Wine Classification

Kevin Nolasco

9/30/2020

Introduction

The purpose of this project is to determine the Class of wine from 13 attributes. The data that is examined in this project is provided by UCI Machine Learning Repository. Each wine was grown in the same region in Italy although they were processed by three different cultivars. The cultivars are represented by three Classes: 1, 2, or 3. The columns of this dataset are as follows:

- Class: This is what we are attempting to predict. Factor
- Alcohol: Numeric
- Malic Acid: Numeric
- Ash: Numeric
- Alcalinity of Ash: Numeric
- Magnesium: Integer
- Total Phenols: Numeric
- Flavanoids: Numeric
- Nonflavanoids Phenols: Numeric
- Proanthocyanins: Numeric
- Color Intensity: Numeric
- Hue: Numeric
- OD280/OD315 of diluted wines: Numeric
- Proline: Numeric

Exploring the Data Set

First, we import the dataset and look at the structure.

```
## Classes 'data.table' and 'data.frame':   178 obs. of  14 variables:
##  $ Class                       : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Alcohol                     : num  14.2 13.2 13.2 14.4 13.2 ...
##  $ Malic_Acid                  : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
##  $ Ash                         : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
##  $ Alcalinity_of_Ash           : num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
##  $ Magnesium                   : int  127 100 101 113 118 112 96 121 97 98 ...
##  $ Total_Phenols               : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
##  $ Flavanoids                  : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
##  $ Nonflavanoids_Phenols       : num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
##  $ Proanthocyanins             : num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
##  $ Color_Intensity             : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
##  $ Hue                         : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
##  $ OD280/OD315_of_diluted wines: num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
##  $ Proline                     : int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

I want to get an idea of the percentage of wines that are Class 1,2, or 3.

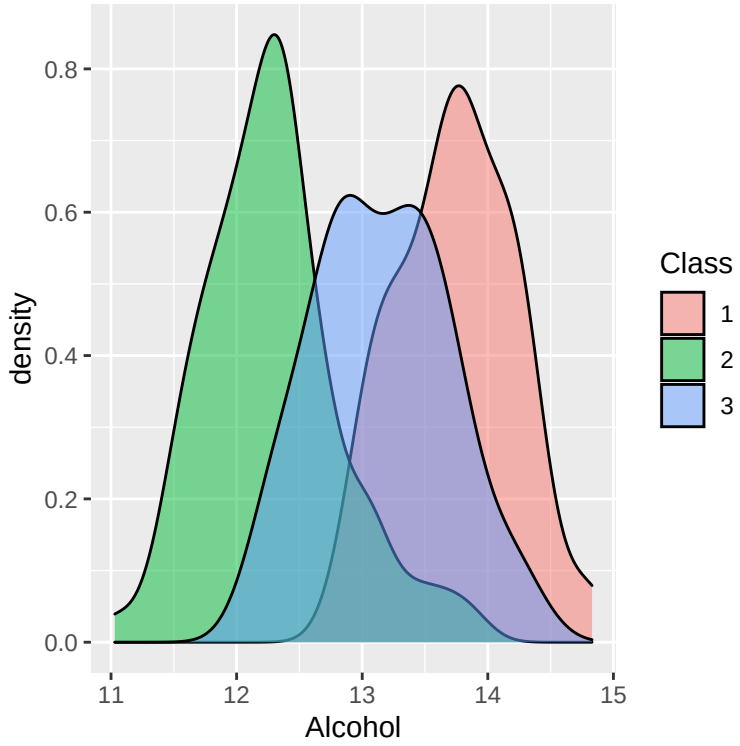
```
## The percentage of Class 1 is:  0.3314607
```

```
## The percentage of Class 2 is:  0.3988764
```

```
## The percentage of Class 3 is:  0.2696629
```

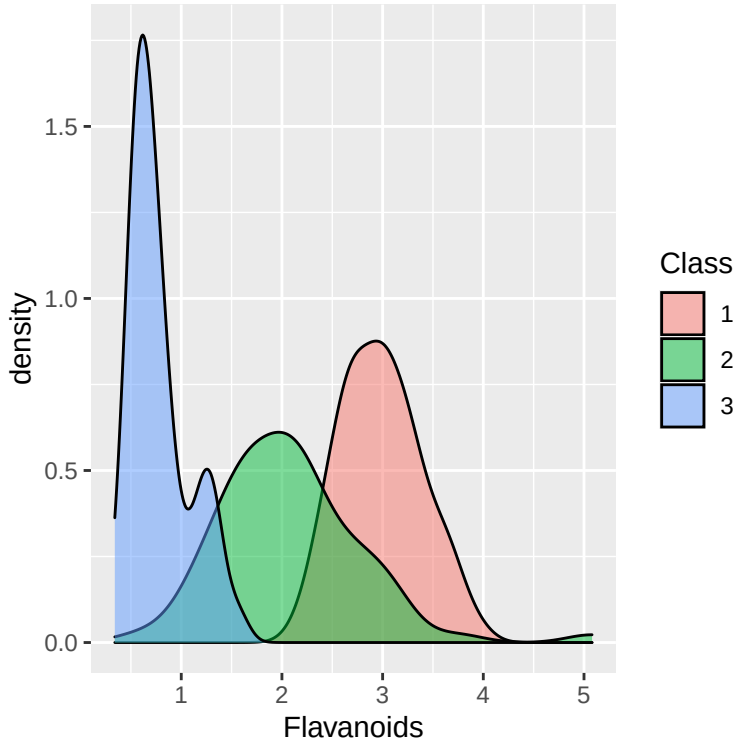
Next, I want to visualize the variables by class. To do this, I will make distributions of the variables and overlap them by class.

```
#alcohol
wine_data %>%
  ggplot(aes(x = Alcohol, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```



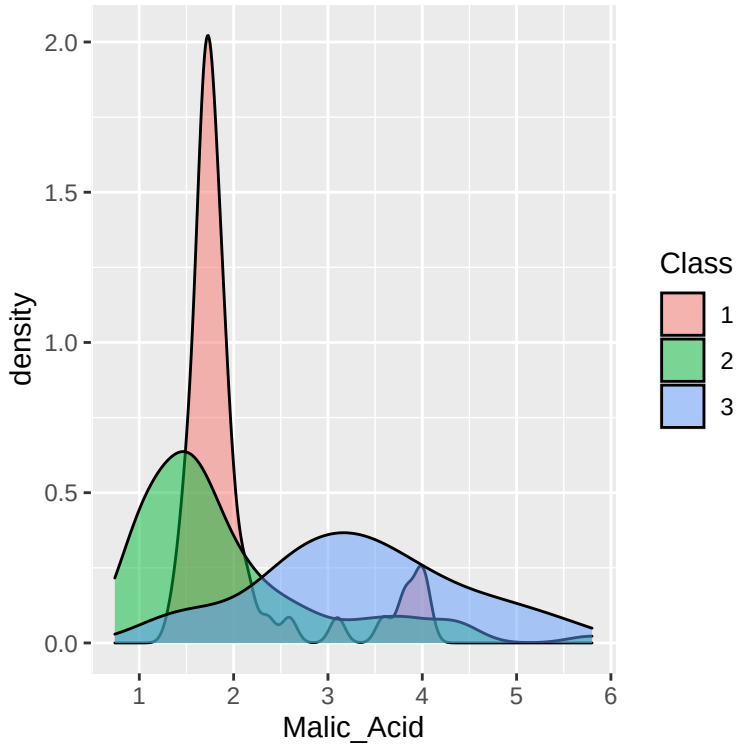

```
#Total Phenols
wine_data %>%
  ggplot(aes(x = Total_Phenols, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```


```
#Flavanoids
wine_data %>%
  ggplot(aes(x = Flavanoids, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```

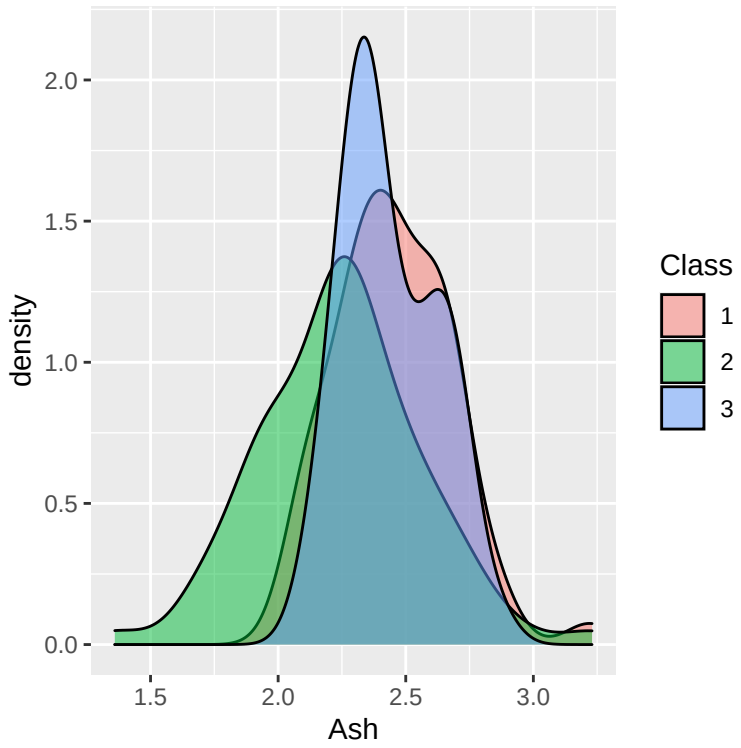


Here, I can see that the distribution of Alcohol, Total Phenols, and Flavanoids by Class is pretty different! I'll keep this in mind.

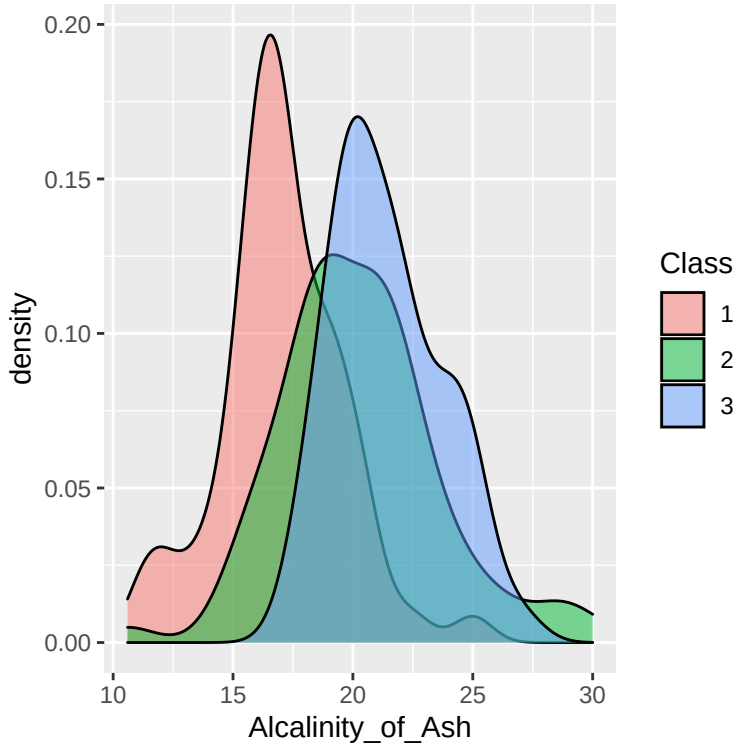
```
#Malic_acid
wine_data %>%
  ggplot(aes(x = Malic_Acid, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```



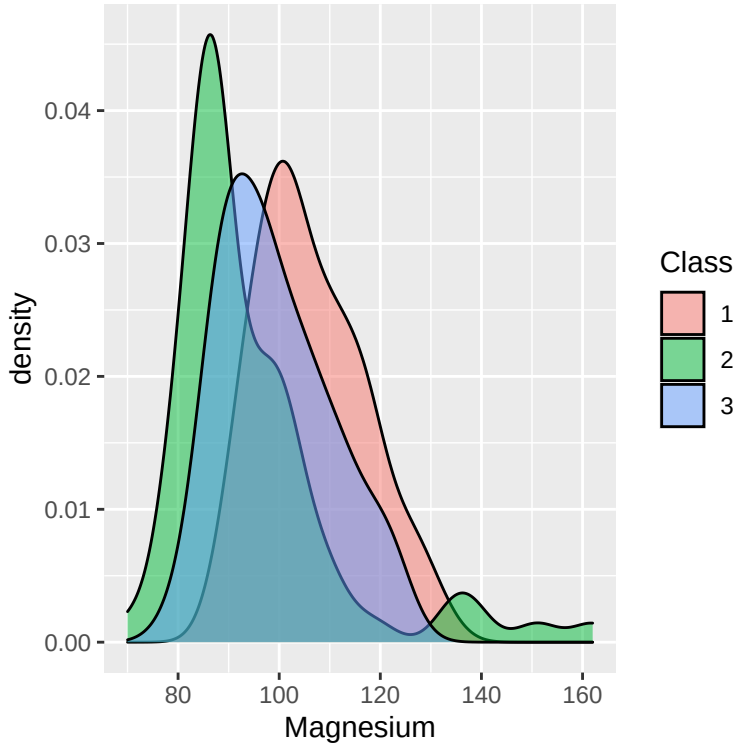

