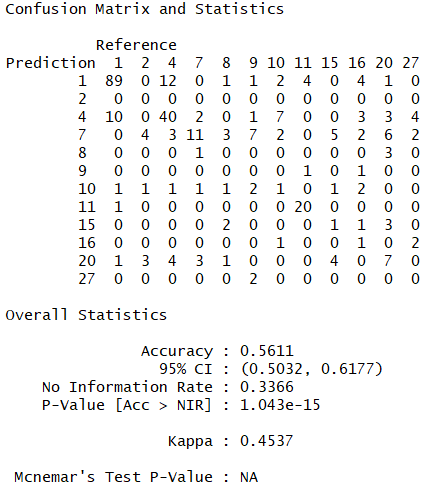
**Model 5.3: Confusion Matrix: style ~ abv+ibu+ounces (minsplit = 19, maxdepth = 11, cp = 0.011) with Bagging**



### Naïve Bayes

The Naïve Bayes classifier is a simple probabilistic classifier which is based on Bayes theorem but with strong assumptions regarding independence. Bayesian probability incorporates the concept of conditional probability, the probabilty of event A given that event B has occurred [denoted as P(A|B)]. Consequently, the naïve Bayes classifier makes a simplifying assumption (hence the name) to allow the computation to scale. With naïve Bayes, we assume that the predictor variables are conditionally independent of one another given the response value.

**Model 6: Confusion Matrix Using Naïve Bayes e1071**



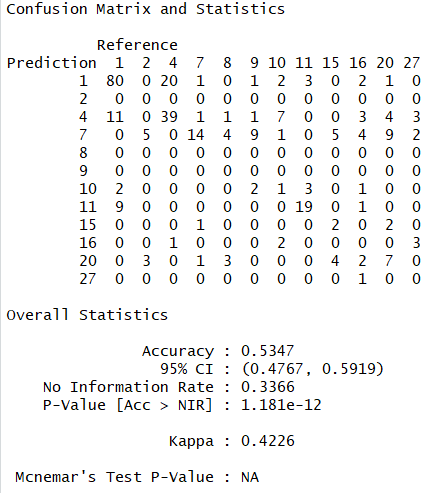
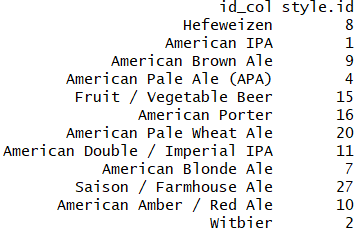
Note: Style is used as dependent variable whereas abv, ibu and ounces are used as independent variable

### Support Vector Machine

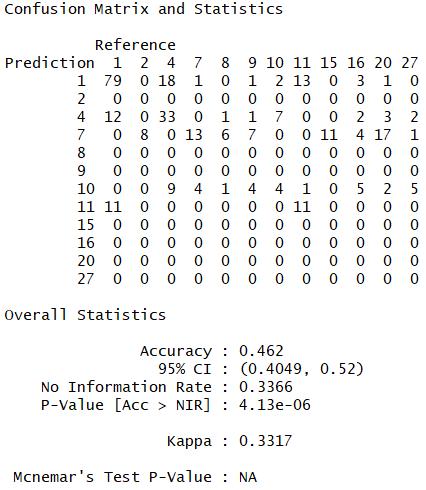
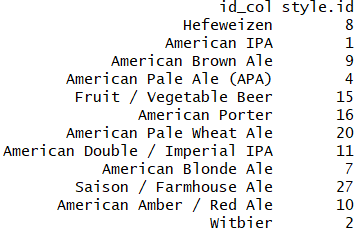
In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. However, they are mostly used in classification problems. SVMs are based on the idea of finding a hyperplane that best divides a dataset into two classes. Support vectors are the data points nearest to the hyperplane, the points of a data set that, if removed, would alter the position of the dividing hyperplane. Because of this, they can be considered the critical elements of a data set.

Think of a hyperplane as a line that linearly separates and classifies a set of data. Intuitively, the further from the hyperplane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as far away from the hyperplane as possible, while still being on the correct side of it. Some data sets can’t be classified in 2D and are instead mapped in a higher dimension (e.g., 3D), this is known as kernelling. In three dimensions, the hyperplane can no longer be a line. It must now be a plane. The idea is that the data will continue to be mapped into higher and higher dimensions until a hyperplane can be formed to segregate it.[[2]](#footnote-2)

**Model 7.1: Model Using SVM with Polynomial Kernel**

**Model 7.2: Model Using SVM with Linear Kernel**

### K-Nearest Neighbor Pre-Processing

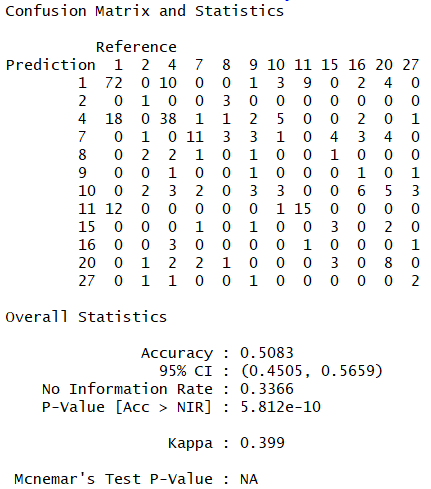
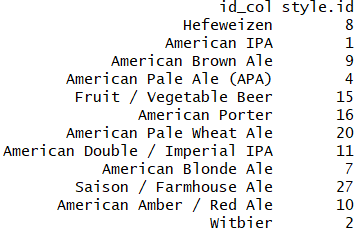
The KNN or k-nearest neighbors’ algorithm is one of the simplest machine learning algorithms and is an example of instance-based learning, where new data are classified based on stored, labeled instances.

More specifically, the distance between the stored data and the new instance is calculated by means of some kind of a similarity measure. This similarity measure is typically expressed by a distance measure such as the Euclidean distance, cosine similarity or the Manhattan distance.

