PSTAT 131 Term Project: Banknote Authentification

Tejal Kolte (5363114) and Samantha Solomon (5510417)

June 4, 2020

Introduction

The data set we used in our analysis is the "Banknote Authentication Data Set" from the UC Irvine Machine Learning Repository. This data was extracted in order to evaluate an authentication procedure for banknotes, and the data set contains information regarding different features of authentic and forged banknotes.

An industrial camera was used to digitize images of various real and counterfeit banknotes and a Wavelet Transform tool was used to extract features from these images. The images have 400 by 400 pixels, and have a resolution of about 660 dpi. The features extracted from the images include the Entropy of the image, the Skewness of the image, the Variance of the image, and the Curtosis of the image.

The principal goal of our analysis is to determine how these variables can be used to predict the authenticity of a banknote and use supervised learning to develop the best binary classifier for such a prediction. We also aim to find the relative importance of each of these variables.

Data

All variables in the "Banknote Authentication Data Set" from the UC Irvine Machine Learning Repository are integer attributes. There are 4 Predictor Variables: Variance (numeric), Skewness(numeric), Curtosis (numeric), and Entropy(numeric). Our Response Variable is Class, a binary response variable (integer). The indicator value 1: indicates the banknote is authentic, and the value 0: indicates the banknote is forged.

A detailed description of each variable:

- Variance (numeric): a measure of spread of the pixels in an image.
- Skewness (numeric): used to make judgements about the surface of an image; measures how "lopsided" the distribution of pixels are in an image.
- Curtosis (numeric): often used as a measure of sparsity, heaviness, of an image; also considering the measure of "tailedness" of a distribution.
- Entropy (numeric): a measure of randomness that characterizes the texture of an image; often involves looking at a histogram to understand the grayscale distribution of an image.

There are 1372 observations in our dataset. From our initial analysis of the data, we found that there are no missing values in the dataset. Thus, it is not necessary to filter our dataset. We converted our response, Class, into a factor variable with levels,"Forged" and "Authentic".

Below is a summary of our data set, "Banknote Authentification Data Set".

```
##     variance         skewness         curtosis         entropy
##  Min.   :-7.042   Min.   :-13.77   Min.   :-5.286   Min.   :-8.548
##  1st Qu.:-1.773   1st Qu.: -1.71   1st Qu.:-1.575   1st Qu.:-2.413
##  Median : 0.496   Median :  2.32   Median : 0.617   Median :-0.587
##  Mean   : 0.434   Mean   :  1.92   Mean   : 1.398   Mean   :-1.192
##  3rd Qu.: 2.821   3rd Qu.:  6.82   3rd Qu.: 3.179   3rd Qu.: 0.395
##  Max.   : 6.825   Max.   : 12.95   Max.   :17.927   Max.   : 2.450
##        class
##  Forged   :762
##  Authentic:610
##
##
##
##
```

We can look at the histograms of our predictors in order to understand the distributions of the measures of each variable on the entire data set. Below are histograms of our predictors:

