Supervised Learning of Forensic Glass Data

Advaith Cheruvu

10/26/2021

Importance of Study

Forensic data is useful in aiding a criminal case. New technological advances have lead to the development of many new sources of forensic data. For example, the glass found at crime scenes can be analyzed to determine the source of the glass. If the glass was broken from a window, then the investigator knows that the suspect had to break into the building and the point of entry can be found. Any predictive algorithm that can predict the source of the glass can save valuable time for the investigators and lead to potential breakthroughs in the case. My goal in this document is to develop a supervised machine learning KNN algorithm that will predict the source of the glass. The dataset I will be examining comes from the UCI machine learning repository (http://archive.ics.uci.edu/ml/machine-learning-databases/glass/glass.data (http://archive.ics.uci.edu/ml/machine-learning-databases/glass/glass.data)).

Glass Data

The glass dataset that I will be using consists of the refractive index, various measures of chemical elements present in the glass, and the source of the glass. The refractive index describes the ratio of the speed of light through a material (in this case glass) compared to the speed of light through a vacuum. This value is highly dependent on the density of the glass. The dataset also has contains values for the amount of sodium (Na), magnesium (Mg), aluminum (Al), silicon (Si), potassium (K), calcium (Ca), barium (Ba), and iron (Fe) present in the glass. These values are measured in weight percent based on corresponding oxide. Finally, the last variable is the source of the glass. The numerical values 1, 2, 3, 5, 6, and 7 correspond to float building windows, non-float building windows, float vehicle windows, container glass, tableware glass, and headlamps, respectively. The “4” corresponds to non-float vehicle windows which does not appear in this dataset. Float glass is made from molten ribbons that are poured onto liquid tin and slowly cooled to form a long ribbon of solid glass. Non-float glass is glass that is not made using this method, mainly including sheet glass and some forms of plate glass. Sheet glass is made from a series of rollers that brings molten glass up vertically and then turned horizontal as the glass cools. Sheet glass has been historically used in situations where thinner glass was more applicable, including residential windows, desktops, etc. The modern glass market is dominated by float glass which is usually cheaper, easier to transport, and allows for larger sizes of thin glass. Therefore, most non-float glass is usually older than float glass.

Setting up Workspace and Installing Packages

First I clean up and set my working space, including setting my working directory and installing packages I will mainly be using ggvis, class, and caret.

```
#
# clean up and set up
rm(list=ls())
setwd("/Users/advai/Documents/DSFS")
#
# libraries
library(class)
library(ggvis)
library(gmodels)
library(tidyverse)
library(caret)
library(GGally)
library(gridExtra)
```

Loading and Looking at the Glass Data

There is no missing data in this dataset. There are 214 observations in this dataset.

```
#
# read in glass data
glass <- read.csv(url("http://archive.ics.uci.edu/ml/machine-learning-databases/glass/glass.dat
a"),  na.strings = "?", header = FALSE)
#
# look at data
head(glass)
```

```
##   V1      V2    V3   V4   V5    V6   V7   V8 V9  V10 V11
## 1  1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00   1
## 2  2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00   1
## 3  3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00   1
## 4  4 1.51766 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00   1
## 5  5 1.51742 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00   1
## 6  6 1.51596 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26   1
```

```
names(glass)
```

```
##  [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11"
```

```
summary(glass)
```

```
##        V1               V2              V3              V4
##  Min.   :  1.00   Min.   :1.511   Min.   :10.73   Min.   :0.000
##  1st Qu.: 54.25   1st Qu.:1.517   1st Qu.:12.91   1st Qu.:2.115
##  Median :107.50   Median :1.518   Median :13.30   Median :3.480
##  Mean   :107.50   Mean   :1.518   Mean   :13.41   Mean   :2.685
##  3rd Qu.:160.75   3rd Qu.:1.519   3rd Qu.:13.82   3rd Qu.:3.600
##  Max.   :214.00   Max.   :1.534   Max.   :17.38   Max.   :4.490
##        V5              V6              V7              V8
##  Min.   :0.290   Min.   :69.81   Min.   :0.0000   Min.   : 5.430
##  1st Qu.:1.190   1st Qu.:72.28   1st Qu.:0.1225   1st Qu.: 8.240
##  Median :1.360   Median :72.79   Median :0.5550   Median : 8.600
##  Mean   :1.445   Mean   :72.65   Mean   :0.4971   Mean   : 8.957
##  3rd Qu.:1.630   3rd Qu.:73.09   3rd Qu.:0.6100   3rd Qu.: 9.172
##  Max.   :3.500   Max.   :75.41   Max.   :6.2100   Max.   :16.190
##        V9             V10              V11
##  Min.   :0.000   Min.   :0.00000   Min.   :1.00
##  1st Qu.:0.000   1st Qu.:0.00000   1st Qu.:1.00
##  Median :0.000   Median :0.00000   Median :2.00
##  Mean   :0.175   Mean   :0.05701   Mean   :2.78
##  3rd Qu.:0.000   3rd Qu.:0.10000   3rd Qu.:3.00
##  Max.   :3.150   Max.   :0.51000   Max.   :7.00
```

```
dim(glass)
```

```
## [1] 214  11
```

