Mine-Craft: Data Mining and Analysis of Craft Beers in America

James Ingram
Joshua Shusterman
Spring 2018

Contents

Background

Craft beers are gaining in popularity in the United States. From 2006 to 2016, craft beers grew their market share nearly 9% against “Big Beer.”[1] Local brewpubs are popular as restaurants that brew their own craft beer onsite, and may traditionally only distribute in small quantities directly to customers.

Beers are categorized into different styles based on a number of factors. Some of these include the brewing and fermenting process such as being brewed hot or cold, and what ingredients are used. This determines the beer’s color, as well as alcohol content by volume (ABV) and international bittering units (IBU).

Data Source

The dataset was chosen from Kaggle, a data science project hosting site that uses a competition format to share datasets, problems, and work toward solutions. One of the datasets hosted on the site is a set of craft beer data.[2]

The data is contained in two files. The first file contains data for over 2,000 beers, and the second file contains data for over 500 breweries. The datasets include the name of the beer, style of the beer, international bittering units (IBU), alcohol content by volume (ABV), and the volume of the beer.

Our Goal

The goal of this project is to perform exploratory data analysis and to create classification models that can predict the style of a beer based on its characteristics.

Initial Approach

We began with the full dataset available on Kaggle. This dataset included 100 classes of beer styles. Since there are so many overlapping styles with similar ABV and IBU levels, only very low accuracy models could be generated from the full dataset.

Updated Approach

Because the full dataset led to models with such low accuracy, we limited our approach to consider only three popular styles of beer that were selected based on results of our descriptive statistical analysis. A new dataset containing only the “American Amber”, “American Double”, and “American IPA” styles of beer was created.

Data Cleaning and Pre-Processing

After removing the NAs from the dataframe, row numbers were reset. Beers$Column1 and beers$id were removed because they weren’t necessary for analysis. The vector beers$name was renamed to beers$beer_name to avoid confusion with

---

[1] GuruFocus.com: Harding loevner commentary: Craft beer is going flat. can craft spirits continue the insurgency? (2017). . Chatham: Newstex. Retrieved from https://search-proquest-com.libezproxy2.syr.edu/docview/1933659154?accountid=14214

[2] https://www.kaggle.com/nickhould/craft-cans/home

breweries$name in subsequent dataframe merging. In the breweries dataframe, “X” was renamed to breweries$brewery_id (to match the column in “beers”) and breweries$name was changed to breweries$brewery_name. The two dataframes were merged using a left outer join on “brewery_id”. This resulted in the information from the breweries dataframe being merged with the beers dataframe where they had corresponding “brewery_id”s. Once merged, the brewery_id column was removed and the columns were re-ordered to create the unified dataframe.

While examining the data, it was noticed that most beer volumes were in multiples of 4 oz, however a few examples had 8.4, 16.9, and 19.2 oz volumes. We decided to transform these into 8, 16, and 20 ounces to conform them with the rest of the data. At this stage, our dataset was considered cleaned and ready for exploratory data analysis.

Exploratory Data Analysis

First, we looked at the frequency of beer styles in our finished dataset. We then decided to explore the beer attributes by looking at their frequency distributions, ranges, and measures of central tendency, starting with “ABV”.