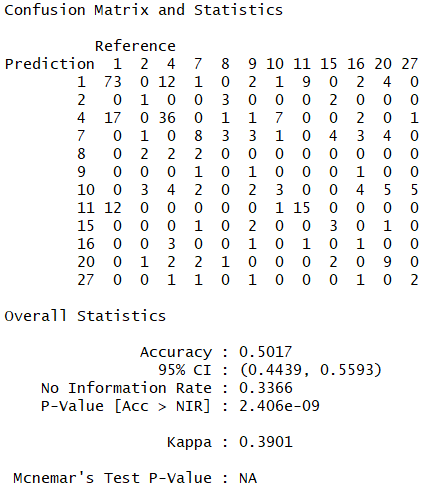
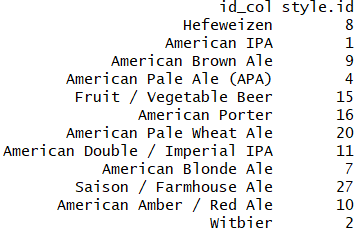
After the distance of the new point to all stored data points has been calculated, the distance values are sorted and the k-nearest neighbors are determined. The labels of these neighbors are gathered and a majority vote or weighted vote is used for classification or regression purposes.

In other words, the higher the score for a certain data point that was already stored, the more likely that the new instance will receive the same classification as that of the neighbor. In the case of regression, the value that will be assigned to the new data point is the mean of its k nearest neighbors.[[3]](#footnote-3)

**Model 8.1: Model using kNN algorithm when k=5 (5 nearest neighbors)**

**Model 8.2: Model using kNN algorithm when k=3 (3 nearest neighbors)**

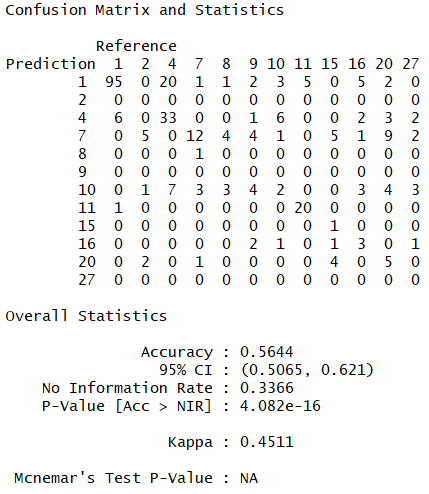
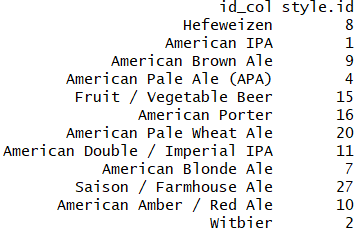
 

### Random Forest Pre-Processing

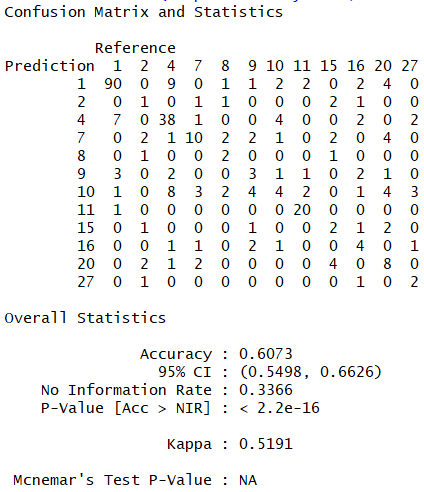
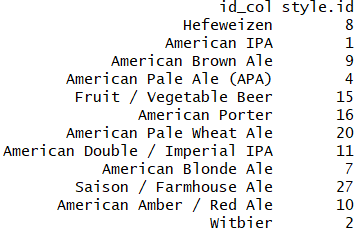
Random Forest is a very powerful ensembling machine learning algorithm which works by creating multiple decision trees and then combining the output generated by each of the decision trees. Decision tree is a classification model which works on the concept of information gain at every node. For all the data points, decision tree will try to classify data points at each of the nodes and check for information gain at each node. It will then classify at the node where information gain is maximum. It will follow this process subsequently until all the nodes are exhausted or there is no further information gain. Decision trees are very simple and easy to understand models; however, they have very low predictive power. In fact, they are called weak learners.

Random Forest works on the same weak learners. It combines the output of multiple decision trees and then finally come up with its own output. Random Forest works on the same principle as Decision Tress; however, it does not select all the data points and variables in each of the trees. It randomly samples data points and variables in each of the tree that it creates and then combines the output at the end. It removes the bias that a decision tree model might introduce in the system. Also, it improves the predictive power significantly.[[4]](#footnote-4)

**Model 9.1: Model Using Random forest with default parameters**

**Model 9.2: Model Using Random forest with mtry=2 parameters**

## Result(s)

### Association Rule Mining

The following insights were inferred from **Model 1** and **Model 2**:

* Model 1 (support = 0.2, confidence = 0.6) didn’t return any rules that included a style
* Model 2 (support = 0.1, confidence = 0.6) returned one rule that included a style
* If “ABV = Strong (6+)” and “IBU = between 61 to 80”, then “Style = American IPA”. Therefore, a relationship between a strong ABV, an IBU between 61 to 80 and American IPAs exists

### Clustering

The following insights were inferred from **Model 3** (K-means) and **Model 4** (HC):

* The “ward” method of Hierarchical Clustering provided the strongest agglomerative coefficient at 0.9439754
* K-means and Hierarchical (using both Agnes and Diana) and three clusters put American IPA and American Pale Ale (APA) into a separate cluster
* Cluster 1 and 2 differs slightly between K-means and Hierarchical Clustering
* Using K-means the following were identified as the optimal number of clusters:
  + Elbow Method = 3
  + Average Silhouette Method = 2
  + Gap Statistic Method = 1
* Using Hierarchical clustering the following were identified as the optimal number of clusters:
  + Elbow Method = 3
  + Average Silhouette Method = 2
  + Gap Statistic Method = 1