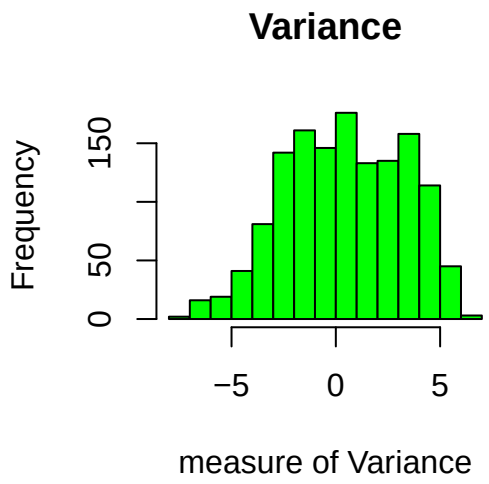

Figure 1: Variance Histogram

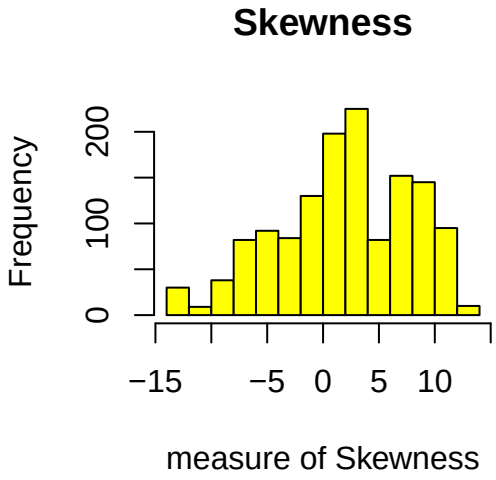

Figure 2: Skewness Histogram

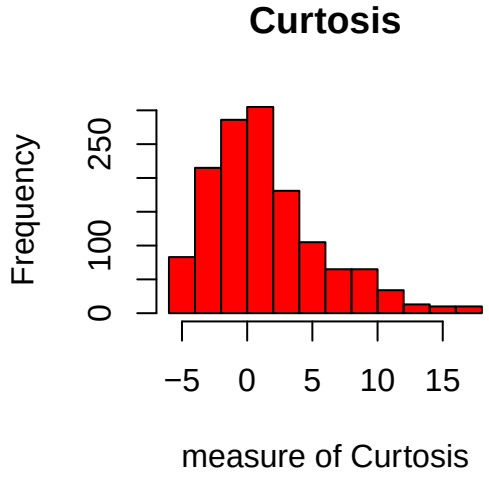

Figure 3: Curtosis Histogram

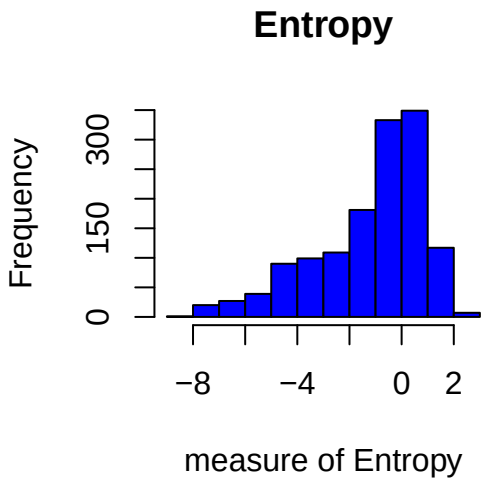

Figure 4: Entropy Histogram

Methods

We plan to use supervised learning to develop a classifier to predict the binary outcome of whether a banknote is authentic or forged. We split our data into three sets: a training set, a validation set, and a test set. We plan to use 50% of the observations for training, 25% of the observations for validation, and the remaining 25% of observations for testing. Therefore, the training set contains 686 observations, the validation set contains 343 observations, and the test set contains 343 observations. We will be using all four covariates (Variance of Image, Skewness of Image, Curtosis of Image, Entropy of Image) in our analysis.

We plan to compare the test error based on the validation set for each model in order to determine the best classifier. We will then determine the test error based on the test set for this classifier.

We will be exploring the following classifiers in our analysis:

- k-Nearest Neighbors
- Decision Trees (Classification Trees)
- Bagging
- Random Forest

We will be determining variable importance based on the Random Forest model.

Data Visualization

Scatterplot

Here we have a scatterplot of the predictors and response, on our training data.

This Scatterplot tells us that the number of forged and authentic banknotes of the dataset are somewhat evenly split among the 4 predictors. Skewness and Curtosis appear to have a clear, negative relationship. Bankotes that have a very high measure of variance are unlikely to be authentic. It also follows that banknotes that have very high measures of Skewness are unlikely to be authentic. Banknotes with higher measures of Curtosis are more likely to be authentic, and less likely to be forged. Based on the Scatterplot, the relationship between Entropy and Class is more difficult to decipher.