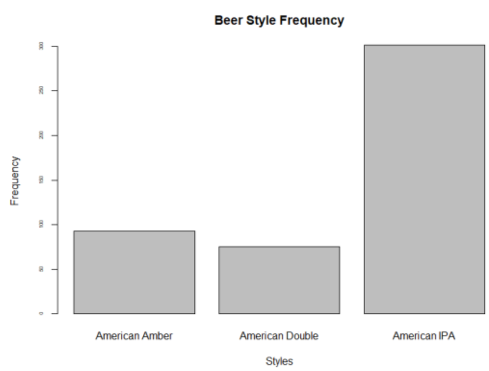

Figure 1: Frequency Distribution of Beer Styles in Data Set

“Alcohol by Volume” variable

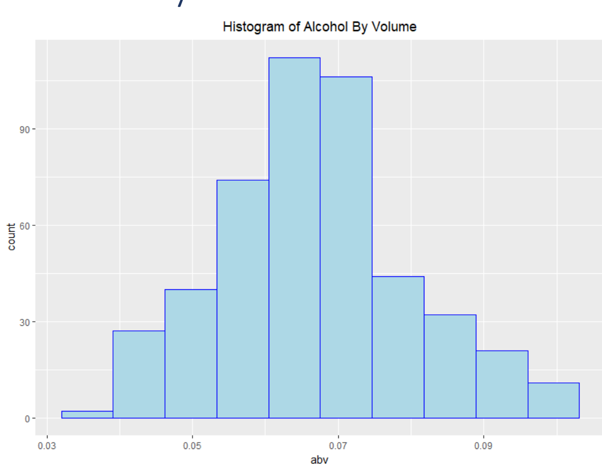

Figure 2: Frequency Distribution of Alcohol by Volume

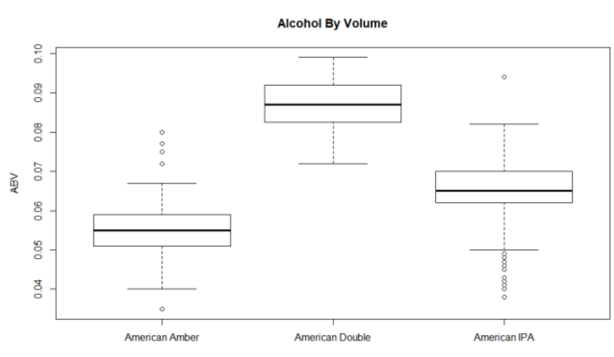

Figure 3: Median and Range of Alcohol by Volume for Each Beer Style

	beer_name	style	abv	ibu	ounces	brewery_name	city	state
24	Hopkick Dropkick	American Double	0.099	115	12	Burn 'Em Brewing	Michigan City	IN
43	Double Trunk	American Double	0.099	101	16	The Dudes' Brewing Company	Torrance	CA
61	Hi-Res	American Double	0.099	111	12	Sixpoint Craft Ales	Brooklyn	NY
65	Upslope Imperial India Pale Ale	American Double	0.099	90	20	Upslope Brewing Company	Boulder	CO
123	Elevation Triple India Pale Ale	American Double	0.099	100	12	Renegade Brewing Company	Denver	CO
133	PRO-AM (2012) (2012)	American Double	0.099	100	12	Southern Star Brewing Company	Conroe	TX
188	GUBNA Imperial IPA	American Double	0.099	100	12	Oskar Blues Brewery	Longmont	CO
359	Chaotic Double IPA	American Double	0.099	93	12	Manzanita Brewing Company	Santee	CA
419	Bad Axe Imperial IPA	American Double	0.098	76	16	Big Wood Brewery	Vadnais Heights	MN
70	Abrasive Ale	American Double	0.097	120	16	Surly Brewing Company	Brooklyn Center	MN

Table 1: Top 10 Beers Ranked by Highest Alcohol by Volume

In Figure 2 we can see that the average ABV is approximately 0.06. Table 1 indicates that all of the top 10 beers ranked by ABV are American Double style. While there is some overlap, we can see in Figure 3 that the three styles have distinct ABV levels, with American Amber having the lowest ABV and American Double having the highest.

"International Bittering Units" variable

Next, we examined International Bittering Units (ibu). The results for analysis of IBU seem to match that of ABV pretty closely. In this case, however, there is an American IPA among the top 10 beers ranked by IBU.

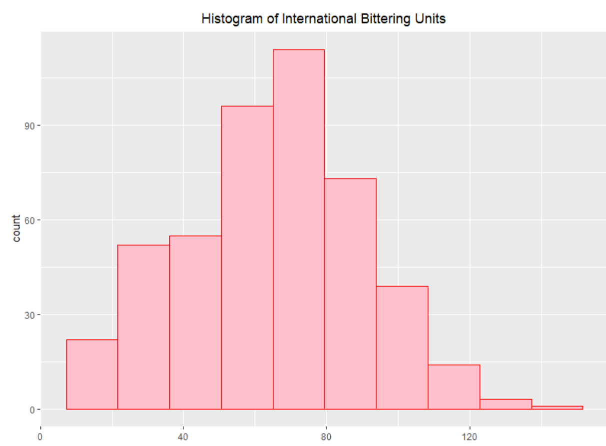

Figure 4: Frequency Distribution of International Bittering Units

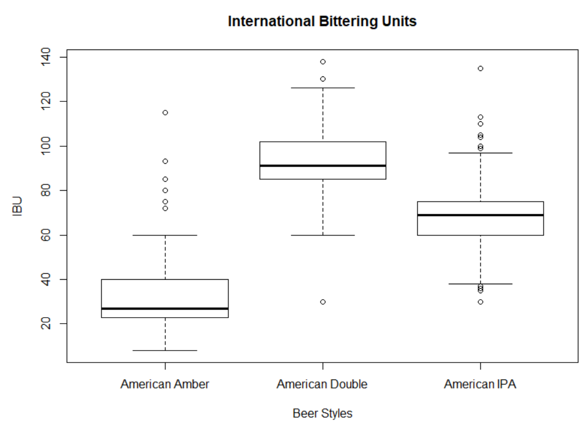

Figure 5: Median and Range of International Bittering Units for Each Beer Style

	beer_name	style	abv	ibu	ounces	brewery_name	city	state
382	Bitter Bitch Imperial IPA	American Double	0.082	138	12	Astoria Brewing Company	Astoria	OR
355	Troopers Alley IPA	American IPA	0.059	135	12	Wolf Hills Brewing Company	Abingdon	VA
262	Dead-Eye DIPA	American Double	0.090	130	16	Cape Ann Brewing Company	Gloucester	MA
113	Bay of Bengal Double IPA (2014)	American Double	0.089	126	12	Christian Moerlein Brewing Company	Cincinnati	OH
70	Abrasive Ale	American Double	0.097	120	16	Surly Brewing Company	Brooklyn Center	MN
295	Heady Topper	American Double	0.080	120	16	The Alchemist	Waterbury	VT
296	Heady Topper	American Double	0.080	120	16	The Alchemist	Waterbury	VT
243	More Cowbell	American Double	0.090	118	16	Buffalo Bayou Brewing Company	Houston	TX
21	Overlord Imperial IPA	American Double	0.085	115	16	Tin Man Brewing Company	Evansville	IN
24	Hopkick Dropkick	American Double	0.099	115	12	Burn 'Em Brewing	Michigan City	IN

Table 2: Top 10 Beers Ranked by Highest International Bittering Units

Again, while there is some overlap, we can see that the three styles have distinct IBU levels. These seem to mirror the ABV level, which implies a correlation between ABV and IBU.

“Ounces” variable

We next examined the distribution and range of the volume of alcohol (in ounces) among the beers. As expected, the 12 oz. size is the most common, with pints being the next common size. When we ranked the top ten beers by volume, there was a lot of diversity among styles, ABV levels, and IBU levels. There may not be a correlation between size of the can of beer and any other attribute. Ounces does not appear to be a useful attribute for our analysis.

