Practice5

Dasha Asienga

Due by midnight, Wednesday, Novemeber 16

Practicing Academic Integrity

If you worked with others or used resources outside of provided course material (anything besides our textbook(s), course materials in Moodle, R help menu) to complete this assignment, please acknowledge them below using a bulleted list.

I acknowledge the following individuals with whom I worked on this assignment:

Name(s) and corresponding problem(s)

-

I used the following sources to help complete this assignment:

Source(s) and corresponding problem(s)

-

Prompt

The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence...if it is correctly identified! (Spiehler 1987) It turns out that glass isn't always easily identifiable. In order to help criminologists classify glass, data was gathered and a series of classification analyses were undertaken using three techniques from class. There are four models to consider from the three techniques (one method has two models).

Below, you will find the results of those analyses. Your challenge is to write up this practice using the provided four models as your results. You'll still need to fill in the methods for them. Some preliminary analysis code has also been provided. Be sure you talk about what you learn from it. Feel free to add in more if you are interested in doing so (making other plots, etc.). In results, you should compare the four models - how they perform and what you learn from their output. Finally, the main goal is to determine what your final choice of model would be after considering all the models. Be sure to discuss your final model's AER and estimated TER (as well as details about that, like if it was generated via CV, a specific type of CV, holdout sample, or another method).

Note that the variable ID is NOT a variable that should be used in the classification. It is just the observation number.

Data Description:
1. ID number (should not be used in the analysis)
2. RI: refractive index
3. Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 3-10)
4. Mg: Magnesium
5. Al: Aluminum
6. Si: Silicon
7. K: Potassium
8. Ca: Calcium
9. Ba: Barium
10. Fe: Iron
11. Type of glass: (class attribute)
– 1 building_windows_float_processed
– 2 building_windows_non_float_processed
– 3 vehicle_windows_float_processed
– 4 vehicle_windows_non_float_processed
– 5 containers
– 6 tableware
– 7 headlamps

Your methods section should BRIEFLY describe each method applied. This shows you understand what the methods do. The methods have subjective cutoffs, but I am supplying these values for you. Instead of justifying these choices, you should just list what decisions you have to make for each method as part of the related description. This demonstrates you understand what values must be user supplied.

In the results section, since all the code to implement the methods is provided, if you feel you would have done something differently, feel free to point that out. In other words, if there's a different model you would have wanted to try, feel free to say so, and why. You will still need some code chunks to do calculations for AER and estimated TER.

Introduction

Purpose of the Analysis

The main aim is to perform a classification analysis on types of glass to aid in criminological investigations. Classifying glass correctly can help crime investigators gain more information on the crime scene and the nature of the crime performed. However, classifying glass isn't always easy. To aid in this, data was gathered and 4 classification models were fit using 3 techniques learned in class. My challenge, then, is to write-up the analysis using these 4 models as the results, providing as much information for the ready and ultimately, picking the best classification model along with justification for it.

The Data Set

The data set used in this analysis has 214 different observations and 11 different variables. The ID number, which will not be used in the analysis, is simply an observation number. There are 9 different quantitative variables that we are available to us to build our classification model: the refractive index (`RI`), Sodium (`Na`), Magnesium (`Mg`), Aluminum (`Al`), Silicon (`Si`), Potassium (`K`), Calcium (`Ca`), Barium (`Ba`), and Iron (`Fe`). All the elements are a unit measurement of weight percentage in corresponding oxide.

We will use these variables to classify glasses into 7 different pre-classified groups: `building_windows_float_processed`, `building_windows_non_float_processed`, `vehicle_windows_float_processed`,

`vehicle_windows_non_float_processed`, `containers`, `tableware`, and `headlamps`. These types of glasses are coded 1 - 7 respectively in the data set under the variable `Type`.

Preliminary Analysis

```
glass <- read.table("https://awagaman.people.amherst.edu/stat240/glass.txt", h = T, sep = ",")
glass <- mutate(glass, Type = factor(Type))
glass <- select(glass, - ID)
```

The Data

Let's begin with the preliminary analysis before proceeding with building some classification models. As seen in the output below, there are 214 different observations. We took out the ID variable, which was just an observation number, so we will proceed with the 10 remaining variables.

