Tomas Santos Yciano

CUS 610: Data Science Concepts & Methods

Dr. Christina Schweikert

11/8/2025

Assignment 7 – Algorithm Comparison

Comparing results and performance: classification and association rule analysis

Include results/answers for the items in red.

Here we will compare the results of different data mining approaches. A sample dataset is posted on Canvas (zoo dataset). You can use R *(include code and any results/visualizations)*.

(A) Analyze the dataset with a decision tree algorithm.

[What pre-processing is needed?]

For the zoo dataset, several key preprocessing steps were necessary to prepare the dataset for the classification algorithms of decision tree, Naïve Bayes, and Random Forest. Firstly, the animal attribute had to be deleted, as it is a unique value and unnecessary to include for this assignment. Secondly, the columns of char type had to be converted into Boolean or logical types. This is because classification algorithms cannot accept a string for true or false; they require the Boolean versions (TRUE or FALSE). Finally, the type attribute needed to be converted into a factor, as it represents categorical data (in this case, the different animal families to which each animal in the dataset belongs). Aside from that, no other cleaning or preprocessing steps were required on the original dataset, as it was already free from NAs, nulls, empty strings, whitespaces, etc. However, the dataset was highly skewed towards mammals and birds, which made up over 60% of the dataset. In real life, this makes sense, as most zoos would house more mammals and birds than other families of animals, even though invertebrates and insects are way more numerous than mammals and birds. Because of this, stratified sampling of the dataset was required to ensure that the resulting training and test datasets were representative of the entire dataset, rather than being dominated by mammals and birds. Below are the results of pre-processing for the classification algorithms, including the decision tree algorithm.

```
> zooData <- read.csv(file="/Users/themi/Downloads/zoo.csv")
```

```
> head(zooData)
    animal  hair feathers  eggs  milk airborne aquatic predator toothed backbone breathes venomous  fins legs  tail domestic
1 aardvark  true    false false  true    false   false     true    true     true     true    false false    4 false    false
2 antelope  true    false false  true    false   false    false    true     true     true    false false    4  true    false
3     bass false    false  true false    false    true     true    true     true    false    false  true    0  true    false
4     bear  true    false false  true    false   false     true    true     true     true    false false    4 false    false
5     boar  true    false false  true    false   false     true    true     true     true    false false    4  true    false
6  buffalo  true    false false  true    false   false    false    true     true     true    false false    4  true    false
  catsize   type
1    true mammal
2    true mammal
3   false   fish
4    true mammal
5    true mammal
6    true mammal
```

```
> str(zooData)
'data.frame':   99 obs. of  18 variables:
 $ animal  : chr  "aardvark" "antelope" "bass" "bear" ...
 $ hair    : chr  "true" "true" "false" "true" ...
 $ feathers: chr  "false" "false" "false" "false" ...
 $ eggs    : chr  "false" "false" "true" "false" ...
 $ milk    : chr  "true" "true" "false" "true" ...
 $ airborne: chr  "false" "false" "false" "false" ...
 $ aquatic : chr  "false" "false" "true" "false" ...
 $ predator: chr  "true" "false" "true" "true" ...
 $ toothed : chr  "true" "true" "true" "true" ...
 $ backbone: chr  "true" "true" "true" "true" ...
 $ breathes: chr  "true" "true" "false" "true" ...
 $ venomous: chr  "false" "false" "false" "false" ...
 $ fins    : chr  "false" "false" "true" "false" ...
 $ legs    : int  4 4 0 4 4 4 4 0 0 4 ...
 $ tail    : chr  "false" "true" "true" "false" ...
 $ domestic: chr  "false" "false" "false" "false" ...
 $ catsize : chr  "true" "true" "false" "true" ...
 $ type    : chr  "mammal" "mammal" "fish" "mammal" ...
```

```
> summary(zooData)
    animal              hair             feathers             eggs              milk             airborne        
 Length:99          Length:99          Length:99          Length:99          Length:99          Length:99         
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character  


   aquatic            predator           toothed            backbone          breathes           venomous        
 Length:99          Length:99          Length:99          Length:99          Length:99          Length:99         
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character  


    fins                legs            tail              domestic           catsize             type          
 Length:99          Min.   :0.000   Length:99          Length:99          Length:99          Length:99         
 Class :character   1st Qu.:2.000   Class :character   Class :character   Class :character   Class :character  
 Mode  :character   Median :4.000   Mode  :character   Mode  :character   Mode  :character   Mode  :character  
                    Mean   :2.838                                                                             
                    3rd Qu.:4.000                                                                             
                    Max.   :8.000                                                                             
```

```
> zooData <- subset(zooData, select = -c(animal))
> head(zooData)
   hair feathers  eggs  milk airborne aquatic predator toothed backbone breathes venomous  fins legs  tail domestic catsize
1  true    false false  true    false   false     true    true     true     true    false false    4 false    false    true
2  true    false false  true    false   false    false    true     true     true    false false    4  true    false    true
3 false    false  true false    false    true     true    true     true    false    false  true    0  true    false   false
4  true    false false  true    false   false     true    true     true     true    false false    4 false    false    true
