

hw04

February 26, 2020

#

Naives Bayer & ID3 Animal Classifier

0.0.1 Author & Notes

Skylar Trendley

Data Mining

Homework 4

0.0.2 Concept Description:

Using Naive Bayes and ID3, create a model that can classify unknown animals into the 7 possible classifications.

0.0.3 Data Collection:

The data has been provided by Perry B. Koob. It consists of 19 attributes: a descriptor attribute(name), 15 boolean attributes, a classification attribute (type), and two numerical attributes.

0.0.4 Example Description:

Attribute Information: 1. animal name: Nominal attribute that indicates the animal. Unique for each instance.

2. hair: Nominal boolean attribute that describes if the animal has hair. 3. feathers: Nominal boolean attribute that describes if the animal has feathers. 4. eggs: Nominal boolean attribute that describes if the animal lays eggs.

5. milk: Nominal boolean attribute that describes if the animal produces milk.

6. airborne: Nominal boolean attribute that describes if the animal has the capability of flight.

7. aquatic: Nominal boolean attribute that describes if the animal has the capability to breathe under water.

8. predator: Nominal boolean attribute that describes if the animal is a predator.

9. toothed: Nominal boolean attribute that describes if the animal has teeth.

10. backbone: Nominal boolean attribute that describes if the animal has a backbone.

11. breathes: Nominal boolean attribute that describes if the animal breathes air.

12. venomous: Nominal boolean attribute that describes if the animal produces venom.

13. fins: Nominal boolean attribute that describes if the animal has fins.

14. legs: Numeric classification that describes the number of legs an animal has (set of values: {0,2,4,5,6,8})

- 15. tail: Nominal boolean attribute that describes if the animal has a tail.
- 16. domestic: Nominal boolean attribute that describes if the animal has been domesticated.
- 17. catsize: Nominal boolean attribute that describes if the animal is catsized.
- 18. gestation: Numeric classification that describes if the animal has live birth (in days)
- 19. type: Nominal classification attribute that describes the species of the animal.

0.0.5 Data Import and Wrangling:

```
[53]: train <-read.csv(file = '../src-data/animal-taxonomy-train.
      ↪csv',stringsAsFactors=TRUE)
test <-read.csv(file = '../src-data/animal-taxonomy-test.
      ↪csv',stringsAsFactors=TRUE)

colnames(test)

#Unused attributes
train$legs <- NULL
train$gestation <- NULL
```

- 1. 'animal.name' 2. 'hair' 3. 'feathers' 4. 'eggs' 5. 'milk' 6. 'airborne' 7. 'aquatic' 8. 'predator'
- 9. 'toothed' 10. 'backbone' 11. 'breathes' 12. 'venomous' 13. 'fins' 14. 'legs' 15. 'tail' 16. 'domestic'
- 17. 'catsize' 18. 'gestation' 19. 'type'

0.0.6 Exploratory Data Analysis:

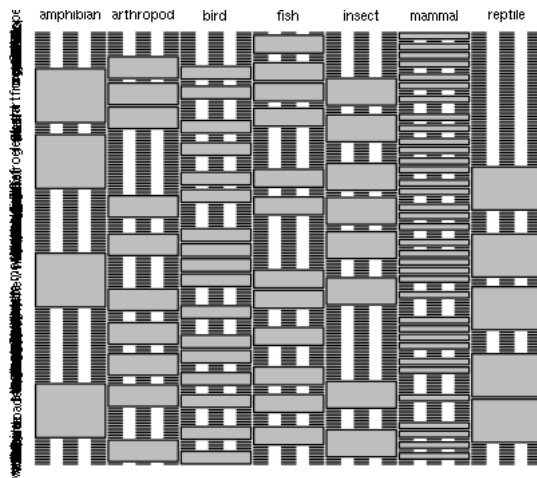
The Naive Bayes and ID3 classifiers use nominal attributes, so each attribute was examined and determined to be nominal or discretized to be nominal.