### Decision Trees

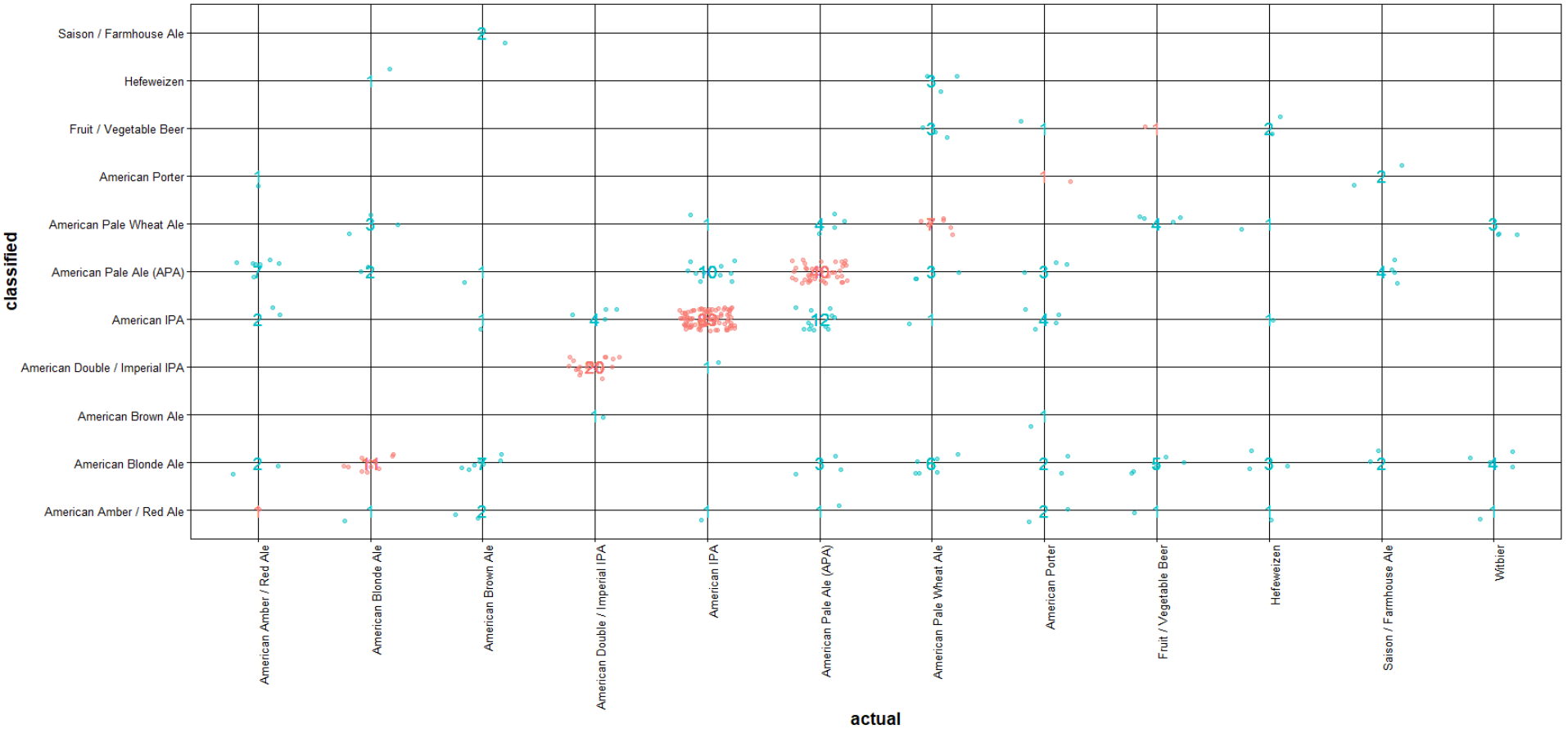
The following insights were inferred from Models 5.1 (default), 5.2 (pruned) and 5.3 (bagging):

* While none of the models proved to be fairly accurate there were incremental improvements at each iteration
  + Accuracy of the default tree (Model 5.1) is 50.87%
  + Accuracy of the pruned tree (Model 5.2) is 51.22%
  + Accuracy of the bagged tree (Model 5.3) is 52.96%
* The test dataset contained 13 styles; however, the decision tree was only able to distinguish between six styles (i.e., the ones with more observations)
* The following styles had the highest accuracy: American Double / Imperial IPA (96%), American IPA (85%) and American Pale Ale (APA) (77%)
* The following style had the lowest accuracy: Hefeweizen (48%)

The accuracy of Model 6 (default settings) is about 56%.

