COSC 757 Data Mining Assignment 2

Mary Snyder
Department of Computer & Information Sciences
College of Science & Mathematics
Towson University
msnyde8@students.towson.edu

ABSTRACT
In this paper, I will be exploring a dataset to become more familiar with data classification through the COSC 757 Data Mining Assignment 2.

Categories and Subject Descriptors
H.2.8 **[Database Management]** Database Applications – *Data mining*

Keywords
Classification; Multivariate; Categorical; Decision Tree Classification; Naïve Bayes Classification; Random Forest Classification; Training and Testing; Holdout Method; Cross-Validation; Bootstrap; Accuracy; Error Rate; Sensitivity; Specificity; Precision; Recall; F Measure;

1. INTRODUCTION
1.1 Dataset
I chose a dataset from the UCI Machine Learning Repository classified for the task of Classification. This Balance Scale Weight & Distance dataset was generated to model psychological experiment results. The dataset contains information regarding a scale either tipped to the right, tipped to the left, or balanced. There are 625 instances with a distribution of 49 balanced, 288 left tipped, and 288 right tipped classifications. The dataset has no missing values and contains 5 attributes: Class Name, Left-Weight, Left-Distance, Right-Weight, and Right-Distance.

1.2 Objective of Analysis
The objective of classification is to predict categorical class labels for data. There are two steps to this process, model construction and model evaluation.

First a model is constructed to classify data based on a training set of values and their respective class labels for a specified classifying attribute. The model can be represented in many ways, such as classification rules, decision trees, or even mathematical formulae. The classification algorithm examines the data set values for the predictor as well as the already classified target variables in the training set. This allows the algorithm to learn which values of the predictor variables are associated with values of the classifying attribute.

Second the constructed model is used to classify new records or records in which the value of the classifying attribute is unknown. A test set, independent of the training set, is used to validate the model. The classifications learned from the training set are used to classify the data in the test set.

2. METHODOLOGY
2.1 Preprocessing
The dataset description gave the correct way to find the classification for the values as the greater of (left-distance * left-weight) and (right-distance * right-weight), with equal values meaning it is balanced. This could be simply be translated into the following formula:

(left-distance * left-weight) – (right-distance * right-weight)

where a result less than zero indicates left-tripped scale, greater than zero a right-tipped scale, and equal to zero a balanced scale. This formula helped me determine which attributes I may wish to exclude or include. In this case since all four attributes played an important role in the classification I used all of them; however, to avoid any results skew for the different classification techniques, I did not apply the formula to the dataset before processing.

2.2 Experiment Design
2.2.1 Holdout Method
In the Holdout Method, the entire data set is randomly partitioned into two independent sets: training set and the test set. The training set is used for model construction and the test set is used to evaluate the accuracy of the constructed model. In this case, the data was partitioned with 70% in the training set and 30% in the test set. Both training and test sets contained examples of each classification type.

2.3 Classification Methods
2.3.1 Decision Tree Classification
Decision Tree classification uses a flowchart-like tree structure for classification. In the tree, each test on an attribute is represented by a tree node, each outcome of the attribute test is represented by a tree branch, and each tree leaf node has a classification label.

The tree is constructed in a divide-and-conquer manner with no backtracking. At the start of tree construction, the training examples are all at the root and are partitioned reclusively based on the provided selected attributes as the construction proceeds. Partitioning is complete when all the samples for a given node belong to the same class, there are no remaining attributes for further partitioning, and there are no samples left to partition.

2.3.2 Naïve Bayes Classification
Naïve Bayes classification uses simple probabilistic classifiers based on Bayes' theorem and assumes the attributes display a strong independence. In general terms, Bayes' theorem describes the probability of an event based on an already observed event.

Bayes' theorem is formally written as follows:

Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**X**):

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$$

The theorem is used to determine the posteriori probability P(H|X) that the hypothesis holds given the observed data sample X, or in simpler terms the likelihood of the hypothesis given prior evidence, for each classification. The classification with the highest probability is assigned for that data.

2.3.3 Random Forest Classification

Random Forest classification is similar to the decision tree classification. The algorithm takes each classifying attribute and generates a decision tree using a random selection of attributes at each node to determine the split. Each sample is fed through each of the decision trees to determine a result classification and each of the resulting classifications are tallied with the most popular classification being assigned.

More formally, each tree is constructed as follows:

Let N be the number of cases in the training set.
Let M be the number of input variables
For a number m < M (constant throughout the forest growing), select m variables at random out of the input variables for each node and use them to split the node.