5  true    false false  true    false   false     true    true     true     true    false false    4  true    false    true
6  true    false false  true    false   false    false    true     true     true    false false    4  true    false    true
    type
1 mammal
2 mammal
3   fish
4 mammal
5 mammal
6 mammal
```

```
> char_columns <- sapply(zooData, is.character)
```

```
> # Converting true/false strings to logical attributes
> zooData[char_columns] <- lapply(zooData[char_columns], function(x) x == "true")
```

```
> # Convert "type" attribute into a factor
> zooData$type <- as.factor(zooData$type)
> str(zooData)
'data.frame':   99 obs. of  17 variables:
 $ hair    : logi  TRUE TRUE FALSE TRUE TRUE TRUE ...
 $ feathers: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ eggs    : logi  FALSE FALSE TRUE FALSE FALSE FALSE ...
 $ milk    : logi  TRUE TRUE FALSE TRUE TRUE TRUE ...
 $ airborne: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ aquatic : logi  FALSE FALSE TRUE FALSE FALSE FALSE ...
 $ predator: logi  TRUE FALSE TRUE TRUE TRUE FALSE ...
 $ toothed : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
 $ backbone: logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
 $ breathes: logi  TRUE TRUE FALSE TRUE TRUE TRUE ...
 $ venomous: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ fins    : logi  FALSE FALSE TRUE FALSE FALSE FALSE ...
 $ legs    : int  4 4 0 4 4 4 4 0 0 4 ...
 $ tail    : logi  FALSE TRUE TRUE FALSE TRUE TRUE ...
 $ domestic: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ catsize : logi  TRUE TRUE FALSE TRUE TRUE TRUE ...
 $ type    : Factor w/ 7 levels "amphibian","bird",..: 6 6 3 6 6 6 6 3 3 6 ...
```

```
> # Checking for NAs, Nulls, Empty Strings, etc.
> colSums(is.na(zooData))
    hair feathers     eggs     milk airborne  aquatic predator  toothed backbone breathes venomous     fins     legs
       0        0        0        0        0        0        0        0        0        0        0        0        0
    tail domestic  catsize     type
       0        0        0        0
> any(sapply(zooData, is.null))
[1] FALSE
> colSums(zooData == "", na.rm = TRUE)
    hair feathers     eggs     milk airborne  aquatic predator  toothed backbone breathes venomous     fins     legs
       0        0        0        0        0        0        0        0        0        0        0        0        0
    tail domestic  catsize     type
       0        0        0        0
> colSums(trimws(zooData) == "", na.rm = TRUE)
```

```
> table(zooData$type)

   amphibian         bird         fish       insect invertebrate       mammal      reptile
           3           20           13            8           10           40            5
```

```
> install.packages("caret", dependencies = c("Depends", "Suggests"))
```

```
Warning: dependencies ‘MassSpecWavelet’, ‘Biobase’, ‘BiocVersion’, ‘BiocStyle’, ‘RTCGA.rnaseq’, ‘DiceOptim’, ‘globaltest’, ‘cmdstan
r’, ‘Rgraphviz’, ‘biomaRt’, ‘KEGGgraph’, ‘sylly.de’, ‘sylly.es’, ‘PCICt’, ‘starsdata’, ‘cairoDevice’, ‘RGtk2’, ‘glmmADMB’, ‘bench
r’, ‘marray’, ‘affy’, ‘limma’, ‘INLA’, ‘rnaturalearthhires’, ‘graph’, ‘Rcampdf’, ‘tm.lexicon.GeneralInquirer’, ‘Biostrings’, ‘Exact
Data’, ‘RDCOMClient’, ‘designmatch’, ‘gurobi’, ‘mlr3proba’, ‘rrelaxiv’, ‘M3C’, ‘spDataLarge’, ‘blavsam’, ‘iwmm’, ‘BRugs’, ‘genefilt
er’, ‘sva’, ‘ALEPlot’, ‘polars’, ‘RandomFields’, ‘taxidata’, ‘shock’, ‘ComplexHeatmap’ are not available

  also installing the dependencies ‘Rmosek’, ‘DTRlearn2’, ‘geomapdata’, ‘REBayes’, ‘attachment’, ‘dockerfiler’, ‘gistr’, ‘lda’, ‘OAI
  Harvester’, ‘naivebayes’, ‘sparsediscrim’, ‘polle’, ‘SurvMetrics’, ‘mlr3misc’, ‘irlba’, ‘spocc’, ‘ecospat’, ‘usdm’, ‘bezier’, ‘aw
  s.ec2metadata’, ‘aws.signature’, ‘sodium’, ‘dr’, ‘Gmisc’, ‘Greg’, ‘GEOmap’, ‘ashr’, ‘golem’, ‘geojsonio’, ‘topicmodels’, ‘discri
  m’, ‘RcppHungarian’, ‘targeted’, ‘aorsf’, ‘ulid’, ‘apcluster’, ‘LPCM’, ‘stream’, ‘fastVoteR’, ‘genalg’, ‘lsa’, ‘synchronicity’, ‘w
  rapr’, ‘rquery’, ‘rqdatatable’, ‘Rtsne’, ‘cccd’, ‘coRanking’, ‘diffusionMap’, ‘pcaL1’, ‘umap’, ‘maxnet’, ‘ENMeval’, ‘archive’, ‘ma
  ptiles’, ‘stddiff’, ‘tableone’, ‘dynamicTreeCut’, ‘DendSer’, ‘circlize’, ‘zipfR’, ‘lavaan.mi’, ‘restriktor’, ‘mcmcse’, ‘rngtools’,
  ‘doRedis’, ‘beepr’, ‘pbmcapply’, ‘ntfy’, ‘shinyvalidate’, ‘brio’, ‘gargle’, ‘shinychat’, ‘irr’, ‘fortunes’, ‘denstrip’, ‘rjson’,
  ‘sommer’, ‘sortable’, ‘LDRTools’, ‘tourr’, ‘pimeta’, ‘robvis’, ‘crossnma’, ‘forestplot’, ‘compute.es’, ‘NlcOptim’, ‘futile.logge
  r’, ‘bspec’, ‘RSEIS’, ‘signal’, ‘bartMachineJARs’, ‘ncvreg’, ‘TTR’, ‘RMySQL’, ‘downloader’, ‘fauxpas’, ‘webmockr’, ‘bs4Dash’, ‘hig
  hcharter’, ‘vioplot’, ‘ccaPP’, ‘udpipe’, ‘finetune’, ‘tidyclust’, ‘lava’, ‘censored’, ‘mlflow’, ‘jsonvalidate’, ‘modules’, ‘packra
  t’, ‘rotor’, ‘xaringan’, ‘revealjs’, ‘ggparty’, ‘mlr3cluster’, ‘mlr3fselect’, ‘mlr3inferr’, ‘precrec’, ‘mlr3measures’, ‘quanteda’,
  ‘stopwords’, ‘bestNormalize’, ‘smotefamily’, ‘NMF’, ‘vtreat’, ‘dimRed’, ‘redux’, ‘biomod2’, ‘geodata’, ‘PCAmixdata’, ‘Boruta’, ‘ca
  rSurv’, ‘wav’, ‘yulab.utils’, ‘DIDmultiplegt’, ‘grf’, ‘FactoInvestigate’, ‘dtplyr’, ‘slider’, ‘tidyterra’, ‘SGPdata’, ‘sdmTMB’, ‘s
  jstats’, ‘smd’, ‘apollo’, ‘fastDummies’, ‘gmnl’, ‘mixl’, ‘cardx’, ‘ggsurvfit’, ‘purrrlyr’, ‘BayesXsrc’, ‘MBA’, ‘dendextend’, ‘wk
  b’, ‘cmm’, ‘ggpcp’, ‘doMC’, ‘languageR’, ‘modeest’, ‘semTools’, ‘nimble’, ‘ggside’, ‘doRNG’, ‘future.callr’, ‘doFuture’, ‘progress
  r’, ‘philentropy’, ‘transport’, ‘glba’, ‘binom’, ‘Laplac ... <truncated>
```

```
  There is a binary version available but the source version is later:
       binary  source needs_compilation
ENMeval  2.0.4 2.0.5.2             FALSE
```

```
Package which is only available in source form, and may need compilation of C/C++/Fortran: ‘Rmpi’
```

```
> # Since animal types are highly unbalanced towards mammals, use stratified sampling
> library(caret)
> set.seed(1)
> trainingIndex <- createDataPartition(zooData$type, p = 0.70, list = FALSE)
> # Splitting zooData into training/testing set with 70/30 split
> trainingData <- zooData[trainingIndex, ]
> testingData <- zooData[-trainingIndex, ]
```

