

hw_3_r

February 25, 2020

#

1R Animal Classifier

0.0.1 Author & Notes

Skylar Trendley

Data Mining

Homework 3

0.0.2 Concept Description:

Train a system to classify animals as either mammal or non-mammal.

0.0.3 Data Collection:

The data has been provided by Perry B. Koob. It consists of 18 attributes: a descriptor attribute(name), 15 boolean attributes, 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)

0.0.5 Data Import and Wrangling:

```
[1]: library(dplyr) #library necessary for %>%rename()
library(OneR) #library necessary for model
library(caret) #library necessary for confusionMatrix
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

intersect, setdiff, setequal, union

Loading required package: lattice

Loading required package: ggplot2

```
[2]: 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'

```
[3]: train$type <- ifelse(train$type == 'mammal', "Mammal", "NotMammal")
train$type <- as.factor(train$type)

test$type <- ifelse(test$type == "mammal", "Mammal", "NotMammal")
```

```
test$type <- as.factor(test$type)
```

```
test
```

A data.frame: 10 × 19

	animal.name	hair	feathers	eggs	milk	airborne	aquatic	predator	too
	<fct>	<lgl>	<lgl>	<lgl>	<lgl>	<lgl>	<lgl>	<lgl>	<lgl>
	dolphin	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
	chicken	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE
	aardvark	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE
	platypus	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE
	flamingo	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE
	dove	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE
	oryx	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE
	lion	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE
	polecat	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE
	dogfish	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE

0.0.6 Exploratory Data Analysis:

The Naive Bayes classifier uses nominal attributes, so each attribute was examined and determined to be nominal or discretized to be nominal.

0.0.7 Mining or Analytics:

```
[4]: prediction <- train
```

```
[5]: #Hair
```

```
hair.count<-as.data.frame(table(train$hair))
hair.true <-as.data.frame(table(train[train$type == "Mammal",]$hair))
hair.false <-as.data.frame(table(train[train$type == "NotMammal",]$hair))

hair.count <-hair.count %>%rename(Label=Var1)
hair.true <-hair.true %>%rename(Label=Var1,Mammal=Freq)
hair.false <-hair.false %>%rename(Label=Var1,NotMammal=Freq)
hair <-merge(hair.count,hair.true, by=Label,all=TRUE)
hair <-merge(hair,hair.false, by=Label,all=TRUE)
hair
```

A data.frame: 2 × 4

	Label	Freq	Mammal	NotMammal
	<fct>	<int>	<int>	<int>
	FALSE	53	1	52
	TRUE	38	34	4

```
[6]: prediction$pred.hair <-NA
prediction$pred.hair[prediction$hair=="TRUE"] <- "Mammal"
prediction$pred.hair[prediction$hair=="FALSE"] <- "NotMammal"
prediction$hair <-NULL
```

```
prediction$pred.hair <-as.factor(prediction$pred.hair)
```

[7]: *#Feathers*

```
feathers.count<-as.data.frame(table(train$feathers))
feathers.true <-as.data.frame(table(train[train$type == "NotMammal",]$feathers))
feathers.false <-as.data.frame(table(train[train$type == "Mammal",]$feathers))

feathers.count <-feathers.count %>%rename(Label=Var1)
feathers.true <-feathers.true %>%rename(Label=Var1,Mammal=Freq)
feathers.false <-feathers.false %>%rename(Label=Var1,NotMammal=Freq)
feathers <-merge(feathers.count,feathers.true, by=Label,all=TRUE)
feathers <-merge(feathers,feathers.false, by=Label,all=TRUE)
feathers
```

A data.frame: 2 × 4

	Label <fct>	Freq <int>	Mammal <int>	NotMammal <int>
	FALSE	74	39	35
	TRUE	17	17	NA

[8]:

```
prediction$pred.feathers <-NA
prediction$pred.feathers[prediction$feathers=="TRUE"] <- "NotMammal"
prediction$pred.feathers[prediction$feathers=="FALSE"] <- "Mammal"
prediction$feathers <-NULL
prediction$pred.feathers <-as.factor(prediction$pred.feathers)
```