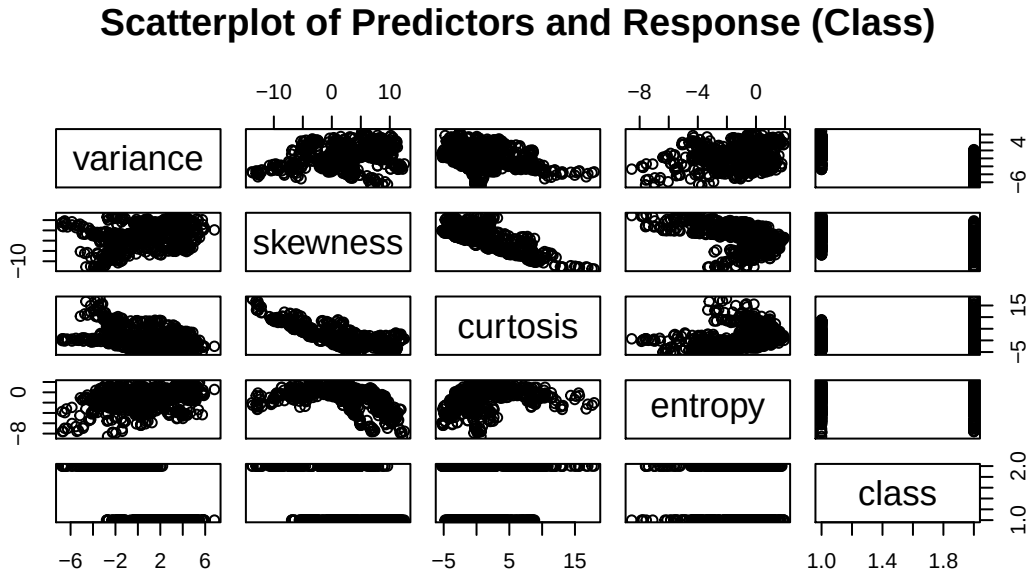

Figure 5: Scatterplot of Variables

Boxplots (on the training set)

- Forged banknotes appear to have higher measures of Variance, on average.
- Forged and Authentic banknotes appear to have similar levels of Entropy, on average.
- Forged Banknotes tend to have higher measures of Skewness, while Authentic banknotes tend to have lower levels of Skewness. The average Skewness of an authentic banknote is close to 0.
- We can see that there is a considerable amount of outliers that belong to the authentic class that have high measures of Curtosis. The average measure of Curtosis for an banknote is close to 0. We may infer that banknotes with higher measures of Curtosis are more likely to be authentic.