Renaming Columns

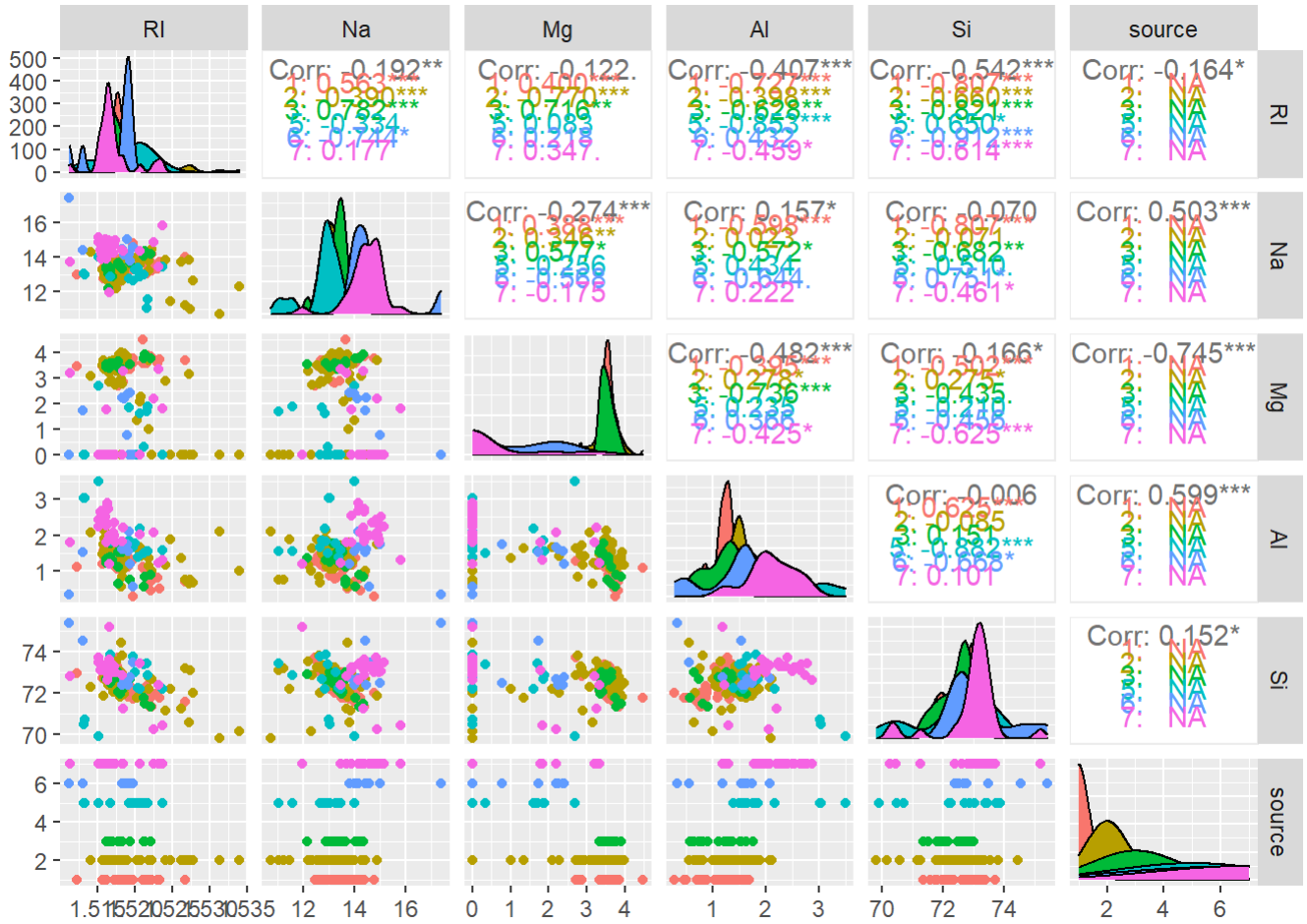
This dataset didn't come downloaded with the names so I added them.

```
#
# column names
names(glass) <- c("ID", "RI", "Na", "Mg", "Al", "Si", "K", "Ca", "Ba", "Fe", "source")
#
# check the glass data names
names(glass)
```

```
##  [1] "ID"     "RI"     "Na"     "Mg"     "Al"     "Si"     "K"      "Ca"
##  [9] "Ba"     "Fe"     "source"
```