**Exhibit 20: Distribution of Naïve Bayes Prediction Error Across Styles**

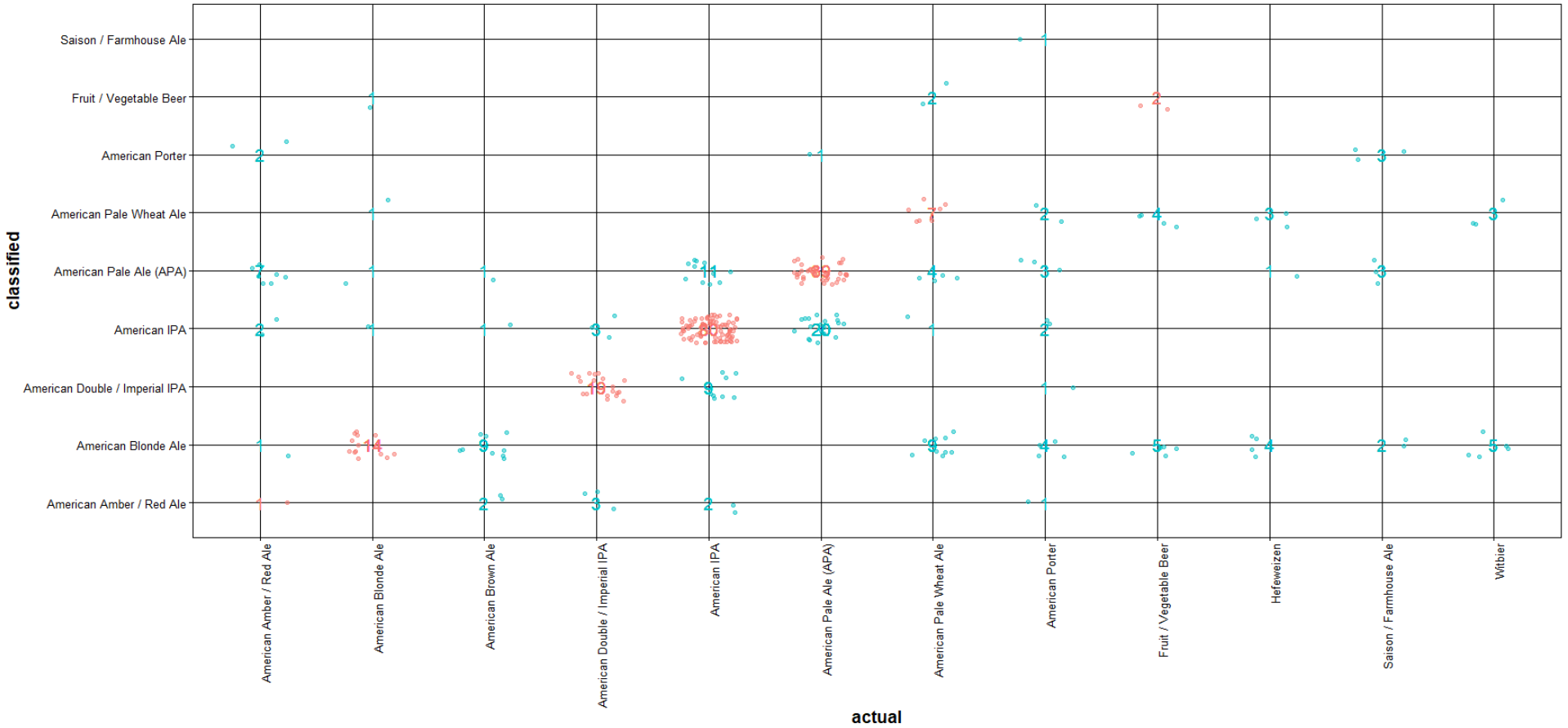
### Naïve Bayes



### Support Vector Machines

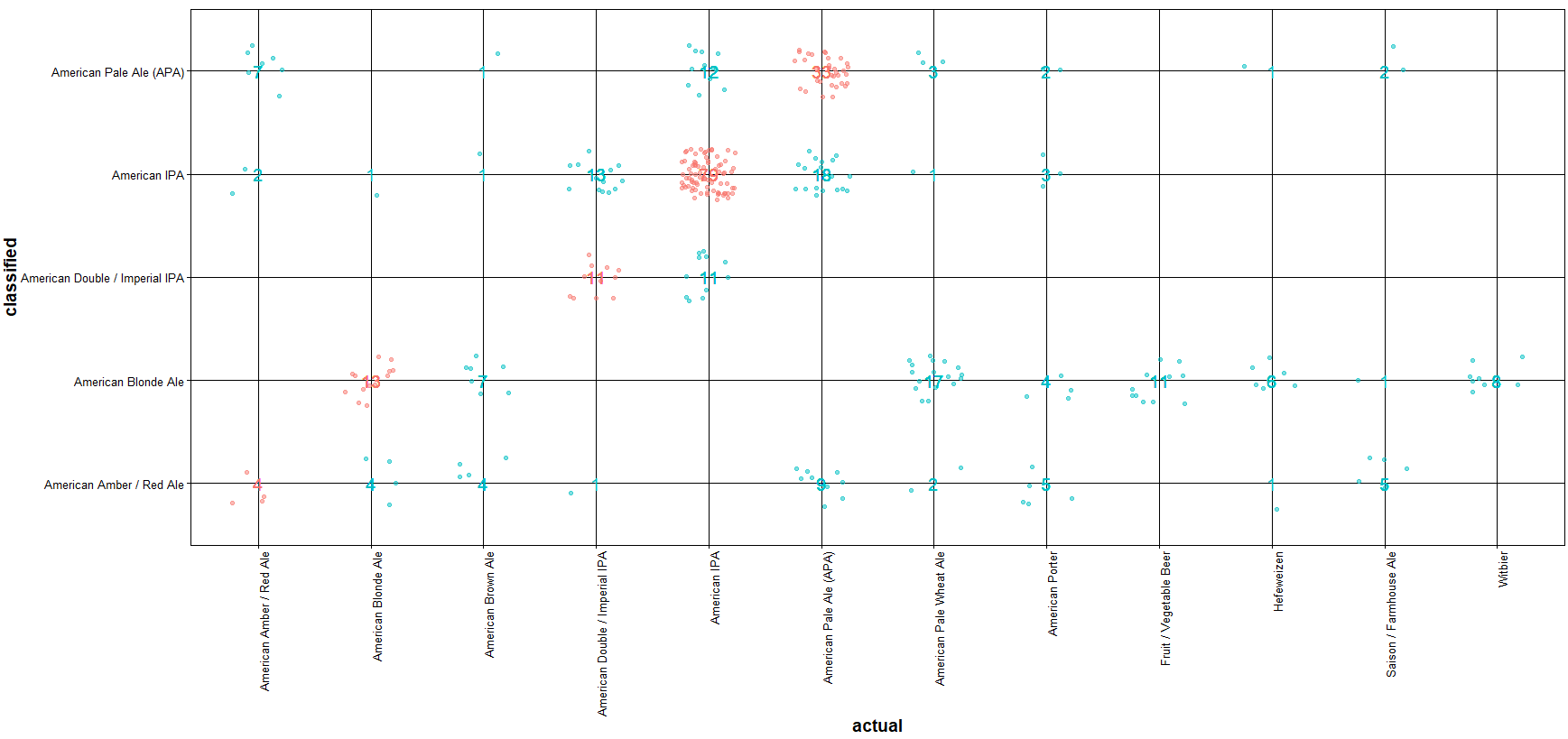
The accuracy of this **Model 7.1** (polynomial kernel) is about 53%.