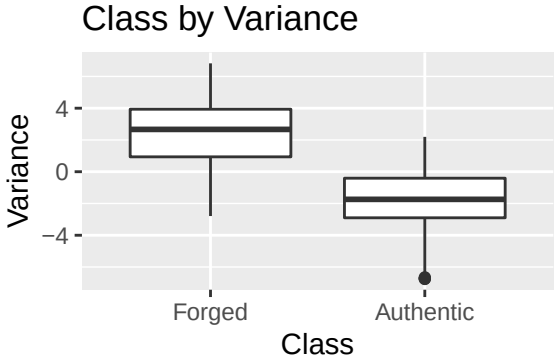

Figure 6: Class by Variance Boxplot

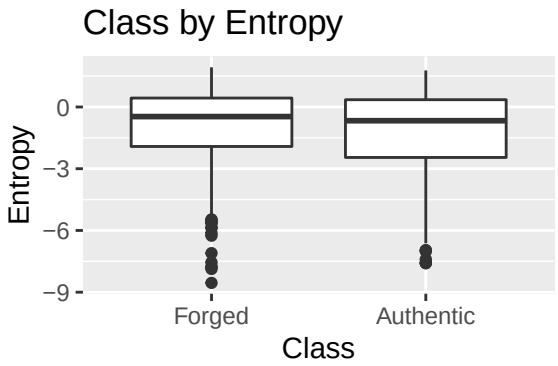

Figure 7: Class by Entropy Boxplot

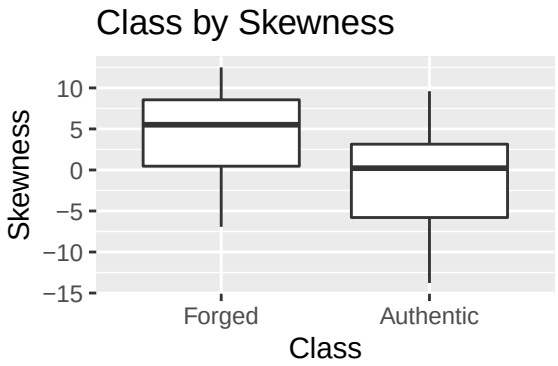

Figure 8: Class by Skewness Boxplot

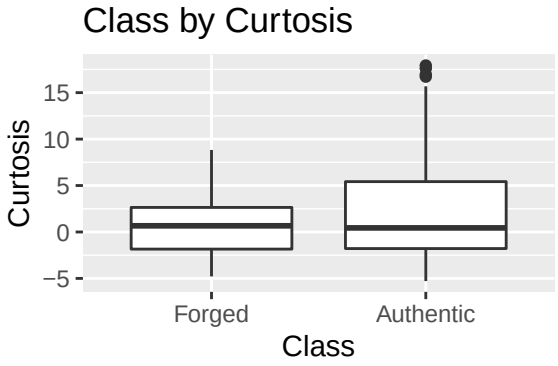

Figure 9: Class by Curtosis Boxplot

Based on the 4 boxplots on our 4 predictors, we can infer that high measures of Variance, high measures of Skewness, and lower measures of Curtosis may indicate that a banknote is forged. The measure of a banknote's Entropy may be more difficult to determine the authenticity of the banknote.

Model Building(Discussion)

Before arriving at our final model, we tried several different classifying models. We first trained our data with k-Nearest Neighbors (k_NN). We then tried different classifiers, progressing from Classifications Trees to Bagging to Random Forest. We generated an error for each classifier using our validation set.

k-Nearest Neighbors

In order to first develop a basic classifier, we decided to implement the k-Nearest Neighbors (k-NN) method. Since there are several different predictors, each with a different relative scale used in measurement, we will rescale the predictors. We will use Leave-One-Out Cross-Validation (LOOCV) to find the best number of neighbors in k-NN. We will be considering values of k ranging from 15 to 30, since there are 646 observations in the training set. This is because we do not want to set k at too high of a value or too low of a value in order to limit the bias and variance, respectively.

The idea behind k-NN is to estimate the conditional probability of Y given an observation $X = x_0$.

$$Pr(Y = j|X = x_0) \approx \frac{1}{k} \sum_{i \in N_0} I(y_i = j)$$

Table 1: Confusion Matrix for k-NN

	Forged	Authentic
Forged	188	0
Authentic	1	154

We determined that the best value of k is 19. Once we found the best value for k, we created the confusion matrix above using the validation set. The rows represent the predicted values and the columns represent the true values. At this value of k, the accuracy rate based on the validation set is 0.9971 and the error rate based on the validation set is 0.0029.

Decision Trees (Classification Trees)

Before using the Random Forest algorithm, we will use the single tree model to classify the banknote as either "Authentic" or "Forged".

Table 2: Confusion Matrix for Decision Trees

	Forged	Authentic
Forged	187	3
Authentic	2	151

After selecting best tree size using 10-fold Cross Validation, we find that the optimal tree size is 10. We then pruned the tree according to this optimal tree size. Then, we found the test error using our pruned tree and

our response values from the validation set. We find that the test error is 0.0146.

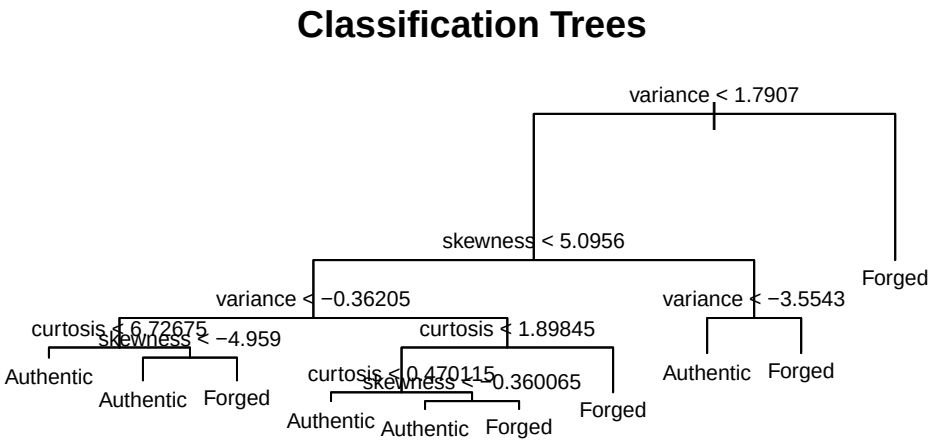

Figure 10: Classification Tree

Bagging

Prior to using the Random Forest Algorithm, we will use Bagging to compare our test error from Classification Trees and Random Forest. Bagging is very similar to Random Forest, but we use $m = p$ instead of $m = \sqrt{p}$. Hence, $m = 4$ here. Below, we plot the Out-Of-Bag error for the Bagged Decision Trees.