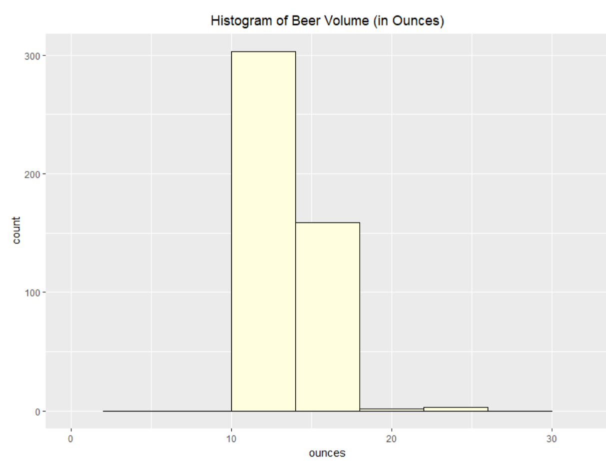

Figure 6: Frequency Distribution of Ounces

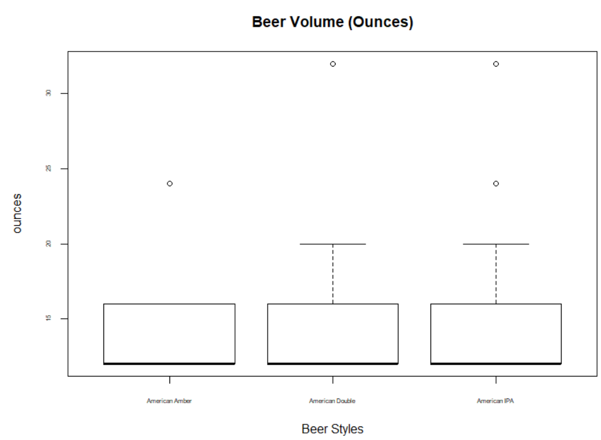

Figure 7: Median and Range for Volume (in Ounces) for Each Beer Style

	beer_name	style	abv	ibu	ounces	brewery_name	city	state
111	Mission IPA	American IPA	0.068	66	32	Mission Brewery	San Diego	CA
112	Shipwrecked Double IPA	American Double	0.092	75	32	Mission Brewery	San Diego	CA
58	Bengali	American IPA	0.065	62	24	Sixpoint Craft Ales	Brooklyn	NY
416	Longboard Island Lager	American Amber	0.046	18	24	Kona Brewing Company	Kona	HI
421	TailGate IPA	American IPA	0.050	44	24	TailGate Beer	San Diego	CA
65	Upslope Imperial India Pale Ale	American Double	0.099	90	20	Upslope Brewing Company	Boulder	CO
232	Tsunami IPA	American IPA	0.072	75	20	Mehana Brewing Co.	Hilo	HI
1	Get Together	American IPA	0.045	50	16	NorthGate Brewing	Minneapolis	MN
2	Pile of Face	American IPA	0.060	65	16	Against the Grain Brewery	Louisville	KY
3	Rico Sauvin	American Double	0.076	68	16	Against the Grain Brewery	Louisville	KY

Table 3: Top Ten Beers Ranked by Highest Volume (in Ounces)

"State" variable

We examined the relationship between the brewery state and the style of beer to see if this variable would be useful for classification purposes. First, we looked at the distribution of beers per state and then the presence or absence of the beer styles in each state.