[9]: *#Eggs*

```
eggs.count<-as.data.frame(table(train$eggs))
eggs.true <-as.data.frame(table(train[train$type == "NotMammal",]$eggs))
eggs.false <-as.data.frame(table(train[train$type == "Mammal",]$eggs))

eggs.count <-eggs.count %>%rename(Label=Var1)
eggs.true <-eggs.true %>%rename(Label=Var1,Mammal=Freq)
eggs.false <-eggs.false %>%rename(Label=Var1,NotMammal=Freq)
eggs <-merge(eggs.count,eggs.true, by=Label,all=TRUE)
eggs <-merge(eggs,eggs.false, by=Label,all=TRUE)
eggs
```

A data.frame: 2 × 4

	Label <fct>	Freq <int>	Mammal <int>	NotMammal <int>
	FALSE	37	2	35
	TRUE	54	54	NA

```
[10]: prediction$pred.eggs <-NA
prediction$pred.eggs[prediction$eggs=="TRUE"] <- "NotMammal"
prediction$pred.eggs[prediction$eggs=="FALSE"] <- "Mammal"
prediction$eggs <-NULL
prediction$pred.eggs <-as.factor(prediction$pred.eggs)
```

```
[11]: #Milk

milk.count<-as.data.frame(table(train$milk))
milk.true <-as.data.frame(table(train[train$type == "Mammal",]$milk))
milk.false <-as.data.frame(table(train[train$type == "NotMammal",]$milk))

milk.count <-milk.count %>%rename(Label=Var1)
milk.true <-milk.true %>%rename(Label=Var1,Mammal=Freq)
milk.false <-milk.false %>%rename(Label=Var1,NotMammal=Freq)
milk <-merge(milk.count,milk.true, by=Label,all=TRUE)
milk <-merge(milk,milk.false, by=Label,all=TRUE)
milk
```

	Label	Freq	Mammal	NotMammal
	<fct>	<int>	<int>	<int>
A data.frame: 2 × 4	FALSE	56	NA	56
	TRUE	35	35	NA

```
[12]: prediction$pred.milk <-NA
prediction$pred.milk[prediction$milk=="TRUE"] <- "Mammal"
prediction$pred.milk[prediction$milk=="FALSE"] <- "NotMammal"
prediction$milk <-NULL
prediction$pred.milk <-as.factor(prediction$pred.milk)
```

```
[13]: #Airborne

airborne.count<-as.data.frame(table(train$airborne))
airborne.true <-as.data.frame(table(train[train$type == "NotMammal",]$airborne))
airborne.false <-as.data.frame(table(train[train$type == "Mammal",]$airborne))

airborne.count <-airborne.count %>%rename(Label=Var1)
airborne.true <-airborne.true %>%rename(Label=Var1,Mammal=Freq)
airborne.false <-airborne.false %>%rename(Label=Var1,NotMammal=Freq)
airborne <-merge(airborne.count,airborne.true, by=Label,all=TRUE)
airborne <-merge(airborne,airborne.false, by=Label,all=TRUE)
airborne
```

	Label	Freq	Mammal	NotMammal
	<fct>	<int>	<int>	<int>
A data.frame: 2 × 4	FALSE	70	37	33
	TRUE	21	19	2

```
[14]: prediction$pred.airborne <-NA
prediction$pred.airborne[prediction$airborne=="TRUE"] <- "NotMammal"
prediction$pred.airborne[prediction$airborne=="FALSE"] <- "Mammal"
prediction$airborne <-NULL
prediction$pred.airborne <-as.factor(prediction$pred.airborne)
```

```
[15]: #Aquatic

aquatic.count<-as.data.frame(table(train$aquatic))
aquatic.true <-as.data.frame(table(train[train$type == "NotMammal",]$aquatic))
aquatic.false <-as.data.frame(table(train[train$type == "Mammal",]$aquatic))

aquatic.count <-aquatic.count %>%rename(Label=Var1)
aquatic.true <-aquatic.true %>%rename(Label=Var1,Mammal=Freq)
aquatic.false <-aquatic.false %>%rename(Label=Var1,NotMammal=Freq)
aquatic <-merge(aquatic.count,aquatic.true, by=Label,all=TRUE)
aquatic <-merge(aquatic,aquatic.false, by=Label,all=TRUE)
aquatic
```