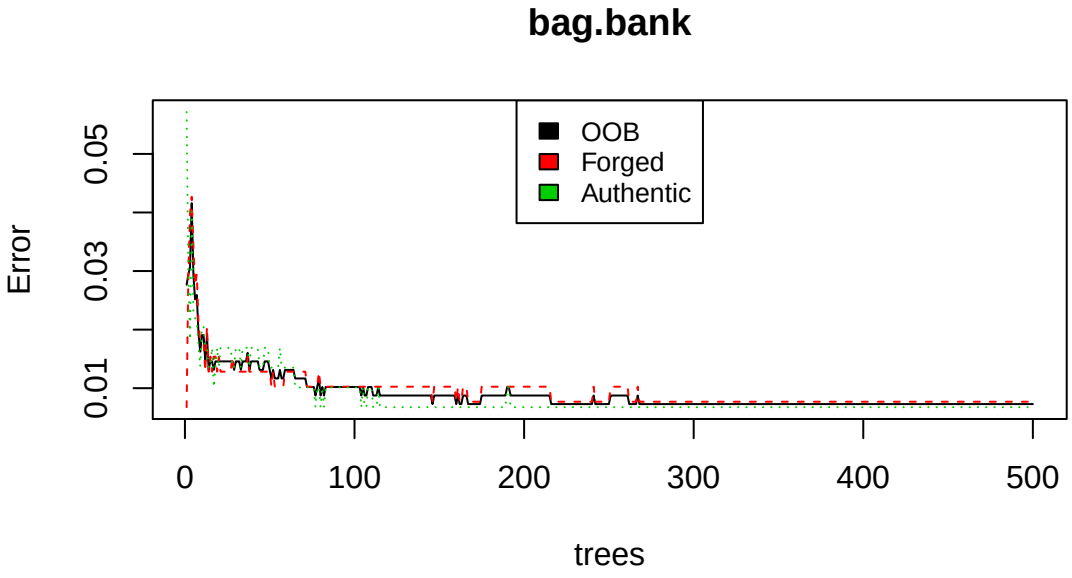

Table 3: Confusion Matrix for Bagging

	Forged	Authentic
Forged	187	0
Authentic	2	154

After using Bagging on our training data and testing with our validation set, we find that the test error is 0.0058. One of the downsides of a Bagged Decision Tree is that it is more difficult to interpret than a Classification or Regression Tree.

Random Forest

We will use the Random Forest algorithm to correct for Decision Trees' tendency to overfit to the training set. Hence, we would be decreasing the high variance that often comes with Classification Trees. Random Forest is an improvement from Bagging because it works to decorrelate bootstrap trees. Random Forest is unique because the algorithm does not allow the consideration of the majority of the predictors at each split. By only considering a subset of the predictors, Random Forest makes the result less variable and more reliable because the algorithm will not favor a strong or moderately strong predictor(s). For Classification, it is necessary to split on $m = \sqrt{p}$ randomly chosen variables. A small m is helpful if the predictors are highly correlated.

In our case, we have $p = 4$ predictors, so we will have $m = 2$ randomly chosen variables. Based on our initial analysis, we may find that Variance is a strong predictor and would be the first split variable for every tree. By using Random Forest to subsample the predictors, our trees will look more distinct and we will have a reduction of variance. Below, we plot the Out-Of-Bag error for the Random Forest.