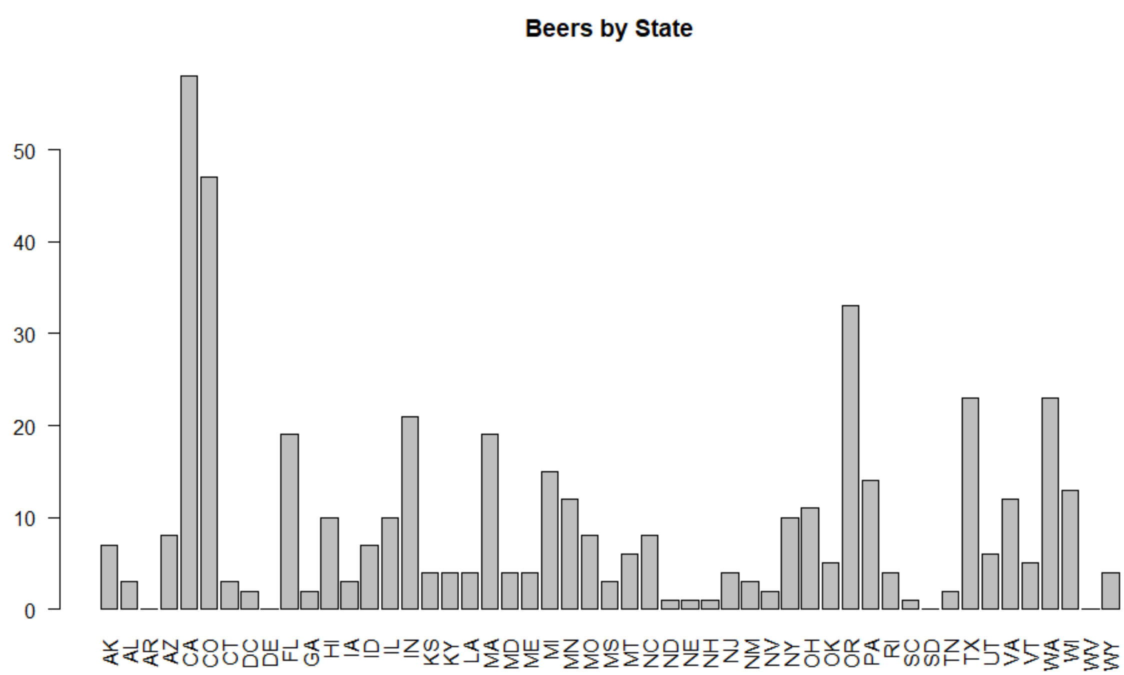

Figure 8: Count of Unique Beers Per State

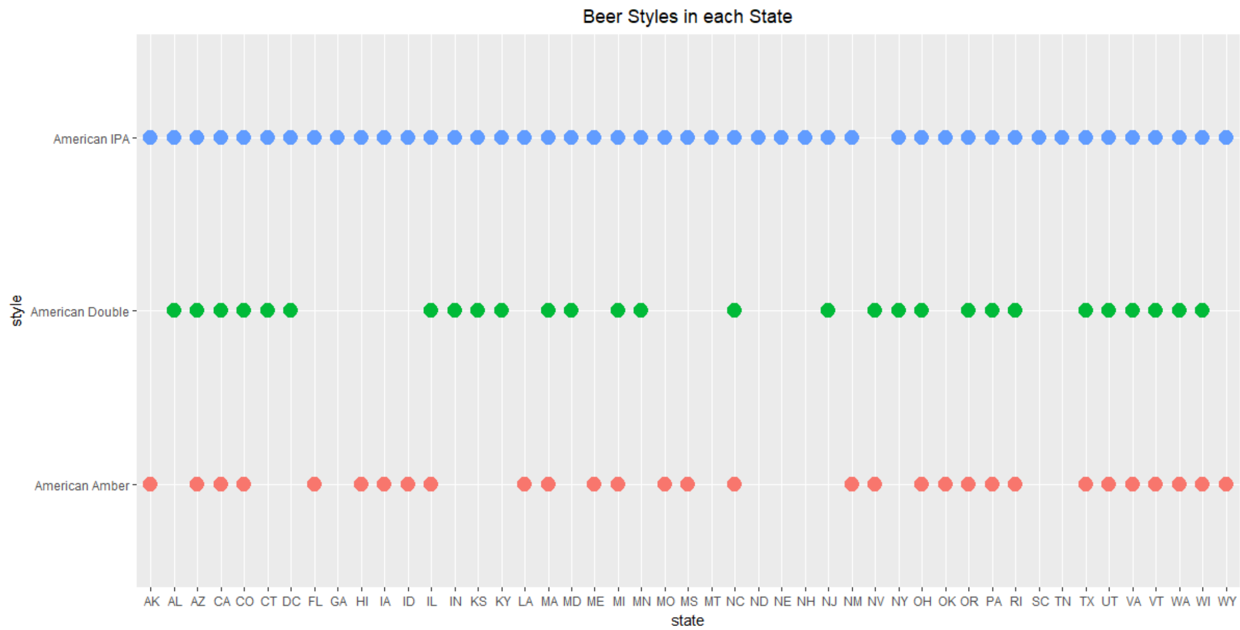

Figure 9: Presence or Absence of Each Beer Style Per State

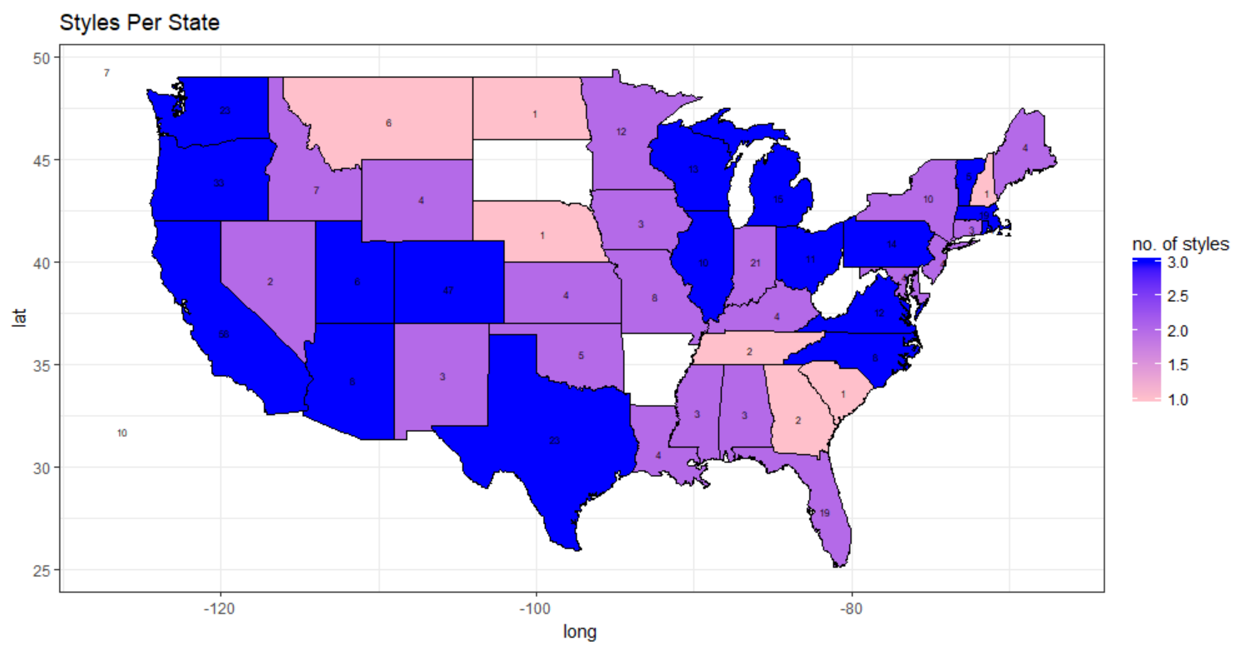

Figure 10: Beer Styles Per State (On US Map)

Very few states have only one style, and no style is unique to a particular state. This means that the state variable may not be useful in classification model building. Initial classification models were built including the “state” variable; however, they did not result in increased accuracy. Therefore, this variable was left out of our final models.

Looking at ABV and IBU together

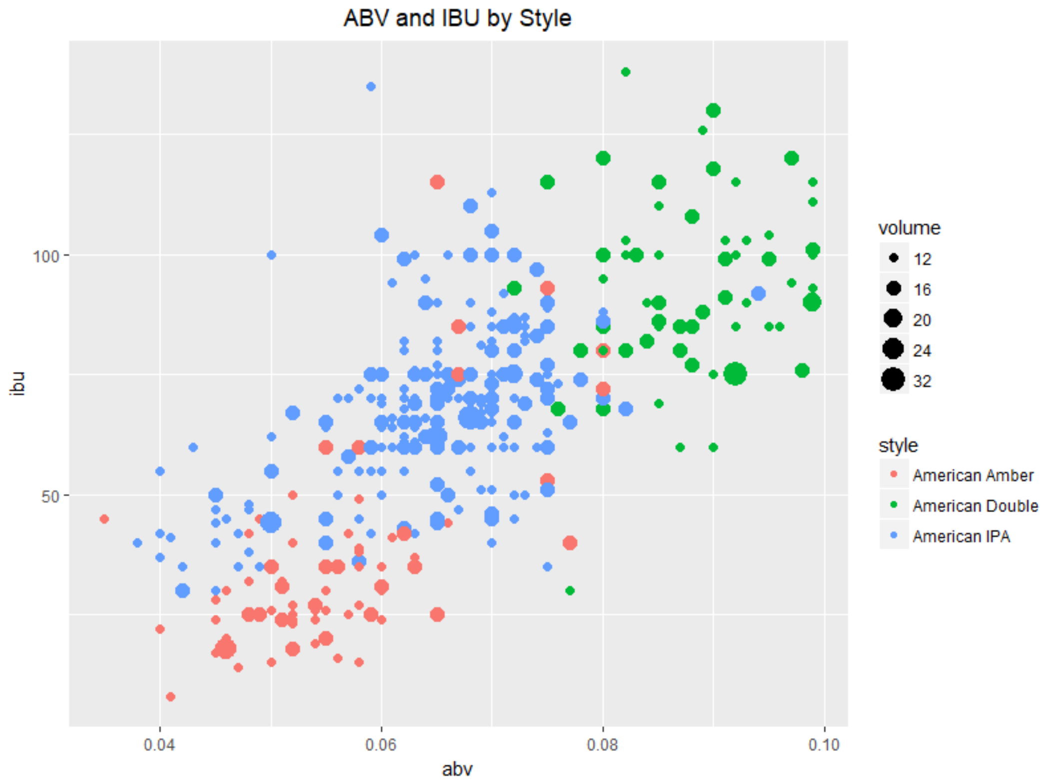

Figure 11: Scatterplot of Alcohol by Volume and International Bittering Units

Figure 11 shows that the chosen styles form three clusters based on ABV and IBU. For this reason, we chose to focus on only these two attributes to build the most parsimonious classification models.

Classification Models

For classification models, we created a trainData dataframe containing only the beer style, ABV, and IBU. We also constructed a training (new_train) and testing (new_validation) data set with a 2/3 cut point.

Decision Tree Models (RWeka J48)

The initial model was a pruned tree using the trainData set to classify the beer style. It had a size of 13, with 7 leaves. We performed 10-fold cross-validation to evaluate the model's accuracy (~88.7%). We attempted to improve the model by increasing the confidence level from 0.25 to 0.5. The same decision tree was produced with about the same accuracy. To discover the effect of pruning, an unpruned tree was created. The size of the unpruned tree was 23, with 12 leaves. This model produced an accuracy of ~87.6%. According to our calculated Information Gain results (to see the importance of each variable in the decision tree) ABV was used first and IBU second in the models.

```
abv <= 0.075
|   ibu <= 35
|   |   ibu <= 28: American Amber (48.0)
|   |   ibu > 28
|   |   |   abv <= 0.05: American IPA (9.0/3.0)
|   |   |   abv > 0.05: American Amber (15.0/1.0)
|   ibu > 35: American IPA (310.0/27.0)
abv > 0.075
|   abv <= 0.08
|   |   ibu <= 91: American IPA (18.0/9.0)
|   |   ibu > 91: American Double (5.0)
|   abv > 0.08: American Double (64.0/2.0)

Number of Leaves  :     7

Size of the tree :      13
```

Figure 12: J48 Decision Tree (default parameters)

```
          pctCorrect         pctIncorrect     pctUnclassified                  kappa
          88.6993603           11.3006397           0.0000000              0.7770953
   meanAbsoluteError  rootMeanSquaredError relativeAbsoluteError rootRelativeSquaredError
           0.1091163            0.2529076          31.2206891             60.5559701
```

Figure 13: 10-fold CV Evaluation of J48 Decision Tree (default parameters)

```
Information Gain:

'abv' 'ibu'
 0.7400649 0.5716396
```

Figure 14: Information Gain

Naïve Bayes Models

RWeka NB

Using the RWeka NB package, we created the following classification model:

```
Naive Bayes Classifier

                          Class
Attribute       American Amber American Double    American IPA
                         (0.2)          (0.16)          (0.64)
===============================================================
abv
  mean                  0.0555          0.0877          0.0648
  std. dev.             0.0085          0.0067          0.0085
  weight sum                93              75             301
  precision             0.0011          0.0011          0.0011

ibu
  mean                 33.9567         93.2494         67.5307
  std. dev.            18.8665         17.3454         15.9636
  weight sum                93              75             301
  precision             1.4607          1.4607          1.4607
```

Figure 15: RWeka Naïve Bayes Model

10-fold CV evaluation demonstrated that the model's accuracy was about 87.6%.

```
         pctCorrect              pctIncorrect          pctUnclassified                    kappa
         87.6332623                12.3667377                0.0000000                0.7677266
  meanAbsoluteError      rootMeanSquaredError    relativeAbsoluteError rootRelativeSquaredError
          0.1137663                 0.2575620               32.5511799               61.6704085
```

Figure 16: 10-fold CV Evaluation of RWeka Naïve Bayes Model

Naïve Bayes using e1071

Next, we built a Naïve Bayes model using R's e1071 package.

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
 American Amber American Double    American IPA
      0.1982942       0.1599147       0.6417910

Conditional probabilities:
                 abv
Y                       [,1]        [,2]
  American Amber  0.05562366 0.008449227
  American Double 0.08769333 0.006655567
  American IPA    0.06480731 0.008494473

                 ibu
Y                     [,1]     [,2]
  American Amber  34.05376 18.90041
  American Double 93.32000 17.53079
  American IPA    67.63455 16.05260
```

Figure 17: e1071 Naïve Bayes Model

Using the model, we predicted the beer styles of the data from the trainData set. The model seems fairly accurate.

NB_pred	American Amber	American Double	American IPA
American Amber	74	0	33
American Double	0	69	3
American IPA	19	6	265

Figure 18: Confusion Matrix from NB Model "style" Prediction of "trainData" Data Set

Similar results were obtained when the model was applied to the "new_train" and "new_validation" data sets.

NB_pred2	American Amber	American Double	American IPA
American Amber	28	0	8
American Double	0	18	1
American IPA	6	3	93

Figure 19: Confusion Matrix from NB Model "style" Prediction of "new_validation" Data Set

Support Vector Machine Models

Linear SVM Model

A linear SVM model was created using “new_train” data with a training accuracy of over 91%.

```
Support Vector Machines with Linear Kernel

312 samples
  2 predictor
  3 classes: 'American Amber', 'American Double', 'American IPA'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 312, 312, 312, 312, 312, 312, ...
Resampling results:

  Accuracy  Kappa
  0.914551  0.8327833

Tuning parameter 'C' was held constant at a value of 1
```

Figure 20: Linear SVM Model

Applying this model to predict the “new_validation” data, we obtained results with ~91% accuracy, as can be seen in the confusion matrix and summary statistics below:

```
Confusion Matrix and Statistics

                 Reference
Prediction        American Amber American Double American IPA
  American Amber              26               1            2
  American Double              0              18            1
  American IPA                 8               2           99

Overall Statistics

               Accuracy : 0.9108
                 95% CI : (0.8549, 0.9504)
    No Information Rate : 0.6497
    P-Value [Acc > NIR] : 2.927e-14

                  Kappa : 0.819
 Mcnemar's Test P-Value : 0.1767

Statistics by Class:

                     Class: American Amber Class: American Double Class: American IPA
Sensitivity                         0.7647                 0.8571              0.9706
Specificity                         0.9756                 0.9926              0.8182
Pos Pred Value                      0.8966                 0.9474              0.9083
Neg Pred Value                      0.9375                 0.9783              0.9375
Prevalence                          0.2166                 0.1338              0.6497
Detection Rate                      0.1656                 0.1146              0.6306
Detection Prevalence                0.1847                 0.1210              0.6943
Balanced Accuracy                   0.8702                 0.9249              0.8944
```

Figure 21: Linear SVM Model Prediction Results

Radial SVM Model

A non-linear (RBF) Support Vector Machine model was also created using the “new_train” data set with varying levels of Confidence (0.25, 0.5, and 1.0). Accuracy levels of 90.9, 90.9, and 91.2% were obtained, respectively.

```
Support Vector Machines with Radial Basis Function Kernel

312 samples
  2 predictor
  3 classes: 'American Amber', 'American Double', 'American IPA'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 312, 312, 312, 312, 312, 312, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.9087421  0.8203480
  0.50  0.9086468  0.8212055
  1.00  0.9123640  0.8289506

Tuning parameter 'sigma' was held constant at a value of 2.861614
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 2.861614 and C = 1.
```

Figure 22: Radial SVM Model

Again, the model was applied to predict the beer styles from the “new_validation” data set. The classification model was approximately 90.4% accurate as can be seen below:

```
Confusion Matrix and Statistics

                 Reference
Prediction        American Amber American Double American IPA
  American Amber              24               0            0
  American Double              1              18            2
  American IPA                 9               3          100

Overall Statistics

               Accuracy : 0.9045
                 95% CI : (0.8473, 0.9455)
    No Information Rate : 0.6497
    P-Value [Acc > NIR] : 1.53e-13

                  Kappa : 0.8032
 Mcnemar's Test P-Value : 0.01694

Statistics by Class:

                     Class: American Amber Class: American Double Class: American IPA
Sensitivity                         0.7059                 0.8571              0.9804
Specificity                         1.0000                 0.9779              0.7818
Pos Pred Value                      1.0000                 0.8571              0.8929
Neg Pred Value                      0.9248                 0.9779              0.9556
Prevalence                          0.2166                 0.1338              0.6497
Detection Rate                      0.1529                 0.1146              0.6369
Detection Prevalence                0.1529                 0.1338              0.7134
Balanced Accuracy                   0.8529                 0.9175              0.8811
```

Figure 23: Radial SVM Model Prediction Results

Model Comparison

All three types of models produced similar results (approx. 88-91% accuracy), with SVM being the most accurate

Conclusion

Once we chose to focus on only 3 styles of beer, and only 2 characteristics, we were able to create classification models that had much higher levels of accuracy than we had when we used the full dataset. When there are larger numbers of classes in the dataset, there is more overlap among the classes, which leads to lower accuracy in the classification models. The three classes that we chose had enough distinction to allow for high levels of accuracy.

Appendix: R Code