3. RESULTS

3.1 Analysis

3.1.1 Decision Tree Classification

The Decision Tree from constructed from the training set was more complicated than I had expected (see Figure 1). When I looked at the below Figure 2 of the relative error and complexity point (CP) it the complexity of the tree made more sense. As the size of the tree grew, the CP continued to decrease as well as the relative error. One interesting part of the resulting tree was even with the increase in tree size, the algorithm still did not determine a great way to classify balanced scales. The resulting Decision Tree from the training set had no leaf nodes with classification balanced despite numerous examples in the training data. The Decision Tree confusion matrix shown in Table 1 also confirms the trouble the Decision Tree classification had showing no balanced classifications for any of the test data.

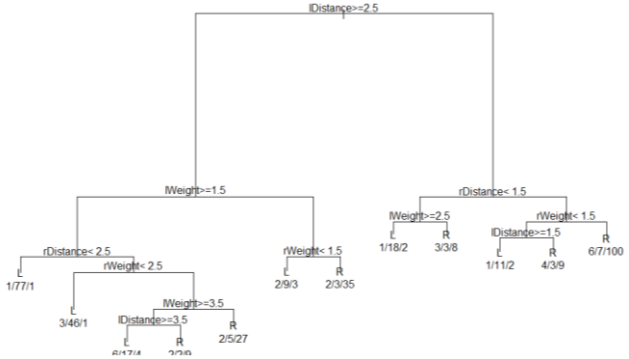

Figure 1. Decision Tree

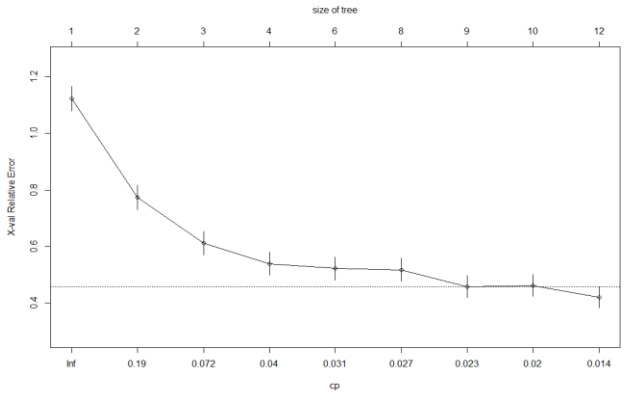

Figure 2. Relative Error and Complexity Point (CP)

Table 1. Decision Tree Confusion Matrix Results

balanceScale_pred	B	L	R
B	0	0	0
L	6	72	10
R	10	15	77

3.1.2 Naïve Bayes Classification

As is shown in Table 2, Naïve Bayes still had trouble classifying balanced scales, with no true positive results for the balanced classification of the test data. It would make sense that this would lead to some error in classifying both left and right tipped scales as well; however, the numbers show those were not the only classification errors. There were additional errors in classifying left-tipped and right-tipped scales in addition to the balance classification errors.

Table 2. Naïve Bayes Confusion Matrix Results

pred	B	L	R
B	0	0	0
L	5	74	14
R	11	13	73

3.1.3 Random Forest Classification

The Random Forest classification while similar to Decision Tree classification did produce different results. While there were still no true positive values for a balanced scale (see Table 3 below), there was lot higher percentage of true positives and conversely a lower percentage of false negatives/false positives overall.

Table 3. Random Forest Classification Results

```
Call:
 randomForest(formula = class ~ rWeight + rDistance + lWeight +
     lDistance, data = trainData, method = "class")
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 14.02%
Confusion matrix:
  B   L   R class.error
B 0  16  17  1.00000000
L 7 189   5  0.05970149
R 9   7 185  0.07960199
```

As shown below in Table 4, each of the four attributes used in the classification (right-weight, right-distance, left-weight, left-distance) have a fairly equal importance.

Table 4. Random Forest Classification Fit Importance

	MeanDecreaseGini
rWeight	54.83161
rDistance	54.47461
lWeight	55.76208
lDistance	58.52633

4. CONCLUSIONS

4.1 Evaluation Metrics

4.1.1 Accuracy and Error Rate

Accuracy is calculated as the percentage of test samples correctly calculated (TP is true positive, TN is true negative):

$$accuracy = \frac{(TP+TN)}{All\ samples}$$

Error rate is calculated as the opposite, or 1- accuracy (FP is false positive, FN is false negative):

$$error\ rate = \frac{(FP+FN)}{All\ samples}$$

4.1.2 Sensitivity and Specificity

Sensitivity is calculated as the true positive (TP) recognition rate:

$$sensitivity = \frac{TP}{P}$$

Specificity is calculated as the true negative (TN) recognition rate:

$$specificity = \frac{TN}{N}$$

Accuracy can be written as a function of both sensitivity and specificity:

$$accuracy = \frac{sensitivity*P}{(P+N)} + \frac{specificity*N}{(P+N)} = \frac{(sensitivity*P)+(specificity*N)}{(P+N)}$$

4.1.3 Precision and Recall

There is an inverse relationship between precision and recall.