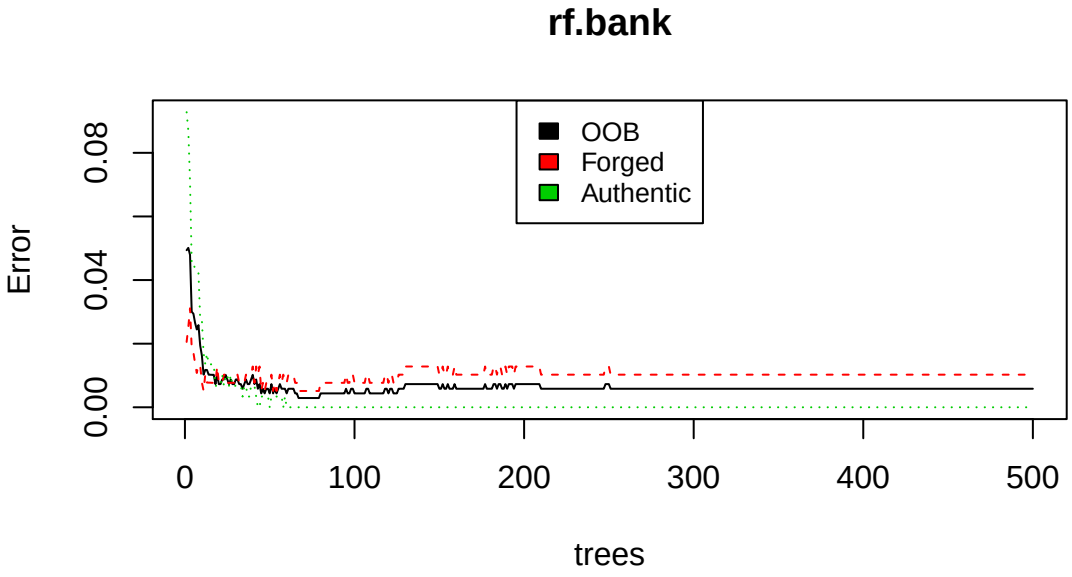

Table 4: Confusion Matrix for Random Forest

	Forged	Authentic
Forged	188	0
Authentic	1	154

After using Random Forest on our training data and testing with our validation set, we find that the test error is 0.0029.

Observation for Classification Trees, Bagging, and Random Forest

We observe that the lowest error rate comes from Random Forest. As mentioned before, Random Forest corrects for Decision Trees' tendency to overfit to the training set-causing for high variance. When we test our models on our validation set, we find that the error rate for Decision Trees is higher than Bagging and Random Forest.

Model Comparison

We will now compare the models based on the validation test error of each model.

Table 5: Validation Set Test Error for Each Model

Classifier	Validation Set Test Error
k-NN	0.00291545189504383
Classification Tree	0.0145772594752187
Bagging	0.00583090379008744
Random Forest	0.00291545189504372

Based on the table above, we can see that the Random Forest classifier had the lowest validation test error. However, this value was very close to that of k-Nearest-Neighbors when we selected k = 19. Although these values were quite close, we determined that the best classifier for our data is the Random Forest model. This is because this classifier resulted in the lowest misclassification error of the validation set. In addition, k-Nearest-Neighbors has some weaknesses, as it does not leverage the importance of some variables over others and tends to suffer from the curse of dimensionality.

Variable Importance

Since we determined that the best classifier for our data is the Random Forest classifier, we will be analyzing the importance of each variable based on this model.

```
##          Forged Authentic MeanDecreaseAccuracy MeanDecreaseGini
## variance  86.00    118.17               121.81           179.20
## skewness  47.78     67.52                70.83            78.73
## curtosis  55.65     58.77                76.81            58.33
## entropy   17.29     19.29                23.95            19.89
```