Data	
testingData	27 obs. of 17 variables
trainingData	72 obs. of 17 variables

[What percentage of instances are correctly classified by your decision tree?] – Compute the accuracy]

The percentage of instances correctly classified by my decision tree was approximately 92.59%, as indicated by the subsequent confusion matrix. Calculating it manually, the percentage of instances correctly classified was approximately 92.59259%, basically the same as what the confusion matrix reported. In addition to the accuracy, it falls within the 95% confidence interval, which ranges from 75.71% to 99.09%. Cohen's Kappa is also at 89.69%, which means that the decision tree model is that much better at classifying than random guessing. Overall, this decision tree model is highly reliable based on these metrics. Below are the results of setting up the decision tree model, including its output and a plotted decision tree to visualize the results.

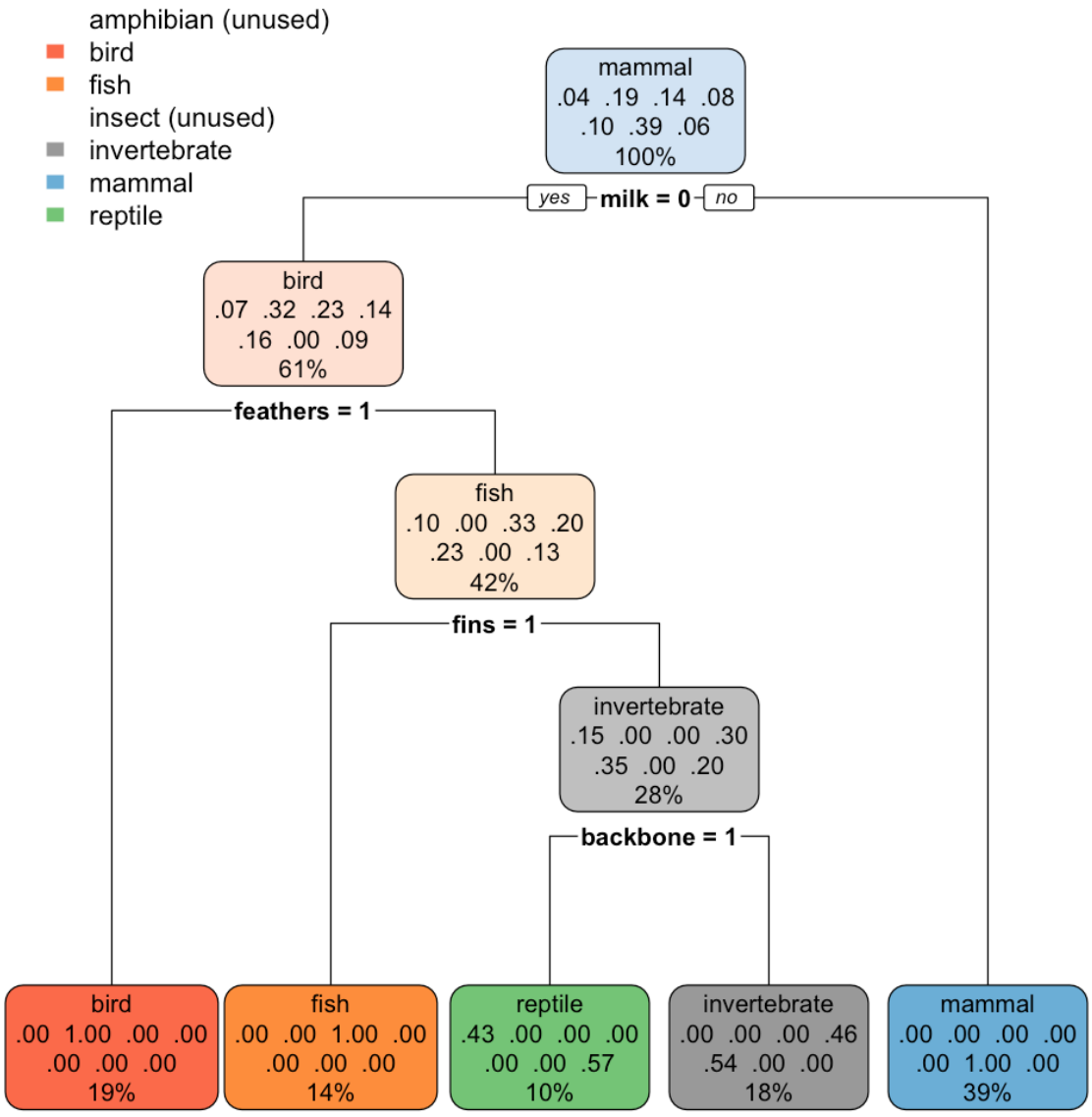
```
> # Decision Tree Analysis using rpart library
> library(rpart)
> library(rpart.plot)
>
> decision_tree_model <- rpart(type ~ ., data = trainingData, method = "class")
> rpart.plot(decision_tree_model)
>
```

```
> printcp(decision_tree_model)

Classification tree:
rpart(formula = type ~ ., data = trainingData, method = "class")

Variables actually used in tree construction:
[1] backbone feathers fins     milk

Root node error: 44/72 = 0.61111

n= 72

        CP nsplit rel error  xerror     xstd
1 0.318182      0   1.00000 1.00000 0.094013
2 0.227273      1   0.68182 0.68182 0.095075
3 0.159091      2   0.45455 0.45455 0.086377
4 0.090909      3   0.29545 0.40909 0.083505
5 0.010000      4   0.20455 0.40909 0.083505
>
```

```
> summary(decision_tree_model)
Call:
rpart(formula = type ~ ., data = trainingData, method = "class")
  n= 72

          CP nsplit rel error    xerror       xstd
1 0.31818182      0 1.0000000 1.0000000 0.09401268
2 0.22727273      1 0.6818182 0.6818182 0.09507500
3 0.15909091      2 0.4545455 0.4545455 0.08637693
4 0.09090909      3 0.2954545 0.4090909 0.08350533
5 0.01000000      4 0.2045455 0.4090909 0.08350533

Variable importance
    milk     eggs     hair  toothed     legs feathers     fins  catsize backbone     tail airborne breathes domestic
      15       14       13       11        9        8        6        6        5        4        4        3        1

Node number 1: 72 observations,    complexity param=0.3181818
  predicted class=mammal        expected loss=0.6111111  P(node) =1
    class counts:     3    14    10     6     7    28     4
   probabilities: 0.042 0.194 0.139 0.083 0.097 0.389 0.056
  left son=2 (44 obs) right son=3 (28 obs)
  Primary splits:
      milk     < 0.5 to the left,  improve=20.69949, (0 missing)
      eggs     < 0.5 to the right, improve=18.49820, (0 missing)
      hair     < 0.5 to the left,  improve=18.29677, (0 missing)
      feathers < 0.5 to the right, improve=14.61015, (0 missing)
      toothed  < 0.5 to the left,  improve=12.92852, (0 missing)
  Surrogate splits:
      eggs    < 0.5 to the right, agree=0.972, adj=0.929, (0 split)
      hair    < 0.5 to the left,  agree=0.958, adj=0.893, (0 split)
      toothed < 0.5 to the left,  agree=0.764, adj=0.393, (0 split)
      catsize < 0.5 to the left,  agree=0.750, adj=0.357, (0 split)
      legs    < 3   to the left,  agree=0.708, adj=0.250, (0 split)

Node number 2: 44 observations,    complexity param=0.2272727
  predicted class=bird          expected loss=0.6818182  P(node) =0.6111111
    class counts:     3    14    10     6     7     0     4
   probabilities: 0.068 0.318 0.227 0.136 0.159 0.000 0.091
  left son=4 (14 obs) right son=5 (30 obs)
  Primary splits:
      feathers < 0.5 to the right, improve=11.772730, (0 missing)
      fins     < 0.5 to the right, improve= 9.772727, (0 missing)
      toothed  < 0.5 to the left,  improve= 8.219156, (0 missing)
      breathes < 0.5 to the right, improve= 8.106061, (0 missing)
      backbone < 0.5 to the right, improve= 7.666028, (0 missing)
  Surrogate splits:
      airborne < 0.5 to the right, agree=0.841, adj=0.500, (0 split)
      domestic < 0.5 to the right, agree=0.705, adj=0.071, (0 split)
      catsize  < 0.5 to the right, agree=0.705, adj=0.071, (0 split)
```

```
Node number 3: 28 observations
  predicted class=mammal        expected loss=0  P(node) =0.3888889
    class counts:     0     0     0     0     0    28     0
   probabilities: 0.000 0.000 0.000 0.000 0.000 1.000 0.000

Node number 4: 14 observations
  predicted class=bird          expected loss=0  P(node) =0.1944444
    class counts:     0    14     0     0     0     0     0
   probabilities: 0.000 1.000 0.000 0.000 0.000 0.000 0.000

Node number 5: 30 observations,    complexity param=0.1590909
  predicted class=fish          expected loss=0.6666667  P(node) =0.4166667
    class counts:     3     0    10     6     7     0     4
   probabilities: 0.100 0.000 0.333 0.200 0.233 0.000 0.133
  left son=10 (10 obs) right son=11 (20 obs)
  Primary splits:
      fins     < 0.5 to the right, improve=8.500000, (0 missing)
      backbone < 0.5 to the right, improve=6.891403, (0 missing)
      toothed  < 0.5 to the right, improve=6.517857, (0 missing)
      tail     < 0.5 to the left,  improve=5.803571, (0 missing)
      breathes < 0.5 to the left,  improve=5.666667, (0 missing)
  Surrogate splits:
      legs     < 2   to the left,  agree=0.867, adj=0.6, (0 split)
      breathes < 0.5 to the left,  agree=0.833, adj=0.5, (0 split)
      toothed  < 0.5 to the right, agree=0.800, adj=0.4, (0 split)
      tail     < 0.5 to the right, agree=0.800, adj=0.4, (0 split)
      backbone < 0.5 to the right, agree=0.767, adj=0.3, (0 split)

Node number 10: 10 observations
  predicted class=fish          expected loss=0  P(node) =0.1388889
    class counts:     0     0    10     0     0     0     0
   probabilities: 0.000 0.000 1.000 0.000 0.000 0.000 0.000

Node number 11: 20 observations,    complexity param=0.09090909
  predicted class=invertebrate  expected loss=0.65  P(node) =0.2777778
    class counts:     3     0     0     6     7     0     4
   probabilities: 0.150 0.000 0.000 0.300 0.350 0.000 0.200
  left son=22 (7 obs) right son=23 (13 obs)
  Primary splits:
      backbone < 0.5 to the right, improve=4.609890, (0 missing)
      aquatic  < 0.5 to the right, improve=3.416667, (0 missing)
      predator < 0.5 to the left,  improve=3.287879, (0 missing)
      legs     < 5   to the right, improve=3.100000, (0 missing)
  Surrogate splits:
      toothed < 0.5 to the right, agree=0.95, adj=0.857, (0 split)
      legs    < 5   to the left,  agree=0.85, adj=0.571, (0 split)
      tail    < 0.5 to the right, agree=0.85, adj=0.571, (0 split)
```

```
Node number 22: 7 observations
  predicted class=reptile       expected loss=0.4285714  P(node) =0.09722222
    class counts:     3     0     0     0     0     0     4
   probabilities: 0.429 0.000 0.000 0.000 0.000 0.000 0.571

Node number 23: 13 observations
  predicted class=invertebrate  expected loss=0.4615385  P(node) =0.1805556
    class counts:     0     0     0     6     7     0     0
   probabilities: 0.000 0.000 0.000 0.462 0.538 0.000 0.000
```



```
> # Creating confusion matrix with predictions
> confusion_matrix <- confusionMatrix(decision_tree_prediction, testingData$type)
> confusion_matrix
Confusion Matrix and Statistics

             Reference
Prediction     amphibian bird fish insect invertebrate mammal reptile
  amphibian            0    0    0      0            0      0       0
  bird                 0    6    0      0            0      0       0
  fish                 0    0    3      0            0      0       0
  insect               0    0    0      0            0      0       0
  invertebrate         0    0    0      2            3      0       0
  mammal               0    0    0      0            0     12       0
  reptile              0    0    0      0            0      0       1

Overall Statistics

               Accuracy : 0.9259
                 95% CI : (0.7571, 0.9909)
    No Information Rate : 0.4444
    P-Value [Acc > NIR] : 1.807e-07

                  Kappa : 0.8969

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: amphibian Class: bird Class: fish Class: insect Class: invertebrate Class: mammal Class: reptile
Sensitivity                        NA      1.0000      1.0000       0.00000              1.0000        1.0000        1.00000
Specificity                         1      1.0000      1.0000       1.00000              0.9167        1.0000        1.00000
Pos Pred Value                     NA      1.0000      1.0000           NaN              0.6000        1.0000        1.00000
Neg Pred Value                     NA      1.0000      1.0000       0.92593              1.0000        1.0000        1.00000
Prevalence                          0      0.2222      0.1111       0.07407              0.1111        0.4444        0.03704
Detection Rate                      0      0.2222      0.1111       0.00000              0.1111        0.4444        0.03704
Detection Prevalence                0      0.2222      0.1111       0.00000              0.1852        0.4444        0.03704
Balanced Accuracy                  NA      1.0000      1.0000       0.50000              0.9583        1.0000        1.00000
```

```
> # Manually calculating accuracy to ensure it's also accurate (quite ironic lol)
> decision_tree_accuracy <- sum(diag(confusion_matrix$table)) / sum(confusion_matrix$table)
> decision_tree_accuracy
[1] 0.9259259
>
```