	Label	Freq	Mammal	NotMammal
	<fct>	<int>	<int>	<int>
A data.frame: 2 × 4	FALSE	58	27	31
	TRUE	33	29	4

```
[16]: prediction$pred.aquatic <-NA
prediction$pred.aquatic[prediction$aquatic=="TRUE"] <- "NotMammal"
prediction$pred.aquatic[prediction$aquatic=="FALSE"] <- "Mammal"
prediction$aquatic <-NULL
prediction$pred.aquatic <-as.factor(prediction$pred.aquatic)
```

```
[17]: #Predator

predator.count<-as.data.frame(table(train$predator))
predator.true <-as.data.frame(table(train[train$type == "Mammal",]$predator))
predator.false <-as.data.frame(table(train[train$type == "
  ↪ "NotMammal",]$predator))

predator.count <-predator.count %>%rename(Label=Var1)
predator.true <-predator.true %>%rename(Label=Var1,Mammal=Freq)
predator.false <-predator.false %>%rename(Label=Var1,NotMammal=Freq)
predator <-merge(predator.count,predator.true, by=Label,all=TRUE)
```

```
predator <-merge(predator,predator.false, bye=Label,all=TRUE)
predator
```

A data.frame: 2 × 4

	Label <fct>	Freq <int>	Mammal <int>	NotMammal <int>
	FALSE	41	18	23
	TRUE	50	17	33

```
[18]: prediction$pred.predator <-NA
prediction$pred.predator[prediction$predator=="TRUE"] <- "Mammal"
prediction$pred.predator[prediction$predator=="FALSE"] <- "NotMammal"
prediction$predator <-NULL
prediction$pred.predator <-as.factor(prediction$pred.predator)
```

```
[19]: #Toothed

toothed.count<-as.data.frame(table(train$toothed))
toothed.true <-as.data.frame(table(train[train$type == "Mammal",]$toothed))
toothed.false <-as.data.frame(table(train[train$type == "NotMammal",]$toothed))

toothed.count <-toothed.count %>%rename(Label=Var1)
toothed.true <-toothed.true %>%rename(Label=Var1,Mammal=Freq)
toothed.false <-toothed.false %>%rename(Label=Var1,NotMammal=Freq)
toothed <-merge(toothed.count,toothed.true, bye=Label,all=TRUE)
toothed <-merge(toothed,toothed.false, bye=Label,all=TRUE)
toothed
```

A data.frame: 2 × 4

	Label <fct>	Freq <int>	Mammal <int>	NotMammal <int>
	FALSE	36	NA	36
	TRUE	55	35	20

```
[20]: prediction$pred.toothed <-NA
prediction$pred.toothed[prediction$toothed=="TRUE"] <- "Mammal"
prediction$pred.toothed[prediction$toothed=="FALSE"] <- "NotMammal"
prediction$toothed <-NULL
prediction$pred.toothed <-as.factor(prediction$pred.toothed)
```

```
[21]: #Backbone

backbone.count<-as.data.frame(table(train$backbone))
backbone.true <-as.data.frame(table(train[train$type == "Mammal",]$backbone))
backbone.false <-as.data.frame(table(train[train$type == "
  ↪ "NotMammal",]$backbone))

backbone.count <-backbone.count %>%rename(Label=Var1)
```

```
backbone.true <-backbone.true %>%rename(Label=Var1,Mammal=Freq)
backbone.false <-backbone.false %>%rename(Label=Var1,NotMammal=Freq)
backbone <-merge(backbone.count,backbone.true, by=Label,all=TRUE)
backbone <-merge(backbone,backbone.false, by=Label,all=TRUE)
backbone
```