0.0.7 Mining or Analytics:

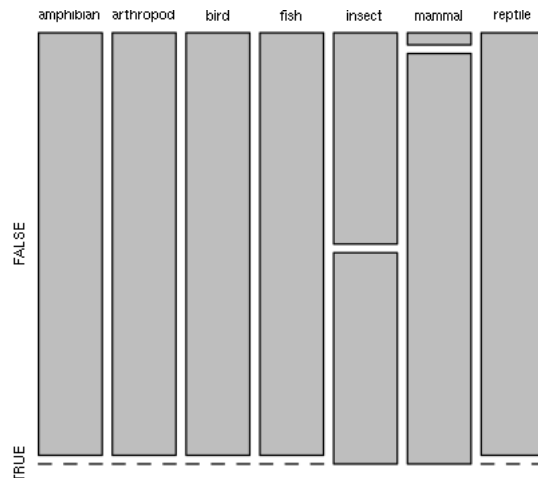
```
[54]: #Naive Bayers Classification
library(klaR) #library necessary for naives bayer classification
nb_mod <- NaiveBayes(type~., data=train)
pred<-predict(nb_mod, test)
```

Warning message in FUN(X[[i]], ...):
 "Numerical 0 probability for all classes with observation 4"

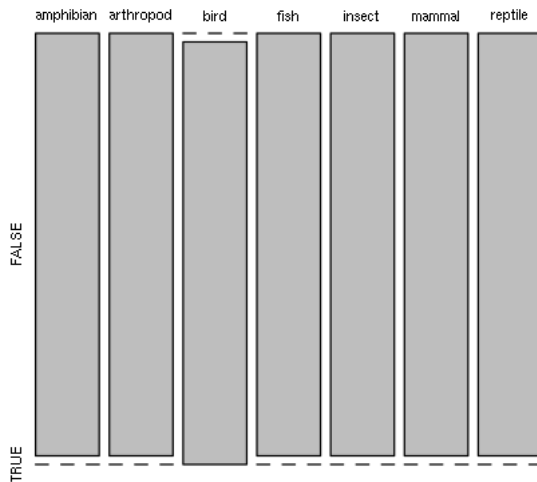
```
[55]: # Plot density of each feature using nb_mod
opar = par(mfrow=c(2, 2), mar=c(4,0,0,0))
plot(nb_mod, main="")
par(opar)
```