Correlation and Pairwise

I decided to look for correlation between any of the variables. To do so, I created a pairwise graph showing the correlation between half of the variables.


RI
Na
Mg
Al
Si
source
Corr: -0.192**
Corr: -0.122.
Corr: -0.407***
Corr: -0.542***
Corr: -0.164*
Corr: -0.274***
Corr: 0.157*
Corr: -0.070
Corr: 0.503***
Corr: -0.482***
Corr: -0.166*
Corr: -0.745***
Corr: -0.006
Corr: 0.599***
Corr: 0.152*


K
Ca
Ba
Fe
source
Corr: -0.318***
Corr: -0.043
Corr: -0.008
Corr: -0.010
Corr: -0.113.
Corr: 0.125.
Corr: 0.001
Corr: -0.059
Corr: 0.575***
Corr: -0.188**

I looked for the correlation constants that had a large magnitude. This included aluminum and source, barium and source, aluminum and refractive index, and silicon and refractive index.

```
#
# Correlation (Barium and Aluminum have high correlation with source)
cor(as.numeric(glass$Al), as.numeric(glass$source))
```

```
## [1] 0.5988292
```

```
cor(as.numeric(glass$Ba), as.numeric(glass$source))
```

```
## [1] 0.5751615
```

```
# Aluminum and Silicon have high negative correlation with Refractive Index
cor(as.numeric(glass$Al), as.numeric(glass$RI))
```

```
## [1] -0.407326
```