[Which classes (in the zoo case, families) are often mistaken for each other?] – View the confusion matrix.

The classes/families that are often mistaken for each other are the insect and invertebrate families, which comprise the only errors made by the decision tree model on the testing dataset. We can see from the results below that the decision tree incorrectly classified two insects as invertebrates. In real life, this isn't really an error, as insects are a type of invertebrate. However, for the purposes of this dataset, since insects and invertebrates are listed as distinct attributes/families, this classification is incorrect. Below is the decision tree model's confusion matrix, which highlights the errors.

	Reference						
Prediction	amphibian	bird	fish	insect	invertebrate	mammal	reptile
amphibian	0	0	0	0	0	0	0
bird	0	6	0	0	0	0	0
fish	0	0	3	0	0	0	0
insect	0	0	0	0	0	0	0
invertebrate	0	0	0	2	3	0	0
mammal	0	0	0	0	0	12	0
reptile	0	0	0	0	0	0	1

(B) Experiment with some of the other classifiers (Naïve Bayes, Random Forest) and see if we can get a better classification performance.

[Which classifier had the best performance? Which performance measure(s) support your claim?]

Between the Naïve Bayes and Random Forest classifiers, the classifier with the best performance was undoubtedly the Random Forest. The Random Forest model achieved an outstanding 96.3% accuracy (96.2963% after manual calculation), while the Naïve Bayes model achieved a solid 88.89% accuracy (same percentage after manual calculation). Additionally, the Random Forest's 95% CI is between 81.03% and 99.99%, indicating that the model is near the upper echelon in the metric. On the other hand, the 95% CI for Naïve Bayes is between 70.84% and 97.65%. This indicates that while the Naïve Bayes accuracy is well within this confidence interval (and thus is a strong model), it is not nearly as strong as the Random Forest in this particular metric. Additionally, Cohen's Kappa for the Random Forest is 94.85%, whereas for Naïve Bayes, it is 85.06%. This means that, aside from random guessing, the Random Forest model performs approximately 10% better than the Naïve Bayes model in terms of real agreement between predictions and actual labels. Finally, when examining the confusion matrix for both models, we observe that the Random Forest model makes a single error, incorrectly classifying a fish as a reptile. On the other hand, the Naïve Bayes confusion matrix indicates that it classified two mammals and one reptile as fish. All in all, the Random Forest classifier comes out on top in this race and is the most accurate classifier/algorithm examined in this assignment. Below are the results for both classifiers, as well as the setup and results for each classifier.

Naïve Bayes Performance Results:

```
> naive_bayes_confusion_matrix <- confusionMatrix(naive_bayes_predictions, testingData$type)
> naive_bayes_confusion_matrix
Confusion Matrix and Statistics

             Reference
Prediction     amphibian bird fish insect invertebrate mammal reptile
  amphibian            0    0    0      0            0      0       0
  bird                 0    6    0      0            0      0       0
  fish                 0    0    3      0            0      2       1
  insect               0    0    0      2            0      0       0
  invertebrate         0    0    0      0            3      0       0
  mammal               0    0    0      0            0     10       0
  reptile              0    0    0      0            0      0       0

Overall Statistics

               Accuracy : 0.8889
                 95% CI : (0.7084, 0.9765)
    No Information Rate : 0.4444
    P-Value [Acc > NIR] : 1.95e-06

                  Kappa : 0.8506

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: amphibian Class: bird Class: fish Class: insect Class: invertebrate Class: mammal Class: reptile
Sensitivity                        NA      1.0000      1.0000       1.00000              1.0000        0.8333        0.00000
Specificity                         1      1.0000      0.8750       1.00000              1.0000        1.0000        1.00000
Pos Pred Value                     NA      1.0000      0.5000       1.00000              1.0000        1.0000            NaN
Neg Pred Value                     NA      1.0000      1.0000       1.00000              1.0000        0.8824        0.96296
Prevalence                          0      0.2222      0.1111       0.07407              0.1111        0.4444        0.03704
Detection Rate                      0      0.2222      0.1111       0.07407              0.1111        0.3704        0.00000
Detection Prevalence                0      0.2222      0.2222       0.07407              0.1111        0.3704        0.00000
Balanced Accuracy                  NA      1.0000      0.9375       1.00000              1.0000        0.9167        0.50000
```