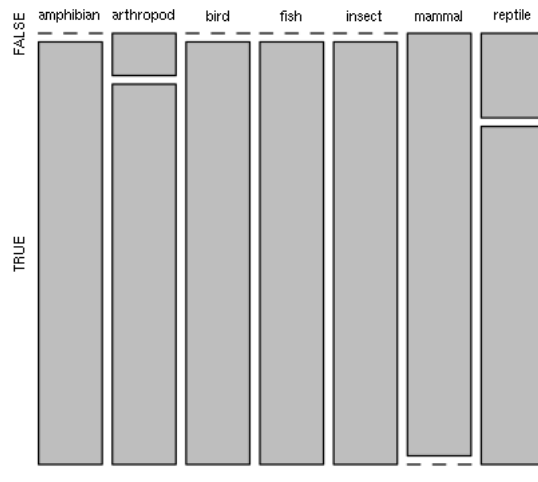
grouping



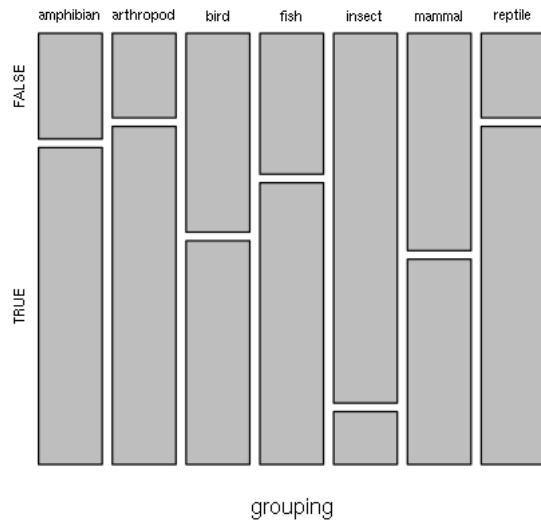
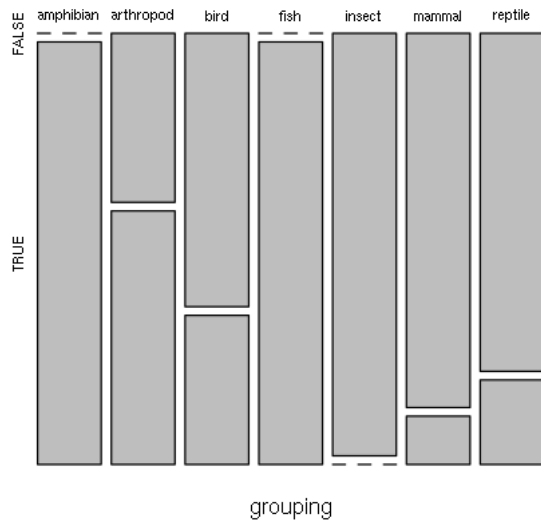
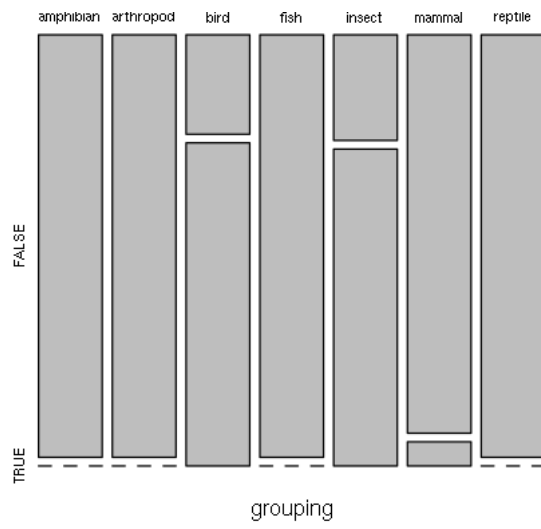
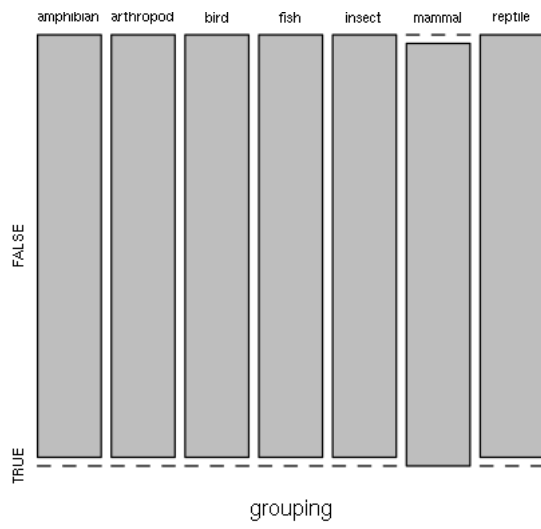
grouping