**Exhibit 21: Distribution of SVM Prediction Error Across Styles (polynomial kernel)**



The accuracy of this **Model 7.2** (linear kernel) is about 46%.

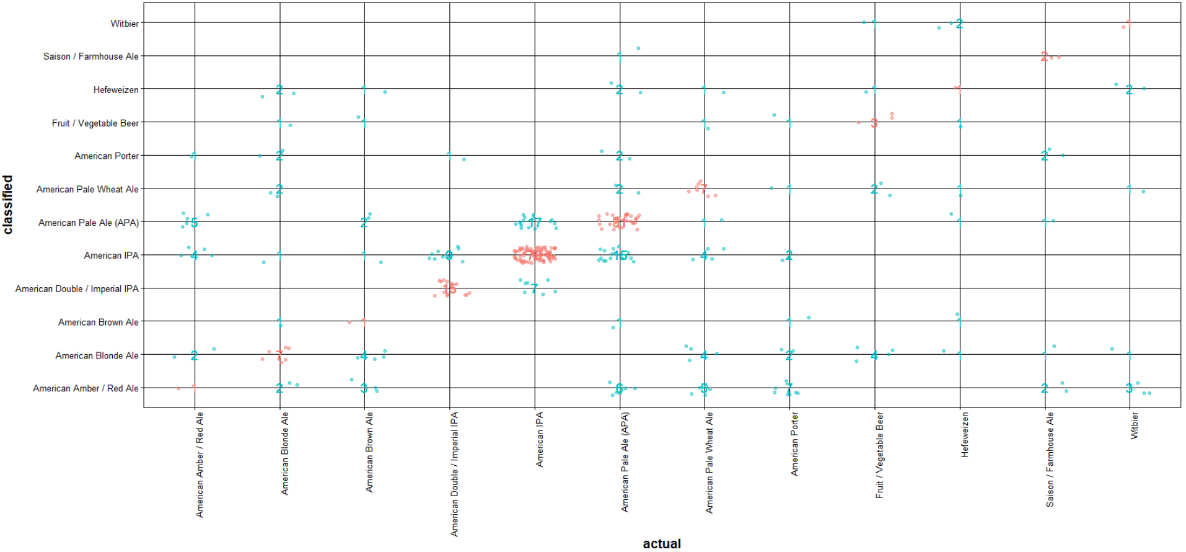
**Exhibit 22: Distribution of SVM Prediction Error Across Styles (linear kernel)**



### K-Nearest Neighbor

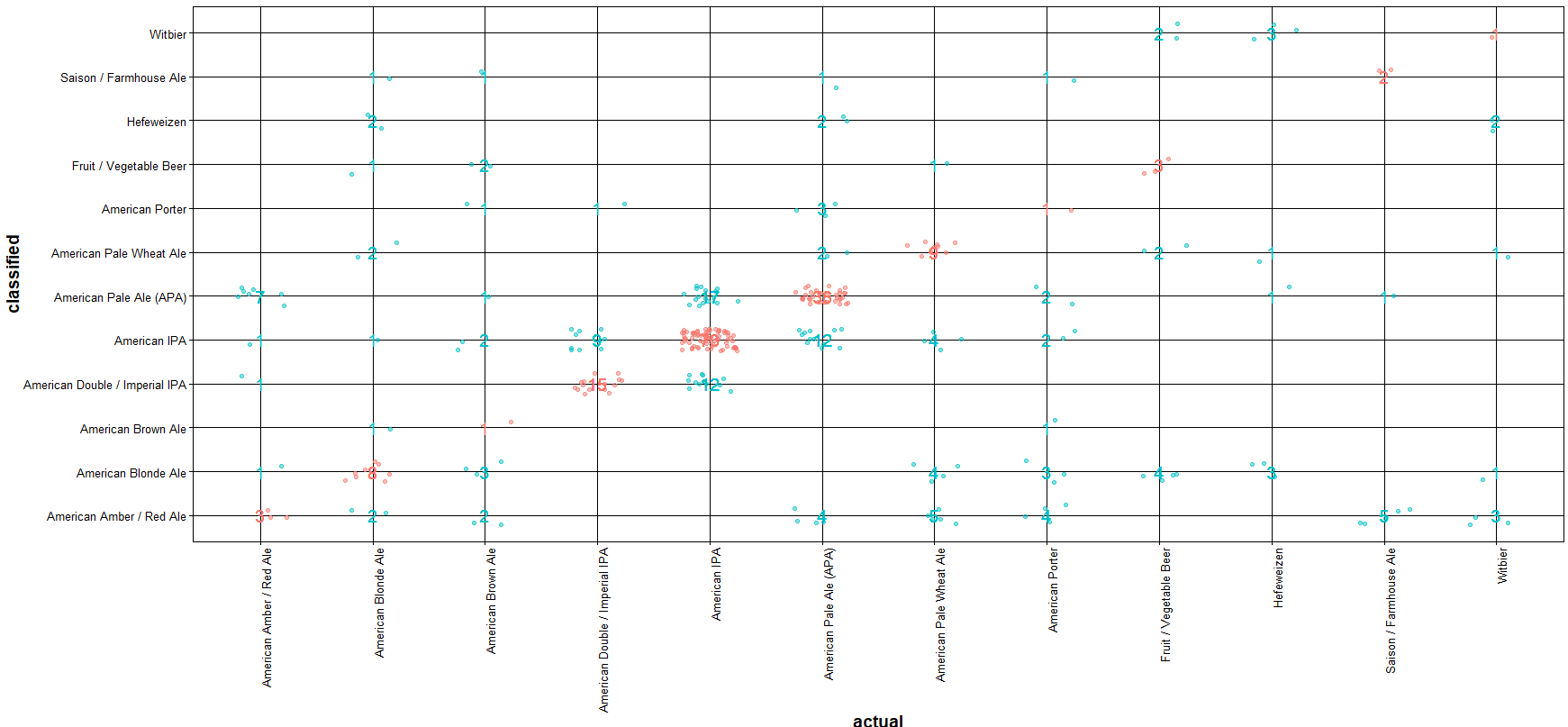
The accuracy of **Model 8.1** (when k=5) is about 50%.

**Exhibit 23: Distribution of kNN Prediction Error Across Styles (k=5)**

****

The accuracy of **Model 8.2** (when k=2) is about 50%.

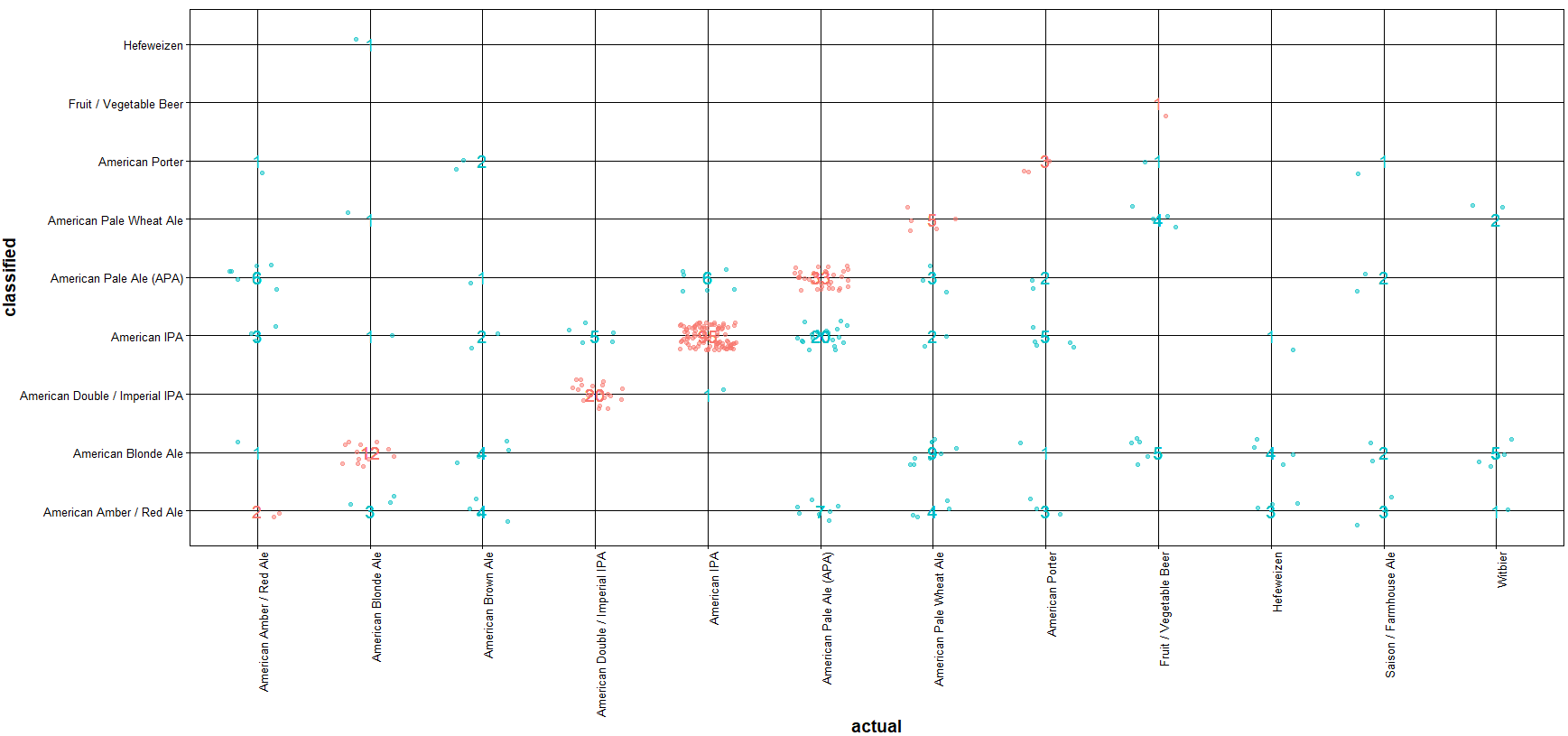
**Exhibit 24: Distribution of kNN Prediction Error Across Styles (k=2)**