```
#####################################
# Install Packages and Load Libraries
#####################################

packages <- c("tidyverse", "RWeka","rJava","caret","e1071","kernlab", "randomForest", "class",
"cluster", "ggplot2", "ggmap", "maps", "mapdata", "maptools", "rstudioapi")
package.check <- lapply(packages, FUN = function(x){
  if(!require(x, character.only=TRUE)){
    install.packages(x, dependencies=TRUE)
    library(x, character.only=TRUE)
  }
})

##############################
# Data Load and Pre-Processing
##############################

setwd(dirname(getActiveDocumentContext()$path))
beers <- read.csv("beers_new.csv")
breweries <- read.csv("breweries.csv")

# examine datasets
str(beers)
sum(is.na(beers)) # find total NAs in beers data set
str(breweries)
sum(is.na(breweries)) # find total NAs in breweries data set

# omit NAs
beers <- na.omit(beers)
rownames(beers) <- NULL # resets row counts after NAs are removed

# remove beers$Column1 and beers$id
beers <- beers[, -1,]
beers <- beers[, -3]
head(beers) # examine structure of beers dataframe

# rename beers$name to beers$beer_name
cnames <- colnames(beers)
cnames[3] <- "beer_name"
colnames(beers) <- cnames
colnames(beers)

# rename breweries$X to breweries$brewery_id and breweries$name to breweries$brewery_name
cnames <- colnames(breweries)
cnames[1] <- "brewery_id"
cnames[2] <- "brewery_name"
colnames(breweries) <- cnames
colnames(breweries)

# merge beers and breweries data sets on brewery_id
df <- merge(x = beers, y = breweries, by = "brewery_id")

# re-order column headings and remove brewery_id from dataframe
df <- df[,c(4,5,2,3,6,7,8,9)]
View(head(df,25))

# standardize ounces
df$ounces[df$ounces==8.4]<-8
```

```
df$ounces[df$ounces==16.9]<-16
df$ounces[df$ounces==19.2]<-20

# create new csv file containing cleaned data set
write.csv(df, file="clean_beer_data.csv")

###########################
# Exploratory Data Analysis
###########################

# frequency of each beer style
barplot(table(df$style), main="Beer Style Frequency", xlab="Styles", ylab="Frequency", cex.axis=0.5)

# ABV
# display frequency distribution of abv
g.abv <- ggplot(df, aes(x=abv))
g.abv <- g.abv + geom_histogram(bins=10, color = "blue", fill="light blue")
g.abv <- g.abv + ggtitle("Histogram of Alcohol By Volume")
g.abv <- g.abv + theme(plot.title = element_text(hjust = 0.5))
g.abv
# sort beers with highest alcohol by volume (display top 10)
sortedAbv <- df[order(-df$abv), ]
View(head(sortedAbv,10))
# Look at ABV by Beer Style
plot(x=df$style, y=df$abv, xlab="Beer Styles", ylab="ABV", cex.axis=1, main="Alcohol By Volumne")

# IBU
# display frequency distribution of ibu
g.ibu <- ggplot(df, aes(x=ibu))
g.ibu <- g.ibu + geom_histogram(bins=10, color = "red", fill="pink")
g.ibu <- g.ibu + ggtitle("Histogram of International Bittering Units")
g.ibu <- g.ibu + theme(plot.title = element_text(hjust = 0.5))
g.ibu
# sort beers with highest international bitter units (display top 10)
sortedIbu <- df[order(-df$ibu), ]
View(head(sortedIbu, 10))
# Look at IBU by Beer Style
plot(x=df$style, y=df$ibu, xlab="Beer Styles", ylab="IBU", cex.axis=1, main="International Bittering
Units")

# display frequency distribution of ounces
g.ounces <- ggplot(df, aes(x=ounces))
g.ounces <- g.ounces + geom_histogram(binwidth = 4, color = "black", fill="light yellow")
g.ounces <- g.ounces + ggtitle("Histogram of Beer Volume (in Ounces)") + xlim(0,32)
g.ounces <- g.ounces + theme(plot.title = element_text(hjust = 0.5))
g.ounces
# sort beers with largest volume in ounces (display top 10)
sortedOunces <- df[order(-df$ounces), ]
View(head(sortedOunces, 10))
# Look at ounces by Beer Style
plot(x=df$style, y=df$ounces, xlab="Beer Styles", ylab="ounces", cex.axis=0.5, main="Beer Volume
(Ounces)")

# relationship between abv and ibu
volume <- as.factor(df$ounces)
qplot(data=df, x=abv, y=ibu, color=style, size=volume, main="ABV and IBU by Style") +
theme(plot.title = element_text(hjust = 0.5))

# Frequency of beers by state
stateBeers <- barplot(table(df$state), main="Beers by State", las=2)
```

```r
# relationship between style and state
qplot(data=df, x=state, y=style, main="Beer Styles in each State", color=df$style, size=3) +
theme(legend.position="none") + theme(plot.title = element_text(hjust = 0.5))

# map visualization of styles per state
# function from: https://favorableoutcomes.wordpress.com/2012/10/19/create-an-r-function-to-convert-
state-codes-to-full-state-name/
stateFromLower <-function(x) {
st.codes<-data.frame(
  state=as.factor(c("AK", "AL", "AR", "AZ", "CA", "CO", "CT", "DC", "DE", "FL", "GA",
                    "HI", "IA", "ID", "IL", "IN", "KS", "KY", "LA", "MA", "MD", "ME",
                    "MI", "MN", "MO", "MS",  "MT", "NC", "ND", "NE", "NH", "NJ", "NM",
                    "NV", "NY", "OH", "OK", "OR", "PA", "PR", "RI", "SC", "SD", "TN",
                    "TX", "UT", "VA", "VT", "WA", "WI", "WV", "WY")),
    full=as.factor(c("alaska","alabama","arkansas","arizona","california","colorado",
                     "connecticut","district of columbia","delaware","florida","georgia",
                     "hawaii","iowa","idaho","illinois","indiana","kansas","kentucky",
                     "louisiana","massachusetts","maryland","maine","michigan","minnesota",
                     "missouri","mississippi","montana","north carolina","north dakota",
                     "nebraska","new hampshire","new jersey","new mexico","nevada",
                     "new york","ohio","oklahoma","oregon","pennsylvania","puerto rico",
                     "rhode island","south carolina","south dakota","tennessee","texas",
                     "utah","virginia","vermont","washington","wisconsin",
                     "west virginia","wyoming"))
  )
  st.x<-data.frame(state=x)
  refac.x<-st.codes$full[match(st.x$state,st.codes$state)]
  return(refac.x)
}
states<-map_data("state")
df$state<-gsub("[[:space:]]", "", df$state)
df$region<-stateFromLower(df$state)