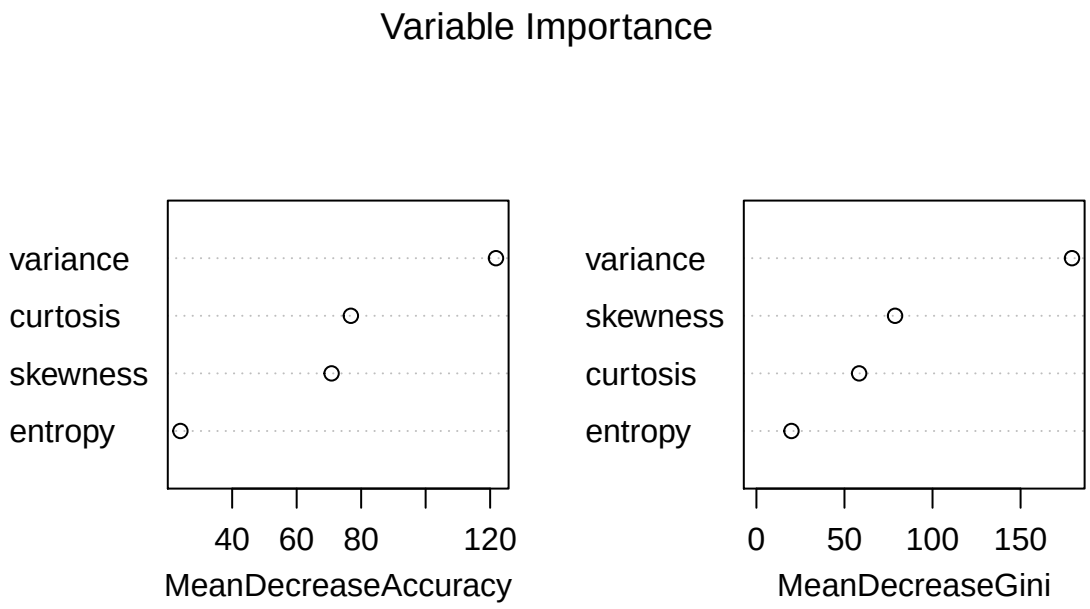

Figure 11: Variable Importance based on Random Forest Model

As shown above, based on both the Mean Decrease in Model Accuracy, and the Mean Decrease in Gini value, the order of importance of each of the four variables is as follows: Variance, Skewness, Curtosis, Entropy.

Final Model

In order to test our final model, we will now apply the Random Forest classifier we examined earlier on our testing set.

Table 6: Confusion Matrix for Random Forest (Test Set)

	Forged	Authentic
Forged	182	0
Authentic	1	160

Table 7: Error Rate Comparison for Random Forest on Validation and Test Set

Classifier	Validation Set Test Error
Random Forest on Validation Set	0.00291545189504372
Random Forest on Testing Set	0.00291545189504372

Observations

After applying the Random Forest classifier on our testing set, we find that the error rate based on this testing set is 0.0029. Therefore, we see that the error rate using the testing set and the error rate using the validation set are the same.

Conclusion

Our final model selected is the Random Forest Algorithm. We found that the classifier using Random Forest had the same error rate on the testing set as it did on the validation set. We see that in both cases, the algorithm only misclassified one observation. Our error rates may be low and similar on different data sets due to the small nature of our sample size. In the "Banknote Authentication Data Set", there are only 1372 observations. In addition, we discovered through Variable Importance that Variance is our most significant predictor. This aligns with our prior speculation as high measures of Variance have a strong influence on the classification of a banknote as forged.

Although our classifier, developed using Random Forest, appears to be quite accurate, we may want to further develop our model on a much larger data set. In order to develop a more complex, flexible model, we need a larger, more diverse set of observations. In order to expand on this research, we may want to gather more predictors to gain a more in-depth understanding of factors that influence the authenticity of a banknote, such as different types of currencies sampled. All in all, we feel that we now have a more sophisticated understanding of the process of developing a classifier using supervised learning.

Appendix

k-Nearest Neighbors

```
# Fit a k-NN model to the training set
# We decided to include all four covariates for our k-NN model.

# banknoteytrain is the observed labels for class on the training set
# banknotextrain is the design matrix

banknoteytrain = train_banknote$class
banknotextrain = train_banknote %>%
  select(variance,skewness,curtosis,entropy)
banknotextrain <- scale(banknotextrain,center = TRUE, scale = TRUE)

meanvec <- attr(banknotextrain,'scaled:center')
sdvec <- attr(banknotextrain,'scaled:scale')

# banknoteyvalid is the observed labels for class on the validation set
# banknotexvalid is the design matrix

banknoteyvalid = valid_banknote$class
banknotexvalid = valid_banknote %>% select(variance,skewness,curtosis,entropy) %>%
  scale(center = meanvec, scale = sdvec)

# Decided to use LOOCV

# Set validation.error (a vector) to save validation errors in future
validation.error = NULL

# Give possible number of nearest neighbours to be considered
# Since we want to limit the variance of the model, we will consider values of k between 15 and 30.

allK = 15:30

# Set random seed
set.seed(943)

# For each number in allK, use LOOCV to find a validation error
for (i in allK){
  pred.banknoteyval = knn.cv(train=banknotextrain, cl=banknoteytrain, k=i)
  validation.error = c(validation.error, mean(pred.banknoteyval!=banknoteytrain))
}

# Best number of neighbors
# if there is a tie, pick larger number of neighbors for simpler model
bestk = max(allK[validation.error == min(validation.error)])

# Find min. validation error
minvalidation.error = min(validation.error)

# Set k = bestk
pred.ybanknotetrain = knn(train=banknotextrain,test=banknotexvalid, cl=banknoteytrain, k=bestk)
```

```r
# Set random seed
set.seed(9472)

# Confusion matrix
conf.matrix = table(Predicted=pred.ybanknotetrain, True=banknoteyvalid)
tab.conf.matrix = kable(conf.matrix,caption="Confusion Matrix for k-NN")
tab.conf.matrix
# conf.matrix

# Test accuracy rate
knnaccuracy = sum(diag(conf.matrix)/sum(conf.matrix))

# Test error rate
knnerror = 1-sum(diag(conf.matrix)/sum(conf.matrix))
```