```
#ash
wine_data %>%
  ggplot(aes(x = Ash, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```



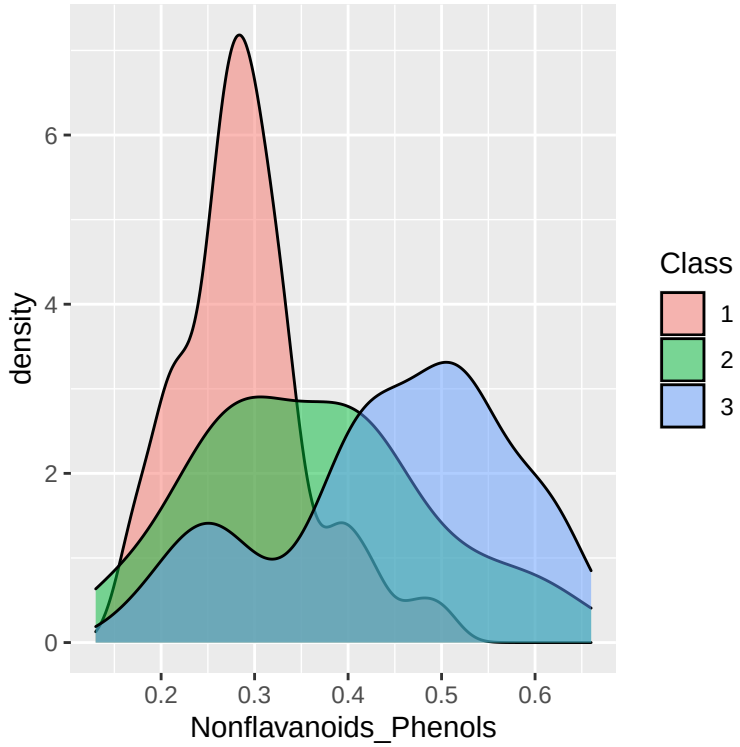

```
#alcalinity of ash
wine_data %>%
  ggplot(aes(x = Alcalinity_of_Ash, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```



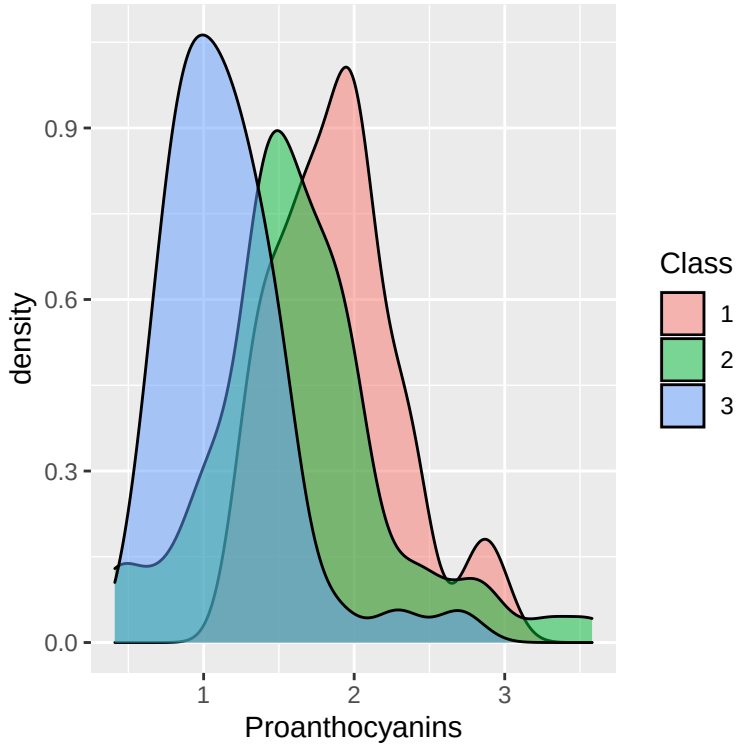
```r
#magnesium
wine_data %>%
  ggplot(aes(x = Magnesium, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```



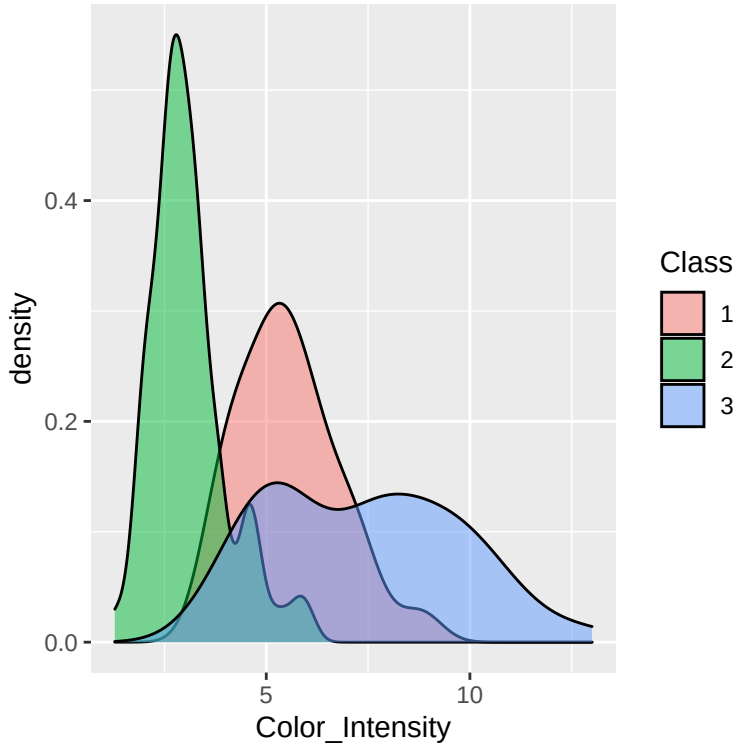

```r
#non-flavanoids phenols
wine_data %>%
  ggplot(aes(x = Nonflavanoids_Phenols, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```



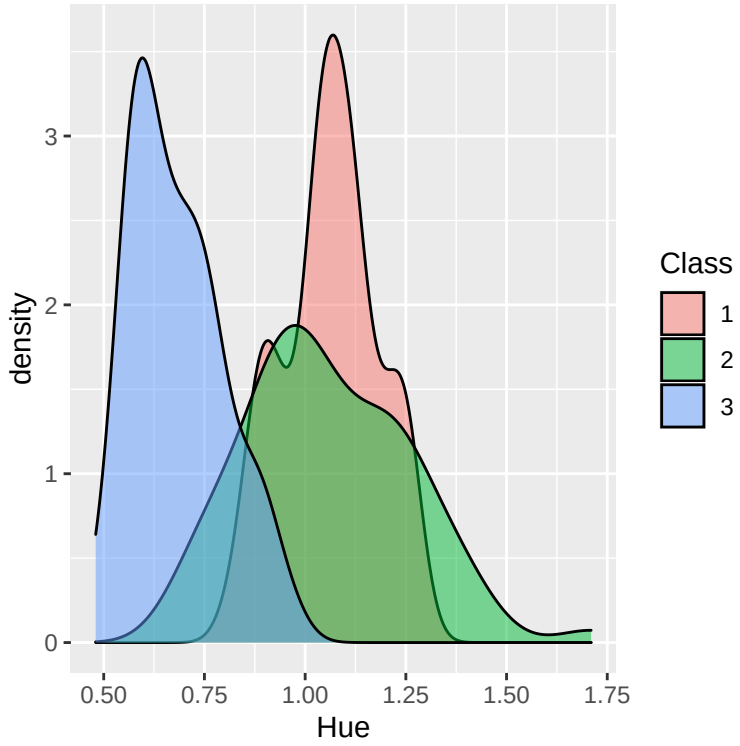