Precision is measured as a percentage of the samples classified with a positive label that are actually positive, or exactness:

$$precision = \frac{TP}{TP + FP}$$

Recall is measured as a percentage of positive samples actually classified with a positive label, or completeness.

$$recall = \frac{TP}{TP + FN}$$

A perfect score would be 1.0 or 100%.

4.1.4 F-Measures

F-measure is a type of accuracy measurement which takes into account both precision and recall, with the resulting score assigned is between 0 and 1.

$$F = \frac{2 * precision * recall}{precision + recall}$$

F-measure can also be a weighted measurement as follows:

$$F = \frac{(1 + \beta^2) * precision * recall}{\beta^2 * precision + recall}$$

4.2 Decision Tree Classification Evaluation Metrics

The overall accuracy for the Decision Tree classification can be calculated using Table 1 as follows:

(0 + 72 + 77) / 190 = 0.784 = 78.4%

The overall error rate for the Decision Tree classification can be calculated using Table 1 as follows:

(16 + 15 + 10) / 190 = 0.216 = 21.6%

The sensitivity for each value for the Decision Tree classification can be calculated using Table 1 as follows:

Balanced: 0 / 0 so it cannot be calculated

Left-tipped: 72 / 88 = 0.818 = 81.8%

Right-tipped: 77 / 102 = 0.755 = 75.5%

The specificity for each value for the Decision Tree classification can be calculated using Table 1 as follows:

Balanced: 174 / 190 = 0.916 = 91.6%

Left-tipped: 87 / 102 = 0.853 = 85.3%

Right-tipped: 78 / 88 = 0.886 = 88.6%

The precision for each value for the Decision Tree classification can be calculated using Table 1 as follows:

Balanced: 0 / (0 + 16) = 0%

Left-tipped: 72 / (72 + 15) = 0.828 = 82.8%

Right-tipped: 77 / (77 + 10) = 0.885 = 88.5%

The recall for each value for the Decision Tree classification can be calculated using Table 1 as follows:

Balanced: 0 / (0 + 0) so it cannot be calculated

Left-tipped: 72 / (72 + 16) = 0.818 = 81.8%

Right-tipped: 77 / (77 + 25) = 0.755 = 75.5%

The F-measure for each value for the Decision Tree classification can be calculated using Table 1 as follows:

Balanced: cannot be calculated since recall could not be calculated

Left-tipped: (2 * 0.828 * 0.818) / (0.828 + 0.818) = 0.823 = 82.3%

Right-tipped: (2 * 0.885 * 0.755) / (0.885 + 0.755) = 0.815 = 81.5%

Overall, the Decision Tree classification seems to have performed with just over a 20% error rate and just below an 80% accuracy rate. The left-tipped and right-tipped values were evaluated very well with rates between 75% and 95%.

4.3 Naïve Bayes Classification Evaluation Metrics

The overall accuracy for the Naïve Bayes classification can be calculated using Table 2 as follows:

(0 + 74 + 73) / 190 = 0.774 = 77.4%

The overall error rate for the Naïve Bayes classification can be calculated using Table 2 as follows:

(16 + 13 + 14) / 190 = 0.226 = 22.6%

The sensitivity for each value for the Naïve Bayes classification can be calculated using Table 2 as follows:

Balanced: 0 / 0 so it cannot be calculated

Left-tipped: 74 / 93 = 0.8 = 80%

Right-tipped: 73 / 97 = 0.75 = 75%

The specificity for each value for the Naïve Bayes classification can be calculated using Table 2 as follows:

Balanced: 174 / 190 = 0.916 = 91.6%

Left-tipped: 74 / 97 = 0.763 = 76.3%

Right-tipped: 73 / 93 = 0.785 = 78.5%

The precision for each value for the Naïve Bayes classification can be calculated using Table 2 as follows:

Balanced: 0 / (0 + 16) = 0%

Left-tipped: 74 / (74 + 13) = 0.85 = 85%

Right-tipped: 73 / (73 + 14) = 0.84 = 84%

The recall for each value for the Naïve Bayes classification can be calculated using Table 2 as follows:

Balanced: 0 / (0 + 0) so it cannot be calculated

Left-tipped: 74 / (74 + 19) = 0.8 = 80%

Right-tipped: 73 / (73 + 24) = 0.75 = 75%

The F-measure for each value for the Naïve Bayes classification can be calculated using Table 2 as follows:

Balanced: cannot be calculated since recall could not be calculated

Left-tipped: (2 * 0.85 * 0.8) / (0.85 + 0.8) = 0.824 = 82.4%

Right-tipped: (2 * 0.85 * 0.75) / (0.84 + 0.75) = 0.802 = 80.2%

Overall, the Naïve Bayes classification seems to have performed with a less than 25% error rate and above 75% accuracy rate. The left-tipped and right-tipped values were evaluated very well with rates between 75% and 85%. The Decision Tree classification seems to have performed better than the Naïve Bayes classification, but it was a very slight difference. Both classifications still appear to have issues classifying the balanced scale values.

4.4 Random Forest Classification Evaluation Metrics

The overall accuracy for the Random Forest classification can be calculated using Table 3 as follows:

(0 + 189 + 185) / 435 = 0.86 = 86%

The overall error rate for the Random Forest classification can be calculated using Table 3 as follows:

(7 + 9 + 16 + 17 + 7 + 5) / 190 = 0.14 = 14%

The sensitivity for each value for the Random Forest classification can be calculated using Table 2 as follows:

Balanced: 0 / 33 = 0 = 0%

Left-tipped: 189 / 201 = 0.94 = 94%

Right-tipped: 185 / 201 = 0.92 = 92%

The specificity for each value for the Random Forest classification can be calculated using Table 3 as follows:

Balanced: 386 / 402 = 0.96 = 96%

Left-tipped: 211 / 234 = 0.902 = 90.2%

Right-tipped: 212 / 234 = 0.906 = 90.6%

The precision for each value for the Random Forest classification can be calculated using Table 3 as follows:

Balanced: 0 / (0 + 16) = 0 = 0%

Left-tipped: 189 / (189 + 23) = 0.892 = 89.2%

Right-tipped: 185 / (185 + 22) = 0.894 = 89.4%

The recall for each value for the Random Forest classification can be calculated using Table 3 as follows:

Balanced: 0 / (0 + 33) = 0 = 0%

Left-tipped: 189 / (189 + 12) = 0.94 = 94%

Right-tipped: 185 / (185 + 16) = 0.92 = 92%

The F-measure for each value for the Random Forest classification can be calculated using Table 3 as follows:

Balanced: (2 * 0 * 0) / (0 + 0) cannot be calculated since recall and prevision are both 0

Left-tipped: (2 * 0.892 * 0.94) / (0.892 + 0.94) = 0.915 = 91.5%

Right-tipped: (2 * 0.894 * 0.92) / (0.894 + 0.92) = 0.907 = 90.7%

Overall, the Random Forest classification seems to have performed even better than the Decision Tree and Naïve Bayes with a less than 15% error rate and above 85% accuracy rate. The left-tipped and right-tipped values were evaluated very well with rates between 89% and 96%. This seems to be the best fit of the all the classifications.

5. REFERENCES

[1] Larose, Daniel and Larose, Chantal D. 2014. *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley-Interscience.

[2] Siegler, R.S. (1976). *Three Aspects of Cognitive Development.* Cognitive Psychology, 8, 481-520.

[3] Klahr, D., & Sieger, R.S. (1978). *The Representation of Children's Knowledge.* H.W. Reese & L. P. Lipsitt (Eds.), *Advances in Child Develpoment and Behavior*, pp. 61-116. New York: Academic Press

[4] Langley, P. (1987). *A General Theory of Discrimination Learning.* D. Klahr, P. Langley, & R. Neches (Eds.), *Production System Models of Learning and Development*, pp. 99-161. Cambridge, MA: MIT Press

[5] Newell, A. (1990). *Unified Theories of Cognition.* Cambridge, MA: Harvard Univeristy Press

[6] McClelland, J.L. (1988). *Parallel Distributed Processing: Implication for Cognition and Development.* Technical Report AIP-47, Department of Psychology, Carnegie-Mellon University

[7] Shultz, T., Mareschal, D., & Schmidt, W. (1994). *Modeling Cognitive Development on Balance Scale Phenomena.* Machine Learning, Vol. 16, pp. 59-88.