Random Forest Performance Results:

```
> random_forest_confusion_matrix <- confusionMatrix(random_forest_predictions, testingData$type)
> random_forest_confusion_matrix
Confusion Matrix and Statistics

             Reference
Prediction     amphibian bird fish insect invertebrate mammal reptile
  amphibian            0    0    0      0            0      0       0
  bird                 0    6    0      0            0      0       0
  fish                 0    0    3      0            0      0       1
  insect               0    0    0      2            0      0       0
  invertebrate         0    0    0      0            3      0       0
  mammal               0    0    0      0            0     12       0
  reptile              0    0    0      0            0      0       0

Overall Statistics

               Accuracy : 0.963
                 95% CI : (0.8103, 0.9991)
    No Information Rate : 0.4444
    P-Value [Acc > NIR] : 1.077e-08

                  Kappa : 0.9485

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: amphibian Class: bird Class: fish Class: insect Class: invertebrate Class: mammal Class: reptile
Sensitivity                        NA      1.0000      1.0000       1.00000              1.0000        1.0000        0.00000
Specificity                         1      1.0000      0.9583       1.00000              1.0000        1.0000        1.00000
Pos Pred Value                     NA      1.0000      0.7500       1.00000              1.0000        1.0000            NaN
Neg Pred Value                     NA      1.0000      1.0000       1.00000              1.0000        1.0000        0.96296
Prevalence                          0      0.2222      0.1111       0.07407              0.1111        0.4444        0.03704
Detection Rate                      0      0.2222      0.1111       0.07407              0.1111        0.4444        0.00000
Detection Prevalence                0      0.2222      0.1481       0.07407              0.1111        0.4444        0.00000
Balanced Accuracy                  NA      1.0000      0.9792       1.00000              1.0000        1.0000        0.50000
```

Naïve Bayes Classification Setup and Results:

```
> # Installing and importing necessary library for Naive Bayes Classification
> install.packages("e1071")
trying URL 'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.5/e1071_1.7-16.tgz'
Content type 'application/x-gzip' length 666194 bytes (650 KB)
==================================================
downloaded 650 KB


The downloaded binary packages are in
        /var/folders/v2/j9qyr40n4p1_tf97m0r491lr0000gn/T//RtmpCoNgmB/downloaded_packages
> library(e1071)
>
```

```
> # Training Naive Bayes Model
> naive_bayes_model <- naiveBayes(type ~ ., data = trainingData)
> naive_bayes_model

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
   amphibian         bird         fish       insect invertebrate       mammal      reptile
  0.04166667   0.19444444   0.13888889   0.08333333   0.09722222   0.38888889   0.05555556

Conditional probabilities:
             hair
Y             FALSE TRUE
  amphibian     1.0  0.0
  bird          1.0  0.0
  fish          1.0  0.0
  insect        0.5  0.5
  invertebrate  1.0  0.0
  mammal        0.0  1.0
  reptile       1.0  0.0

             feathers
Y             FALSE TRUE
  amphibian       1    0
  bird            0    1
  fish            1    0
  insect          1    0
  invertebrate    1    0
  mammal          1    0
  reptile         1    0

             eggs
Y                   FALSE       TRUE
  amphibian    0.00000000 1.00000000
  bird         0.00000000 1.00000000
  fish         0.00000000 1.00000000
  insect       0.00000000 1.00000000
  invertebrate 0.14285714 0.85714286
  mammal       0.96428571 0.03571429
  reptile      0.00000000 1.00000000
```

```
             milk
Y             FALSE TRUE
  amphibian       1    0
  bird            1    0
  fish            1    0
  insect          1    0
  invertebrate    1    0
  mammal          0    1
  reptile         1    0

             airborne
Y                  FALSE       TRUE
  amphibian    1.00000000 0.00000000
  bird         0.21428571 0.78571429
  fish         1.00000000 0.00000000
  insect       0.33333333 0.66666667
  invertebrate 1.00000000 0.00000000
  mammal       0.92857143 0.07142857
  reptile      1.00000000 0.00000000

             aquatic
Y                 FALSE      TRUE
  amphibian    0.0000000 1.0000000
  bird         0.7142857 0.2857143
  fish         0.0000000 1.0000000
  insect       1.0000000 0.0000000
  invertebrate 0.2857143 0.7142857
  mammal       0.8571429 0.1428571
  reptile      1.0000000 0.0000000

             predator
Y                 FALSE      TRUE
  amphibian    0.3333333 0.6666667
  bird         0.5714286 0.4285714
  fish         0.4000000 0.6000000
  insect       1.0000000 0.0000000
  invertebrate 0.1428571 0.8571429
  mammal       0.4642857 0.5357143
  reptile      0.2500000 0.7500000

             toothed
Y                  FALSE       TRUE
  amphibian    0.00000000 1.00000000
  bird         1.00000000 0.00000000
  fish         0.00000000 1.00000000
  insect       1.00000000 0.00000000
  invertebrate 1.00000000 0.00000000
  mammal       0.03571429 0.96428571
  reptile      0.25000000 0.75000000
```

```
              backbone
Y              FALSE TRUE
  amphibian        0    1
  bird             0    1
  fish             0    1
  insect           1    0
  invertebrate     1    0
  mammal           0    1
  reptile          0    1

              breathes
Y                  FALSE      TRUE
  amphibian    0.0000000 1.0000000
  bird         0.0000000 1.0000000
  fish         1.0000000 0.0000000
  insect       0.0000000 1.0000000
  invertebrate 0.7142857 0.2857143
  mammal       0.0000000 1.0000000
  reptile      0.0000000 1.0000000

              venomous
Y                  FALSE      TRUE
  amphibian    0.6666667 0.3333333
  bird         1.0000000 0.0000000
  fish         0.9000000 0.1000000
  insect       0.8333333 0.1666667
  invertebrate 0.7142857 0.2857143
  mammal       1.0000000 0.0000000
  reptile      0.7500000 0.2500000

              fins
Y                   FALSE       TRUE
  amphibian    1.00000000 0.00000000
  bird         1.00000000 0.00000000
  fish         0.00000000 1.00000000
  insect       1.00000000 0.00000000
  invertebrate 1.00000000 0.00000000
  mammal       0.92857143 0.07142857
  reptile      1.00000000 0.00000000

              legs
Y                  [,1]     [,2]
  amphibian    4.000000 0.000000
  bird         2.000000 0.000000
  fish         0.000000 0.000000
  insect       6.000000 0.000000
  invertebrate 4.571429 3.408672
  mammal       3.571429 0.997351
  reptile      2.000000 2.309401
```

```
             tail
Y                 FALSE      TRUE
  amphibian    0.6666667 0.3333333
  bird         0.0000000 1.0000000
  fish         0.0000000 1.0000000
  insect       1.0000000 0.0000000
  invertebrate 0.8571429 0.1428571
  mammal       0.1071429 0.8928571
  reptile      0.0000000 1.0000000

             domestic
Y                 FALSE      TRUE
  amphibian    1.0000000 0.0000000
  bird         0.7857143 0.2142857
  fish         0.9000000 0.1000000
  insect       0.8333333 0.1666667
  invertebrate 1.0000000 0.0000000
  mammal       0.8214286 0.1785714
  reptile      1.0000000 0.0000000

             catsize
Y                 FALSE      TRUE
  amphibian    1.0000000 0.0000000
  bird         0.5714286 0.4285714
  fish         0.7000000 0.3000000
  insect       1.0000000 0.0000000
  invertebrate 0.8571429 0.1428571
  mammal       0.2500000 0.7500000
  reptile      0.7500000 0.2500000
```

```
> # Predict using testing data
> naive_bayes_predictions <- predict(naive_bayes_model, testingData)
> naive_bayes_predictions
 [1] mammal       mammal       fish         fish         invertebrate mammal       fish         bird         mammal
[10] mammal       mammal       bird         bird         insect       bird         fish         mammal       fish
[19] mammal       fish         bird         invertebrate bird         invertebrate mammal       insect       mammal
Levels: amphibian bird fish insect invertebrate mammal reptile
```

```
> naive_bayes_confusion_matrix <- confusionMatrix(naive_bayes_predictions, testingData$type)
> naive_bayes_confusion_matrix
Confusion Matrix and Statistics

             Reference
Prediction     amphibian bird fish insect invertebrate mammal reptile
  amphibian            0    0    0      0            0      0       0
  bird                 0    6    0      0            0      0       0
  fish                 0    0    3      0            0      2       1
  insect               0    0    0      2            0      0       0
  invertebrate         0    0    0      0            3      0       0
  mammal               0    0    0      0            0     10       0
  reptile              0    0    0      0            0      0       0

Overall Statistics

               Accuracy : 0.8889
                 95% CI : (0.7084, 0.9765)
    No Information Rate : 0.4444
    P-Value [Acc > NIR] : 1.95e-06

                  Kappa : 0.8506

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: amphibian Class: bird Class: fish Class: insect Class: invertebrate Class: mammal Class: reptile
Sensitivity                        NA      1.0000      1.0000       1.00000              1.0000        0.8333        0.00000
Specificity                         1      1.0000      0.8750       1.00000              1.0000        1.0000        1.00000
Pos Pred Value                     NA      1.0000      0.5000       1.00000              1.0000        1.0000            NaN
Neg Pred Value                     NA      1.0000      1.0000       1.00000              1.0000        0.8824        0.96296
Prevalence                          0      0.2222      0.1111       0.07407              0.1111        0.4444        0.03704
Detection Rate                      0      0.2222      0.1111       0.07407              0.1111        0.3704        0.00000
Detection Prevalence                0      0.2222      0.2222       0.07407              0.1111        0.3704        0.00000
Balanced Accuracy                  NA      1.0000      0.9375       1.00000              1.0000        0.9167        0.50000
```

```
             Reference
Prediction    amphibian bird fish insect invertebrate mammal reptile
  amphibian           0    0    0      0            0      0       0
  bird                0    6    0      0            0      0       0
  fish                0    0    3      0            0      2       1
  insect              0    0    0      2            0      0       0
  invertebrate        0    0    0      0            3      0       0
  mammal              0    0    0      0            0     10       0
  reptile             0    0    0      0            0      0       0
```

```
> # Verifying Naive Bayes accuracy
> naive_bayes_accuracy <- sum(diag(naive_bayes_confusion_matrix$table)) / sum(naive_bayes_confusion_matrix$table)
> naive_bayes_accuracy
[1] 0.8888889
```