```
#Proanthocyanins
wine_data %>%
  ggplot(aes(x = Proanthocyanins, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```



```r
#Color Intensity
wine_data %>%
  ggplot(aes(x = Color_Intensity, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```

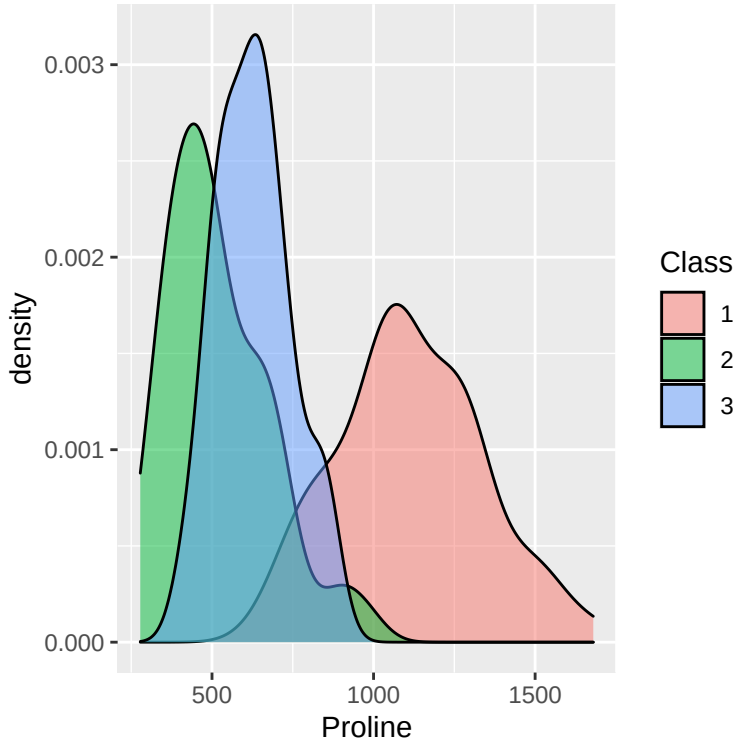



```r
#Hue
wine_data %>%
  ggplot(aes(x = Hue, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```




```
#OD280/OD315_of_diluted wines
wine_data %>%
  ggplot(aes(x = `OD280/OD315_of_diluted wines`, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```

```r
#proline
wine_data %>%
  ggplot(aes(x = Proline, group = Class, fill = Class)) +
  geom_density(alpha = 0.5)
```



The rest of the variables have similar distributions by class. There is nothing that strikes out as different.

Methods/Analysis

To create a predictive model, I created three simple (and naive) models followed by two machine learning models. Before we jump into the models, we separate the dataset into a training and testing set.

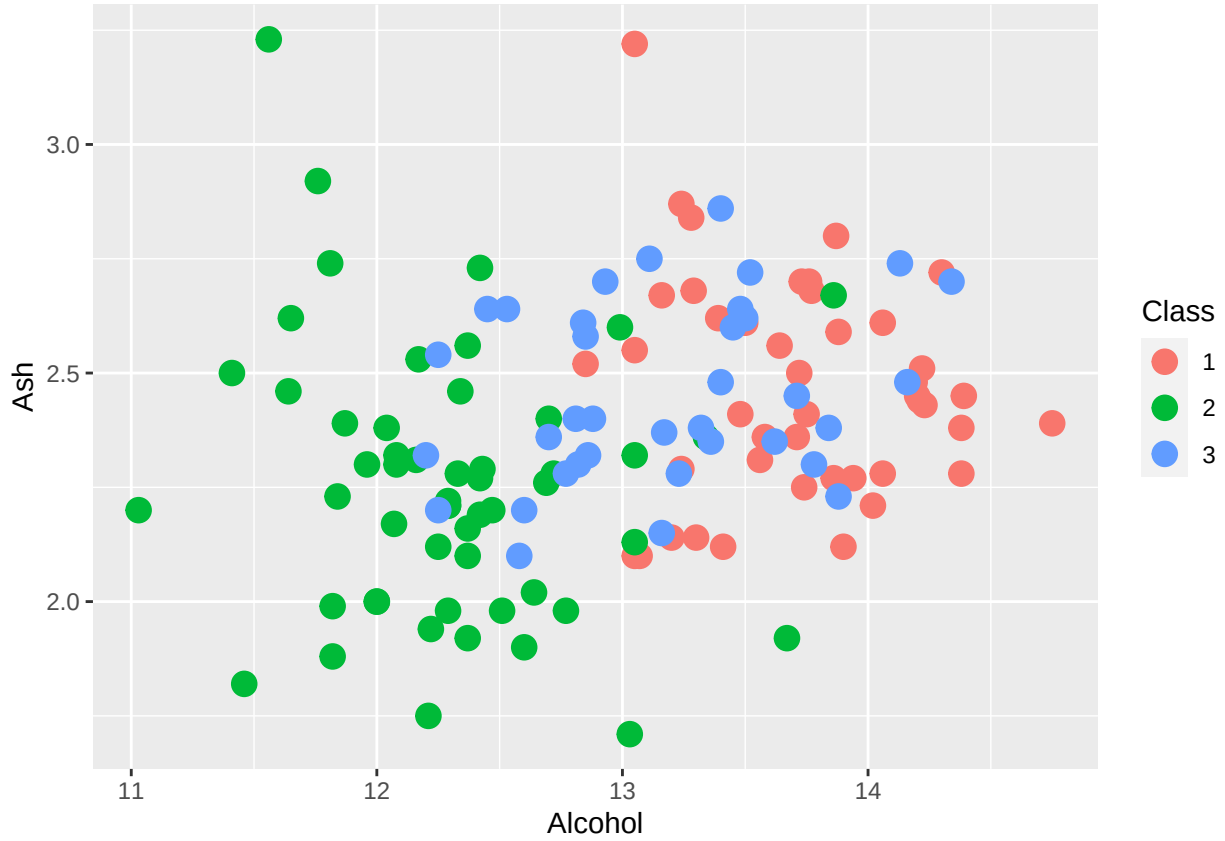
```r
#the names for the columns I specified were bad
colnames(wine_data) <- make.names(colnames(wine_data))
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(wine_data$Class, times = 1, p = 0.25, list = FALSE)

train_set <- wine_data[-test_index]
test_set <- wine_data[test_index,]
```

If/Else Models

Alcohol Vs Ash

First, let's plot Alcohol vs Ash. Ash's distribution by class showed very little differences. This initial model is meant to be simple.

```r
train_set %>%
  ggplot(aes(x = Alcohol, y = Ash, col = Class)) +
  geom_point(size = 4)
```



There appears to be groups in the data. The groups are not exclusive, which implies that the If/Else model cannot be 100% accurate.

The first model is built as follows: Wines with alcohol less than or equal to 12.75 are classified as "Class 2". Otherwise, Wines with Alcohol less than or equal to 13.75 are classified as "Class 3". Otherwise, the Wine is classified as "Class 1". These cut-off values are visually inspected.

```
alcohol_model <- function(val){
  if(val <= 12.75){
    2
  } else if(val <= 13.75){
    3
  } else 1
}

y_hat_alc <- sapply(test_set$Alcohol, alcohol_model)
y_hat_alc <- as.factor(y_hat_alc)
method1 <- mean(y_hat_alc == test_set$Class)*100
```

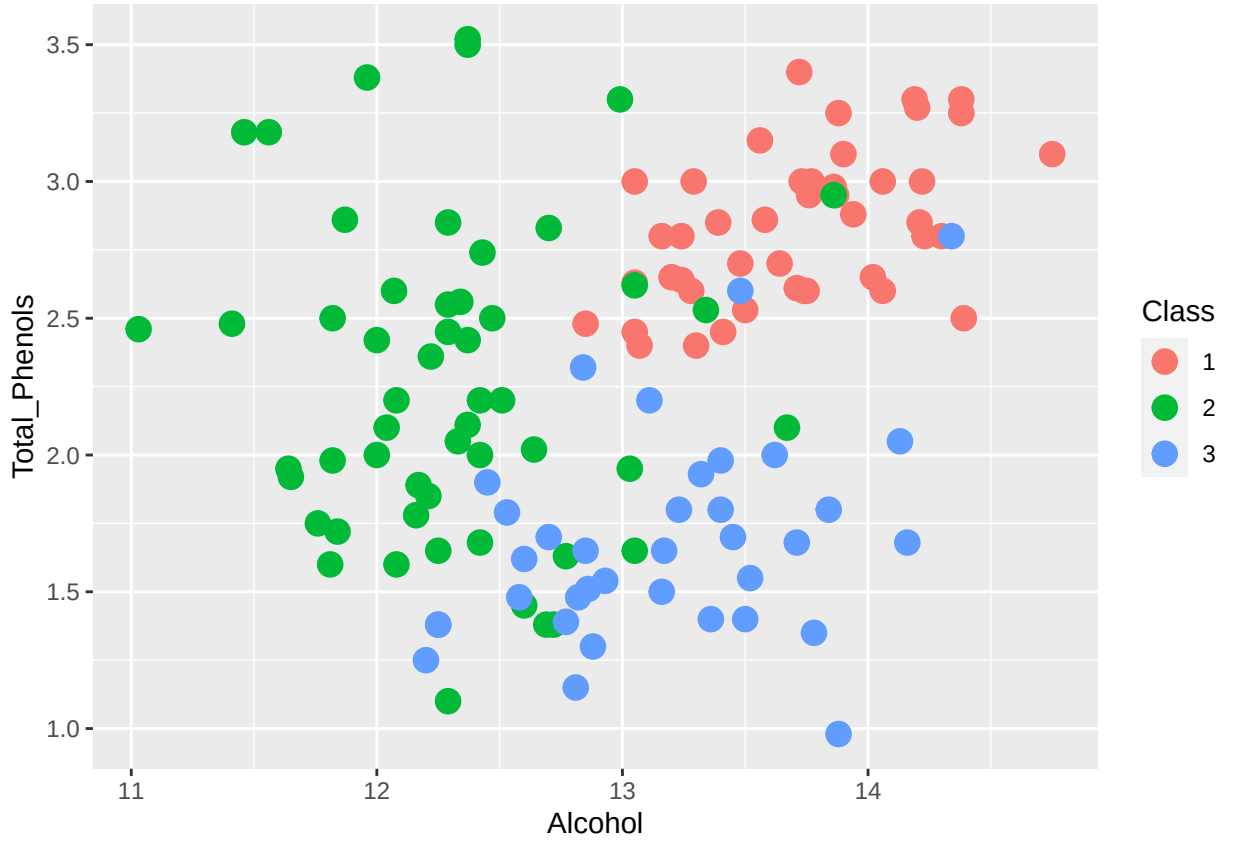
```
##         Methods Accuracy
## 1 Just Alcohol 77.77778
```

With this simple model, we get an accuracy of about 78%.

Alcohol Vs Total Phenols

For the next model, we consider Alcohol and Total Phenols. Let's plot.

```
train_set %>%
  ggplot(aes(x = Alcohol, y = Total_Phenols, col = Class)) +
  geom_point(size = 4)
```



There also appear to be gourps in this plot. The groups are a bit more distinct compared to Alcohol vs Ash so we will consider Total Phenols when constructing the If/Else model.

The next model is built as follows: Wines with alcohol less than or equal to 12.75 are classified as "Class 2". Otherwise, Wines with Total Phenols greater than or equal to 2.5 are classified as "Class 1". Otherwise, the Wine is classified as "Class 3". These cut-off values are visually inspected.

```
alc_phen <- function(alc, phen){
  if (alc <= 12.75){
    2
  } else if(phen >= 2.5){
    1
  } else 3
}

y_hat_alc_phen <- mapply(alc_phen, test_set$Alcohol, test_set$Total_Phenols)
y_hat_alc_phen <- as.factor(y_hat_alc_phen)
method2 <- mean(y_hat_alc_phen == test_set$Class)*100
```

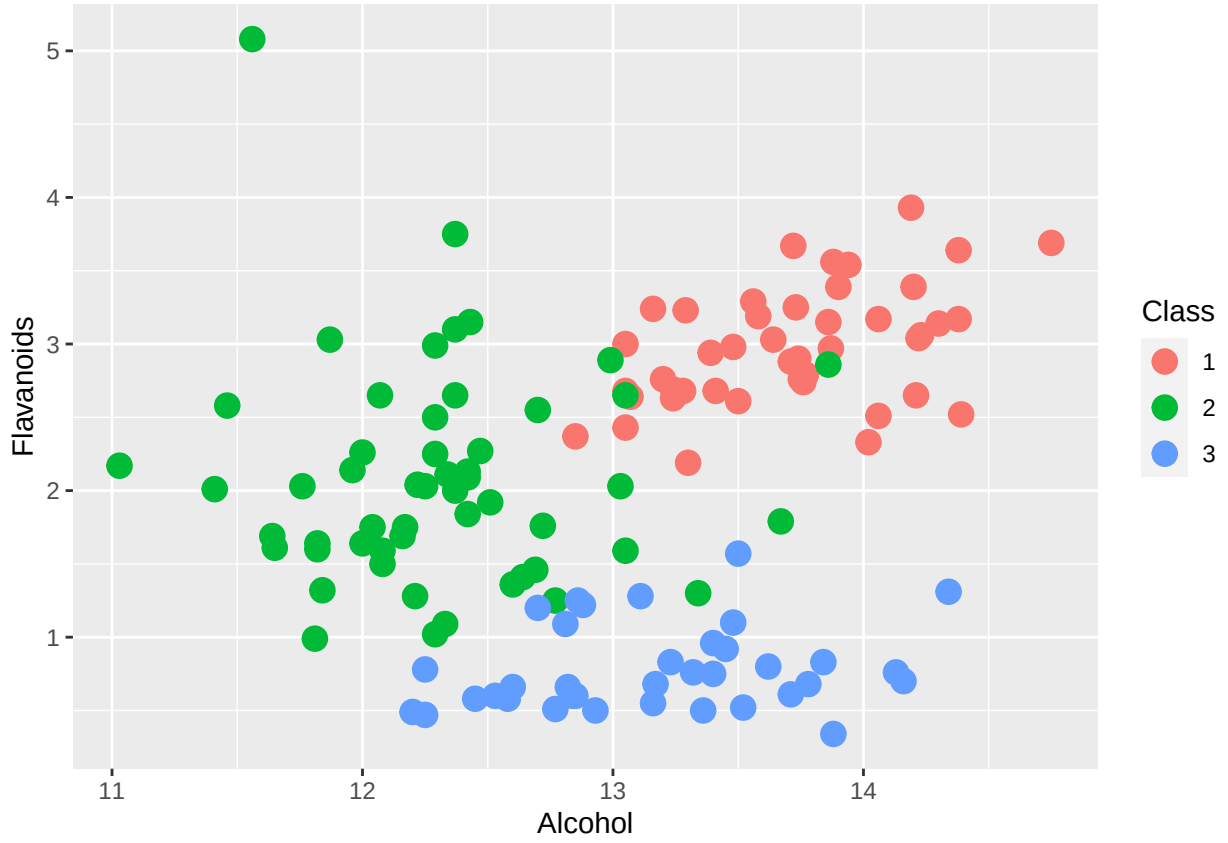
```
##                     Methods Accuracy
## 1              Just Alcohol 77.77778
## 2 Alcohol and Total Phenols 80.00000
```

With this improved model, our accuracy increases to 80%!

Alcohol Vs Flavanoids

For the final If/Else model, we consider Alcohol and Flavanoids. Let's plot.

```
train_set %>%
  ggplot(aes(x = Alcohol, y = Flavanoids, col = Class)) +
  geom_point(size = 4)
```



This plot also shows groups! For this If/Else model, we consider both Alcohol and Flavanoids.

The model is built as follows: Wines with flavanoids greater than 2.5 are classified as "Class 1". Otherwise, Wines with alcohol less than or equal to 12.5 are classified as "Class 2". Otherwise, the wine is labeled as "Class 3".

```
alc_flav <- function(alc, flav){
  if(flav > 2.5){
    1
  } else if(alc <= 12.5){
    2
  } else
    3
}

y_hat_alc_flav <- mapply(alc_flav, test_set$Alcohol, test_set$Flavanoids)
y_hat_alc_flav <- as.factor(y_hat_alc_flav)
method3 <- mean(y_hat_alc_flav == test_set$Class)*100
```

```
##                    Methods Accuracy
## 1             Just Alcohol 77.77778
```