A data.frame: 2 × 4

	Label <fct>	Freq <int>	Mammal <int>	NotMammal <int>
	FALSE	18	NA	18
	TRUE	73	35	38

```
[22]: prediction$pred.backbone <-NA
prediction$pred.backbone[prediction$backbone=="TRUE"] <- "Mammal"
prediction$pred.backbone[prediction$backbone=="FALSE"] <- "NotMammal"
prediction$backbone <-NULL
prediction$pred.backbone <-as.factor(prediction$pred.backbone)
```

```
[23]: #Breathes

breathes.count<-as.data.frame(table(train$breathes))
breathes.true <-as.data.frame(table(train[train$type == "Mammal",]$breathes))
breathes.false <-as.data.frame(table(train[train$type ==
  ↳ "NotMammal",]$breathes))

breathes.count <-breathes.count %>%rename(Label=Var1)
breathes.true <-breathes.true %>%rename(Label=Var1,Mammal=Freq)
breathes.false <-breathes.false %>%rename(Label=Var1,NotMammal=Freq)
breathes <-merge(breathes.count,breathes.true, by=Label,all=TRUE)
breathes <-merge(breathes,breathes.false, by=Label,all=TRUE)
breathes
```

A data.frame: 2 × 4

	Label <fct>	Freq <int>	Mammal <int>	NotMammal <int>
	FALSE	20	NA	20
	TRUE	71	35	36

```
[24]: prediction$pred.breathes <-NA
prediction$pred.breathes[prediction$breathes=="TRUE"] <- "Mammal"
prediction$pred.breathes[prediction$breathes=="FALSE"] <- "NotMammal"
prediction$breathes <-NULL
prediction$pred.breathes <-as.factor(prediction$pred.breathes)
```

```
[25]: #Venomous

venomous.count<-as.data.frame(table(train$venomous))
venomous.true <-as.data.frame(table(train[train$type == "NotMammal",]$venomous))
venomous.false <-as.data.frame(table(train[train$type == "Mammal",]$venomous))
```



```

venomous.count <-venomous.count %>%rename(Label=Var1)
venomous.true <-venomous.true %>%rename(Label=Var1,Mammal=Freq)
venomous.false <-venomous.false %>%rename(Label=Var1,NotMammal=Freq)
venomous <-merge(venomous.count,venomous.true, bye=Label,all=TRUE)
venomous <-merge(venomous,venomous.false, bye=Label,all=TRUE)
venomous

```

A data.frame: 2 × 4

	Label <fct>	Freq <int>	Mammal <int>	NotMammal <int>
	FALSE	83	48	35
	TRUE	8	8	NA

```

[26]: prediction$pred.venomous <-NA
prediction$pred.venomous[prediction$venomous=="TRUE"] <- "NotMammal"
prediction$pred.venomous[prediction$venomous=="FALSE"] <- "Mammal"
prediction$venomous <-NULL
prediction$pred.venomous <-as.factor(prediction$pred.venomous)

```

[27]: *#Fins*

```

fins.count<-as.data.frame(table(train$fins))
fins.true <-as.data.frame(table(train[train$type == "NotMammal",]$fins))
fins.false <-as.data.frame(table(train[train$type == "Mammal",]$fins))

fins.count <-fins.count %>%rename(Label=Var1)
fins.true <-fins.true %>%rename(Label=Var1,Mammal=Freq)
fins.false <-fins.false %>%rename(Label=Var1,NotMammal=Freq)
fins <-merge(fins.count,fins.true, bye=Label,all=TRUE)
fins <-merge(fins,fins.false, bye=Label,all=TRUE)
fins