# text data for maps
counts<-as.data.frame(table(df$state)) # no. of observations per state
colnames(counts)<-c("state.abb","count")
txt <- data.frame(state.center, state.abb)
d1<-txt
d2<-counts
lab<-merge(d1,d2, by = "state.abb", all=FALSE)
rm(counts,txt,d1,d2)

plot.data <- inner_join(states, agg, by = "region")

df$styles<-as.character(df$style)
df.new<-within(df,{no.styles<-ave(styles,region,FUN=function(x) length(unique(x)))})
agg<-subset(df.new,select=c("region","no.styles"))
agg<-unique(agg)
agg$no.styles<-as.numeric(paste(agg$no.styles))
plot.data <- inner_join(states, agg, by = "region")

ggplot(data = plot.data, mapping = aes(x = long, y = lat, group = group)) +
  coord_fixed(1.3) +   geom_polygon(data = plot.data, aes(fill = no.styles), color = "white") +
  geom_polygon(color = "black", fill = NA) +theme_bw() +labs( title="Styles Per State") +
  scale_fill_gradientn("no. of styles",colors=c("pink","blue" )) +
  theme(axis.text = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        panel.border = element_blank(),
```

```
        panel.grid = element_blank(),
        axis.title = element_blank())+
  geom_text(data = lab, aes(x = x, y = y, label = count, group = NULL), size = 2)+theme_bw()

#######################
# Classification Models
#######################

# create data set with only style, abv, and ibu
trainData <- df[,-1] #remove beer name
trainData <- trainData[,-4:-7] #remove ounces, brewery name, city, and state

# hold-out method to measure performance
randIndex <- sample(1:dim(trainData)[1])
cutpoint2_3 <- floor(2*dim(trainData)[1]/3)
new_train <- trainData[randIndex[1:cutpoint2_3], ]
new_validation <- trainData[randIndex[(cutpoint2_3+1):dim(trainData)[1]],]
train_labels <- new_train$style

#########################################
# Decision Tree Modeling (J48 from RWeka)
#########################################

# J48 decision tree model with default values
model_dt <- J48(style~.,data=trainData)
model_dt

# use 10-fold cross-validation to evaluate model_dt
e <- evaluate_Weka_classifier(model_dt, numFolds=10, seed=1, class=TRUE)
e$details

# improve model with increased confidence
model_dt2 <- J48(style~.,data=trainData, control=Weka_control(U=FALSE, M=2, C=0.5))
model_dt2

# use 10-fold cross-validation to evaulate model_dt2
e2 <- evaluate_Weka_classifier(model_dt2, numFolds=10, seed=1, class=TRUE)
e2$details

# examine the effect of pruning by creating unpruned tree
model_dt3 <- J48(style~.,data=trainData, control=Weka_control(U=TRUE, M=2))
model_dt3

# use 10-fold cross-validation to evaulate model_dt3
e3 <- evaluate_Weka_classifier(model_dt3, numFolds=10, seed=1, class=TRUE)
e3$details

# evaluate information gain
ig <- InfoGainAttributeEval(style~.,data=trainData)
cat("Information Gain:\n\n'abv' 'ibu' \n", ig)

##################################
# Naive Bayes Classification Model
##################################

# RWeka NB Model with defualt values
NB <- make_Weka_classifier("weka/classifiers/bayes/NaiveBayes")
model_nb <- NB(style~.,data=trainData)
model_nb
```

```
# use 10-fold cross-validation to evaulate the model_nb
e_nb <- evaluate_Weka_classifier(model_nb, numFolds=10, seed=1, class=TRUE)
NB_results <- e_nb$details
NB_results

# Naive Bayes using e1071
NB_model <- naiveBayes(style~.,data=trainData)
NB_model
NB_pred <- predict(NB_model, trainData)
table(NB_pred, trainData$style)

NB_model2 <- naiveBayes(style~.,data=new_train)
NB_model2
NB_pred2 <- predict(NB_model2, new_validation)
table(NB_pred2, new_validation$style)

###############################
# Support Vector Machines Model
###############################

# SVM linear model using caret package
set.seed(1818)
train_control <- trainControl(method="cv", number=10)
fit_svm_linear <- train(style~.,data=new_train,
                        method="svmLinear",
                        traControl = train_control,
                        na.action=na.pass)
# cross-validation performance
fit_svm_linear

pred_svm_linear <- predict(fit_svm_linear, newdata=new_validation)
table(pred_svm_linear)

confusionMatrix(pred_svm_linear, new_validation$style)

# SVM RBF model using caret package
fit_svm_rbf <- train(style~.,data=new_train,
                     method="svmRadial",
                     traControl = train_control,
                     na.action=na.pass)
# cross-validation performance
fit_svm_rbf

pred_svm_rbf <- predict(fit_svm_rbf, newdata=new_validation)
table(pred_svm_rbf)

confusionMatrix(pred_svm_rbf, new_validation$style)
```