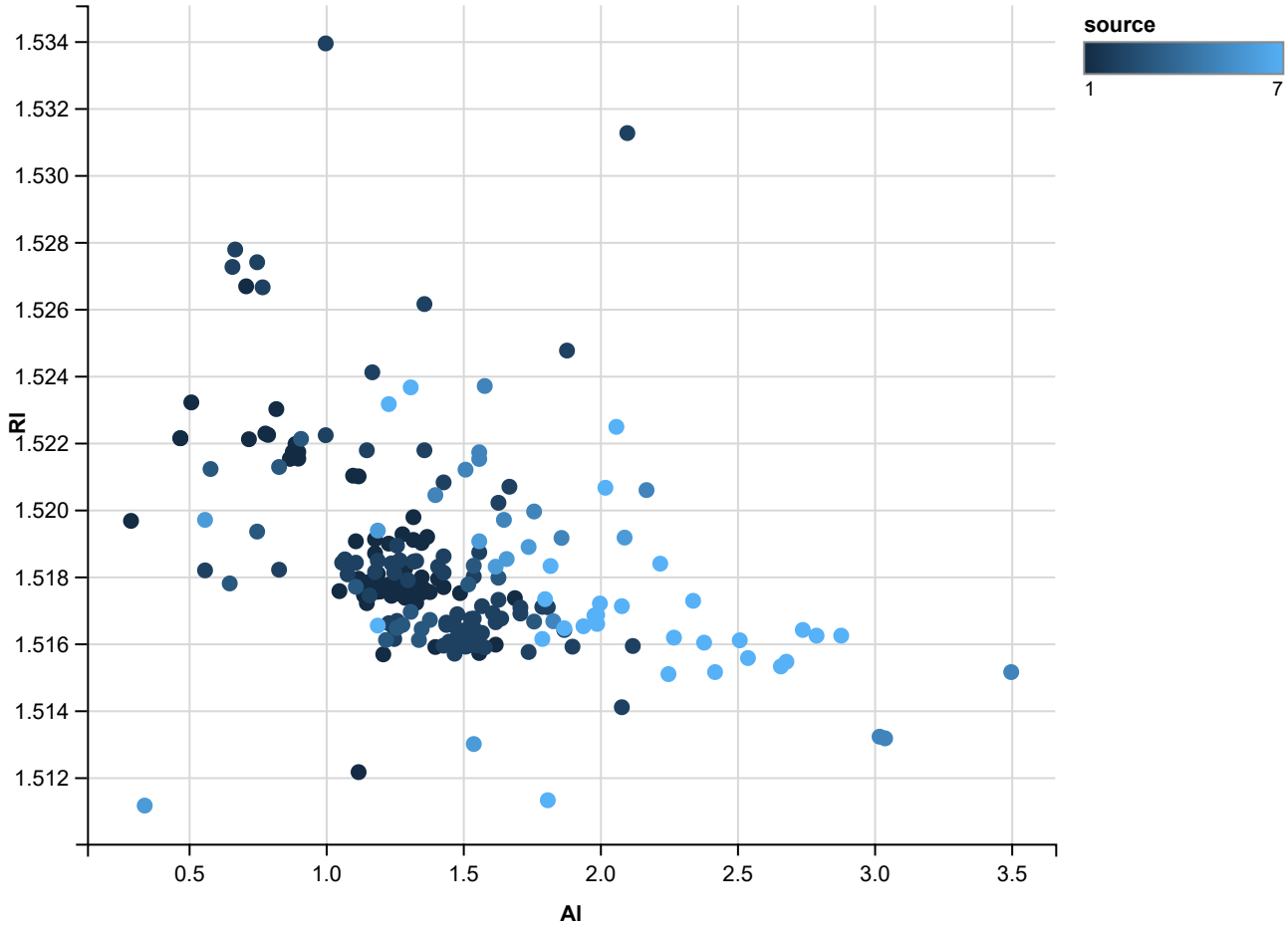
```
cor(as.numeric(glass$Si), as.numeric(glass$RI))
```

```
## [1] -0.5420522
```