One of the variables, `Type`, codes the type of glass for each observation. There are 7 different types of glasses. Type 1 and type 2 have the most number of observations – there are 70 and 76 observations in these glass types respectively. From the data dictionary, these are `building_windows_float_processed` and `building_windows_non_float_processed`, so it seems most glasses at crime sites are from building windows.

Types 3, 5, and 7 have 17, 13, and 29 observations respectively and are also similar in group size. These are `vehicle_windows_float_processed`, `containers`, and `headlamps`. However, these groups contain less than a third of the observations in the large group.

Finally, type 6, which encodes glass from tableware, has only 9 observations. We will need to be careful in our analysis if small class sizes is a concern.

Note that no observations were encoded as type 4, `vehicle_windows_non_float_processed`, so it will not be relevant for our analysis.

```
dim(glass)
```

```
## [1] 214  10
```

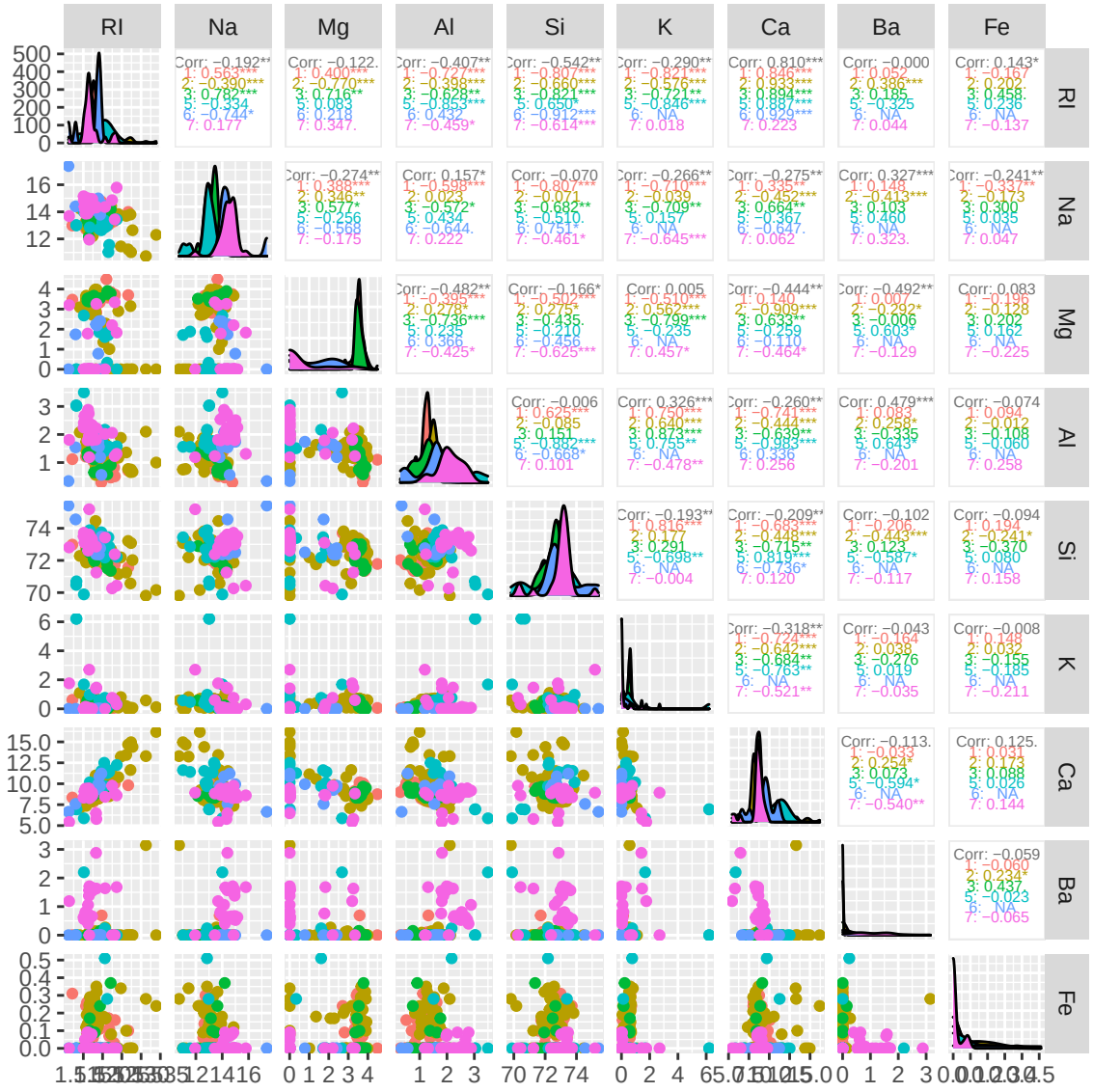
```
tally(~ Type, data = glass)
```

```
## Type
##  1  2  3  5  6  7
## 70 76 17 13  9 29
```

Multivariate Relationships

From the scatter-plot matrix below, we observe that different glasses have different peak densities for some of the quantitative variables, suggesting that some of these variables are useful in classifying or differentiating between different types of glass. Some, however, have a significant amount of overlap and may only be useful in the presence of other variables. We also notice that there are moderate to strong correlations between some of the variables, suggesting that not all of the variables may be needed in the models, especially in situations where we may have to select a few variables to proceed with. However, something important to note is that some of the variables have underlying skewed distributions, which may be a concern if we choose to run linear discriminant analysis (LDA) or quadratic discriminant analysis (QDA), which both require the variables to have multivariate normal distributions. We may proceed with caution but this will be important to take note of.

```
ggpairs(glass, columns = 1:9, ggplot2::aes(color = Type),
        upper = list(continuous = wrap("cor", size = 2)))
```



Now that we have a preliminary understanding of the data, let's proceed to run and assess some classification models.

Methods

Classification

In previous reports, we used clustering to find natural groups in our data. Clustering uses unsupervised learning to find groups when we don't know what the groups are. In many ways, classification can validate a clustering solution. This is because classification uses supervised learning to find rules to assign observations to different groups, that is, the different groups are known and pre-specified. It is similar to regression, except that the response variable is categorical (a group with many levels). Logistic regression, in fact, is a classification method where the response only has 2 levels.

For our purposes, the 9 quantitative variables in our data set – the refractive index (`RI`), Sodium (`Na`), Magnesium (`Mg`), Aluminum (`Al`), Silicon (`Si`), Potassium (`K`), Calcium (`Ca`), Barium (`Ba`), and Iron (`Fe`) – will be used to classify observations into the 6 different types of glasses, with the type of glass being our response variable.

Different classification techniques make different decision rules to accomplish classification. In this analysis, we will create 4 models using 3 different techniques: decision tree, random forest, and linear discriminant analysis (LDA).

Tree Model

Classification trees are also known as decision trees, which create binary splits on predictor variables to cut the predictor space into hyper-cubes. Ideally, the end cubes only contain 1 class. If the condition is satisfied, we move to the left of the tree, and if not, we move to the right. Different hyper-cubes per class is allowed.

One can change the `CP` value if they want to grow a larger tree or prune a tree (make it smaller). It is initially set at 0.01. A user can also set `minsplit`, which determines the minimum number of observations that can be in a node in order to make a split, and `minbucket`, which determines the minimum number of elements that must be in a node after a split. Additionally, the user can determine whether to turn cross-validation on, as well as how many folds are desired, or whether to turn it off and instead use a holdout sample.

Random Forest

Classification tress have 2 major problems: they are greedy (they don't think one step ahead) and unstable (they are extremely sensitive to changes in the data set). Random forests were designed to solve these problems by growing many trees, each built on a bootstrapped data set.

There are two main choices to make: `ntree`, which sets the number of trees to grow, and `mtry`, which sets the number of variables to be used at each split. The variables are randomly chosen at each split. Because of variable selection, this method also allows us to assess which variables are most important in the classification process.

Because this process is run on bootstrapped data sets, the out-of-bootstrap (OOB) observations from each run form a natural test set.

If `mtry = p`, then the technique is otherwise known as bagging.

Linear Discriminant Analysis (LDA)

Finally, we will also run LDA, which looks for linear combinations of variables that separate the various classes well. LDA, however, has 2 key assumptions: multivariate normal distributions (with different means) and equal covariance. It uses all variables given to the algorithm. Like classification trees, a user can turn cross-validation on or choose to use a holdout sample instead.

Assessing the Models

We will use the apparent error rate (AER) and the estimated true error rate (TER) as our main criteria for assessing the models. The AER tends to be over-optimistic because it is based on the training set, so the estimated TER may be a stronger statistic as it is based on the testing set.

Results

Let's fit 4 different models using the 3 methods described above.

The root node error is determined by finding the error rate if we classified everything into the largest class. In this case, it would be $\frac{214-76}{214} = 0.645 = 64.5\%$. We want our models to have a lower error rate than this.

Tree Model

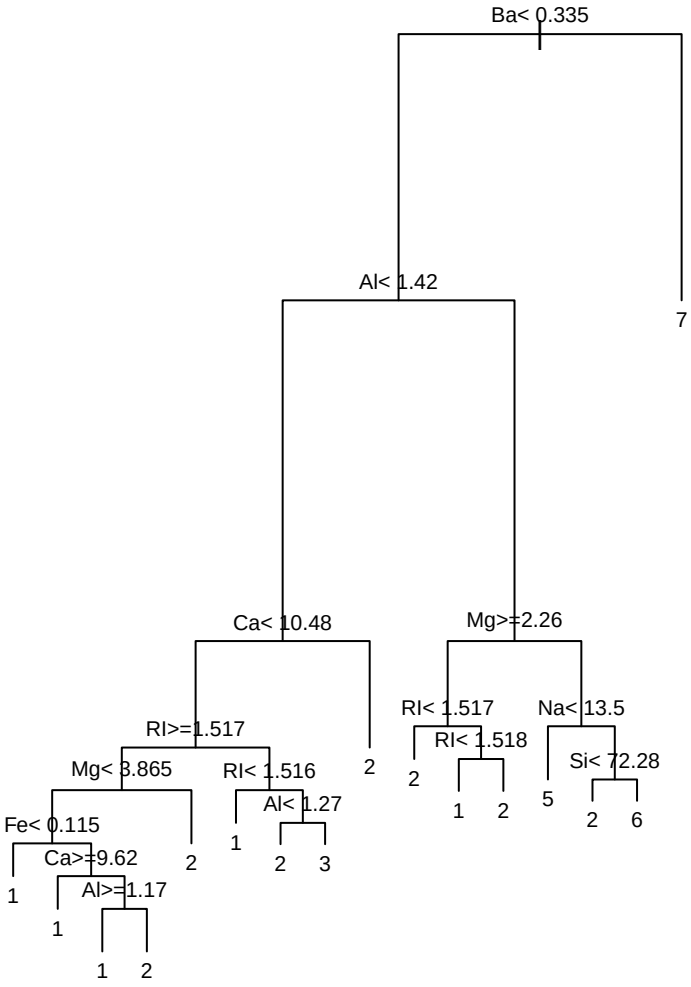
The classification tree model has set `minsplit = 8` and `minbucket = 3`, meaning that the minimum number of observations in order to make a split is 8, and each bucket will contain at least 3 observations after a split. `xval = 214`, which means that the model is running Leave One Out Cross Validation (LOOCV), with 214 folds for the 214 observations.

All quantitative variables but `K` (Potassium) were used in the actual construction of the tree (note that all variables were available to fit the model).

```
g.control <- rpart.control(minsplit = 8, minbucket = 3, xval = 214)
g.treeorig <- rpart(Type ~ ., data = glass, method = "class", control = g.control)
printcp(g.treeorig)
```

```
##
## Classification tree:
## rpart(formula = Type ~ ., data = glass, method = "class", control = g.control)
##
## Variables actually used in tree construction:
## [1] Al Ba Ca Fe Mg Na RI Si
##
## Root node error: 138/214 = 0.6449
##
## n= 214
##
##         CP nsplit rel error xerror    xstd
## 1 0.20652      0    1.0000 1.0000 0.05073
## 2 0.07246      2    0.5870 0.6014 0.05165
## 3 0.05797      3    0.5145 0.5507 0.05073
## 4 0.03623      4    0.4565 0.4783 0.04896
## 5 0.03261      5    0.4203 0.5072 0.04973
## 6 0.02174      7    0.3551 0.4493 0.04809
## 7 0.01449      8    0.3333 0.4493 0.04809
## 8 0.01087     13    0.2609 0.4420 0.04785
## 9 0.01000     15    0.2391 0.4420 0.04785
```

```
plot(g.treeorig)
text(g.treeorig, cex = 0.7)
```



\ There are 16 different end nodes, which is quite large. Glass of type 7 is quite different from all the other types. Glass of type 1 and 2 seem to be the hardest to classify and require a lot of different rules, as there are a lot of different end nodes for these 2 types of glasses. From the preliminary analysis, these made up the largest groups and there may be a lot of variability within the groups.

This model has an AER of 0.154 (15.4%) and an estimated TER of 0.285 (28.5%).

```
0.6449 * 0.2391
```

```
## [1] 0.154196
```

```
0.6449 * 0.4420
```

```
## [1] 0.285046
```

Random Forest

The random forest model has `mtry = 3` and `ntree = 1000`, meaning that each split, 3 variables are chosen at random, and 1000 trees will be grown in this forest. This model will also assess the importance of variables as well as calculate the proximity measure among the rows. All variables were used to fit the model.

From the confusion matrix, this model has an AER of 0%, but from the R output, the estimated TER is 19.63%. This is much lower than the estimated TER of the classification tree, so this model seems to be performing better on the data set.

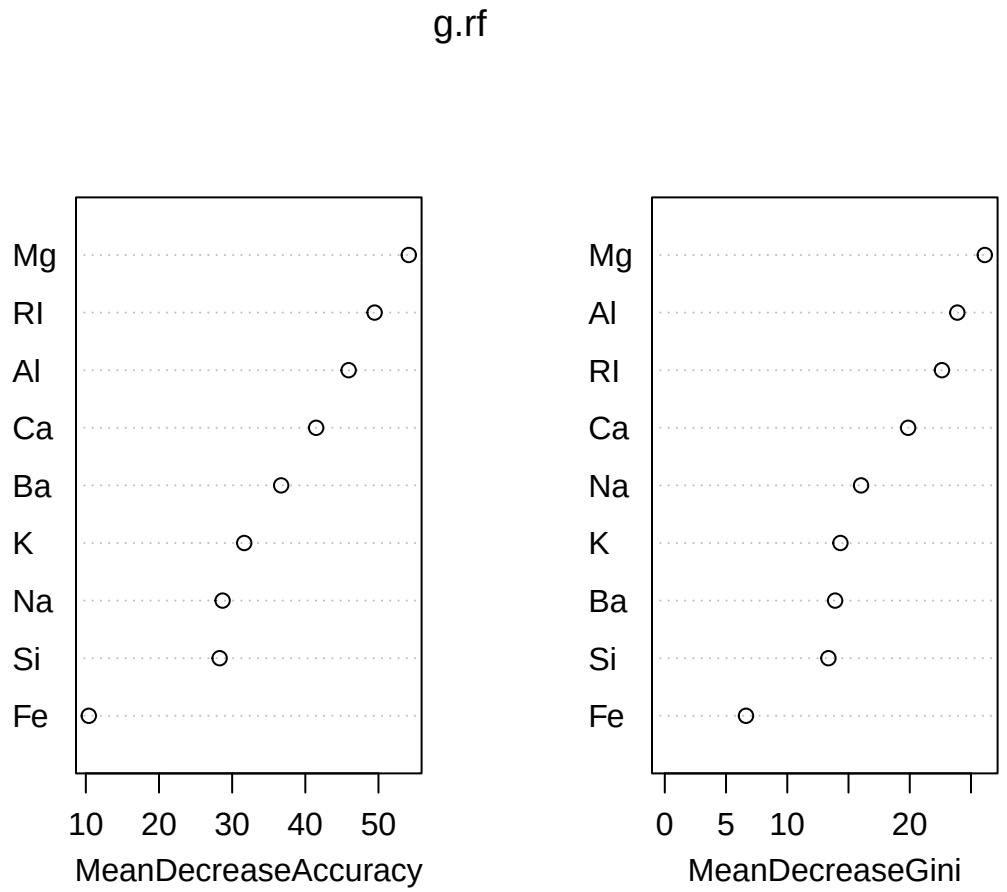
```
set.seed(240)
g.rf <- randomForest(Type ~ ., data = glass, mtry = 3, ntree = 1000,
                     importance = T, proximity = T)
g.rf
```

```
##
## Call:
##  randomForest(formula = Type ~ ., data = glass, mtry = 3, ntree = 1000,      importance = T, proximi
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 19.63%
## Confusion matrix:
##    1  2 3  5 6  7 class.error
## 1 63  6 1  0 0  0    0.100000
## 2 10 60 1  3 1  1    0.210526
## 3  8  3 6  0 0  0    0.647059
## 5  0  2 0 10 0  1    0.230769
## 6  0  2 0  0 7  0    0.222222
## 7  1  2 0  0 0 26    0.103448
```

```
table(glass$Type, predict(g.rf, glass))
```

```
##
##      1  2  3  5  6  7
##   1 70  0  0  0  0  0
##   2  0 76  0  0  0  0
##   3  0  0 17  0  0  0
##   5  0  0  0 13  0  0
##   6  0  0  0  0  9  0
##   7  0  0  0  0  0 29
```

```
varImpPlot(g.rf)
```



We also tested for variable importance, as seen in the plot above. Variable importance is measured by how often a variable was used in the decision whenever it was randomly picked, as measured by 2 indexes: mean decrease accuracy and mean decrease gini. From the plot, iron seems to be the least important variable by a notable degree. The most important variable is magnesium (`Mg`) on both indexes.

LDA (2 models)

Finally, let's fit a linear discriminant model. This method does not require a lot of user-specified choices, but we are using cross-validation to assess our solution instead of the holdout sample approach. Leave One Out Cross-Validation (LOOCV) is the automatic type of cross-validation set.

This model uses all the variables in the data set.

```
#LDA Model 1
g.lda <- MASS::lda(Type ~ ., data = glass)
g.lda
```

```
## Call:
## lda(Type ~ ., data = glass)
##
## Prior probabilities of groups:
##         1         2         3         5         6         7
## 0.3271028 0.3551402 0.0794393 0.0607477 0.0420561 0.1355140
```

```
##
## Group means:
##        RI      Na       Mg      Al      Si        K       Ca         Ba
## 1 1.51872 13.2423 3.552429 1.16386 72.6191 0.447429  8.79729 0.01271429
## 2 1.51862 13.1117 3.002105 1.40816 72.5980 0.521053  9.07368 0.05026316
## 3 1.51796 13.4371 3.543529 1.20118 72.4047 0.406471  8.78294 0.00882353
## 5 1.51893 12.8277 0.773846 2.03385 72.3662 1.470000 10.12385 0.18769231
## 6 1.51746 14.6467 1.305556 1.36667 73.2067 0.000000  9.35667 0.00000000
## 7 1.51712 14.4421 0.538276 2.12276 72.9659 0.325172  8.49138 1.04000000
##          Fe
## 1 0.0570000
## 2 0.0797368
## 3 0.0570588
## 5 0.0607692
## 6 0.0000000
## 7 0.0134483
##
## Coefficients of linear discriminants:
##           LD1       LD2        LD3         LD4         LD5
## RI 311.691252 29.391039 356.018831 246.8572080 -804.655394
## Na   2.381216  3.165080   0.459679   6.9243514    2.398751
## Mg   0.740382  2.985872   1.572884   6.8498390    2.800295
## Al   3.337742  1.724740   2.202467   6.4192364    0.937134
## Si   2.451652  3.006351   1.702619   7.5422030    0.956299
## K    1.571495  1.862016   1.286113   8.0761130    2.820993
## Ca   1.006310  2.372913   0.647520   6.6966357    3.711086
## Ba   2.314095  3.443199   2.596498   6.4384927    4.407706
## Fe  -0.511457  0.216639   1.202607  -0.0447494   -1.302921
##
## Proportion of trace:
##    LD1    LD2    LD3    LD4    LD5
## 0.8145 0.1169 0.0413 0.0163 0.0111
```

Just like in PCA, we can explain ~93% of the variance using the first 2 linear discriminants. In looking at the coefficients, the refractive index, sodium, aluminum, silicon, and barium seem to be most important. However, we will not interpret the output much further.

```
g.ldapred <- predict(g.lda, glass)
temptable1 <- table(glass$Type, g.ldapred$class)
temptable1
```

```
##
##      1  2  3  5  6  7
##   1 52 15  3  0  0  0
##   2 17 54  0  3  2  0
##   3 11  6  0  0  0  0
##   5  0  5  0  7  0  1
##   6  1  2  0  0  6  0
##   7  1  2  0  1  0 25
```

From the confusion matrix above fit on the training set, the AER is 0.327 (32.7%), which is the highest so far.

```
tempsum1 <- as.numeric(sum(as.matrix(temptable1)) - sum(diag(as.matrix(temptable1))))
tempsum1/sum(as.matrix(temptable1))
```

```
## [1] 0.327103
```

```
g.ldaCV <- MASS::lda(Type ~ ., data = glass, CV = T)
temptable2 <- table(glass$Type, g.ldaCV$class)
temptable2
```

```
##
##      1  2  3  5  6  7
##   1 51 16  3  0  0  0
##   2 18 52  0  3  2  1
##   3 11  6  0  0  0  0
##   5  0  6  0  6  0  1
##   6  1  2  0  0  5  1
##   7  1  2  0  1  0 25
```

The estimated TER is slightly higher at 0.35 (35%).

```
tempsum2 <- as.numeric(sum(as.matrix(temptable2)) - sum(diag(as.matrix(temptable2))))
tempsum2/sum(as.matrix(temptable2))
```

```
## [1] 0.350467
```

This model has the highest error rates of the 3 models fit. Let's try fitting one more model using the 4 most important variables from the random forests model output: `Mg`, `Al`, `Ri`, and `Ca`. These 4 variables were the most important according to both indexes and while they may not be an indicator of what's the best for every method, it is worth a try in order to get a stronger LDA solution.

```
#LDA Model 2
g.lda2 <- MASS::lda(Type ~ Mg + Al + RI + Ca, data = glass)
g.lda2
```

```
## Call:
## lda(Type ~ Mg + Al + RI + Ca, data = glass)
##
## Prior probabilities of groups:
##         1         2         3         5         6         7
## 0.3271028 0.3551402 0.0794393 0.0607477 0.0420561 0.1355140
##
## Group means:
##         Mg      Al      RI       Ca
## 1 3.552429 1.16386 1.51872  8.79729
## 2 3.002105 1.40816 1.51862  9.07368
## 3 3.543529 1.20118 1.51796  8.78294
## 5 0.773846 2.03385 1.51893 10.12385
## 6 1.305556 1.36667 1.51746  9.35667
## 7 0.538276 2.12276 1.51712  8.49138
##
```

```
## Coefficients of linear discriminants:
##           LD1        LD2         LD3        LD4
## Mg  -1.516548  -0.759509   -0.387167  -0.334894
## Al   0.856963  -2.231237   -1.913203  -0.639961
## RI 249.709427 263.124154 -443.961557 372.444190
## Ca  -1.035750  -1.421003    0.673720  -0.325831
##
## Proportion of trace:
##    LD1    LD2    LD3    LD4
## 0.9195 0.0538 0.0249 0.0018
```

Now, the first 2 discriminants explain ~97% of the variance, which means that almost all of the variation is accounted for. The refractive index still seems to dominate the other variables.

```
g.ldapred2 <- predict(g.lda2, glass)
temptable3 <- table(glass$Type, g.ldapred2$class)
temptable3
```

```
##
##      1  2  3  5  6  7
##   1 51 18  1  0  0  0
##   2 21 50  0  3  2  0
##   3 11  6  0  0  0  0
##   5  0  3  0  7  0  3
##   6  0  4  0  2  3  0
##   7  1  3  0  2  0 23
```

The AER of this model is 0.379 (37.9%).

```
tempsum3 <- as.numeric(sum(as.matrix(temptable3)) - sum(diag(as.matrix(temptable3))))
tempsum3/sum(as.matrix(temptable3))
```

```
## [1] 0.373832
```

```
g.ldaCV2 <- MASS::lda(Type ~ Mg + Al + RI + Ca, data = glass, CV = T)
temptable4 <- table(glass$Type, g.ldaCV2$class)
temptable4
```

```
##
##      1  2  3  5  6  7
##   1 49 20  1  0  0  0
##   2 22 48  0  4  2  0
##   3 11  6  0  0  0  0
##   5  0  3  0  6  1  3
##   6  0  4  0  2  2  1
##   7  1  3  0  2  1 22
```

The estimated TER is 0.407 (40.7)%. This is a very poor model, and LDA overall does not seem to be a good technique for this data set. Likely, one of the 2 conditions was violated – as we saw in the preliminary analysis, a lot of the underlying distributions were very skewed and not univariate normal.

```
tempsum4 <- as.numeric(sum(as.matrix(temptable4)) - sum(diag(as.matrix(temptable4))))
tempsum4/sum(as.matrix(temptable4))
```

```
## [1] 0.406542
```

Preferred Model

Ultimately, the random forests model had the lowest estimated true error rate (TER) of 19.63%. The AER was 0%. Because this model had the lowest estimated TER of the 4 models, it suggests that the model is robust enough to classify the types of glass correctly about 80% of the time.

Conclusion

The main aim was to perform a classification analysis on types of glass to aid in criminological investigations. Classifying glass correctly can help crime investigators gain more information on the crime scene and the nature of the crime performed. However, classifying glass isn't always easy. To aid in this, data was gathered and 4 classification models were fit using 3 techniques learned in class

We compared a decision tree model, a random forest model, and 2 linear discriminant models. The second LDA model used the 4 most important variables from the random forests model. Out of all 4 models, the random forest classification model is the final choice with an apparent error rate of 0% and an estimated true error rate of 19.63%. This was the lowest of all 4 models. While the error rate is still not quite ideal, the model is robust enough to classify the types of glass correctly at least about 80% of the time, and can prove as a helpful aid in crime scene investigations as well as hasten the process of classifying glass.

If I was coding these myself, I would try fitting a 5th model with nearest neighbors classifiers to determine whether we can obtain a slightly better solution. I also read about ways to optimize choice of `mtry` and `ntree` for the random forests model for my application article. I would either employ those techniques, or try out a few different values for these in order to pick values that would minimize error rates. Finally, I would not run LDA as it was predetermined that there were major skews in the univariate distributions, violating a major assumption, or perhaps only select the variables that were closest to univariate normal to include in the LDA solution.

Nevertheless, this was a great and very applicable analysis, highlighting the importance of statistics in virtually every field, including criminological investigations for something as arbitrary as classifying glass!