### Random Forest

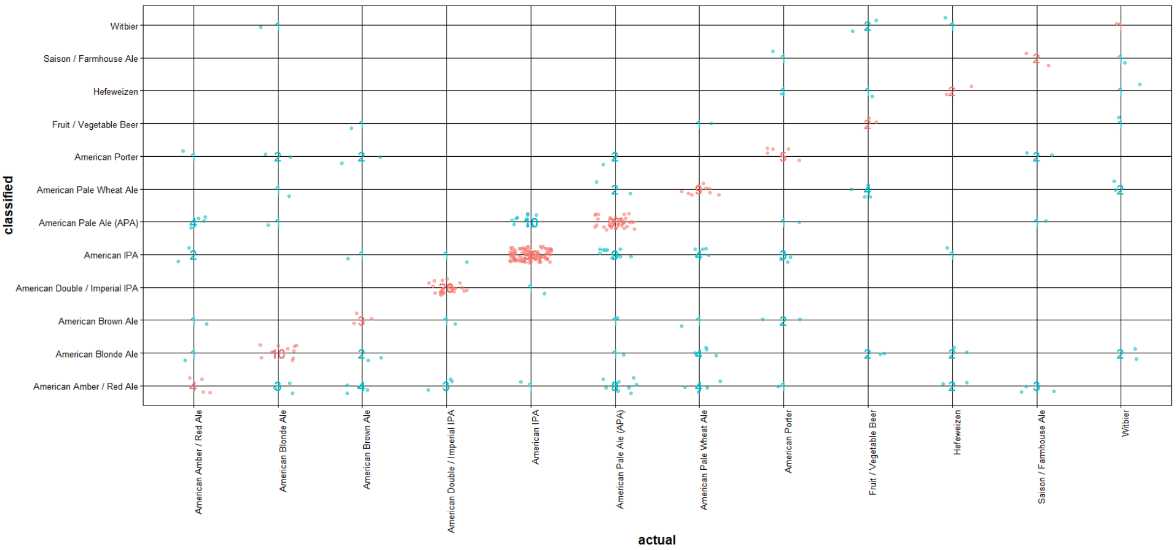
The accuracy of this **Model 9.1** (default parameters) is about 56%.

**Exhibit 25: Distribution of Random Forest Prediction Error Across Styles (default parametrs)**



The accuracy of this Model 9 (when mtry=2) is about 61%.

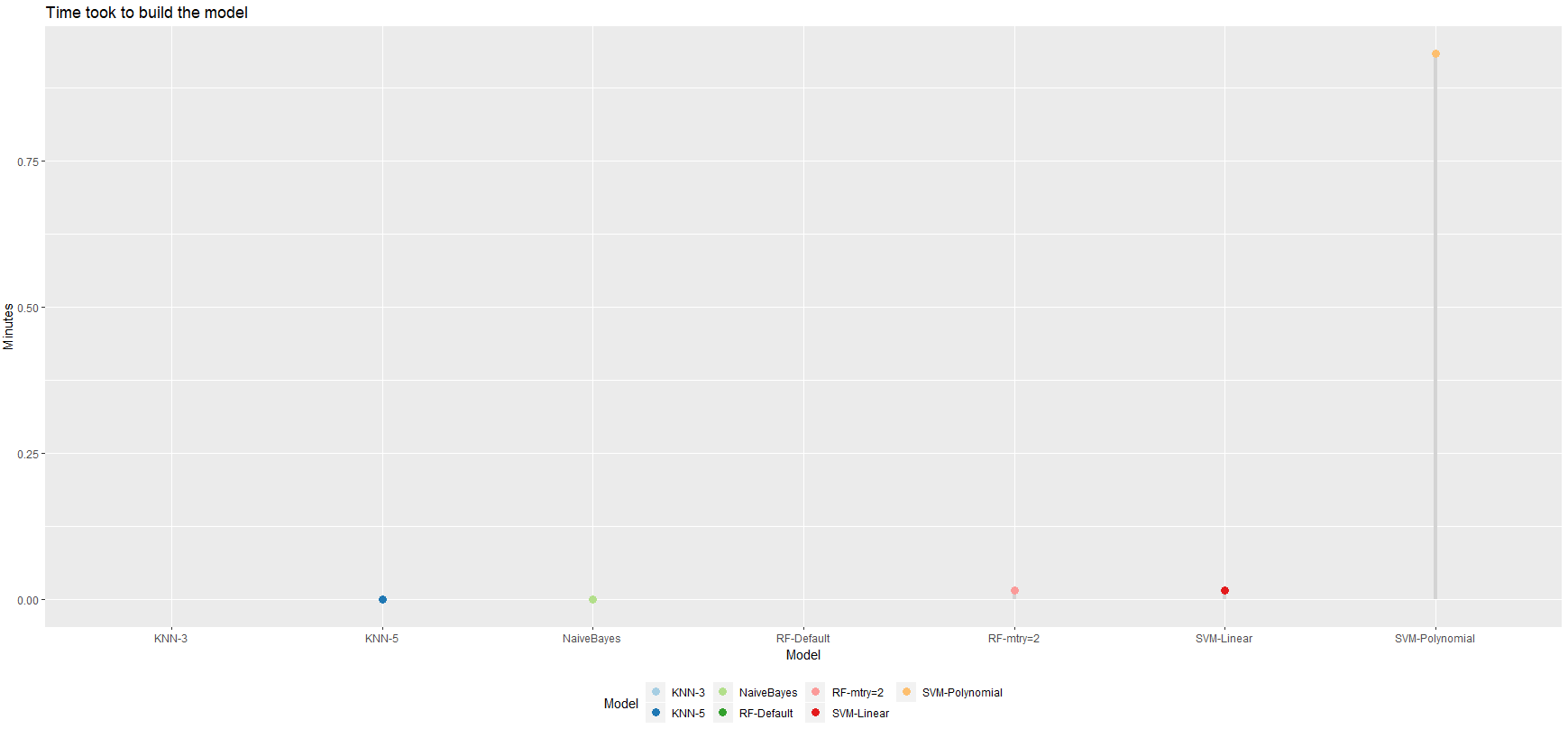
**Exhibit 26: Distribution of Random Forest Prediction Error Across Styles (mtry=2)**