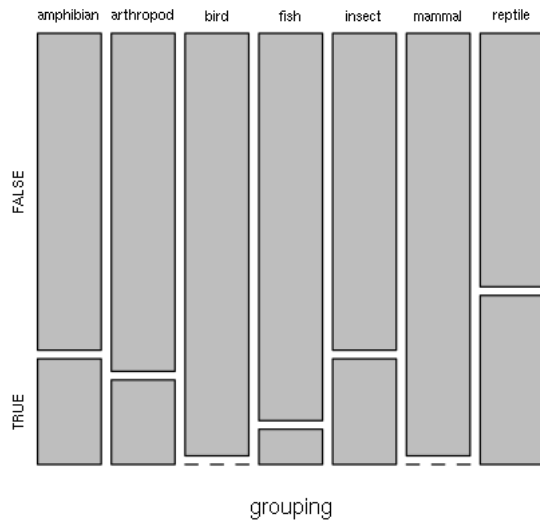
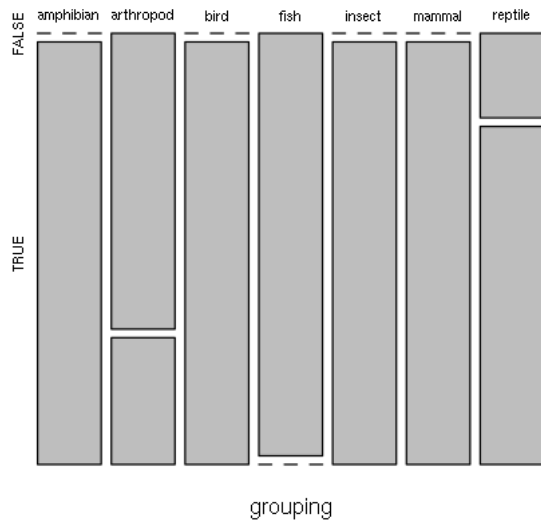
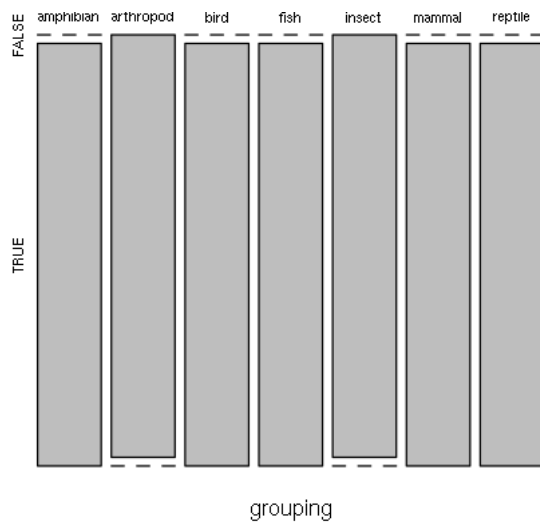
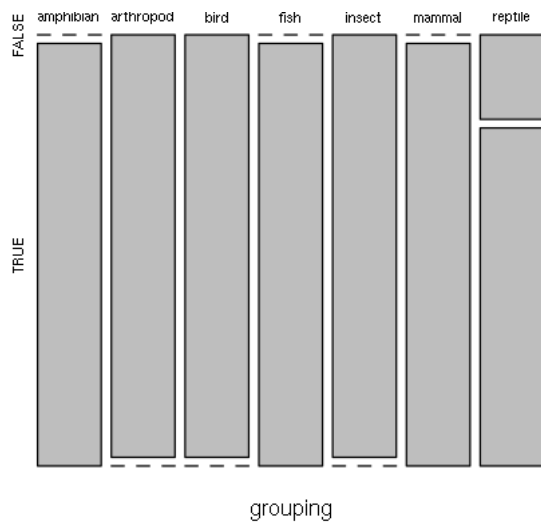