Scatter Plots

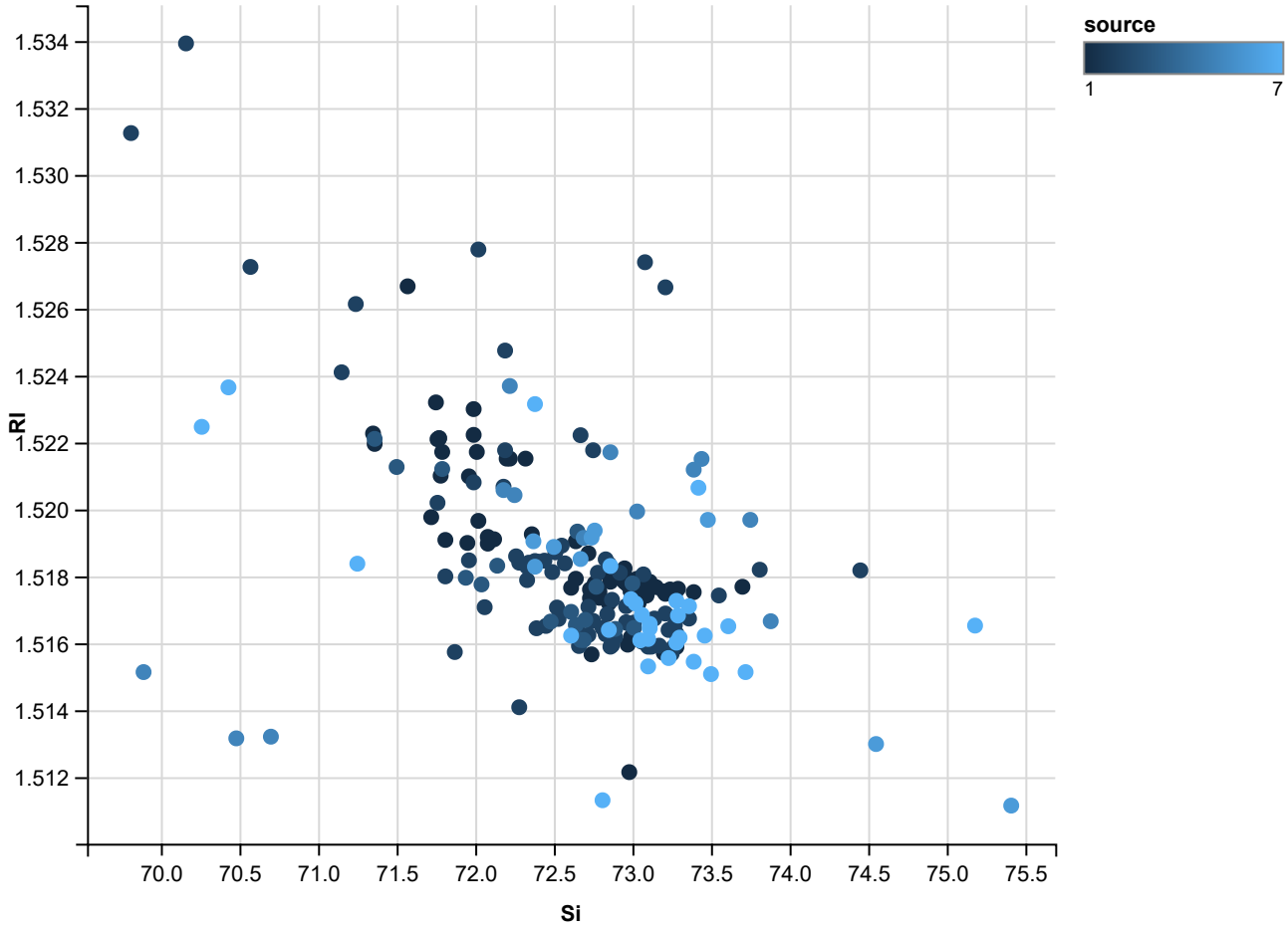
I decided to create a few scatter plots to highlight the relationship between a few variables. The scatter plot below plots aluminum vs the refractive index with the data points colored based on the source. The plot indicates some relationship between aluminum and the source of the glass. This is because the color changes from darker blue to brighter blue moving as the amount of aluminum increases. This could suggest that larger pieces of glass used for buildings and cars do not contain as much aluminum as smaller glass items. The refractive index does not seem to be related to the source of the glass.

```
#
# scatter plots
# Al vs RI
glass %>% ggvis(x= ~Al, y=~RI, fill = ~source) %>% layer_points()
```