****

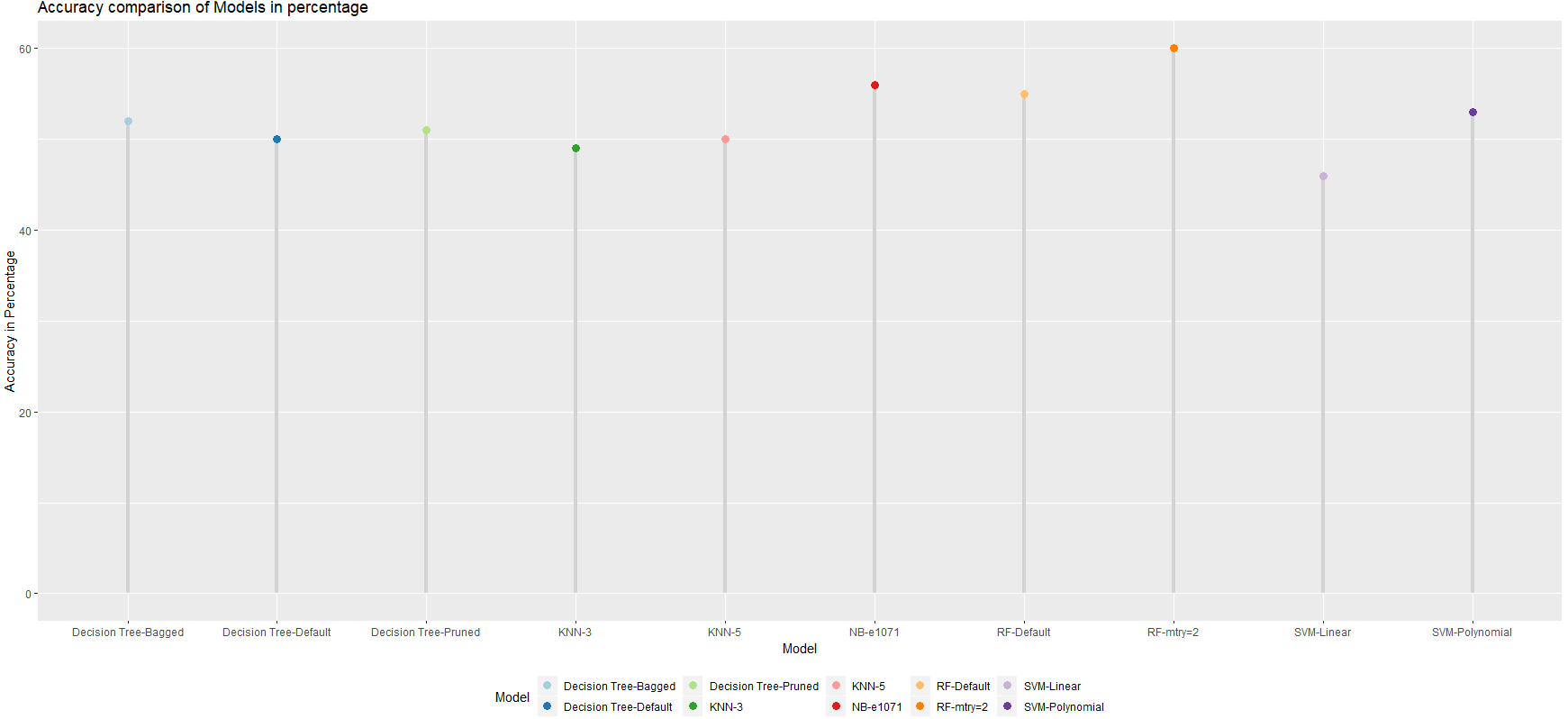
### Supervised Learned Model Comparison

Exhibits 27 and 28 show the average time to build the models and average accuracy of the models after performing three cross validation when using the following four supervised learned models: Naïve Bayes, Support Vector Machines, K-nearest neighbor and Random Forests. Note, the best results of each of the models were used in the exhibits.

**Exhibit 27: Average Time to Build Models**



**Exhibit 28: Average Accuracy of Models**



# Conclusion

Annually the Brewers Association, the not-for-profit trade group dedicated to promoting and protecting America’s craft brewers releases their “Beer Style Guidelines”. In 2019 there were over 1,000 edits, including style additions, consolidations and removals. For example:

* Style Additions
  + Juicy or Hazy Strong Pale Ale
  + Contemporary Belgian-Style Gueuze Lambic
  + Franconian-Style Rotbier
  + American-Style India Pale Lager
* Style Consolidations
  + Pale and Dark American-Belgo-Style Ale styles
  + Kellerbier or Zwickelbier Ale and Lager styles
  + Breslau-Style Pale and Dark Schoeps styles
  + American-Style Light and Dark Wheat Beer styles
  + Wood-and Barrel-Aged Pale to Amber, Dark and Strong
* Style Removals
  + American-Style Ice Lager[[5]](#footnote-5)

These annual changes are a result of craft brewers using innovation techniques and non-traditional ingredients to develop more complex tastes and styles. As a result, ABV, IBU and ounces alone are not good predictors of style. Using a variety of unsupervised and supervised machine learning techniques only returned a 61% accuracy (Random Forests model). The one consistency among the models is that American IPA and American Pale Ale (APA) are the most distinct beer styles from all other styles.

To develop a more reliable model in predicting beer style the following metrics and ingredients should be taken into consideration:

* Standard Reference Method (color)
* Gravity
* pH of the water used
* Type of hops used
* Type of malt used
* Yeast strain used
* Brewing and fermentation process

1. Source: <https://www.brewersassociation.org/statistics/national-beer-sales-production-data/> [↑](#footnote-ref-1)
2. Source: <http://blog.aylien.com/support-vector-machines-for-dummies-a-simple/> [↑](#footnote-ref-2)
3. Source: <https://www.datacamp.com/community/tutorials/machine-learning-in-r> [↑](#footnote-ref-3)
4. Source: <https://www.r-bloggers.com/how-to-implement-random-forests-in-r/> [↑](#footnote-ref-4)
5. Source: <https://www.brewersassociation.org/press-releases/brewers-association-releases-2019-beer-style-guidelines/> [↑](#footnote-ref-5)