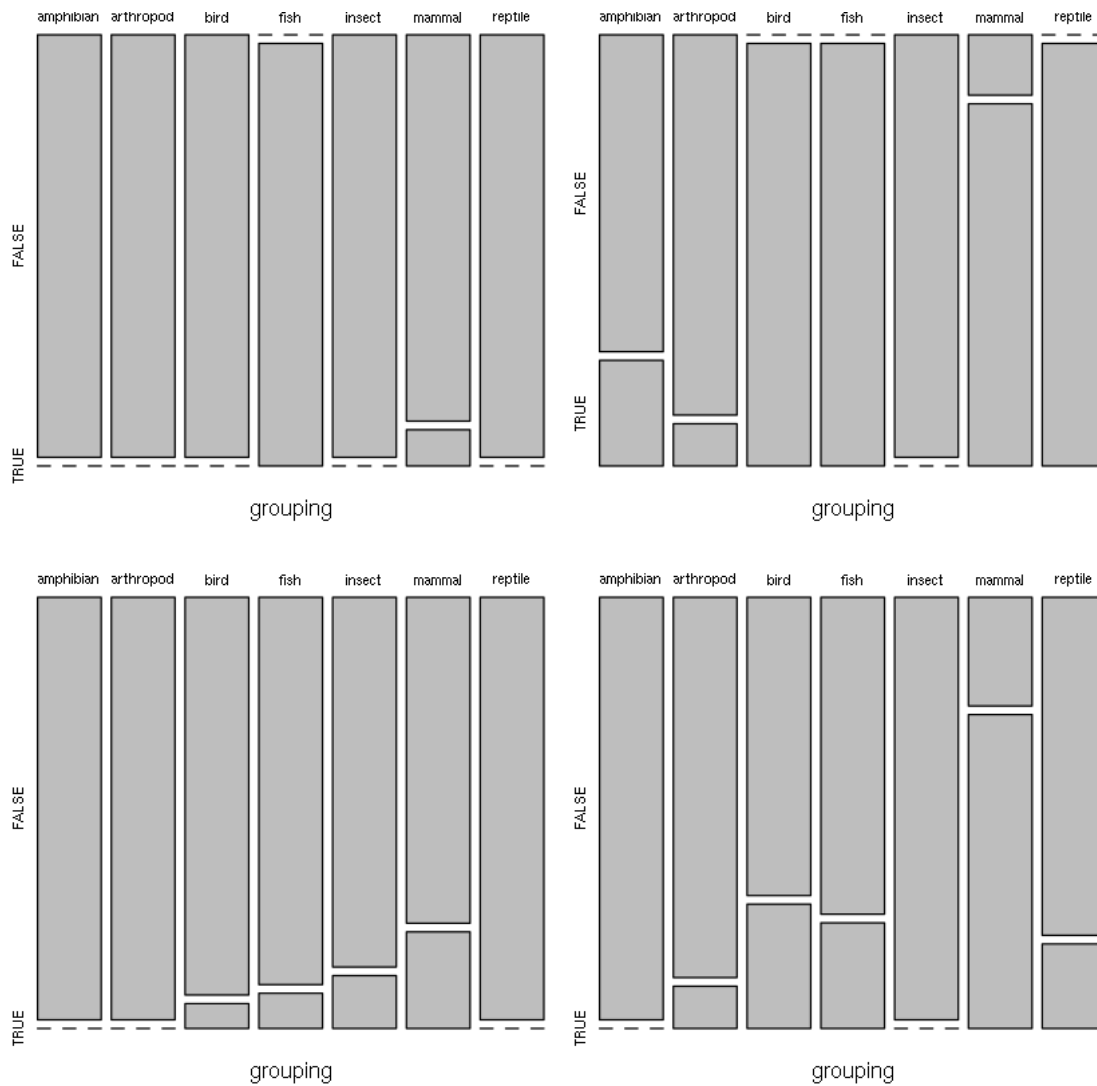
grouping



grouping







```
[62]: #ID3 Classification
library(rpart)
library(rpart.plot)
library(party)
library(caret)

tree = ctree(train$type ~ ., data=train)
predict = predict(tree)

predict
test$type
```

1. mammal 2. fish 3. mammal 4. mammal 5. mammal 6. mammal 7. fish 8. fish 9. mammal 10. mam-