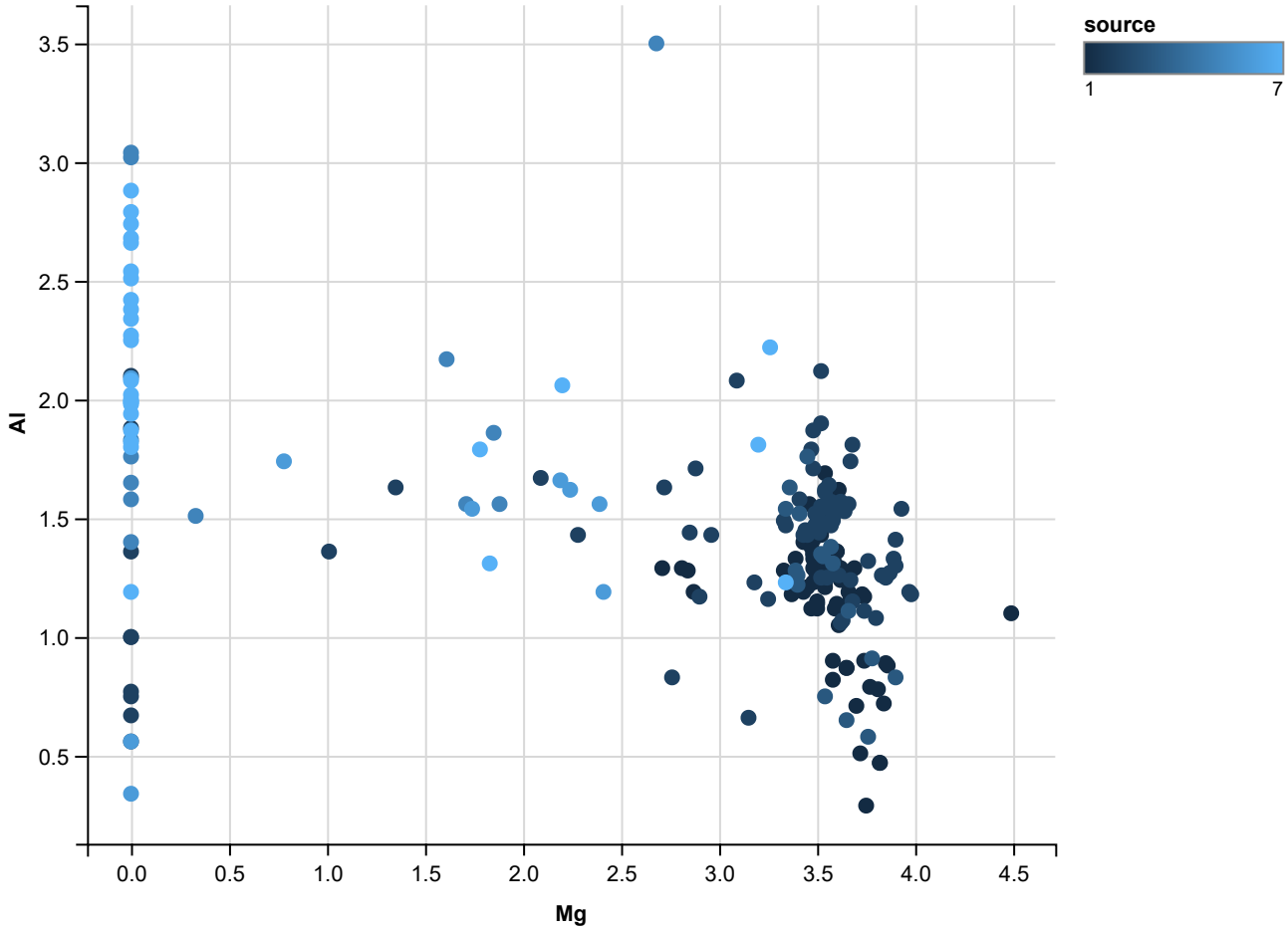
This scatter plot replaces aluminum with silicon which seems to be more related to the refractive index compared to aluminum. Silicon also seems to be somewhat related to the source since most of the brighter (more commercially used glass products) data points are found with high amounts of silicon and a low refractive index. This could indicate that commercially used glass products tends to have more silicon and is less refractive.

```
# Si vs RI
glass %>% ggvis(x= ~Si, y=~RI, fill = ~source) %>% layer_points()
```



This scatter plot shows the relationship between magnesium and aluminum. This plot differs from the previous plots because it does not contain the refractive index. Instead, I wanted to look at how two element amounts can differ within various sources of glass. The data points seem to have more variety based on the amount of aluminum present, but not as much variety based on the amount of magnesium present. Most of the data points tend to have either 0% or 3.5% magnesium, whereas the aluminum amounts vary significantly. Also, most of the building and car window glass have 3.5% magnesium while glass used for small products have 0% magnesium. This could indicate that larger glass uses more magnesium while smaller glass uses less magnesium.

```
# Mg vs Al
glass %>% ggvis(x= ~Mg, y=~Al, fill = ~source) %>% layer_points()
```



KNN Supervised Learning

Now that some possible relationships and correlations have been discovered, I can begin constructing a supervised learning algorithm. This machine learning algorithm will be using K-Nearest Neighbor (KNN).

Setting the Test and Training Sets

Since I will be using the same dataset for testing and training the data, I split the data into 70% training and 30% testing.

```
#
# random seed
set.seed(34827)
index <- sample(2, nrow(glass), replace=TRUE, prob=c(0.7, 0.3))
index
```

```
##   [1] 1 1 1 1 1 2 2 1 1 1 1 2 1 1 1 2 1 1 1 1 1 2 1 2 1 2 2 2 2 1 1 1 1 1 2 1 1
##  [38] 1 2 1 1 2 2 1 2 2 1 1 1 2 1 1 2 2 1 1 1 2 1 2 1 2 1 1 1 1 1 2 1 1 1 2 2
##  [75] 1 2 1 2 1 1 1 2 2 1 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 2 1 1
## [112] 1 2 1 2 2 1 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 2 2 1 1 1 2 2 1 1 1 1 1
## [149] 1 2 1 1 2 1 1 2 2 2 1 2 1 1 2 2 1 1 1 1 1 2 1 2 2 1 1 2 1 1 2 1 2 2 1 1
## [186] 1 1 2 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 1 1 1 1 1 2 1 2
```

```
#
# training set
glass.training <- glass[index==1, 1:11]
#
# test set
glass.test <- glass[index==2, 1:11]
#
# Test and training sets take up 30% and 70% of the original data, respectively.
dim(glass.test)
```

```
## [1] 68 11
```

```
dim(glass.training)
```

```
## [1] 146  11
```

```
dim(glass)
```

```
## [1] 214  11
```

Test and Training Labels

The test and training labels are the sources of glass since this is what we want the model to predict.

```
#
# create glass training labels
glass.trainLabels <- glass[index==1,11]
#
# glass test labels
glass.testLabels <- glass[index==2, 11]
```

Building the Model

Here I am assigning the model to train based off of the training dataset, test based off of the testing dataset, and a k value. This k value is very important because it requires a balance between overfitting and underfitting. Overfitting is when the model fits to the noise of the dataset. This is not favorable because when it comes to test the model, it would only be looking for the specific values of the variables and miss the underlying pattern. The opposite problem is underfitting where the model ignores too much of the data and does not capture the underlying pattern. I chose a K value of 7 because after testing a few various K values, 7 offered the least amount of errors.

```
#
# build model
glass_pred <- knn(train = glass.training, test = glass.test, cl = glass.trainLabels, k=7)
#
# Inspect `glass_pred`
glass_pred
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [39] 2 2 2 2 2 2 2 3 3 3 3 3 3 5 5 5 5 5 5 6 6 6 7 7 7 7 7 7 7 7
## Levels: 1 2 3 5 6 7
```