```

A data.frame: 2 × 4

	Label <fct>	Freq <int>	Mammal <int>	NotMammal <int>
	FALSE	76	44	32
	TRUE	15	12	3

```

[28]: prediction$pred.fins <-NA
prediction$pred.fins[prediction$fins=="TRUE"] <- "NotMammal"
prediction$pred.fins[prediction$fins=="FALSE"] <- "Mammal"
prediction$fins <-NULL
prediction$pred.fins <-as.factor(prediction$pred.fins)

```

[29]: *#Tail*

```

tail.count<-as.data.frame(table(train$tail))

```

```

tail.true <-as.data.frame(table(train[train$type == "Mammal",]$tail))
tail.false <-as.data.frame(table(train[train$type == "NotMammal",]$tail))

tail.count <-tail.count %>%rename(Label=Var1)
tail.true <-tail.true %>%rename(Label=Var1,Mammal=Freq)
tail.false <-tail.false %>%rename(Label=Var1,NotMammal=Freq)
tail <-merge(tail.count,tail.true, by=Label,all=TRUE)
tail <-merge(tail,tail.false, by=Label,all=TRUE)
tail

```

A data.frame: 2 × 4

	Label <fct>	Freq <int>	Mammal <int>	NotMammal <int>
FALSE	FALSE	25	5	20
TRUE	TRUE	66	30	36

```

[30]: prediction$pred.tail <-NA
prediction$pred.tail[prediction$tail=="TRUE"] <- "Mammal"
prediction$pred.tail[prediction$tail=="FALSE"] <- "NotMammal"
prediction$tail <-NULL
prediction$pred.tail <-as.factor(prediction$pred.tail)

```

```

[31]: #Domestic

domestic.count<-as.data.frame(table(train$domestic))
domestic.true <-as.data.frame(table(train[train$type == "Mammal",]$domestic))
domestic.false <-as.data.frame(table(train[train$type ==
  ↪ "NotMammal",]$domestic))

domestic.count <-domestic.count %>%rename(Label=Var1)
domestic.true <-domestic.true %>%rename(Label=Var1,Mammal=Freq)
domestic.false <-domestic.false %>%rename(Label=Var1,NotMammal=Freq)
domestic <-merge(domestic.count,domestic.true, by=Label,all=TRUE)
domestic <-merge(domestic,domestic.false, by=Label,all=TRUE)
domestic

```

A data.frame: 2 × 4

	Label <fct>	Freq <int>	Mammal <int>	NotMammal <int>
FALSE	FALSE	80	27	53
TRUE	TRUE	11	8	3

```

[32]: prediction$pred.domestic <-NA
prediction$pred.domestic[prediction$domestic=="TRUE"] <- "Mammal"
prediction$pred.domestic[prediction$domestic=="FALSE"] <- "NotMammal"
prediction$domestic <-NULL
prediction$pred.domestic <-as.factor(prediction$pred.domestic)

```

[33]: *#Catsized*

```
catsize.count<-as.data.frame(table(train$catsize))
catsize.true <-as.data.frame(table(train[train$type == "Mammal",]$catsize))
catsize.false <-as.data.frame(table(train[train$type == "NotMammal",]$catsize))

catsize.count <-catsize.count %>%rename(Label=Var1)
catsize.true <-catsize.true %>%rename(Label=Var1,Mammal=Freq)
catsize.false <-catsize.false %>%rename(Label=Var1,NotMammal=Freq)
catsize <-merge(catsize.count,catsize.true, by=Label,all=TRUE)
catsize <-merge(catsize,catsize.false, by=Label,all=TRUE)
catsize
```

A data.frame: 2 × 4

	Label <fct>	Freq <int>	Mammal <int>	NotMammal <int>
FALSE	FALSE	55	9	46
TRUE	TRUE	36	26	10

[34]:

```
prediction$pred.catsize <-NA
prediction$pred.catsize[prediction$catsize=="TRUE"] <- "Mammal"
prediction$pred.catsize[prediction$catsize=="FALSE"] <- "NotMammal"
prediction$catsize <-NULL
prediction$pred.catsize <-as.factor(prediction$pred.catsize)
```