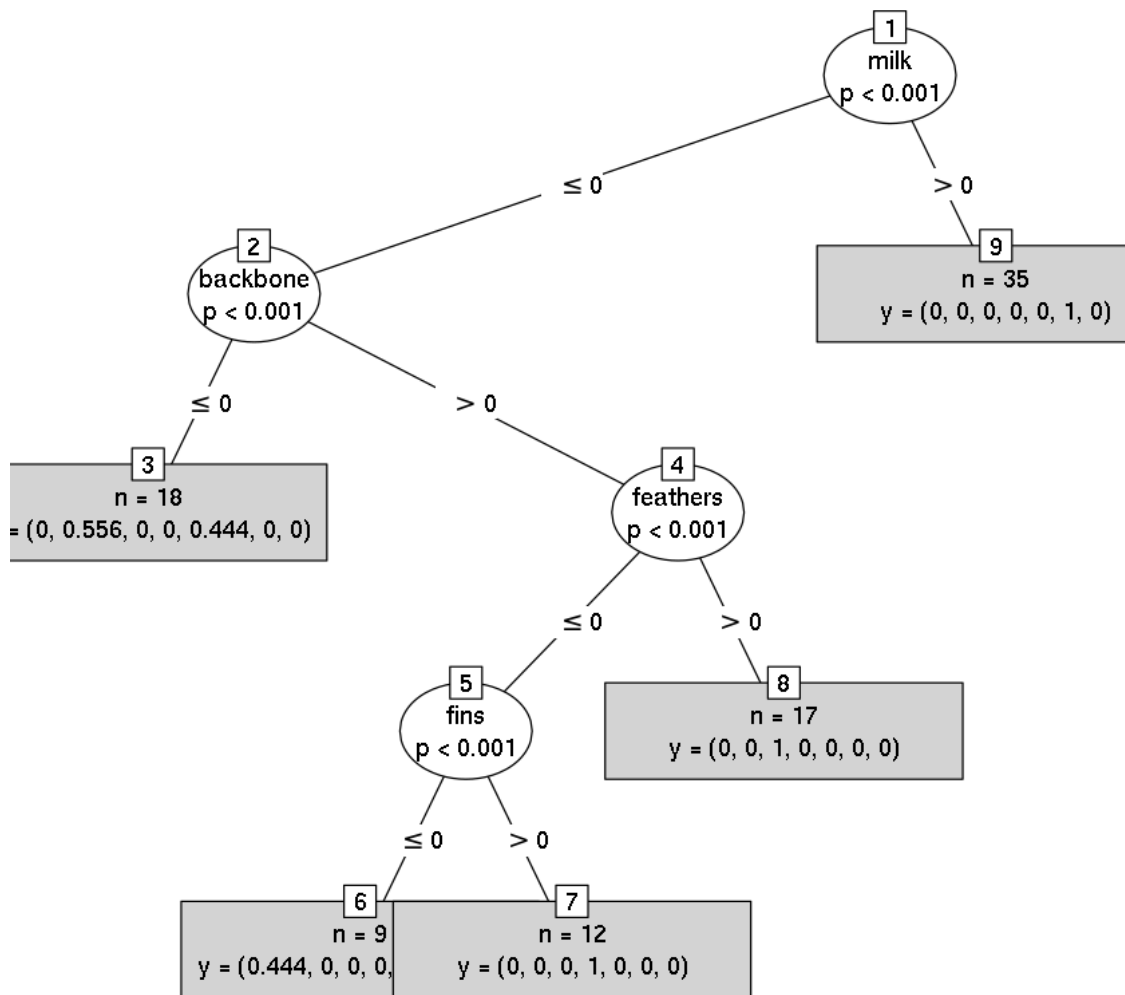
mal 11. fish 12. arthropod 13. arthropod 14. arthropod 15. bird 16. mammal 17. bird 18. mammal 19. arthropod 20. reptile 21. reptile 22. mammal 23. mammal 24. mammal 25. arthropod 26. mammal 27. mammal 28. bird 29. fish 30. mammal 31. mammal 32. bird 33. fish 34. arthropod 35. arthropod 36. bird 37. arthropod 38. bird 39. mammal 40. arthropod 41. mammal 42. mammal 43. mammal 44. mammal 45. arthropod 46. reptile 47. arthropod 48. mammal 49. bird 50. bird 51. bird 52. bird 53. fish 54. fish 55. reptile 56. mammal 57. mammal 58. mammal 59. mammal 60. mammal 61. mammal 62. bird 63. arthropod 64. fish 65. mammal 66. mammal 67. reptile 68. arthropod 69. bird 70. bird 71. reptile 72. arthropod 73. fish 74. bird 75. mammal 76. arthropod 77. fish 78. bird 79. arthropod 80. reptile 81. reptile 82. reptile 83. fish 84. mammal 85. mammal 86. bird 87. mammal 88. arthropod 89. mammal 90. arthropod 91. bird

Levels: 1. 'amphibian' 2. 'arthropod' 3. 'bird' 4. 'fish' 5. 'insect' 6. 'mammal' 7. 'reptile'

1. mammal 2. bird 3. mammal 4. mammal 5. bird 6. bird 7. mammal 8. mammal 9. mammal 10. fish

Levels: 1. 'bird' 2. 'fish' 3. 'mammal'

```
[57]: plot(tree, type="simple")
```



0.0.8 Evaluation

By using the trained data set and a confusion matrix, we can obtain the accuracy and error rate of both models to determine the strength of our model.

```
[58]: #Naives Bayers Confusion Matrix
      confusionMatrix(test$type, pred$class)
```

```
Warning message in levels(reference) != levels(data):
"longer object length is not a multiple of shorter object length"
Warning message in confusionMatrix.default(test$type, pred$class):
"Levels are not in the same order for reference and data. Refactoring data to
match."
```


Confusion Matrix and Statistics

	Reference						
Prediction	amphibian	arthropod	bird	fish	insect	mammal	reptile
amphibian	0	0	0	0	0	0	0
arthropod	0	0	0	0	0	0	0
bird	0	0	3	0	0	0	0
fish	0	0	0	1	0	0	0
insect	0	0	0	0	0	0	0
mammal	0	0	0	0	0	6	0
reptile	0	0	0	0	0	0	0

Overall Statistics

Accuracy : 1
 95% CI : (0.6915, 1)
 No Information Rate : 0.6
 P-Value [Acc > NIR] : 0.006047