Decision Trees (Classification Tree)

```r
set.seed(42)
tree.bank = tree(formula = class ~., data = train_banknote)
summary(tree.bank)

# 10-fold CV for selecting best tree size
tree.bank.cv = cv.tree(tree.bank, FUN=prune.misclass, K=10)

# Best size
best.cv = min(tree.bank.cv$size[tree.bank.cv$dev==min(tree.bank.cv$dev)])

# Prune the tree to the optimal size
tree.bank.prune = prune.misclass(tree.bank, best=best.cv)

# Confusion Matrix for the pruned tree
tree.error = table(treePred=predict(tree.bank.prune, newdata=valid_banknote, type="class"),
truth=banknoteyvalid)
tab.tree.error = kable(tree.error,caption="Confusion Matrix for Decision Trees")
tab.tree.error
test.tree.error = 1 - sum(diag(tree.error))/sum(tree.error)
test.tree.error
```

Bagging

```r
set.seed(837)
# start with bagging for m = p, meaning m = 4
bag.bank = randomForest(class ~ ., data=train_banknote, mtry=4, importance=TRUE)
bag.bank

plot(bag.bank); legend("top", colnames(bag.bank$err.rate),col=1:4,cex=0.8,fill=1:4)
yhat.bag = predict(bag.bank, newdata = valid_banknote)

# Confusion matrix for Bagging
bag.error = table(pred = yhat.bag, truth = banknoteyvalid)
```

```
tab.bag.error = kable(bag.error,caption="Confusion Matrix for Bagging")
tab.bag.error
test.bag.error = 1 - sum(diag(bag.error))/sum(bag.error)
test.bag.error
```

Random Forest

```
set.seed(773)
# random Forest so m = sqrt(p) = 2
rf.bank = randomForest(class ~ ., data=train_banknote, mtry=2, importance=TRUE)
rf.bank

plot(rf.bank); legend("top", colnames(rf.bank$err.rate),col=1:4,cex=0.8,fill=1:4)
yhat.rf = predict(rf.bank, newdata = valid_banknote)

# Confusion matrix for RF
rf.error = table(pred = yhat.rf, truth = valid_banknote$class)
tab.rf.error = kable(rf.error,caption="Confusion Matrix for Random Forest")
tab.rf.error
test.rf.error = 1 - sum(diag(rf.error))/sum(rf.error)
test.rf.error
```

Final Model: Random Forest Classifier applied to Test Set

```
set.seed(770)
# random Forest so m = sqrt(p) = 2
rf.bank = randomForest(class ~ ., data=train_banknote, mtry=2, importance=TRUE)

yhat.rf.final = predict(rf.bank, newdata = test_banknote)

# Confusion matrix for RF
conf.matrix.rf.final = table(pred = yhat.rf.final, truth = test_banknote$class)
tab.rf.error.final = kable(conf.matrix.rf.final,
                           caption="Confusion Matrix for Random Forest (Test Set)")
tab.rf.error.final
rf.error.final = 1 - sum(diag(conf.matrix.rf.final))/sum(conf.matrix.rf.final)
rf.error.final
```

References

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. https://archive.ics.uci.edu/ml/datasets/banknote+authentication#