Creating a Dataframe

Here I create a dataframe that contains the data from the testing set and the predicted results from the model. This way, I can judge whether the model effectively learned the pattern of the data. I can do this by looking for any discrepancies between the observed and predicted values.

```
#
# Put glass.testLabels in a data frame
glassTestLabels <- data.frame(glass.testLabels)
dim(glassTestLabels)
```

```
## [1] 68  1
```

```
#
# merge glass_pred and glass.testLabels
merge <- data.frame(glass.testLabels, glass_pred)
dim(merge)
```

```
## [1] 68  2
```

```
#
# specify column names for final data
names <- colnames(glass.test)
finaldata <- cbind(glass.test, merge)
dim(finaldata)
```

```
## [1] 68 13
```

```
names(finaldata) <- c(names, "Observed Glass Source", "Predicted Glass Source")
# look at final data
head(finaldata)
```

```
##    ID      RI    Na   Mg   Al    Si    K   Ca Ba   Fe source
## 6   6 1.51596 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26      1
## 7   7 1.51743 13.30 3.60 1.14 73.09 0.58 8.17  0 0.00      1
## 12 12 1.51763 12.80 3.66 1.27 73.01 0.60 8.56  0 0.00      1
## 16 16 1.51761 12.81 3.54 1.23 73.24 0.58 8.39  0 0.00      1
## 22 22 1.51966 14.77 3.75 0.29 72.02 0.03 9.00  0 0.00      1
## 24 24 1.51751 12.81 3.57 1.35 73.02 0.62 8.59  0 0.00      1
##    Observed Glass Source Predicted Glass Source
## 6                      1                      1
## 7                      1                      1
## 12                     1                      1
## 16                     1                      1
## 22                     1                      1
## 24                     1                      1
```

```
#
# Checking model accuracy
x <- 0
ifelse(finaldata$`Observed Glass Source` != finaldata$`Predicted Glass Source`, x <- x+1, x <-
 x)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [39] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
paste0("This model is ", round((1-(x/68))*100, 2), "% accurate over the testing set.")
```

```
## [1] "This model is 98.53% accurate over the testing set."
```

```
#
# Checking model accuracy
x <- 0
ifelse(finaldata$`Observed Glass Source` != finaldata$`Predicted Glass Source`, x <- x+1, x <-
 x)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [39] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
paste0("This model is ", round((1-(x/68))*100, 2), "% accurate over the testing set.")
```

```
## [1] "This model is 98.53% accurate over the testing set."
```

Conclusion

The model actually had one mistake at row 163. The model predicted that the glass would be source 5 when it was actually source 3. This gives the model an accuracy of 97.06% over the testing set. This high accuracy means that it can be reliably used by investigators to quickly determine the source of any glass. After the forensics lab

tests for the chemical elements from the dataset (as well as check for refractive index), they could input their findings into this algorithm. Even if the forensics lab can already determine the source of the glass without this algorithm, this algorithm offers another layer of confidence. In either case, time and resources will be saved and hopefully be put to a better use. Furthermore, based on the correlation between some of the variables, PCA analysis could have proven effective in reducing the variables in this dataset. Perhaps this can be done in a future study or project.

Work Cited

Dalpiaz, David. “R For Statistical Learning.” Chapter 12 k-Nearest Neighbors, 28 Oct. 2020, daviddalpiaz.github.io/r4sl/knn-class.html.

“Forensic Glass Comparison: Background Information Used in Data Interpretation.” FBI, FBI, 28 June 2011, archives.fbi.gov/archives/about-us/lab/forensic-science-communications/fsc/april2009/review.

Hill, Lyle. “Float Glass vs. Flat Glass vs. Plate Glass vs. Sheet Glass.” Glass.com, 7 Jan. 2021, info.glass.com/float-flat-plate-sheet-glass/.

Srivastava, Tavish. “K Nearest Neighbor: Knn Algorithm: KNN in Python & R.” Analytics Vidhya, 18 Oct. 2020, www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/.