Kappa : 1

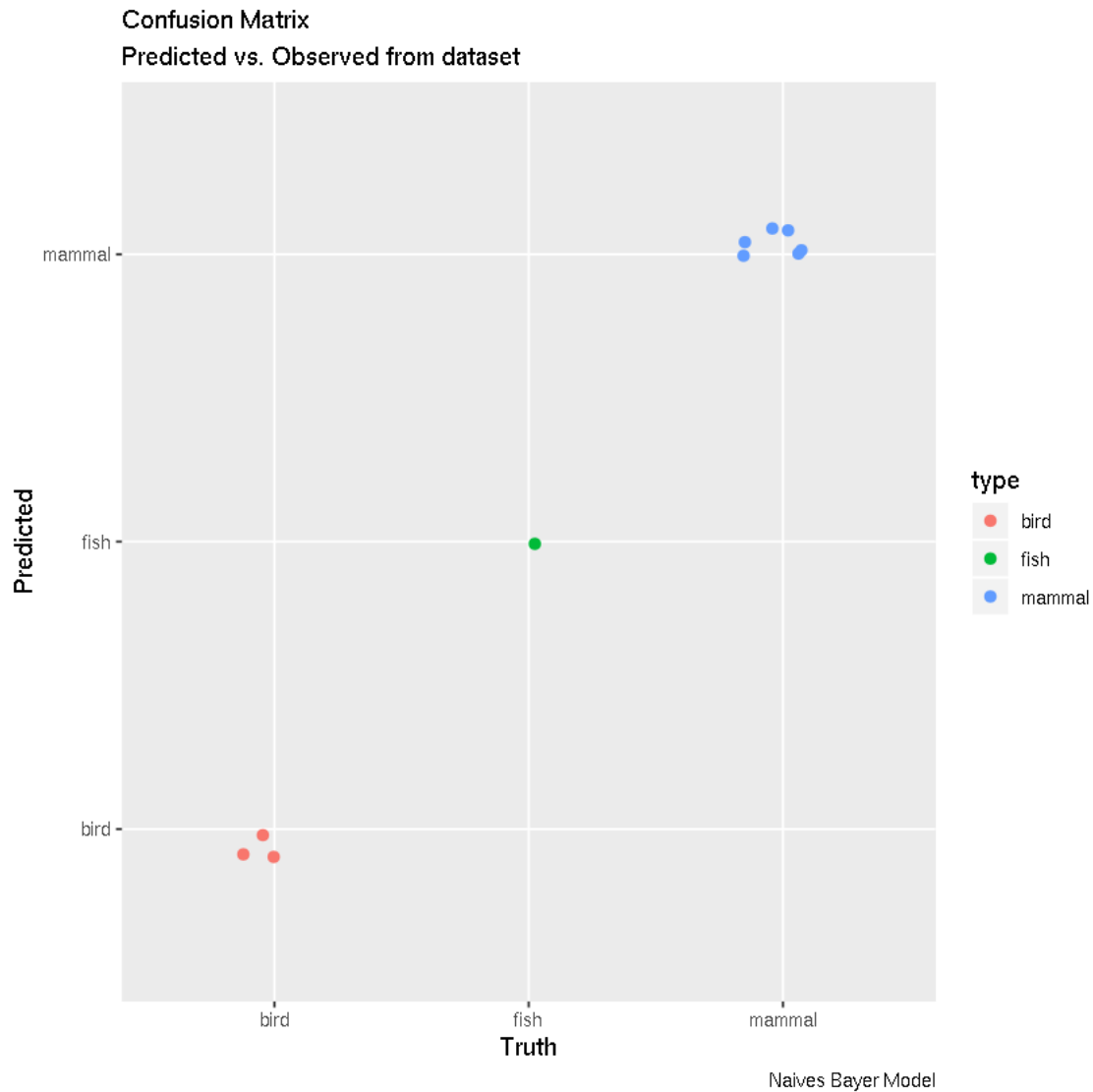
McNemar's Test P-Value : NA

Statistics by Class:

	Class: amphibian	Class: arthropod	Class: bird	Class: fish
Sensitivity	NA	NA	1.0	1.0
Specificity	1	1	1.0	1.0
Pos Pred Value	NA	NA	1.0	1.0
Neg Pred Value	NA	NA	1.0	1.0
Prevalence	0	0	0.3	0.1
Detection Rate	0	0	0.3	0.1
Detection Prevalence	0	0	0.3	0.1
Balanced Accuracy	NA	NA	1.0	1.0
	Class: insect	Class: mammal	Class: reptile	
Sensitivity	NA	1.0	NA	
Specificity	1	1.0	1	
Pos Pred Value	NA	1.0	NA	
Neg Pred Value	NA	1.0	NA	
Prevalence	0	0.6	0	
Detection Rate	0	0.6	0	
Detection Prevalence	0	0.6	0	
Balanced Accuracy	NA	1.0	NA	

```
[59]: library(ggplot2)
      test$pred <- pred$class
      ggplot(test, aes(type, pred, color = type)) +
```

```
geom_jitter(width = 0.2, height = 0.1, size=2) +
labs(title="Confusion Matrix",
      subtitle="Predicted vs. Observed from dataset",
      y="Predicted",
      x="Truth",
      caption="Naives Bayer Model")
```



```
[61]: # ID3 Confusion Matrix
confusionMatrix(test$type, predict)
```

Warning message in levels(reference) != levels(data):
"longer object length is not a multiple of shorter object length"

```
Warning message in confusionMatrix.default(test$type, predict):  
"Levels are not in the same order for reference and data. Refactoring data to  
match."
```

```
Error in table(data, reference, dnn = dnn, ...): all arguments must have  
↳the same length
```

```
Traceback:
```

1. confusionMatrix(test\$type, predict)
2. confusionMatrix.default(test\$type, predict)
3. table(data, reference, dnn = dnn, ...)
4. stop("all arguments must have the same length")

```
[ ]:
```

0.0.9 Results

Through testing each classification method via confusion matrices and testing error rate and accuracy, the advantages and disadvantages of both methods were demonstrated. It seems that Naives Bayer performed better for this particular dataset. The decision tree method typically is better for larger datasets, as it seemed to overfit the data for this particular example. I would therefore recommend the Naives Bayer method to the client.

0.0.10 References

<https://www.machinelearningplus.com/predictive-modeling/how-naive-bayes-algorithm-works-with-example-and-full-code/#8buildinganaivebayesclassifierinr>

<http://ml-tutorials.kyrcha.info/dt.html>

```
[ ]:
```