[35]: *#Error Rate*

```
prediction$pred.hair.correct =ifelse(prediction$type ==prediction$pred.hair,0,1)
prediction$pred.feathers.correct =ifelse(prediction$type ==prediction$pred.
  ↳feathers,0,1)
prediction$pred.eggs.correct =ifelse(prediction$type ==prediction$pred.eggs,0,1)
prediction$pred.milk.correct =ifelse(prediction$type ==prediction$pred.milk,0,1)
prediction$pred.airborne.correct =ifelse(prediction$type ==prediction$pred.
  ↳airborne,0,1)
prediction$pred.aquatic.correct =ifelse(prediction$type ==prediction$pred.
  ↳aquatic,0,1)
prediction$pred.predator.correct =ifelse(prediction$type ==prediction$pred.
  ↳predator,0,1)
prediction$pred.toothed.correct =ifelse(prediction$type ==prediction$pred.
  ↳toothed,0,1)
prediction$pred.backbone.correct =ifelse(prediction$type ==prediction$pred.
  ↳backbone,0,1)
prediction$pred.breathes.correct =ifelse(prediction$type ==prediction$pred.
  ↳breathes,0,1)
prediction$pred.venomous.correct =ifelse(prediction$type ==prediction$pred.
  ↳venomous,0,1)
prediction$pred.fins.correct =ifelse(prediction$type ==prediction$pred.fins,0,1)
```

```

prediction$pred.tail.correct =ifelse(prediction$type ==prediction$pred.tail,0,1)
prediction$pred.domestic.correct =ifelse(prediction$type ==prediction$pred.
  ↪domestic,0,1)
prediction$pred.catsize.correct =ifelse(prediction$type ==prediction$pred.
  ↪catsize,0,1)

#Print the Results
print(paste("Error Rate of Hair Rules:",sum(prediction$pred.hair.correct)/
  ↪nrow(prediction)))
print(paste("Error Rate of Feathers Rules:",sum(prediction$pred.feathers.
  ↪correct)/nrow(prediction)))
print(paste("Error Rate of Eggs Rules:",sum(prediction$pred.eggs.correct)/
  ↪nrow(prediction)))
print(paste("Error Rate of Milk Rules:",sum(prediction$pred.milk.correct)/
  ↪nrow(prediction)))
print(paste("Error Rate of Airborne Rules:",sum(prediction$pred.airborne.
  ↪correct)/nrow(prediction)))
print(paste("Error Rate of Aquatic Rules:",sum(prediction$pred.aquatic.correct)/
  ↪nrow(prediction)))
print(paste("Error Rate of Predator Rules:",sum(prediction$pred.predator.
  ↪correct)/nrow(prediction)))
print(paste("Error Rate of Toothed Rules:",sum(prediction$pred.toothed.correct)/
  ↪nrow(prediction)))
print(paste("Error Rate of Backbone Rules:",sum(prediction$pred.backbone.
  ↪correct)/nrow(prediction)))
print(paste("Error Rate of Breathes Rules:",sum(prediction$pred.breathes.
  ↪correct)/nrow(prediction)))
print(paste("Error Rate of Venomous Rules:",sum(prediction$pred.venomous.
  ↪correct)/nrow(prediction)))
print(paste("Error Rate of Fins Rules:",sum(prediction$pred.fins.correct)/
  ↪nrow(prediction)))
print(paste("Error Rate of Tail Rules:",sum(prediction$pred.tail.correct)/
  ↪nrow(prediction)))
print(paste("Error Rate of Domestic Rules:",sum(prediction$pred.domestic.
  ↪correct)/nrow(prediction)))
print(paste("Error Rate of Catsize Rules:",sum(prediction$pred.catsize.correct)/
  ↪nrow(prediction)))