Random Forest Classification Setup and Results:

```
> #Installing randomForest package to run the Random Forest algo
> install.packages("randomForest")
trying URL 'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.5/randomForest_4.7-1.2.tgz'
Content type 'application/x-gzip' length 260492 bytes (254 KB)
==================================================
downloaded 254 KB


The downloaded binary packages are in
        /var/folders/v2/j9qyr40n4p1_tf97m0r491lr0000gn/T//RtmpCoNgmB/downloaded_packages
> library(randomForest)
randomForest 4.7-1.2
Type rfNews() to see new features/changes/bug fixes.

Attaching package: ‘randomForest’

The following object is masked from ‘package:ggplot2’:

    margin
```

```
> # Training Random Forest Model with 200 trees
> set.seed(1)
> random_forest_model <- randomForest(type ~ ., data = trainingData, ntree = 200)
> # Predicting using the Random Forest Algo
> random_forest_predictions <- predict(random_forest_model, testingData)
> random_forest_model

Call:
 randomForest(formula = type ~ ., data = trainingData, ntree = 200)
               Type of random forest: classification
                     Number of trees: 200
No. of variables tried at each split: 4

        OOB estimate of  error rate: 6.94%
Confusion matrix:
            amphibian bird fish insect invertebrate mammal reptile class.error
amphibian           2    0    0      0            0      0       1   0.3333333
bird                0   14    0      0            0      0       0   0.0000000
fish                0    0   10      0            0      0       0   0.0000000
insect              0    0    0      6            0      0       0   0.0000000
invertebrate        0    0    0      2            5      0       0   0.2857143
mammal              0    0    0      0            0     28       0   0.0000000
reptile             1    1    0      0            0      0       2   0.5000000
> random_forest_predictions
          4           6           9          13          14          18          20          22          31
     mammal      mammal        fish        fish invertebrate      mammal      mammal        bird      mammal
         34          35          36          40          41          58          59          63          65
     mammal      mammal        bird        bird      insect        bird        fish      mammal      mammal
         67          75          77          80          82          84          95          96          97
     mammal        fish        bird invertebrate        bird invertebrate      mammal      insect      mammal
Levels: amphibian bird fish insect invertebrate mammal reptile
>
```

```
> random_forest_confusion_matrix <- confusionMatrix(random_forest_predictions, testingData$type)
> random_forest_confusion_matrix
Confusion Matrix and Statistics

             Reference
Prediction    amphibian bird fish insect invertebrate mammal reptile
  amphibian           0    0    0      0            0      0       0
  bird                0    6    0      0            0      0       0
  fish                0    0    3      0            0      0       1
  insect              0    0    0      2            0      0       0
  invertebrate        0    0    0      0            3      0       0
  mammal              0    0    0      0            0     12       0
  reptile             0    0    0      0            0      0       0

Overall Statistics

               Accuracy : 0.963
                 95% CI : (0.8103, 0.9991)
    No Information Rate : 0.4444
    P-Value [Acc > NIR] : 1.077e-08

                  Kappa : 0.9485

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: amphibian Class: bird Class: fish Class: insect Class: invertebrate Class: mammal Class: reptile
Sensitivity                        NA      1.0000      1.0000       1.00000              1.0000        1.0000        0.00000
Specificity                         1      1.0000      0.9583       1.00000              1.0000        1.0000        1.00000
Pos Pred Value                     NA      1.0000      0.7500       1.00000              1.0000        1.0000            NaN
Neg Pred Value                     NA      1.0000      1.0000       1.00000              1.0000        1.0000        0.96296
Prevalence                          0      0.2222      0.1111       0.07407              0.1111        0.4444        0.03704
Detection Rate                      0      0.2222      0.1111       0.07407              0.1111        0.4444        0.00000
Detection Prevalence                0      0.2222      0.1481       0.07407              0.1111        0.4444        0.00000
Balanced Accuracy                  NA      1.0000      0.9792       1.00000              1.0000        1.0000        0.50000
```

```
Confusion Matrix and Statistics

              Reference
Prediction     amphibian bird fish insect invertebrate mammal reptile
  amphibian            0    0    0      0            0      0       0
  bird                 0    6    0      0            0      0       0
  fish                 0    0    3      0            0      0       1
  insect               0    0    0      2            0      0       0
  invertebrate         0    0    0      0            3      0       0
  mammal               0    0    0      0            0     12       0
  reptile              0    0    0      0            0      0       0
```

```
> random_forest_accuracy <- sum(diag(random_forest_confusion_matrix$table)) / sum(random_forest_confusion_matrix$table)
> random_forest_accuracy
[1] 0.962963
```

(C) Analyze with Apriori

[Is any preprocessing needed before running the Apriori algorithm? Hint- for any numerical attributes.]

To perform the Apriori algorithm correctly, several steps needed to be taken with the dataset. First, a copy of the original, cleaned, and preprocessed dataset had to be made and saved as a different data frame. This was done to ensure the reproducibility of the previous results from the decision tree, Naïve Bayes, and Random Forest classifiers. Second, a few attributes that were unnecessary for running Apriori were removed: type and legs. This is because the algorithm's nature is unsupervised; the type attribute only works for supervised classification algorithms, as previously demonstrated. Additionally, the legs attribute is a numeric attribute, and it is the only one in the dataset. The Apriori algorithm expects categorical data in its transaction-like format, which is not well-suited for numerical attributes. Third, each remaining attribute had to be converted from logical to character types, where if a value is true, we replaced it with the name of the attribute. For values that are false, we replace them with NA. These pre-processing steps paved the way for a successful run of the Apriori algorithm. The setup and results are presented below.

```
> #Preparing Dataset for Apriori
> zooDataApriori <- zooData
> install.packages("arules")
trying URL 'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.5/arules_1.7-11.tgz'
Content type 'application/x-gzip' length 2736021 bytes (2.6 MB)
==================================================
downloaded 2.6 MB


The downloaded binary packages are in
        /var/folders/v2/j9qyr40n4p1_tf97m0r491lr0000gn/T//RtmpCoNgmB/downloaded_packages
> library(arules)
```

```
Loading required package: Matrix

Attaching package: ‘arules’

The following objects are masked from ‘package:base’:

    abbreviate, write
```

```
> zooDataApriori <- subset(zooDataApriori, select = -c(type, legs))
>
```

zooData	99 obs. of 17 variables	
zooDataApriori	99 obs. of 15 variables	

```
> # Preparing columns, function to convert all true values to column names, NA if false
> zooDataApriori[] <- lapply(names(zooDataApriori), function(col) {
+     ifelse(zooDataApriori[[col]] == TRUE, col, NA)
+ })
> summary(zooDataApriori)
     hair             feathers            eggs              milk            airborne          aquatic
 Length:99          Length:99          Length:99          Length:99          Length:99          Length:99
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
   predator           toothed            backbone           breathes           venomous             fins
 Length:99          Length:99          Length:99          Length:99          Length:99          Length:99
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
     tail             domestic           catsize
 Length:99          Length:99          Length:99
 Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character
> str(zooDataApriori)
'data.frame':   99 obs. of  15 variables:
 $ hair    : chr  "hair" "hair" NA "hair" ...
 $ feathers: chr  NA NA NA NA ...
 $ eggs    : chr  NA NA "eggs" NA ...
 $ milk    : chr  "milk" "milk" NA "milk" ...
 $ airborne: chr  NA NA NA NA ...
 $ aquatic : chr  NA NA "aquatic" NA ...
 $ predator: chr  "predator" NA "predator" "predator" ...
 $ toothed : chr  "toothed" "toothed" "toothed" "toothed" ...
 $ backbone: chr  "backbone" "backbone" "backbone" "backbone" ...
 $ breathes: chr  "breathes" "breathes" NA "breathes" ...
 $ venomous: chr  NA NA NA NA ...
 $ fins    : chr  NA NA "fins" NA ...
 $ tail    : chr  NA "tail" "tail" NA ...
 $ domestic: chr  NA NA NA NA ...
 $ catsize : chr  "catsize" "catsize" NA "catsize" ...
> head(zooDataApriori)
  hair feathers eggs milk airborne aquatic predator toothed backbone breathes venomous fins tail domestic catsize
1 hair     <NA> <NA> milk     <NA>    <NA> predator toothed backbone breathes     <NA> <NA> <NA>     <NA> catsize
2 hair     <NA> <NA> milk     <NA>    <NA>     <NA> toothed backbone breathes     <NA> <NA> tail     <NA> catsize
3 <NA>     <NA> eggs <NA>     <NA> aquatic predator toothed backbone     <NA>     <NA> fins tail     <NA>    <NA>
4 hair     <NA> <NA> milk     <NA>    <NA> predator toothed backbone breathes     <NA> <NA> <NA>     <NA> catsize
5 hair     <NA> <NA> milk     <NA>    <NA> predator toothed backbone breathes     <NA> <NA> tail     <NA> catsize
6 hair     <NA> <NA> milk     <NA>    <NA>     <NA> toothed backbone breathes     <NA> <NA> tail     <NA> catsize
>
```

```
0 hair     <NA> <NA> milk     <NA>    <NA>     <NA> toothed backbone breathes     <NA> <NA> tail     <NA> catsize
> #Running Apriori, starting with support of 30% and confidence of 60% for first run
> rules <- apriori(zoo_data_transactions, parameter = list(supp = 0.30, conf = 0.60))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.6    0.1    1 none FALSE            TRUE       5     0.3      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 29

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[15 item(s), 99 transaction(s)] done [0.00s].
sorting and recoding items ... [10 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 done [0.00s].
writing ... [254 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> # Displaying Top 10 rules by lift
> rules_sorted_top10_lift <- sort(rules, by = "lift", decreasing = TRUE)
> inspect(rules_sorted_top10_lift[1:10])
     lhs                             rhs    support   confidence coverage  lift  count
[1]  {hair, toothed}              => {milk} 0.3737374 1          0.3737374 2.475 37
[2]  {hair, tail}                 => {milk} 0.3333333 1          0.3333333 2.475 33
[3]  {hair, backbone}             => {milk} 0.3838384 1          0.3838384 2.475 38
[4]  {hair, toothed, breathes}    => {milk} 0.3737374 1          0.3737374 2.475 37
[5]  {hair, toothed, tail}        => {milk} 0.3232323 1          0.3232323 2.475 32
[6]  {hair, toothed, backbone}    => {milk} 0.3737374 1          0.3737374 2.475 37
[7]  {hair, breathes, tail}       => {milk} 0.3333333 1          0.3333333 2.475 33
[8]  {hair, backbone, breathes}   => {milk} 0.3838384 1          0.3838384 2.475 38
[9]  {hair, backbone, tail}       => {milk} 0.3333333 1          0.3333333 2.475 33
[10] {toothed, breathes, catsize} => {milk} 0.3030303 1          0.3030303 2.475 30
```

```
> #Running Apriori, support of 50% and confidence of 80% for second run
> rules <- apriori(zoo_data_transactions, parameter = list(supp = 0.50, conf = 0.80))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5     0.5      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 49

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[15 item(s), 99 transaction(s)] done [0.00s].
sorting and recoding items ... [6 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [13 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> # Displaying Top 10 rules by lift
> rules_sorted_top10_lift <- sort(rules, by = "lift", decreasing = TRUE)
> inspect(rules_sorted_top10_lift[1:10])
     lhs                     rhs        support   confidence coverage  lift     count
[1]  {toothed}            => {backbone} 0.5959596 1.0000000  0.5959596 1.222222 59
[2]  {toothed, tail}      => {backbone} 0.5252525 1.0000000  0.5252525 1.222222 52
[3]  {tail}               => {backbone} 0.7474747 0.9866667  0.7575758 1.205926 74
[4]  {backbone}           => {tail}     0.7474747 0.9135802  0.8181818 1.205926 74
[5]  {breathes, tail}     => {backbone} 0.6060606 0.9836066  0.6161616 1.202186 60
[6]  {backbone, breathes} => {tail}     0.6060606 0.8955224  0.6767677 1.182090 60
[7]  {toothed}            => {tail}     0.5252525 0.8813559  0.5959596 1.163390 52
[8]  {toothed, backbone}  => {tail}     0.5252525 0.8813559  0.5959596 1.163390 52
[9]  {breathes}           => {backbone} 0.6767677 0.8589744  0.7878788 1.049858 67
[10] {backbone}           => {breathes} 0.6767677 0.8271605  0.8181818 1.049858 67
>
```

```
> #Running Apriori, support of 10% and confidence of 70% for third run
> rules <- apriori(zoo_data_transactions, parameter = list(supp = 0.10, conf = 0.70))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.7    0.1    1 none FALSE            TRUE       5     0.1      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 9

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[15 item(s), 99 transaction(s)] done [0.00s].
sorting and recoding items ... [14 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].
writing ... [1196 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> # Displaying Top 10 rules by lift
> rules_sorted_top10_lift <- sort(rules, by = "lift", decreasing = TRUE)
> inspect(rules_sorted_top10_lift[1:10])
     lhs                                          rhs        support   confidence coverage  lift     count
[1]  {eggs, aquatic, toothed, tail}            => {fins}     0.1313131 0.9285714  0.1414141 5.407563 13
[2]  {eggs, aquatic, toothed, backbone, tail}  => {fins}     0.1313131 0.9285714  0.1414141 5.407563 13
[3]  {eggs, airborne, tail}                    => {feathers} 0.1616162 1.0000000  0.1616162 4.950000 16
[4]  {eggs, airborne, backbone}                => {feathers} 0.1616162 1.0000000  0.1616162 4.950000 16
[5]  {eggs, airborne, breathes, tail}          => {feathers} 0.1616162 1.0000000  0.1616162 4.950000 16
[6]  {eggs, airborne, backbone, breathes}      => {feathers} 0.1616162 1.0000000  0.1616162 4.950000 16
[7]  {eggs, airborne, backbone, tail}          => {feathers} 0.1616162 1.0000000  0.1616162 4.950000 16
[8]  {eggs, airborne, backbone, breathes, tail} => {feathers} 0.1616162 1.0000000  0.1616162 4.950000 16
[9]  {aquatic, toothed, tail}                  => {fins}     0.1616162 0.8421053  0.1919192 4.904025 16
[10] {aquatic, toothed, backbone, tail}        => {fins}     0.1616162 0.8421053  0.1919192 4.904025 16
```

(D) Comparing results.

[Which analysis was most useful for the dataset, and why? Include relevant results.]

Among all of the algorithms analyzed in this assignment, the Random Forest classifier was by far the most accurate. The Random Forest classifier achieved the highest accuracy among the three classifiers, with a 96.3% accuracy, while also achieving a Cohen's Kappa value of 94.85%. This indicates that there is near-perfect agreement between predictions and actual labels beyond random guessing. The decision tree (92.6% accuracy, 89.69% Kappa) and Naïve Bayes classifiers (88.89% accuracy, 85.06% Kappa) performed very well in their own right but produced more misclassifications than the Random Forest. No matter how you slice it, the Random Forest model demonstrated superior performance and reliability. This is due to its nature as an ensemble classification algorithm, creating many trees (n = 200 trees in the case of this assignment) and averaging across all of them and combining predictions through majority voting.

The Apriori algorithm, on the other hand, broke sharply from the supervised classifiers with its unsupervised, rule-generating nature. Instead of predicting animal families, it revealed frequent and positively correlated feature-to-feature rules that align with our reality. A rule that caught my attention was two rules tied with the highest amount of lift throughout my three runs: {eggs, aquatic, toothed, tail} => fins, and {eggs, aquatic, toothed, backbone, tail} => fins. These two rules appeared on my third run, with a support of at least 10% and a confidence of at least 70%. They both had a lift of 5.407563, with a support of 13%, and a confidence of 92.857%. These rules indicate that animals that are aquatic, lay eggs, and have teeth and tails almost always possess fins. These rules perfectly describe fish for what they are, which shows the Apriori algorithm captured the essence of an animal family without supervision.

In conclusion, the supervised classification algorithms, especially the Random Forest classifier, were most helpful in predicting the family to which an animal belongs. The Apriori algorithm, on the other hand, was most useful in identifying frequent rules between features and explaining why the aforementioned classifications occur. Combined, the supervised and unsupervised learning algorithms analyzed in this assignment have given us a complete picture of the zoo dataset.

Classification Model Results:

```
> #Creating a table comparing the results of the classification methods
> comparison_table_classification <- data.frame(
+     Model = c("Decision Tree", "Naive Bayes", "Random Forest"),
+     Accuracy = c(decision_tree_accuracy, naive_bayes_accuracy, random_forest_accuracy)
+     )
> print(comparison_table_classification)
          Model  Accuracy
1 Decision Tree 0.9259259
2   Naive Bayes 0.8888889
3 Random Forest 0.9629630
```

Apriori Algorithm Results (3 runs):

```
0 hair      <NA> <NA> milk      <NA>     <NA>      <NA> toothed backbone breathes      <NA> <NA> tail      <NA> catsize
> #Running Apriori, starting with support of 30% and confidence of 60% for first run
> rules <- apriori(zoo_data_transactions, parameter = list(supp = 0.30, conf = 0.60))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.6    0.1    1 none FALSE            TRUE       5     0.3      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 29

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[15 item(s), 99 transaction(s)] done [0.00s].
sorting and recoding items ... [10 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 done [0.00s].
writing ... [254 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> # Displaying Top 10 rules by lift
> rules_sorted_top10_lift <- sort(rules, by = "lift", decreasing = TRUE)
> inspect(rules_sorted_top10_lift[1:10])
     lhs                              rhs    support   confidence coverage  lift  count
[1]  {hair, toothed}                  => {milk} 0.3737374 1          0.3737374 2.475 37
[2]  {hair, tail}                     => {milk} 0.3333333 1          0.3333333 2.475 33
[3]  {hair, backbone}                 => {milk} 0.3838384 1          0.3838384 2.475 38
[4]  {hair, toothed, breathes}        => {milk} 0.3737374 1          0.3737374 2.475 37
[5]  {hair, toothed, tail}            => {milk} 0.3232323 1          0.3232323 2.475 32
[6]  {hair, toothed, backbone}        => {milk} 0.3737374 1          0.3737374 2.475 37
[7]  {hair, breathes, tail}           => {milk} 0.3333333 1          0.3333333 2.475 33
[8]  {hair, backbone, breathes}       => {milk} 0.3838384 1          0.3838384 2.475 38
[9]  {hair, backbone, tail}           => {milk} 0.3333333 1          0.3333333 2.475 33
[10] {toothed, breathes, catsize} => {milk} 0.3030303 1          0.3030303 2.475 30
```

```
> #Running Apriori, support of 50% and confidence of 80% for second run
> rules <- apriori(zoo_data_transactions, parameter = list(supp = 0.50, conf = 0.80))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5     0.5      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 49

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[15 item(s), 99 transaction(s)] done [0.00s].
sorting and recoding items ... [6 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [13 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> # Displaying Top 10 rules by lift
> rules_sorted_top10_lift <- sort(rules, by = "lift", decreasing = TRUE)
> inspect(rules_sorted_top10_lift[1:10])
     lhs                     rhs        support   confidence coverage  lift     count
[1]  {toothed}            => {backbone} 0.5959596 1.0000000  0.5959596 1.222222 59
[2]  {toothed, tail}      => {backbone} 0.5252525 1.0000000  0.5252525 1.222222 52
[3]  {tail}               => {backbone} 0.7474747 0.9866667  0.7575758 1.205926 74
[4]  {backbone}           => {tail}     0.7474747 0.9135802  0.8181818 1.205926 74
[5]  {breathes, tail}     => {backbone} 0.6060606 0.9836066  0.6161616 1.202186 60
[6]  {backbone, breathes} => {tail}     0.6060606 0.8955224  0.6767677 1.182090 60
[7]  {toothed}            => {tail}     0.5252525 0.8813559  0.5959596 1.163390 52
[8]  {toothed, backbone}  => {tail}     0.5252525 0.8813559  0.5959596 1.163390 52
[9]  {breathes}           => {backbone} 0.6767677 0.8589744  0.7878788 1.049858 67
[10] {backbone}           => {breathes} 0.6767677 0.8271605  0.8181818 1.049858 67
>
```

```
> #Running Apriori, support of 10% and confidence of 70% for third run
> rules <- apriori(zoo_data_transactions, parameter = list(supp = 0.10, conf = 0.70))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.7    0.1    1 none FALSE            TRUE       5     0.1      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 9

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[15 item(s), 99 transaction(s)] done [0.00s].
sorting and recoding items ... [14 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].
writing ... [1196 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> # Displaying Top 10 rules by lift
> rules_sorted_top10_lift <- sort(rules, by = "lift", decreasing = TRUE)
> inspect(rules_sorted_top10_lift[1:10])
     lhs                                          rhs        support   confidence coverage  lift     count
[1]  {eggs, aquatic, toothed, tail}            => {fins}     0.1313131 0.9285714  0.1414141 5.407563 13
[2]  {eggs, aquatic, toothed, backbone, tail}  => {fins}     0.1313131 0.9285714  0.1414141 5.407563 13
[3]  {eggs, airborne, tail}                    => {feathers} 0.1616162 1.0000000  0.1616162 4.950000 16
[4]  {eggs, airborne, backbone}                => {feathers} 0.1616162 1.0000000  0.1616162 4.950000 16
[5]  {eggs, airborne, breathes, tail}          => {feathers} 0.1616162 1.0000000  0.1616162 4.950000 16
[6]  {eggs, airborne, backbone, breathes}      => {feathers} 0.1616162 1.0000000  0.1616162 4.950000 16
[7]  {eggs, airborne, backbone, tail}          => {feathers} 0.1616162 1.0000000  0.1616162 4.950000 16
[8]  {eggs, airborne, backbone, breathes, tail} => {feathers} 0.1616162 1.0000000  0.1616162 4.950000 16
[9]  {aquatic, toothed, tail}                  => {fins}     0.1616162 0.8421053  0.1919192 4.904025 16
[10] {aquatic, toothed, backbone, tail}        => {fins}     0.1616162 0.8421053  0.1919192 4.904025 16
```