```
## 2 Alcohol and Total Phenols 80.00000
## 3    Alcohol and Flavanoids 77.77778
```

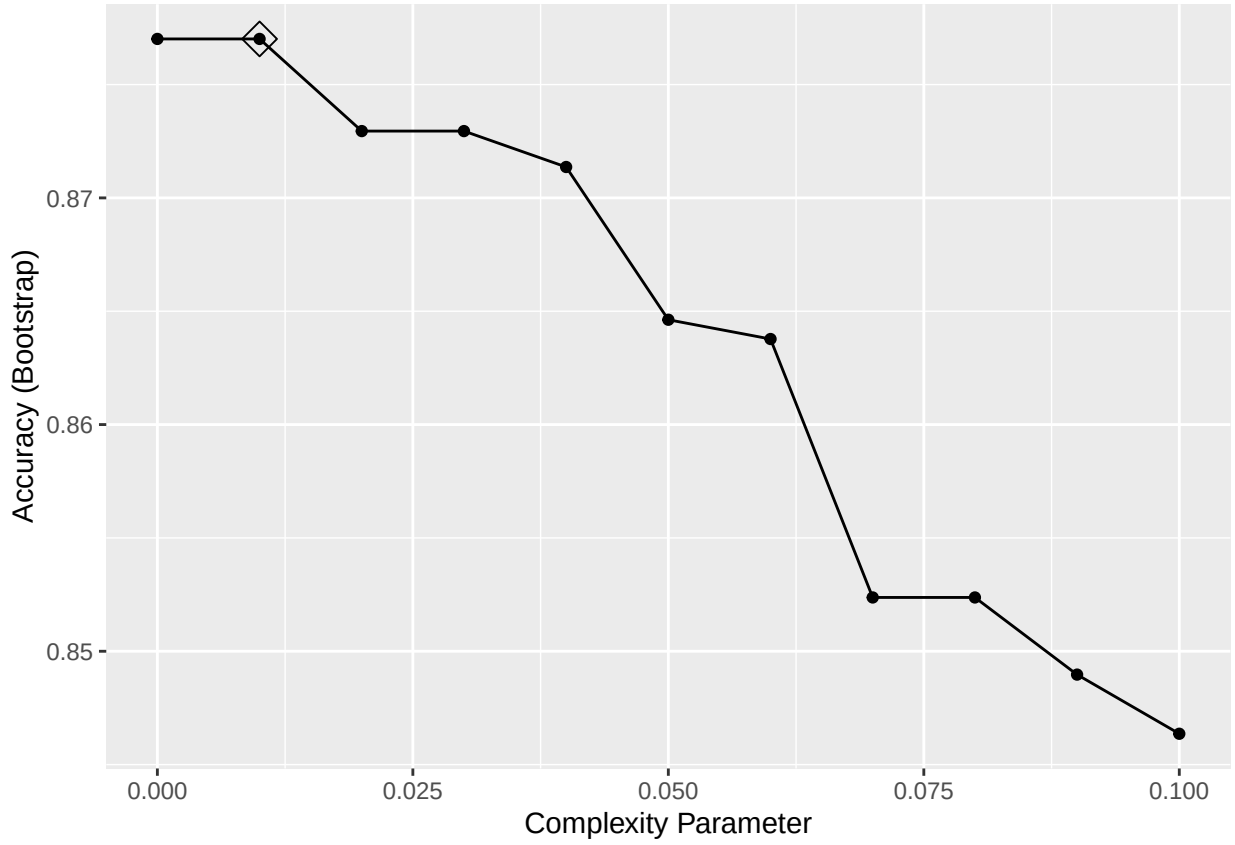
This method gave an accuracy of 77%, similar to the first model.

Machine Learning Models

Decision Tree

The first machine learning model is a Decision Tree. The model is constructed using the **train()** function that is part of the *caret* package. The train function uses Cross-Validation to find the optimal Complexity Parameter.

```
set.seed(3, sample.kind = "Rounding") #so that we get consistent results everytime.
train_rpart <- train(Class ~ .,
                     method = "rpart",
                     tuneGrid = data.frame(cp = seq(0,0.1,0.01)),
                     data = train_set)

ggplot(train_rpart, highlight = TRUE)
```

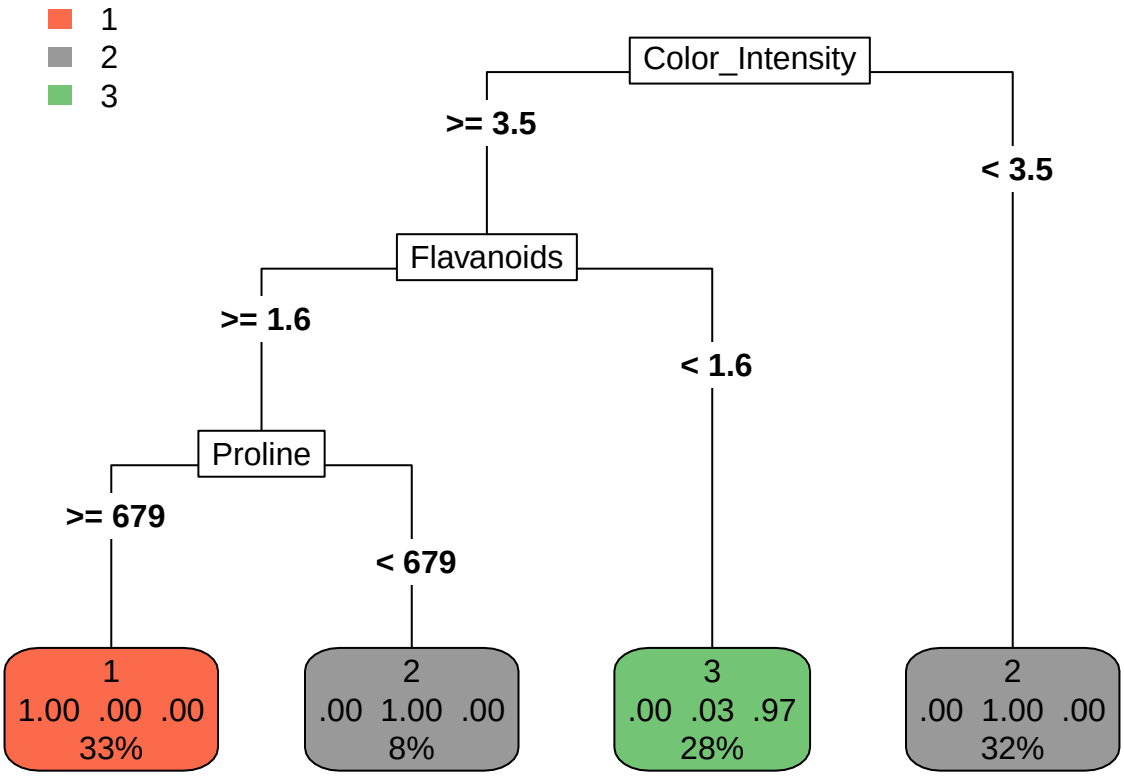



```
y_hat_rpart <- predict(train_rpart, test_set) #we make a prediction

method4 <- mean(y_hat_rpart == test_set$Class)*100
```

```
##                     Methods Accuracy
## 1              Just Alcohol 77.77778
## 2 Alcohol and Total Phenols 80.00000
```

```
## 3    Alcohol and Flavanoids 77.77778
## 4              Decision Tree 93.33333
```

By using this method, we obtain an accuracy of 93%! We can view the final model to see the decisions that the model created.

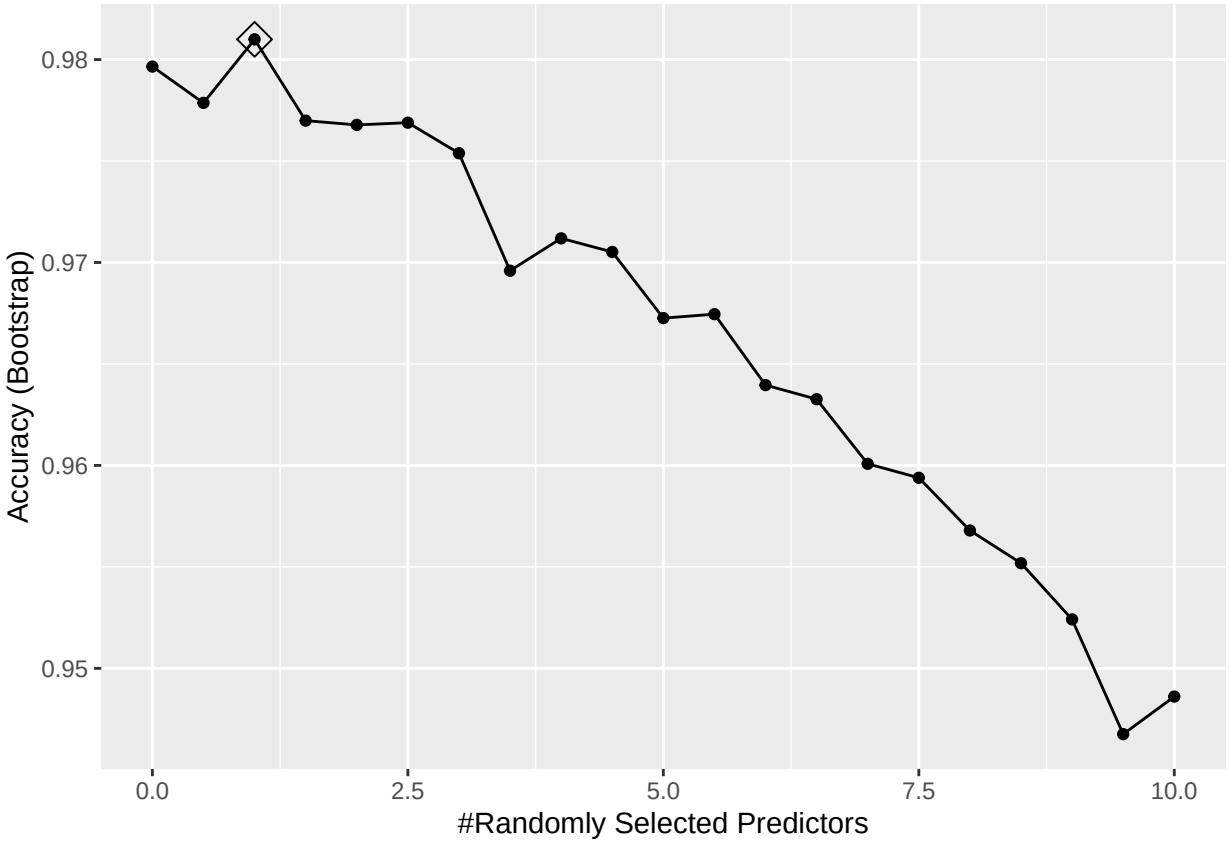


As we can see, Flavanoids, Color Intensity, and Proline were the most important variables for this model.

Random Forest

The next model is an extension of a Decision Tree. We implement the Random Forest model through the caret's train() function. The train function uses Cross-Validation to find the optimal mtry Parameter.

```
set.seed(6, sample.kind = "Rounding")#so that we get consistent results everytime.
train_rf <- train(Class ~ .,
                  data = train_set,
                  model = "rf",
                  tuneGrid = data.frame(mtry = seq(0,10,0.5)))

ggplot(train_rf, highlight = TRUE)
```




```
y_hat_rf <- predict(train_rf, test_set)
method5 <- mean(y_hat_rf == test_set$Class)*100
```

```
##                     Methods  Accuracy
## 1              Just Alcohol  77.77778
## 2 Alcohol and Total Phenols  80.00000
## 3    Alcohol and Flavanoids  77.77778
## 4             Decision Tree  93.33333
## 5             Random Forest 100.00000
```

From this model, our accuracy jumps to 100%! We can observe the variable importance as follows.

```
##                               MeanDecreaseGini
## Flavanoids                            9.363674
## Color_Intensity                       9.157813
## Alcohol                               8.181030
## OD280.OD315_of_diluted.wines          8.151966
## Hue                                   8.089092
## Proline                               7.992047
## Total_Phenols                         6.932498
## Alcalinity_of_Ash                     5.764173
## Magnesium                             5.446868
## Malic_Acid                            4.977307
## Proanthocyanins                       4.889111
## Nonflavanoids_Phenols                 4.313364
## Ash                                   3.714328
```

Here we can see the Flavanoids, Color Intensity, and Alcohol had the highest variable importance.

Results

Below is a table summarizing the accuracies of each method by using the train and test sets.

```
##                      Methods  Accuracy
## 1    Alcohol and Flavanoids  77.77778
## 2              Just Alcohol  77.77778
## 3 Alcohol and Total Phenols  80.00000
## 4             Decision Tree  93.33333
## 5             Random Forest 100.00000
```

I would like to test the Random Forest model with the entire dataset as follows.

```
final_preds <- predict(train_rf, wine_data)
final_method <- mean(final_preds == wine_data$Class)*100
cat("The accuracy of Random Forest on the entire dataset is: ",final_method,"%")
```

```
## The accuracy of Random Forest on the entire dataset is:  100 %
```

Conclusion

By applying a Random Forest algorithm, I was able to perfectly predict the classes of all wines! The results of this project can be extended to classify unknown wine's by their chemical compounds. In order to do that, this data set must be much more detailed. All of the wines that were explored in this project were from the same region in Italy; it is possible that wines from a different country could have similar chemical compounds as these wines which would require a more sophisticated predictive model.

References

Aeberhard, Stefan. et al (2020). UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/wine]. Irvine, CA: University of California, School of Information and Computer Science.