```

```

[1] "Error Rate of Hair Rules: 0.0549450549450549"
[1] "Error Rate of Feathers Rules: 0.428571428571429"
[1] "Error Rate of Eggs Rules: 0.021978021978022"
[1] "Error Rate of Milk Rules: 0"
[1] "Error Rate of Airborne Rules: 0.428571428571429"
[1] "Error Rate of Aquatic Rules: 0.340659340659341"
[1] "Error Rate of Predator Rules: 0.56043956043956"
[1] "Error Rate of Toothed Rules: 0.21978021978022"

```

```
[1] "Error Rate of Backbone Rules: 0.417582417582418"
[1] "Error Rate of Breathes Rules: 0.395604395604396"
[1] "Error Rate of Venomous Rules: 0.527472527472527"
[1] "Error Rate of Fins Rules: 0.516483516483517"
[1] "Error Rate of Tail Rules: 0.450549450549451"
[1] "Error Rate of Domestic Rules: 0.32967032967033"
[1] "Error Rate of Catsize Rules: 0.208791208791209"
```

We can see that the milk rules have the lowest error rate, so we then choose it as the One Rule.

```
[36]: df_train <-train
      df_train$type <-NULL

      model <-OneR(df_train)
      model_prediction <-predict(model,test)
```

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.

```
[37]: #prepare the list of classes from the test data for evaluation
reference <-as.data.frame(test$type)
colnames(reference) <-c("class")
reference <-as.factor(reference$class)

testing <-as.data.frame(test$hair)
colnames(testing) <- c("milk")

testing$pred[testing$milk=="TRUE"] <- "Mammal"
testing$pred[testing$milk=="FALSE"] <- "NotMammal"

testing <-as.factor(testing$pred)

confusionMatrix(testing, reference)
```

Confusion Matrix and Statistics

	Reference	
Prediction	Mammal	NotMammal
Mammal	5	0
NotMammal	1	4

```

      Accuracy : 0.9
      95% CI : (0.555, 0.9975)
      No Information Rate : 0.6
      P-Value [Acc > NIR] : 0.04636
```

```
      Kappa : 0.8
```

McNemar's Test P-Value : 1.00000

Sensitivity : 0.8333
Specificity : 1.0000
Pos Pred Value : 1.0000
Neg Pred Value : 0.8000
Prevalence : 0.6000
Detection Rate : 0.5000
Detection Prevalence : 0.5000
Balanced Accuracy : 0.9167

'Positive' Class : Mammal

```
[38]: eval_model(model_prediction, test)
```

Confusion matrix (absolute):

	Actual			
Prediction	Mammal	NotMammal	UNSEEN	Sum
Mammal	0	0	0	0
NotMammal	0	0	0	0
UNSEEN	6	4	0	10
Sum	6	4	0	10

Confusion matrix (relative):

	Actual			
Prediction	Mammal	NotMammal	UNSEEN	Sum
Mammal	0.0	0.0	0.0	0.0
NotMammal	0.0	0.0	0.0	0.0
UNSEEN	0.6	0.4	0.0	1.0
Sum	0.6	0.4	0.0	1.0

Accuracy:

0 (0/10)

Error rate:

1 (10/10)

Error rate reduction (vs. base rate):

-1.5 (p-value = 1)

0.0.9 Results

Unfortunately, there was an issue with the `eval_model` that resulted in an extra ‘unseen’ column that I am unsure how to fix. The trend was likely to be that the model had a 91% accuracy rate, meaning it is a relatively strong model. If more time had been spent debugging, I believe the model would have also reflected the 91% accuracy rate, resulting in a final model of:

Milk -> Mammal

No -> Not Mammal

0.0.10 References

<https://mst.instructure.com/courses/44481/files/folder/examples?preview=1961310>

<https://www.rdocumentation.org/packages/OneR/versions/2.2>

<https://cran.r-project.org/web/packages/OneR/index.html>

<https://stackoverflow.com/questions/34705917/conda-how-to-install-r-packages-that-are-not-available-in-r-essentials>

[]: