Taxonomic Classification of Mammals using a Recursive Random Forest Model with Imputation & Oversampling Strategies

Ryan J. Cooper

10/1/2020

Contents

Introduction **2**

Primer on Taxonomy . . . 3

Strengths of CART & Random Forests in Taxonomy . . . 3

Limits & Challenges . . . 4

Model Goals . . . 4

Data Setup **4**

Load Packages . . . 5

Load & Process Mammal Data . . . 5

Vernacular Name Lookup Function for Species Binomials . . . 5

Vernacular Name Index for Orders . . . 5

Data Wrangling & Feature Selection . . . 6

Analysis **6**

Examining Data for Completeness . . . 6

Examining Species Counts . . . 11

Visualization of Trait Distributions . . . 12

Examining Correlations . . . 16

Model Construction **20**

Assessing the Model Scores . . . 20

Create Data Partitions . . . 20

Complete Case Analysis vs. Available Case Analysis . . . 20

Baseline Model . . . 21

Reading the Confusion Matrix Visualization Grid . . . 21

Classification Tree with NA's . . . 22

Random Forests 28
Imputing 0's 29
Oversampling 32
Imputing a Grouped Mean 34
Model Analysis 37

Encapsulation of the Model **37**
Predict Order - Basic Model 37
Predict Genus - Available Case Model 37

Results **38**
Calling the Basic Model 38
Assessing the Basic Model 38
Calling the Advanced Available Case Model 40
Assessing the Advanced Model 43

Conclusion **44**
Performance Tuning 44
Interactive Demonstration 44
Commentary 45
References 45

Introduction

This project was developed for the HarvardX Data Science Professional, Machine Learning Capstone Project.

Data available from the Ecological Society of America (ESA) PanTHERIA dataset contains descriptive attributes concerning over 5,400 individual mammal species. Additional data derived from The Global Biodiversity Information Facility (GBIF) contains extended attributes and details that will support exploration of the Pantheria dataset. These data sources will be used in the construction and training of a machine learning classification model which will try to correctly classify mammals into their most likely taxonomy given a set of identifying *Physical, Behavioral and Reproductive* characteristics.

- Physical characteristics include body mass, head length, and forearm length.
- Behavioral characteristics include population density, diet breadth and trophic level, geographic locale, and activity cycle.
- Reproductive characteristics include longevity, gestation, and weaning ages.

The data set also contains data related to many other characteristics, but many of the columns do not have very complete data. The combination of these various predictors should be very informative about an animals taxonomy. Greater numbers of available predictors to train on generally should increase accuracy of predictions. However, with over 50 possible original columns available, it is necessary to identify and utilize the predictors that are most relevant, and select predictors for which there is a substantial amount of data available.

To demonstrate the concept of a classification and regression tree, we will first implement the RPart package - which enable the visualization of a single decision tree using conditional probability to select the relevant variables and cut off points for each split of the tree. This should give us a good sense of how a single tree would work, and provide some insight into how the model will perform when expanded to an using random forests.

The general formula for conditional probability in a classification and regression tree (CART) is:

$pk(x) = Pr(Y = k|X = x), fork = 1, ..., K$

After demonstration of the basic concept, we will shift to a random forests approach that will go beyond the CART concept by combining many trees, enabling individual trees to work together, providing greater overall accuracy, but with some loss of transparency. Random forests can be surprisingly accurate in multiple outcome classification problems.

Primer on Taxonomy

Biological *taxonomy* is the practice of classification of living organisms by related characteristics. Every organism belongs to a pyramid of taxa starting with the *domain* and *kingdom* at the top, with each successive (lower) level being more specific/descriptive, and generally having fewer species in it. The most specific taxonomic level in general use is *Species.*

The *Species* is expressed as a binomial (two word) title which incorporates the *Genus* into the title. The *Species* name will formally have the first word capitalized and the second word in lower case, which is the word that differentiates this species from others in the same *Genus.*

For example, a blue whale is of the *Genus:* Balaenoptera, *Species: Balaenoptera musculus.* The last word of the species name is also referred to as the *specific epithet.* In text, the species is traditionally italicized. However, for the purpose of this study, species names will not always be italicized.

In some references the *Genus* part of the binomial may be shortened to a single initial, for example, *Balaenoptera musculus* may be shortened to *B. musculus.*

The 8 main levels or *Ranks* of taxonomy are:* Domain > Kingdom > Phylum > Class > Order > Family > Genus > Species

Example Taxonomy of a Blue Whale: Eukarya > Animalia > Chordata > Mammalia > Cetacea > Balaenopteridea > Balaenoptera > musculus

All data in the ESA dataset is concerning animals from the *Class* Mammalia (mammals). The focus of this project will be to generally identify the remaining taxonomic ranks: *Order, Family, or Genus* given a limited number of variables available for prediction.

- Please note, since the advent of widespread genetic testing, some re-organization of taxa has been occurring to better represent *monophyletic groups* or *clades* which share a common ancestor. One example of this is the Order *Soricomorpha*, which appears in the Pantheria dataset, but has been reclassified since. The Pantheria dataset was not selected for relevance and currency - but because its hierarchical structure, sparseness, and imbalanced classes makes it an interesting subject for demonstration of random forest models.

Strengths of CART & Random Forests in Taxonomy

If the purpose of taxonomy is to differentiate species by combinations of unique markers, then a CART / RF approach should work well. Some individual characteristics are very predictive, when classes are very distinct such as the Cetacea - a mammal with a body mass of greater than 100,000,000 grams is *always* going to be a whale.

Other characteristics like the trophic level, terrestriality, diet, or gestational age may be less predictive, unless these characteristics are used in conjunction with other characteristics. A CART / Random Forest approach should reveal which variables are most important, and provide good prediction accuracy by finding the combinations of observations that most efficiently answer the question - which order/family/genus does this mammal belong to?

Limits & Challenges

This model is not intended as a species-level classification tool, as it contains only one row with median values for any given species. A reasonable population of data is needed for any predictor class.

The data is incomplete and any analysis performed may suffer from the *"curse of dimensionality"* - of the characteristics selected, many observations are present on only a portion of the rows. As the number of columns increases, the data becomes relatively sparse.

The number of species per taxonomical order varies greatly, with some orders having only 1 row/species, and some having many individual species. This results in *unbalanced classes*, which may impact the accuracy of the final model. For example, randomly guessing chiroptera (bats) or Rodentia (rats, mice, moles, etc) has a much greater chance of being correct vs guessing an obscure order with only a few species.

Model Goals

Since there are many possible outcomes, it is necessary to frame the problem carefully: What are we trying to accomplish - what are the goals of the model?

- Accuracy: A measure of overall accuracy is one of the goals - we would like a model that predicts the correct order or the correct family most of the time.
- Diversity of Outcomes: We would like a model that will predict a diverse number of orders. Given the presence of many imbalanced classes, we could make a model that predicts Rodentia or Carnivora every time, and it would be correct much of the time. But what good is a model that looks at an elephant and predicts it is a rat or a wolf? We must balance the need to include minority classes. Increasing the selection of minority classes can decrease accuracy - So accuracy and diversity of outcomes are two competing forces in this model.
- Balance: We would like a model that balances precision and recall. The F1 score is a good measure of this. We will record the average of the balanced F1 score for predicted class. If 8 classes are predicted there will be 8 F1 scores. The mean of those scores will be recorded and considered during model analysis.
- Recursion: Given that there are so many families - the number of possible *family* or *genus* outcomes is too great without segmentation or recursion. We can determine the most likely order, then filter the families and re-run the model using a more focused set of data. This could be done again to predict at the genus level as well.
- Flexibility: We would like a classification model that can be flexible and take a range of inputs with as little or as much data as is available. This may necessitate a more complex approach to modeling that can adapt to a variable number of inputs, and handle missing data effectively.

Data Setup

In this section we will load data, libraries, and packages, and set up the data for analysis.

Load Packages

In this section, we will load all packages and external libraries used by this project. This project uses several key utility packages including GGPlot2 for visualizations, dplyr for data manipulation, and the caret package for various functions. The randomforest and rpart packages contain key features used in the model construction and testing process.

Load & Process Mammal Data

In this section we will preprocess the data from the ESA dataset. The ESA data contains over 50 different variables. For the purposes of this study, we're selecting a subset of variables where observations are present for 800+ rows. The variables selected will include the taxonomic data and 12 predictor physical and behavioral characteristics.

The columns with enough observations will be copied into a new dataset and renamed for easier readability, and the data types and values will be cleaned up to work correctly with R.

There are 5416 total original rows of data in the Pantheria file.

Vernacular Name Lookup Function for Species Binomials

Here we set up a lookup function to find the vernacular / common name for a Species binomial.

lookup
The common name for Balaenoptera musculus is Blue Whale
The common name for Rattus rattus is Black Rat
The common name for Canis lupus is Gray Wolf

This function has returned the correct common names of the various species binomials we have passed in.

Vernacular Name Index for Orders

We will also set up an index of common names for the taxonomical orders in the Mammalia family. Again, this is done to make the exploration of the data easier to understand.

taxo_order	taxo_order_desc
Afrosoricida	Tenrecs
Artiodactyla	Even-toed ungulates
Carnivora	Bears & Wolves
Cetacea	Whales & Dolphins
Chiroptera	Bats
Cingulata	Armadillos & Anteaters
Dasyuromorphia	Carnivorous Marsupials
Dermoptera	Flying Lemurs
Didelphimorphia	Opossums
Diprotodontia	Herbivorous Marsupials

Data Wrangling & Feature Selection

In this section, we are mutating the data to be more friendly to work with. We will drop some of the fields that are not going to be used by the model.

Analysis

In this section we will analyze the Pantheria data to thoroughly understand the distribution, completeness, and other properties of the data.

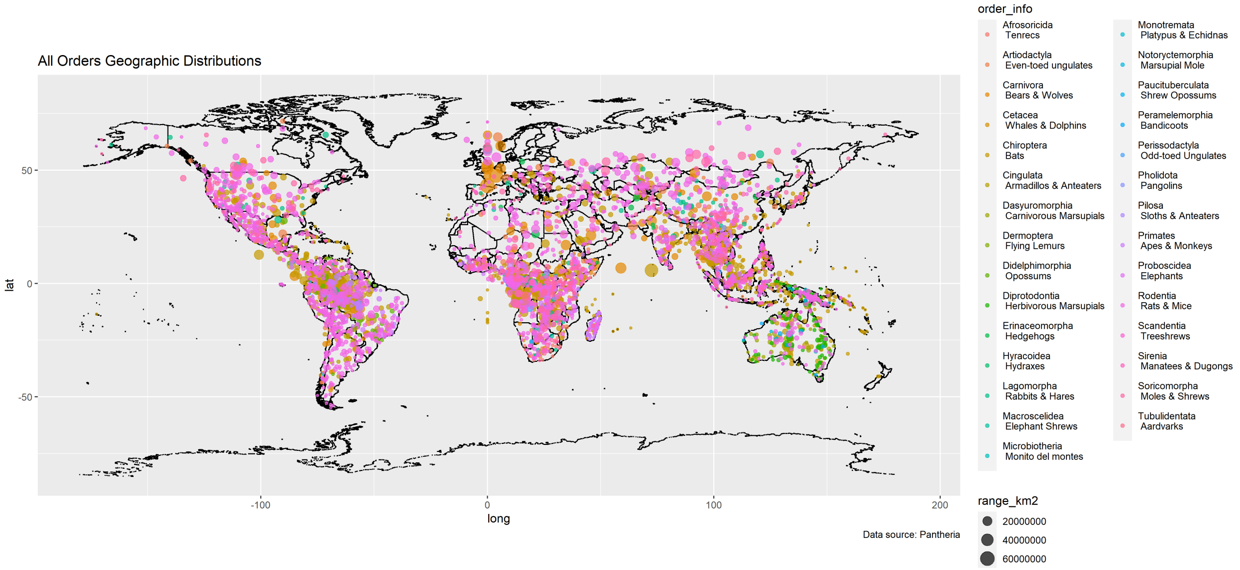

All Orders Geographic Distributions
lat
50
0
-50
long
-100
0
100
200
Data source: Pantheria
order_info
Afrosoricida Tenrecs
Artiodactyla Even-toed ungulates
Carnivora Bears & Wolves
Cetacea Whales & Dolphins
Chiroptera Bats
Cingulata Armadillos & Anteaters
Dasyuromorphia Carnivorous Marsupials
Dermoptera Flying Lemurs
Didelphimorphia Opossums
Diprotodontia Herbivorous Marsupials
Erinaceomorpha Hedgehogs
Hyracoidea Hydraxes
Lagomorpha Rabbits & Hares
Macroscelidea Elephant Shrews
Microbiotheria Monito del montes
Monotremata Platypus & Echidnas
Notoryctemorphia Marsupial Mole
Paucituberculata Shrew Opossums
Peramelemorphia Bandicoots
Perissodactyla Odd-toed Ungulates
Pholidota Pangolins
Pilosa Sloths & Anteaters
Primates Apes & Monkeys
Proboscidea Elephants
Rodentia Rats & Mice
Scandentia Treeshrews
Sirenia Manatees & Dugongs
Soricomorpha Moles & Shrews
Tubulidentata Aardvarks
range_km2
20000000
40000000
60000000

Figure 1: world map

This map shows the vast scale of the Pantheria database - it contains data related to thousands of species from all over the world. The dataset contains a latitude/longitude (geocode) for the center point of each species' known range, which have been plotted on these maps.

Some orders like Diprotodontia (Marsupials) are found only in Australia - Dermoptera (Flying Lemus) in Indonesia & the Phillipines, Proboscidea (Elephants) in Africa and India & Tubulidentata (Aardvarks) which occur only in Sub Saharan africa.

On the other hand, some species like Chiroptera (Bats) Carnivora (Meat Eaters), Rodentia (Rats & Mice), and Lagomorpha (Rabbits) are found all over the globe. Cetacea (Whales) and Sirenia (Manatees & Dugongs) have no geocodes or range data recorded, presumably because they are marine mammals.

Examining Data for Completeness

Now that we've created our "tiny zoo" data set, we will analyze the details, examine the data for completeness, and try to determine what Machine Learning methods would be most appropriate to produce the most accurate predictions. The data used in this project comes from a scientific database which was derived from many sources.

The original source of each data reference may be found by accessing the metadata link in the references section and finding the corresponding numeric codes. The values provided by Pantheria are already highly

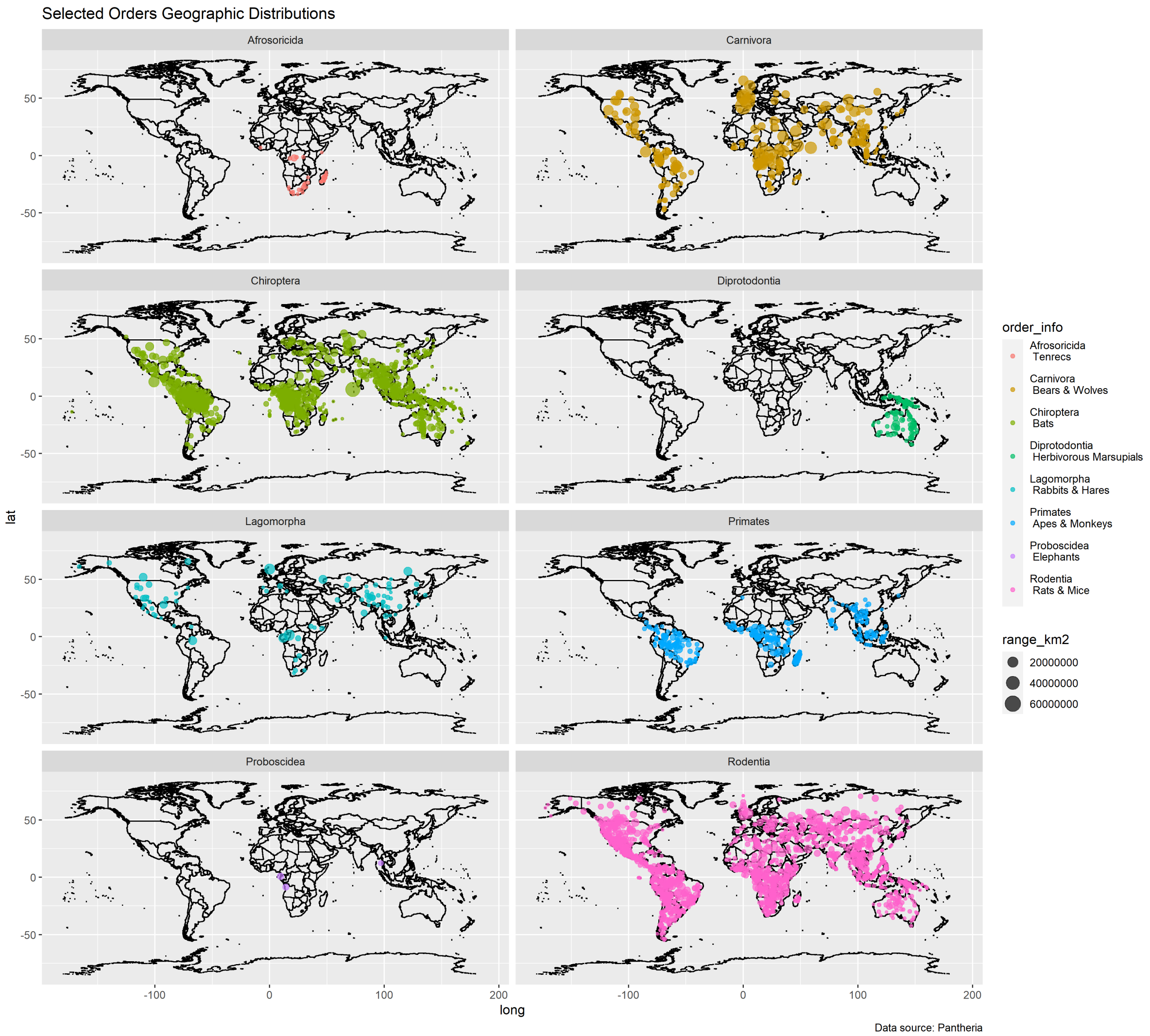
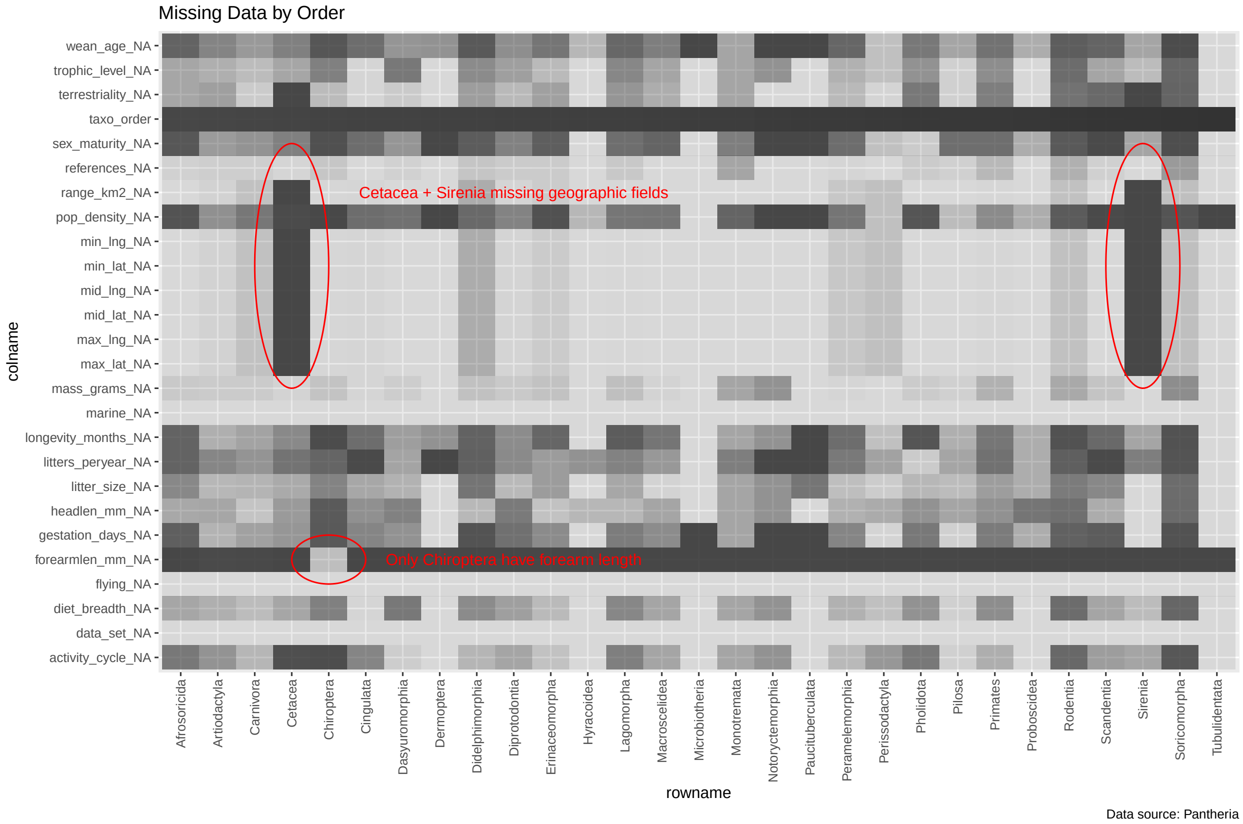


Figure 2: region map

reduced and, in some cases, the values are the product of a regression model. The machine learning model we are building already has a lot of the data collection work “baked in” - and we have just one row per species.



Data source: Pantheria

The dataset contains 5416 total rows with columns containing the following data points:

Commonly recorded attributes:

- headlen_mm - Head & body length in mm
- mass_grams - Mass in grams
- litter_size - Number of offspring per litter
- litters_peryear - Number of litters per pear
- gestation_days - Length of gestational period in days
- longevity_months - Lifespan in months
- diet_breadth - Number of types of food sources
- pop_density - Number of individuals per square km
- trophic_level - Level in the food chain
- sex_maturity - Days to maturity
- wean_age - Days to weaning
- activity cycle - Diurnal (active during the day) / Nocturnal (active at night) etc.

Chiroptera (Bats) only:

- forearmlen_mm - Length of forearm (wing) - a key indicator in bats

Land dwellers only have geo coordinates & terrestriality:

- terrestriality - Above ground/under ground dwelling

- min_lat - minimum latitude
- max_lat - maximum latitude
- min_lng - minimum longitude
- max_lng - maximum longitude
- mid_lat - mid range latitude
- mid_lng - mid range longitude
- range_km2 - mid range longitude

The column with the fewest missing entries is body mass. Aside from the geocodes and forearm length, the column with the most missing entries is population density.

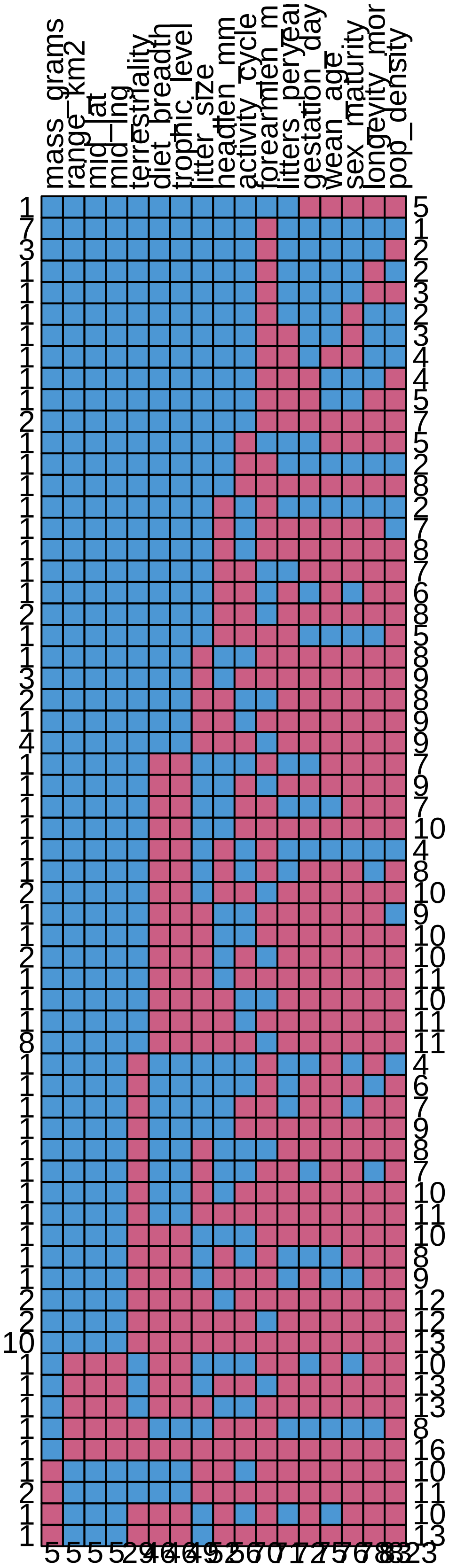
The patterns of missingness may be influenced by differences in the data collection processes for various traits. It is probably easier to ascertain certain physical values like mass (which can be measured fairly instantly) vs something like population density or longevity - which requires observation of a group, long periods of time, or complex studies to observe and record.

It also makes sense that certain characteristics would not be as widely recorded for orders where it is not easily measured or not as useful, or not applicable. In particuler - flying mammals Chiroptera (Bats) and marine mammals - Cetacea (Whales) and Sirenia (Manatees & Dugongs) have some key differences in which columns are commonly recorded.

Only 269 species have all 12 common characteristics recorded. Considering the relatively small number of *complete rows* in the Pantheria data, a *complete case analysis* would be of limited use. This approach could only incorporate a small fraction of the total observations, and the number of possible outcome classifications would include very few outcomes. This method would also require more complete data to be provided to make a prediction.

We will remove any rows that are missing all of the 12 common predictors. The original dataset contains 1278 empty rows that will be removed.

Now we will examine the pattern of missingness - are most rows missing the same columns?


mass_grams
range_km2
mid_lat
mid_lng
terrestriality
diet_breadth
trophic_level
litter_size
head_len_mm
activity_cycle
forearm_len_m
litters_peryear
gestation_days
wean_age
sex_maturity
longevity_mon
pop_density
1
7
3
1
1
1
1
1
1
1
2
1
1
1
1
1
1
1
1
2
1
1
3
2
1
4
1
1
1
1
1
1
2
1
1
2
1
1
1
8
1
1
1
1
1
1
1
1
1
1
1
2
2
10
1
1
1
1
1
1
2
1
1
5
1
2
2
3
2
3
4
4
5
7
5
2
8
2
7
8
7
6
8
5
8
9
8
9
9
7
9
7
10
4
8
10
9
10
10
11
10
11
11
4
6
7
9
8
7
10
11
10
8
9
12
12
13
10
13
13
8
16
10
11
10
13
5
5
5
5
29
46
46
49
52
56
70
71
72
75
76
78
83
823

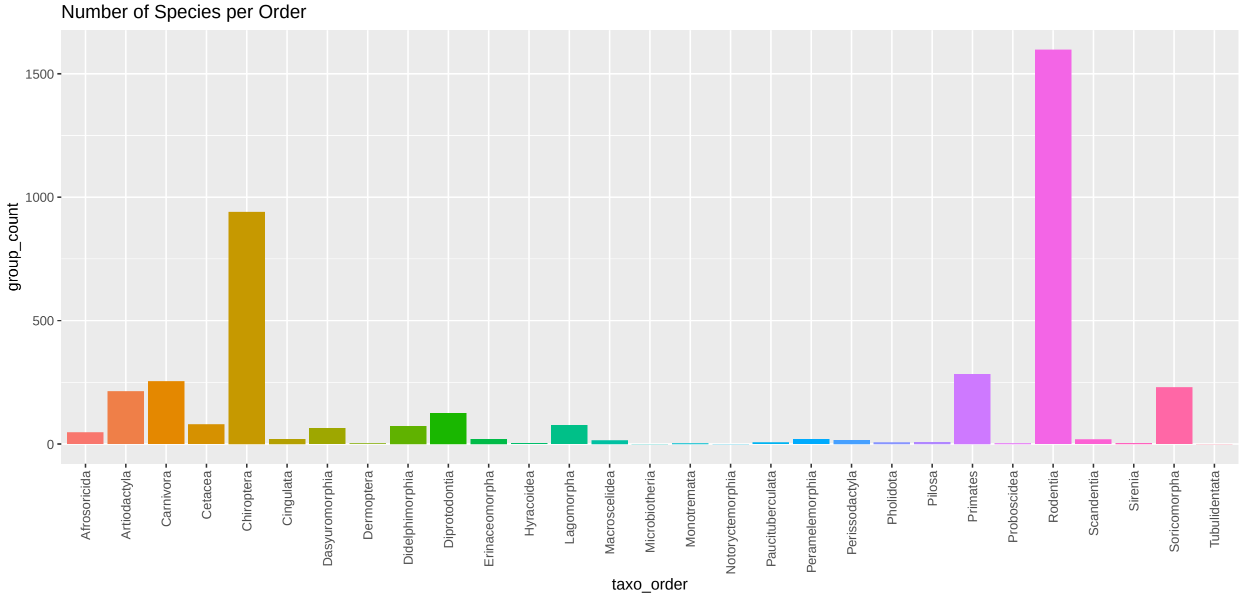
Selecting just 100 random rows, there are many missing values (red), and many different patterns of completeness (blue) in data. The number on the left is the number of rows with that pattern of missingness. The number on the right is how many columns are missing.

Rather than omitting observations with missing data, we will attempt to either replace NA values with other values using various imputation strategies, or to shape a model which trains using only the available data columns. Values may be replaced with a 0 or an estimated value.

Since taxonomical classification attempts to group similar organisms, it should be reasonable to assume that a particular species characteristics (if missing) could be imputed based on an average of other species in the same genus, family, or order. Imputation of missing values could improve the number of possible outcomes by simulating data points that were not recorded for any individual species within a given taxa. We will experiment with how well the system performs with various imputation methods applied.

Examining Species Counts

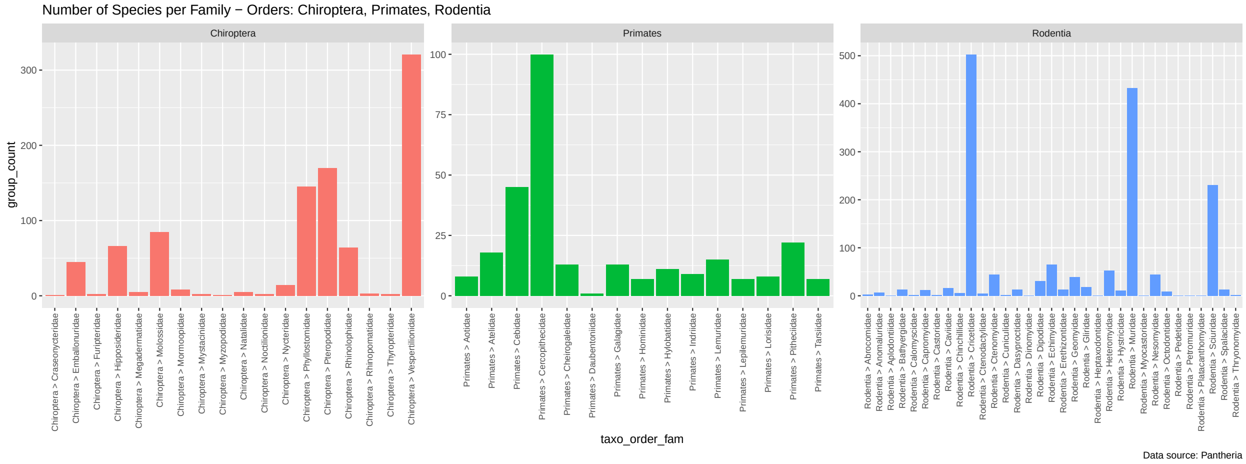
The charts below will examine the count, distribution and ranges of values in our data. In this section we will consider the number of species per order.



Data source: Pantheria

The median species count is 20 species per order. We will consider any groups with fewer than the median group count, to be minority classes. This will be the basis for oversampling the training set. Of 29 orders in the data, 14 could be considered minority groups.

We are now performing the same analysis as above, but at the family level. Since there are so many outcomes, we will need to filter this chart to a subset of orders. We will just examine the 3 largest orders - Rodentia, Chiroptera and Primates.

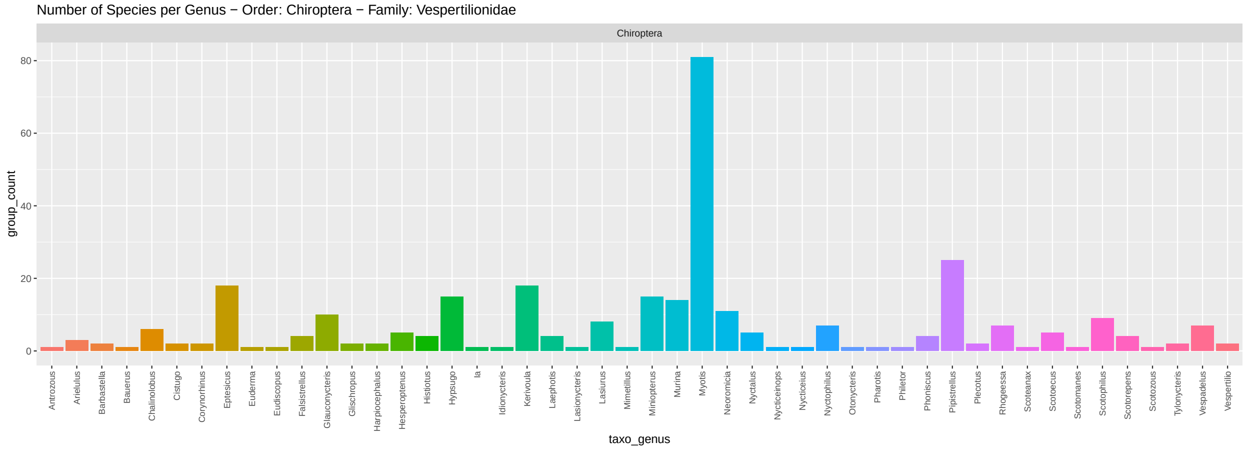


Data source: Pantheria

Clearly the classes are very imbalanced. Some have one or two species, and others have hundreds.

The median number of families per order across all species is: 7. We may also use this detail as the basis for some imputation strategies to be explored later in this report.

We are now performing the same analysis as above, but at the genus level.



Data source: Pantheria

Once again we can see that there is significant imbalance in the number of genus groups per family.

Visualization of Trait Distributions

In the following plots we will examine the taxonomic data to visualize how the distributions vary between each order. First we will examine some of the raw data, to make sure the values agree with general knowledge.

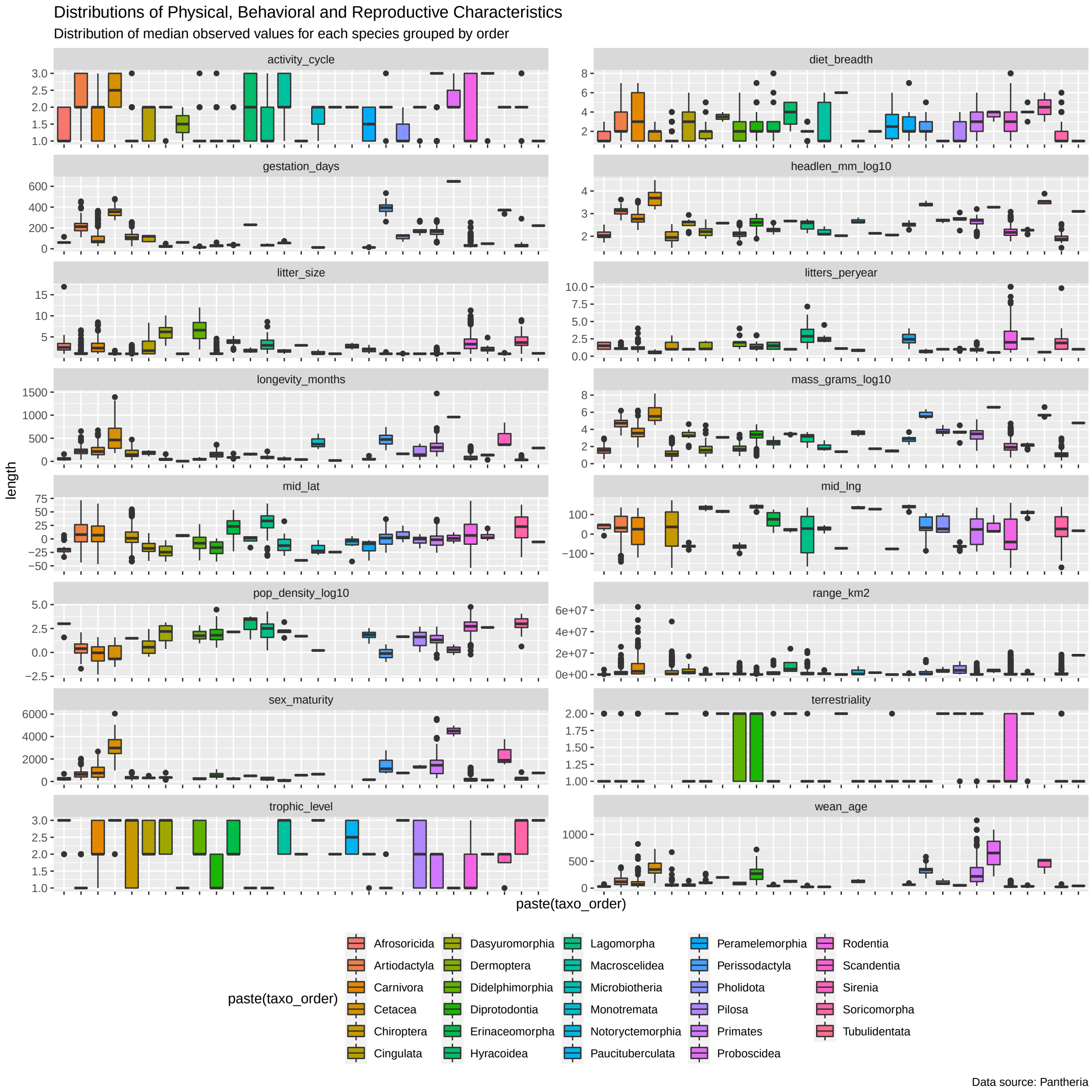
stat	maxspecies_vernacular	maxstat	minspecies_vernacular	minstat
mass_grams	Blue Whale	154321304.50	Bumblebee bat	1.9600000
headlen_mm	Blue Whale	30480.00	Bumblebee bat	30.9900000
forearmlen_mm	Large Flying Fox	200.00	Mouse-like Pipistrelle	23.0000000
litter_size	Tail-less Tenrec	16.89	African Bush Elephant	0.8400000
litters_peryear	Royle s Mountain Vole	10.00	Sperm Whale	0.2500000
longevity_months	Human	1470.00	Sunda Flying Lemur	3.4500000
gestation_days	African Bush Elephant	660.00	New Guinean echidna	10.0000000
wean_age	Chimpanzee	1260.81	Southern Moutain Viscacha	1.9400000

stat	maxspecies_vernacular	maxstat	minspecies_vernacular	minstat
pop_density	Eurasian Water Vole	57067.85	Hooded Seal	0.0048221
trophic_level	Congo Golden Mole	3.00	Pronghorn	1.0000000
terrestriality	Lesser Hedgehog Tenrec	2.00	Hottentot Golden Mole	1.0000000
diet_breadth	Eastern hedgehog	8.00	Congo Golden Mole	1.0000000
activity_cycle	Impala	3.00	Stuhlmann s Golden Mole	1.0000000
range_km2	Red Fox	63034304.34	Bramble Cay Melomys	0.0001874
mid_lat	Musk ox	71.68	Magellanic Pygmy Rice Rat	-53.7500000

The table above examines the data based on the minimum and maximum of each variable. Most of the details in the the table seem to agree with general knowledge. Mass and head size reveal the huge Blue whale vs. tiny, lightweight Bumblebee Bats. Examining Litter Size, Litters per year, Gestation, and Weaning age - Large mammals including Elephants and Sperm Whales, which reproduce infrequent small litters (K-selected species) appear opposite diminuitive Mice, Tenrec, and Echidnas (r-selected species) which are much more prolific in reproductive frequency. Chimpanzees are another K-selected species with a long weaning age vs. the very fast reproducing Viscacha, that wean their young for a very short period. Population density statistics indicate the Hooded Seal as the most solitary vs highly communal Water Voles.

The following box plots and scatter plots show considerable diversity in the distributions, types, and ranges of the various data points for each taxonomical *order*. There are 29 orders represented in our dataset. The box plots indicate the inter-quartile ranges of each "creature feature" including continuous-valued variables like mass & head size, as well as discrete-valued variables like terrestriality, diet breadth, and activity cycle. The aim of a machine learning model would be to find patterns in these features that are predictive in a way that's useful.

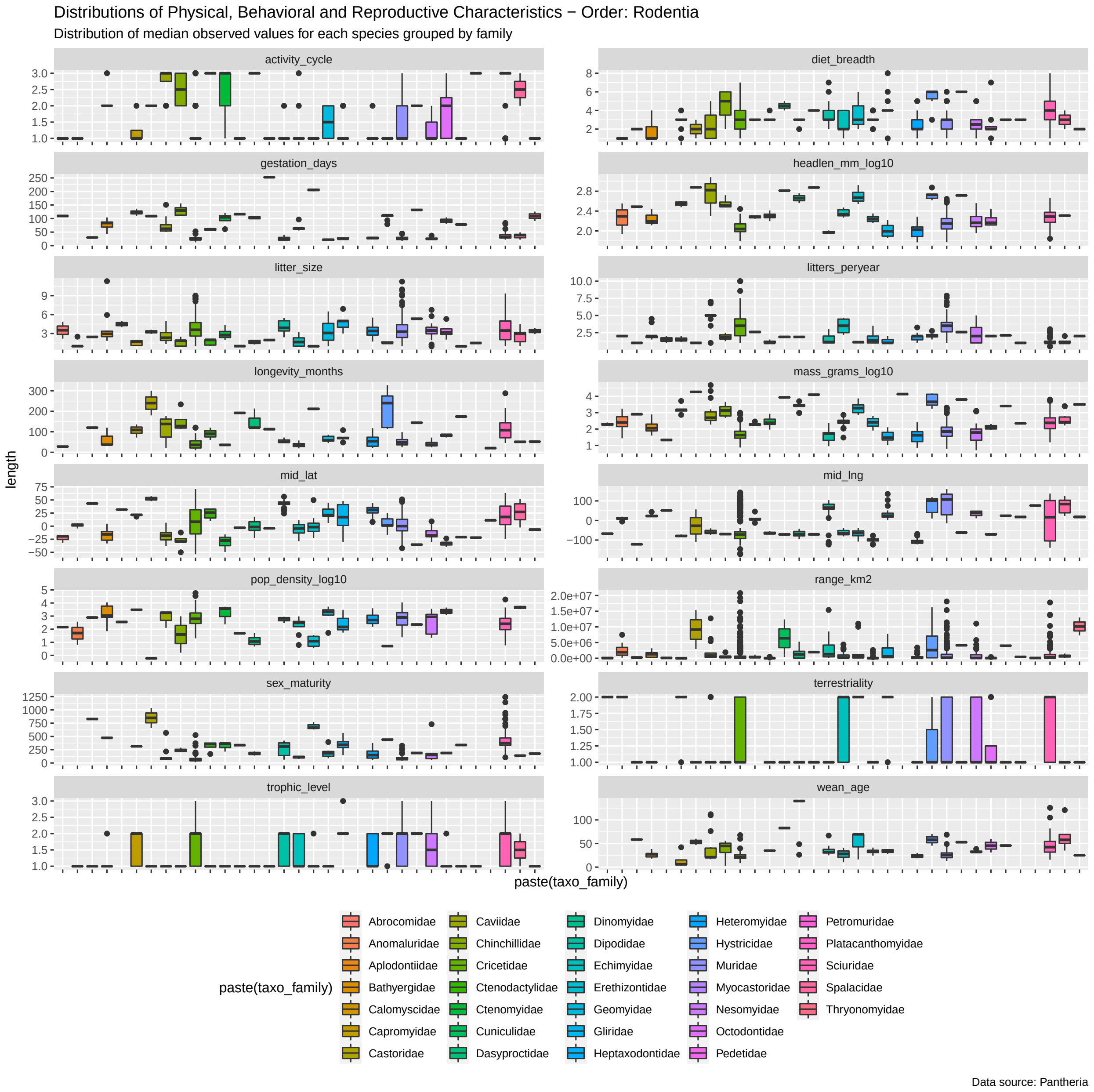
In general - determing the taxonomical order is not terribly difficult - For example - sharp canine teeth mark Carnivora. Flying mammals are Chiroptera. Hooved animals with even number of toes are always perissodactyla. A classification tool determining orders may be of limited value - though it could be useful in some cases where two orders may contain similar species. For example, Scandentia, Soricomorpha, Paucituberculata, Macroscelidea all contain shrews. So we might expect to see more confusion between these similar orders.


Distributions of Physical, Behavioral and Reproductive Characteristics

Distribution of median observed values for each species grouped by order

Data source: Pantheria

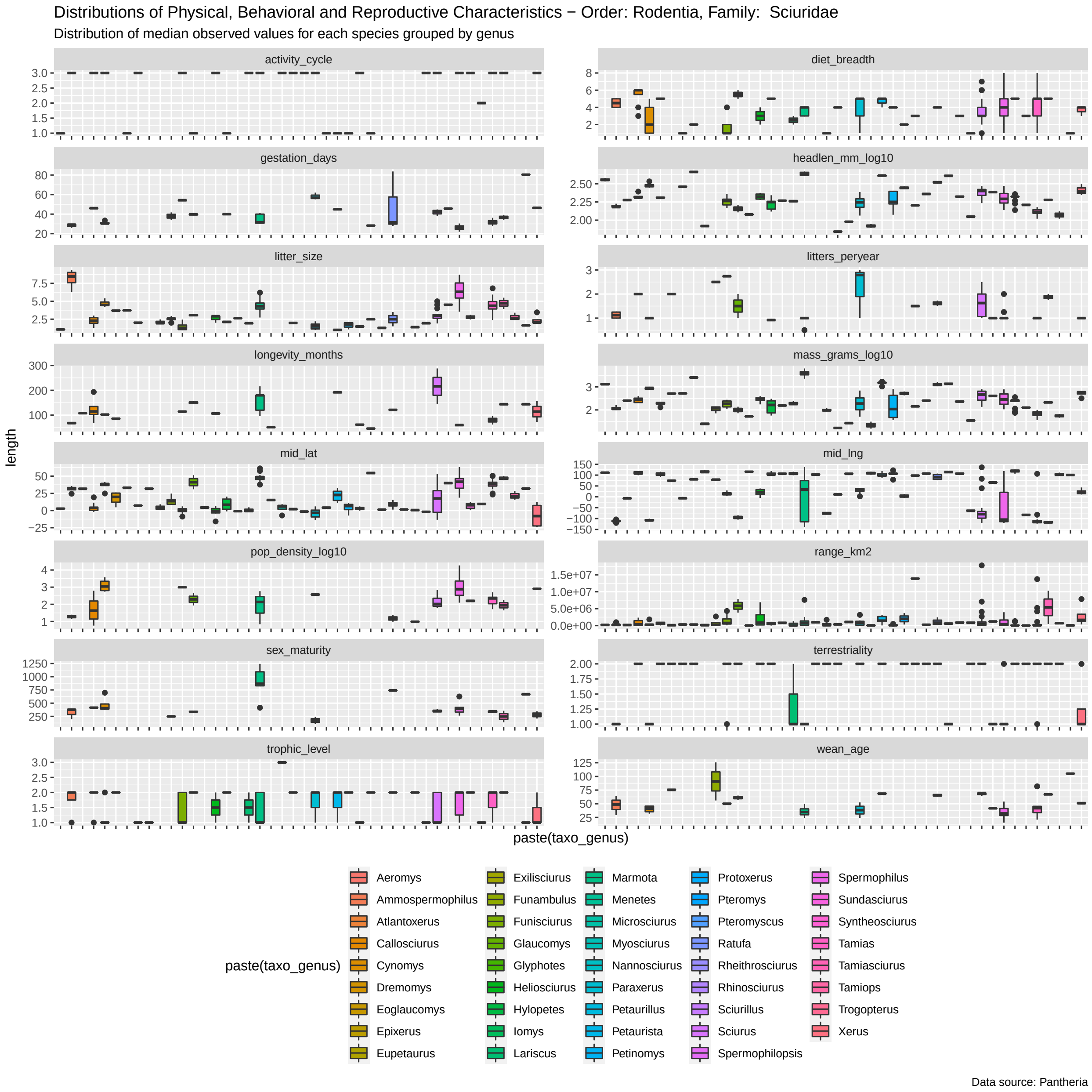
The difference between Families are much more specific, and can be used to differentiate more similar species. We will examine some characteristics of families within the largest order, Rodentia.


Distributions of Physical, Behavioral and Reproductive Characteristics − Order: Rodentia

Distribution of median observed values for each species grouped by family

Data source: Pantheria

Predicting the family correctly could be more helpful in distinguishing fairly similar species - for example voles (Cricetidae), Rats (Muridae), and Gophers (Geomyidae) are all similar families in order Rodentia.



And finally, looking at one family of Rodents - we can see that the data becomes more sparse at the genus level, with many columns containing a lot of missing data. There also appear to be many genus's that have only one recording as indicated by a horizontal bar. So the genus is probably the most specific level that would be practical to predict in a machine learning context.

These charts suggest that a recursive model might be practical - where the first model will train on all data to predict the order, then retrain once an order has been predicted with data that only includes the families in that order. Orders with more complete data will produce more accurate results when recursed. The charts below visualize some of the most highly correlated variables and how their distributions are grouped at the order level.

Examining Correlations

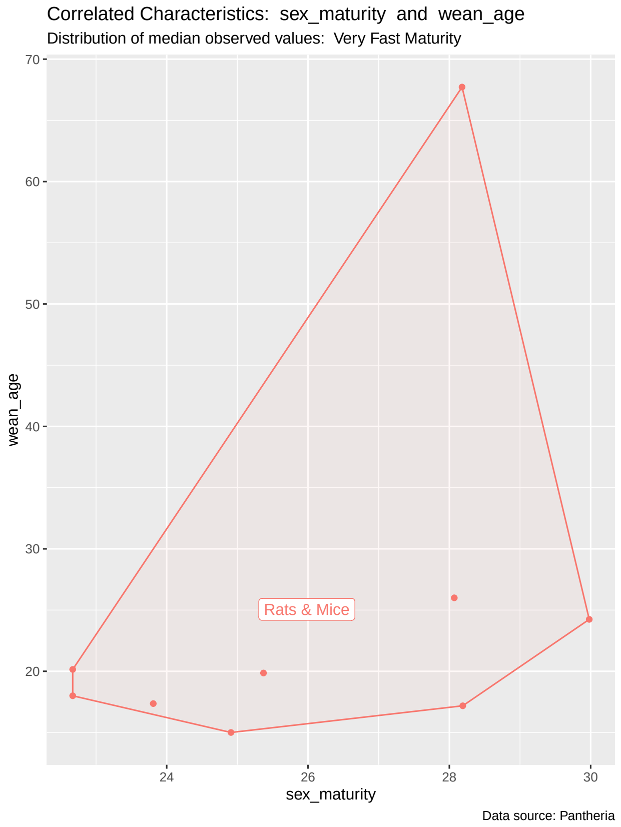
Now we will can how closely the various predictors are correlated.

```
## # A tibble: 16 x 17
##    rowname mass_grams headlen_mm litter_size litters_peryear gestation_days
##    <chr>        <dbl>      <dbl>       <dbl>           <dbl>          <dbl>
##  1 mass_g~   NA           0.762      -0.0464         -0.0750        0.187
##  2 headle~    0.762      NA          -0.203          -0.227         0.548
##  3 litter~   -0.0464     -0.203      NA               0.251        -0.552
##  4 litter~   -0.0750     -0.227       0.251          NA            -0.422
##  5 gestat~    0.187       0.548      -0.552          -0.422        NA
##  6 wean_a~    0.0801      0.266      -0.368          -0.359         0.552
##  7 sex_ma~    0.244       0.502      -0.400          -0.449         0.719
##  8 longev~    0.367       0.670      -0.456          -0.417         0.696
##  9 pop_de~   -0.0217     -0.147       0.204           0.253        -0.185
## 10 diet_b~   -0.0348     -0.0365      0.162           0.0332       -0.0205
## 11 terres~   -0.139      -0.318      -0.326          -0.133        -0.0290
## 12 trophi~    0.0640      0.0789      0.0951         -0.159        -0.0940
## 13 activi~    0.0207      0.129      -0.0862         -0.178         0.271
## 14 range_~    0.0260      0.152       0.0230         -0.0571       -0.00330
## 15 mid_lat    0.00797    -0.0296      0.297           0.0566       -0.144
## 16 mid_lng    0.0278      0.0634     -0.0926         -0.0864        0.0722
## # ... with 11 more variables: wean_age <dbl>, sex_maturity <dbl>,
## #   longevity_months <dbl>, pop_density <dbl>, diet_breadth <dbl>,
## #   terrestriality <dbl>, trophic_level <dbl>, activity_cycle <dbl>,
## #   range_km2 <dbl>, mid_lat <dbl>, mid_lng <dbl>
```

This table demonstrates that there are strong correlations between certain sets of values:

- Longevity and sex maturity: 78%
- Mass and head length: 76%
- Weaning age and sex maturity: 74%
- Sex maturity and gestation days: 72%
- Longevity and gestation days: 70%
- Longevity and head length: 67%
- Weaning age and gestation days: 55%
- Litter size and gestation days: -55%

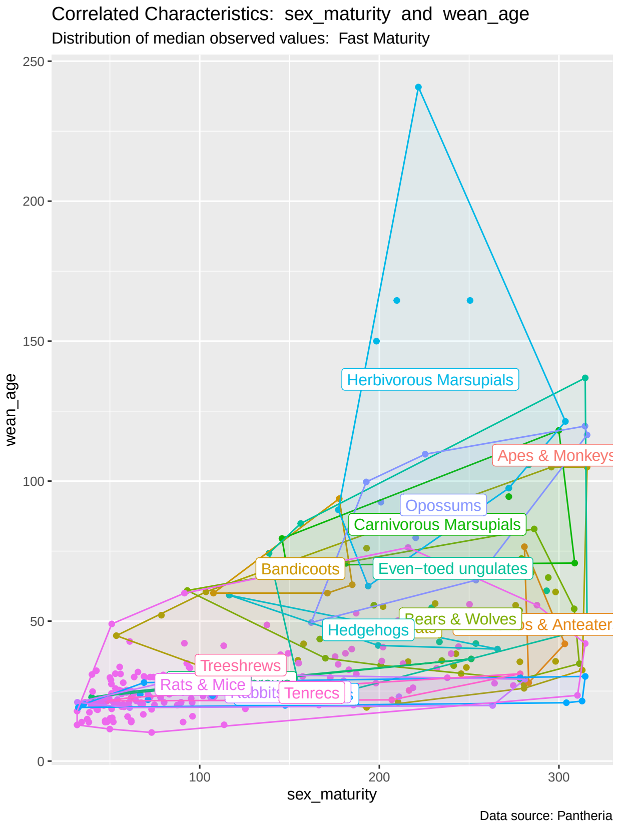
These correlations suggest that some fields may be candidates for use of linear regression to impute some missing values.



Correlated Characteristics: sex_maturity and wean_age

Distribution of median observed values: Very Fast Maturity

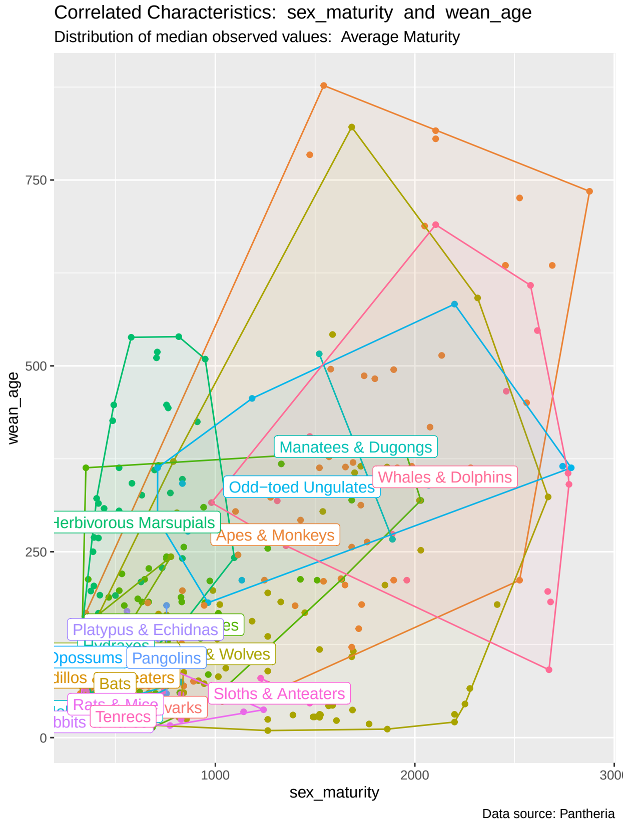
Data source: Pantheria



Correlated Characteristics: sex_maturity and wean_age

Distribution of median observed values: Fast Maturity

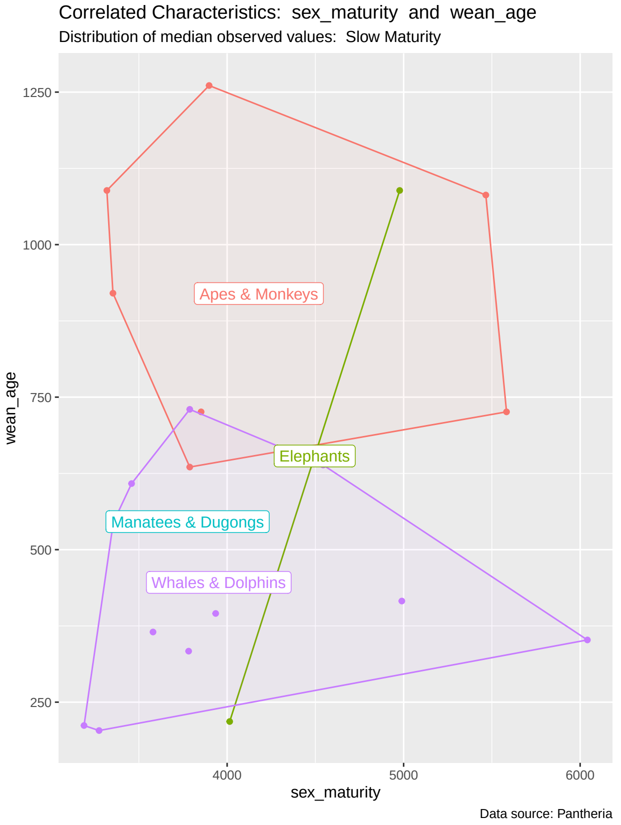
Data source: Pantheria



Correlated Characteristics: sex_maturity and wean_age

Distribution of median observed values: Average Maturity

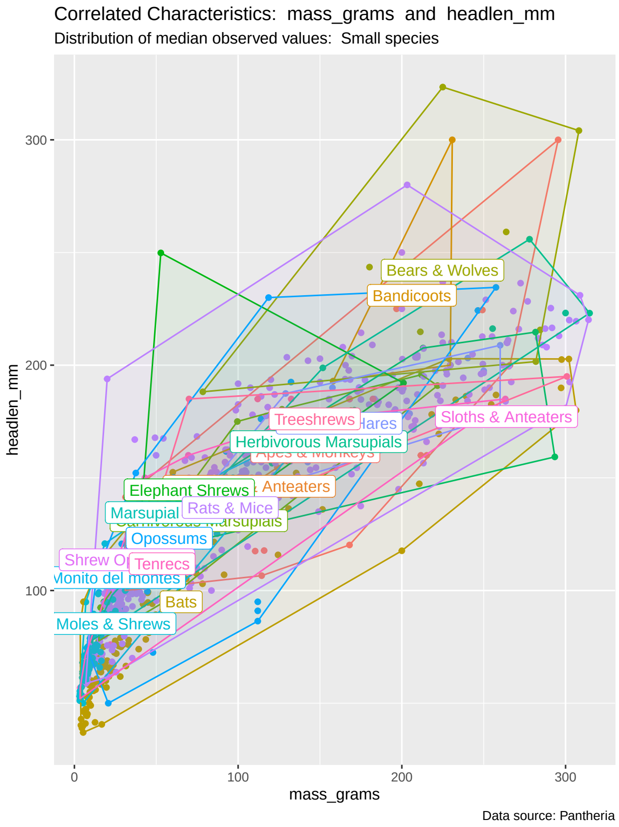
Data source: Pantheria



Correlated Characteristics: sex_maturity and wean_age

Distribution of median observed values: Slow Maturity

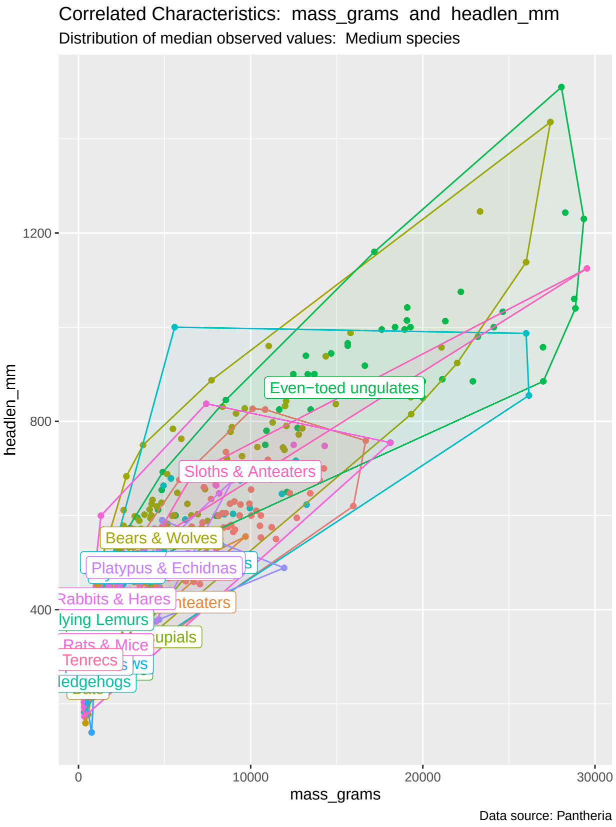
Data source: Pantheria



Correlated Characteristics: mass_grams and headlen_mm

Distribution of median observed values: Small species

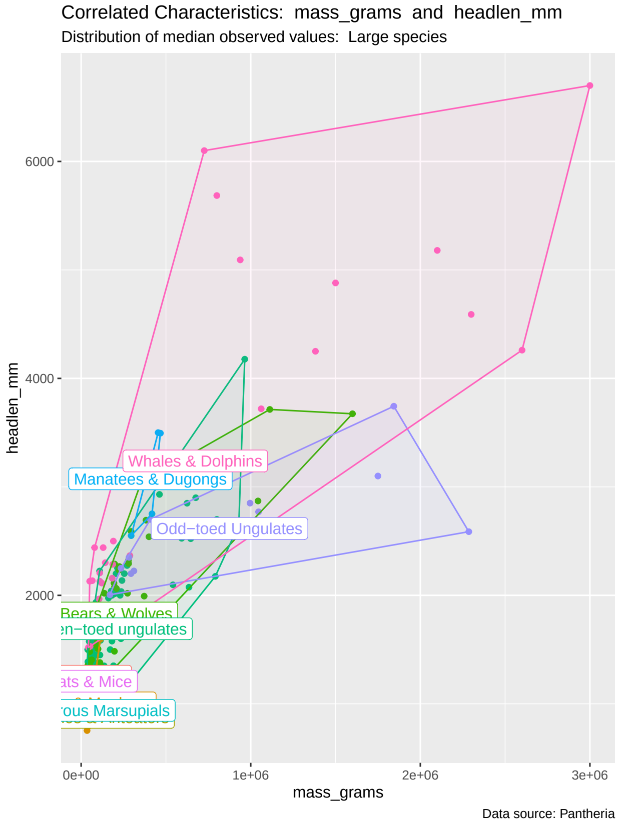
Data source: Pantheria



Correlated Characteristics: mass_grams and headlen_mm

Distribution of median observed values: Medium species

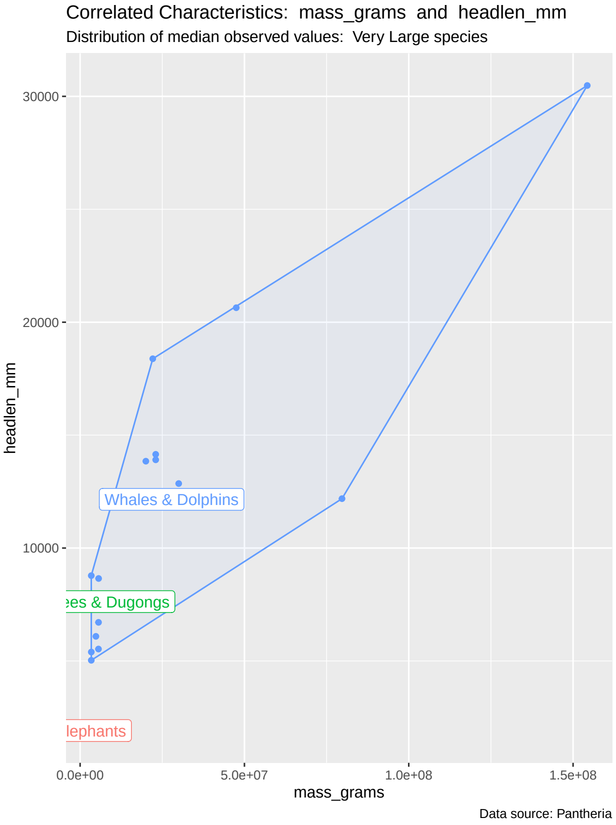
Data source: Pantheria



Correlated Characteristics: mass_grams and headlen_mm

Distribution of median observed values: Large species

Data source: Pantheria



Correlated Characteristics: mass_grams and headlen_mm

Distribution of median observed values: Very Large species

Data source: Pantheria

Examining the most highly correlated values reveals that there are some patterns of distribution between orders and families, but there is also significant overlap. Many areas appear where the distribution of different classes are intermingled. Based on this analysis, a classification tree or random forest approach shoud be useful in finding the important predictive variables to locate splits and isolate areas where groupings exist.

Model Construction

In this section we will begin constructing and testing models. The first approach will be use a Classification and Regression Tree (CART). We will assess the accuracy values of the CART, and then examine how a random forest model may improve the accuracy. In the following section we create a "final" tables of data which has ONLY the class/factor columns, and the various predictors. We will create a set with 0's imputed, with NA's and one with MEAN imputation.

To create a test and training data set that have the same classes represented, we will have to stratify the data by order, then select a portion to of each order for training, and the rest for testing. To maximize the number of observations and to ensure a fair number of challenges - we will start by splitting the data into 50% test and 50% training data for each order. The splitstackshape package enables stratified selection of random data points to create a hold-out data set that would contain a large variety of classes to challenge the machine learning model.

Assessing the Model Scores

The following key metrics relate to the predictions we are making:

- Sensitivity - True Positives - how often the model predicts the right order/family
- Specificity - True Negatives - how often the model avoids predicting this class incorrectly
- Precision: True Positives / True Positives + False Positives
- Recall: True Positives / True Positives + False Negatives
- F1 Score: 2 * ((Precision * Recall) / (Precision + Recall))

So our goal will be do accomplish several objectives - to *increase the number of columns with 1 or more predictions*, to improve the *Overall Accuracy* of the model; to achieve a good average *F1 score* for those classes which were predicted, balancing precision and recall across all predicted classes. Several function will be set up to analyze and assess the results. For each iteration in the model construction, we will create a grid-style heatmap plot for the confusion matrix and run a function to measure accuracy and other key metrics we defined.

Create Data Partitions

We are splitting the test and training data using the stratified method so there are some samples of the same orders in both test and training data sets.

The partition has taken the original data table of 4136 original rows and reserved a hold-out of 289 to assess the final model score. The remaining 3847 rows have been further partitioned into a set of 3077 training rows and 770 test rows to use for the model construction and tuning process. There are 25 taxonomical orders present in the test data. The hold-out validation data represents unknown information - we will not use this data for any other purpose except to test the performance of the final model.

Complete Case Analysis vs. Available Case Analysis

We can limit the decision tree to rows containing a *complete* set of predictors, dropping all rows that have any missing values. This would be termed complete case analysis. This method is used in a random forest model if NA values are encountered in the training data. Complete case analysis may provide the greatest overall accuracy, since it uses the greatest number of predictors and requires all outcomes to have training observations matching all predictors. But it will not be able to produce predictions for every row of test data.

Another option would be to use only training rows with data in all the same columns as the test data. This would require that different models be created to accept varying numbers and combinations of predictors available. In these cases, only *available* predictors (those columns provided in the test set) would be utilized in training the model. This would be a more complex model which would adapt to missing observations by running a custom model for each combination of available data columns. The final "advanced model" will use this method.

Baseline Model

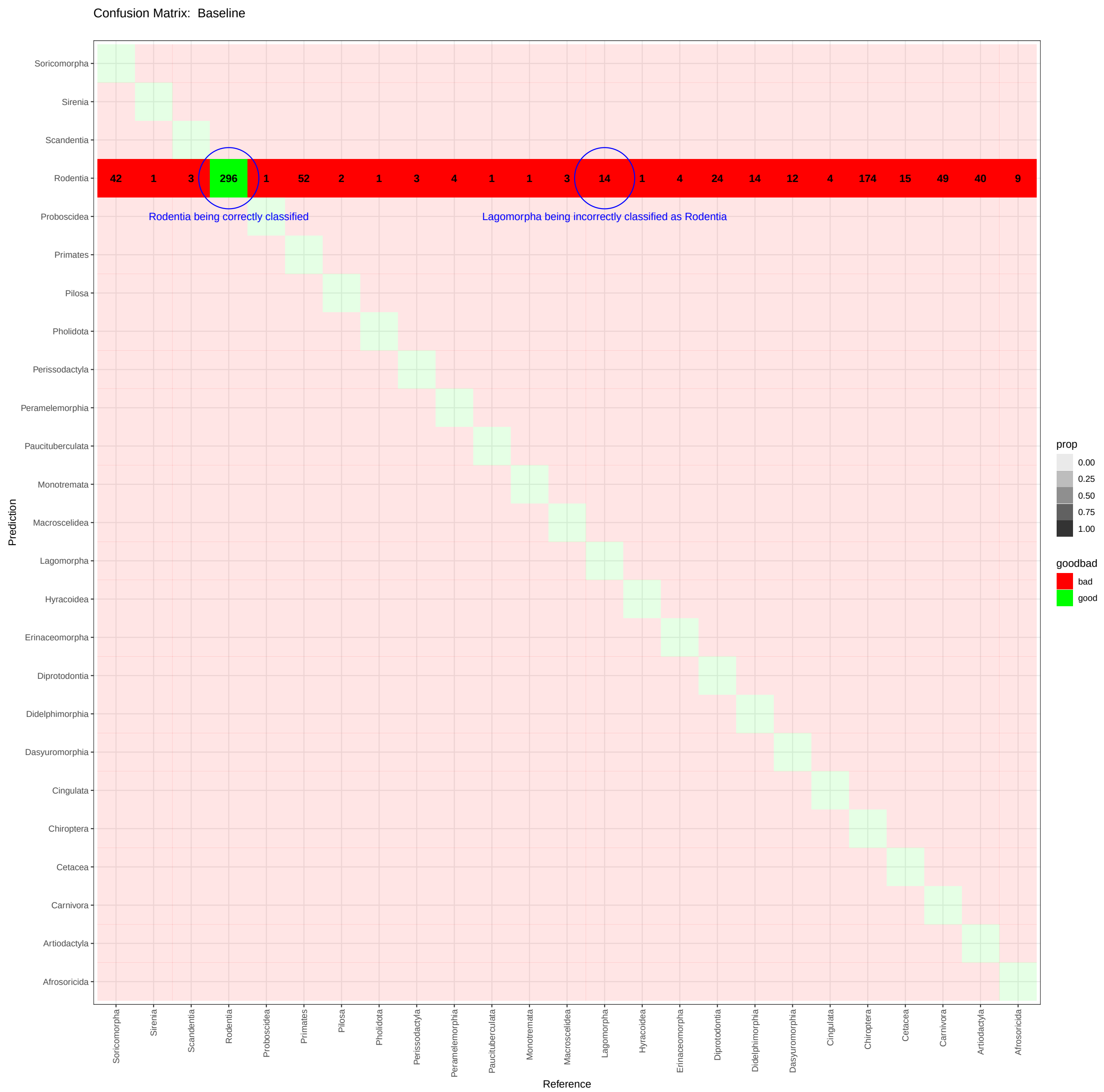
As explained already - a model that predicts the order is not super useful. Also, there are a few orders which contain the majority of species - so you could simply predict one of these and be right most of the time. So we have established some criteria beyond just accuracy.

To demonstrate how accurate a model would be that predict the most common classes - the following code will apply a prediction of Rodentia on every mammal and we will see how well it scores. This would be similar to a naive Bayes approach if the outcomes were continuous values. Since the outcomes are discreet - we will use the most common class as the prediction.

model	dataset	Predicted	Accuracy	F1_Mean
1- Baseline	With 0's	1	0.3844156	0.5553471

Reading the Confusion Matrix Visualization Grid

The chart below visualizes the confusion matrix. A number in any cell other than a green cell, indicates a wrong guess and a number in a green cell is a correct guess. The reference or correct classes are listed on the y axis and the predictions made for each reference class are listed horizontally along the x axis. Annotations have been added on some of the confusion matrix visuals to call out specific data points or assist the reader in understanding how to read these charts.



Data source: Pantheria

This approach is obviously not a great choice, unless you think every Mammal is a Rat! But you can see that we would be right almost 40% of the time if you always predict this one majority class. We will call 40% our threshold value or baseline. Now that we have established a baseline, we will build some more useful models and see if we can predict with greater than 40% accuracy at the order level.

Classification Tree with NA's

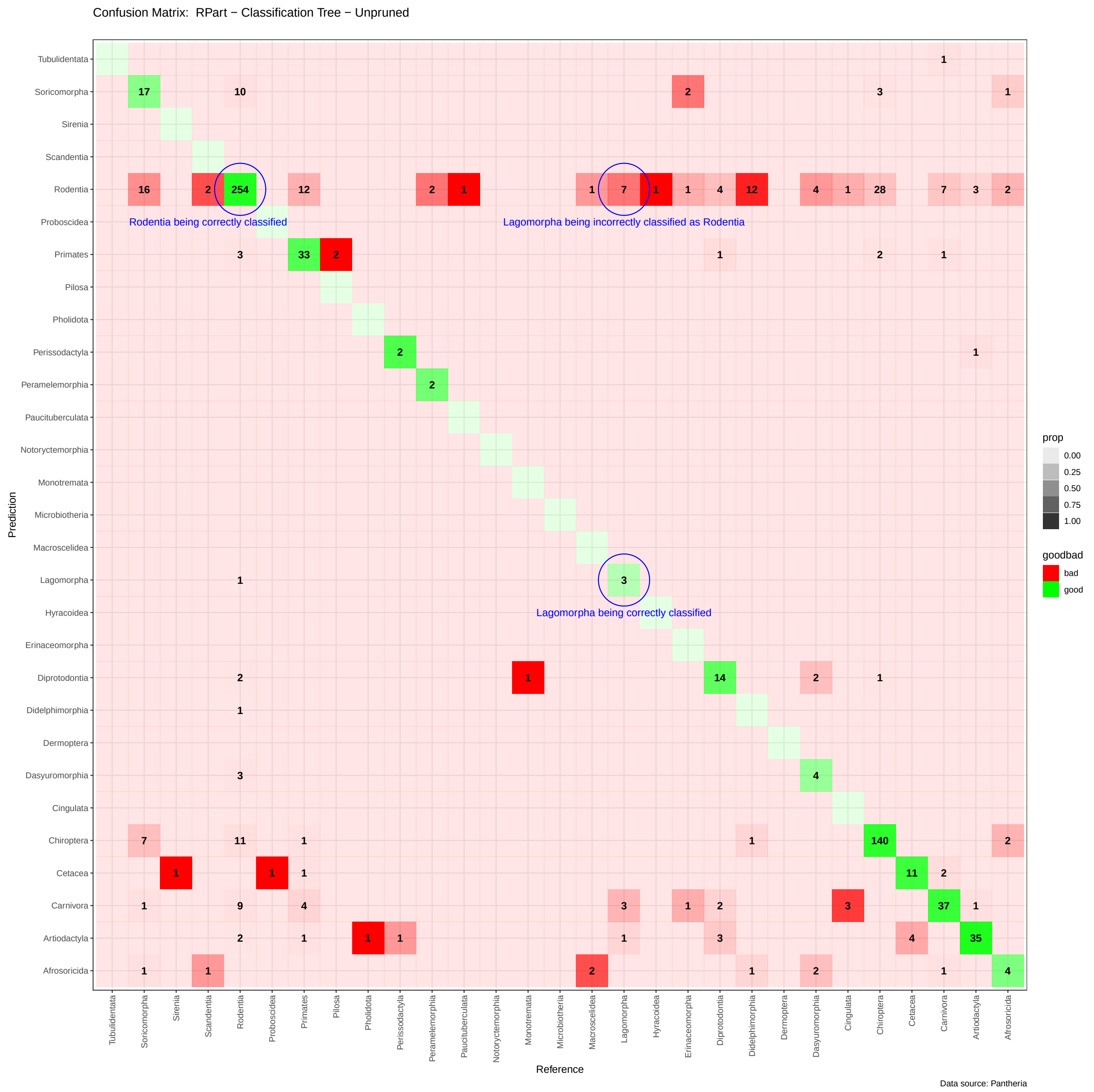
The test set contains most of the 29 orders in the training set. We'll start with a simple classification tree model.

The RPart package provides features to visualize how the tree splits on each variable.

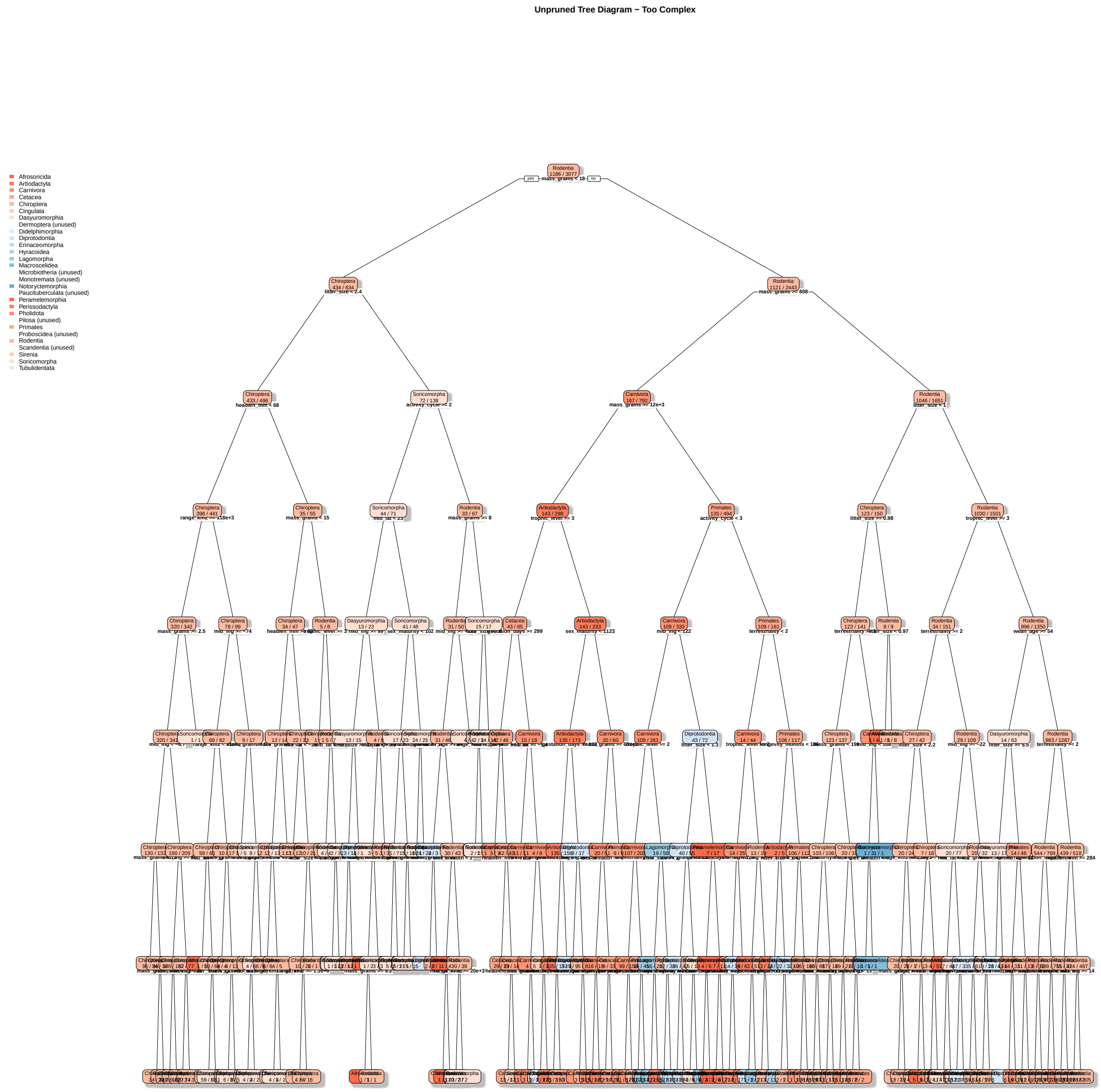
The model has been trained, now we will apply a prediction using test data. This will take the test data set, remove the order column, and get a prediction based on the remaining columns.

model	dataset	Predicted	Accuracy	F1_Mean
1- Baseline	With 0's	1	0.3844156	0.5553471
2- RPART Classification Tree	With 0's	13	0.7220779	0.6191113

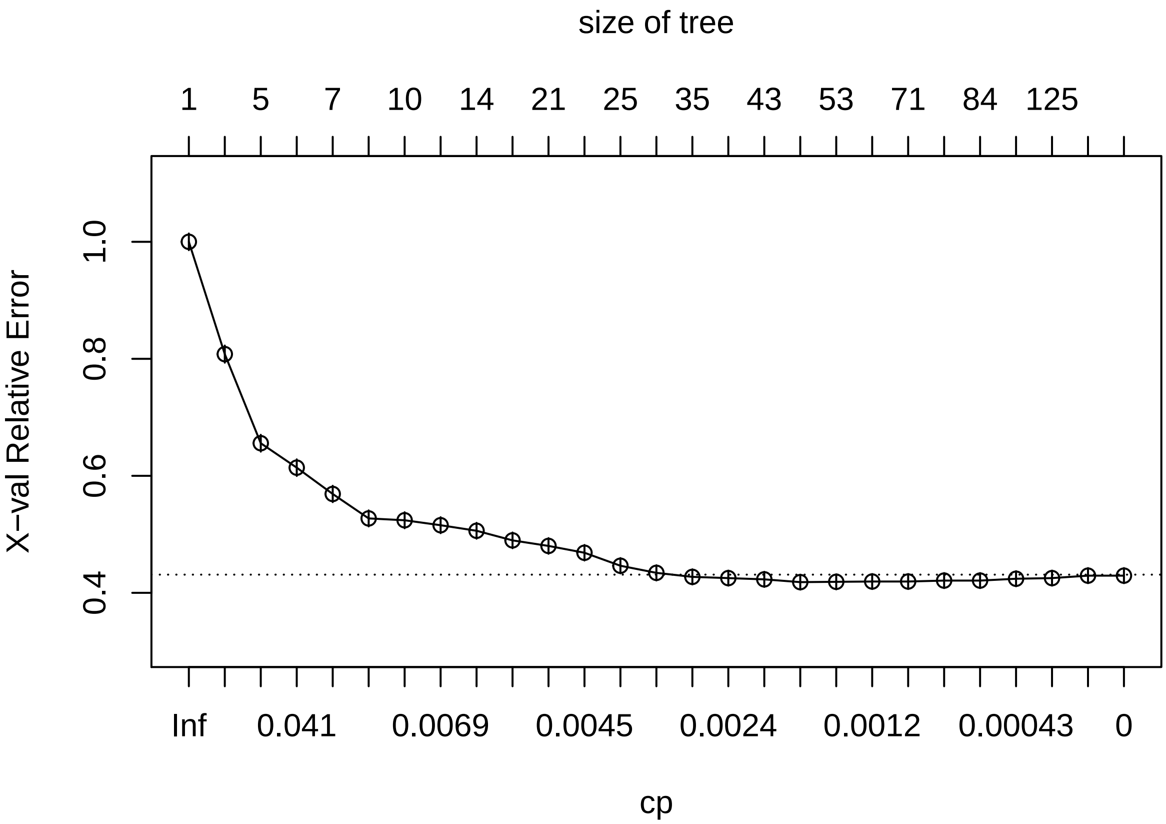
This CART results in about 70% accuracy. This is pretty good for a single decision tree - but this decision tree is overly complex and overtraining. It would need to be pruned to be more flexible.


Confusion Matrix: RPart − Classification Tree − Unpruned

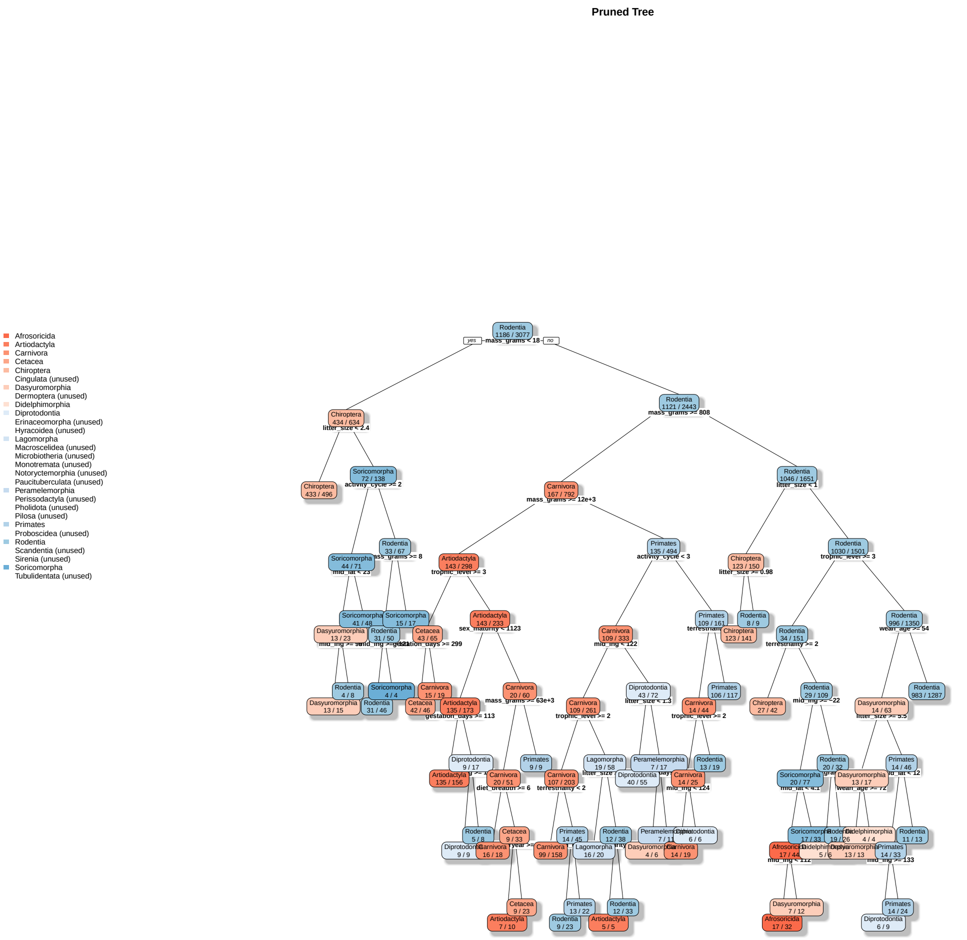
Data source: Pantheria



The unpruned classification tree is very complex, and many orders are excluded. This plot demonstrates one challenge of using decision trees with sparse data over many outcomes. The classification above could be considered over-fitted. It is too complex and the number of splits is too great. It's going to match the training data very closely. Pruning the tree could reduce complexity to a more practical level to make the model more robust.



According to the RPart documentation for the plotcp feature “A good choice of cp for pruning is often the leftmost value for which the mean lies below the horizontal line.” Considering the plotcp above a reasonable complexity parameter would be around .002 which is near the left most reading where the X-val Relative Error curve crosses the dotted line.

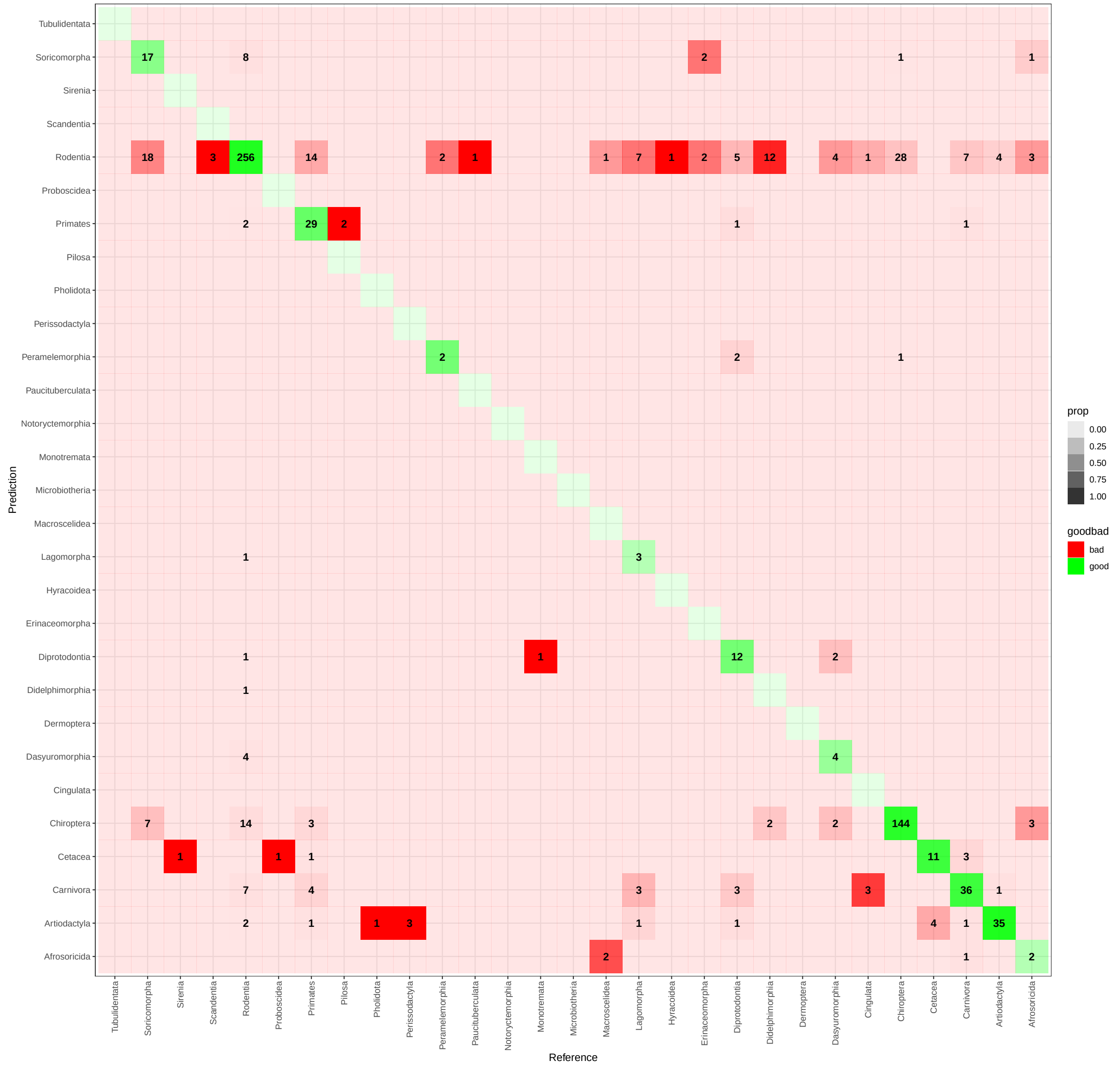


We will now make a prediction using the pruned tree.

model	dataset	Predicted	Accuracy	F1_Mean
1- Baseline	With 0's	1	0.3844156	0.5553471
2- RPART Classification Tree	With 0's	13	0.7220779	0.6191113
3- RPART Classification Tree - Pruned	With 0's	12	0.7155844	0.5797862

The confusion matrix for the pruned tree is shown below. The results appear similar to the initial CART model, but with slightly lower overall accuracy at 0.7155844.

Confusion Matrix: RPart − Classification Tree − Pruned



Data source: Pantheria

Pruning the tree produces a much more compact tree, les complex tree - but only about 1/3 to 1/2 of the orders will “make the cut” and be included in the pruned CART model.

The code below allows you to select a random row from the data and try to predict the outcome using our pruned CART model.

This code will try produce one random prediction using the rpart pruned CART model:

```
## [1] "Afrosoricida"
```

```
## [1] "Afrosoricida"
```

Random Forests

Having established the possible benefits of a CART approach, A random forests ensemble method may provide even better results and a more robust model. One challenge to using this method with the data we have, is that random forests are sensitive to NA values. With NA values in the dataset, the random forest will apply complete case analysis. Even though there are 29 different orders in the training data, since many of them have at least one NA values on every row, fewer than half of these orders will *ever* be selected by the model. We will take a look at the difference between a dataset with NA's, 0's or other imputed values.

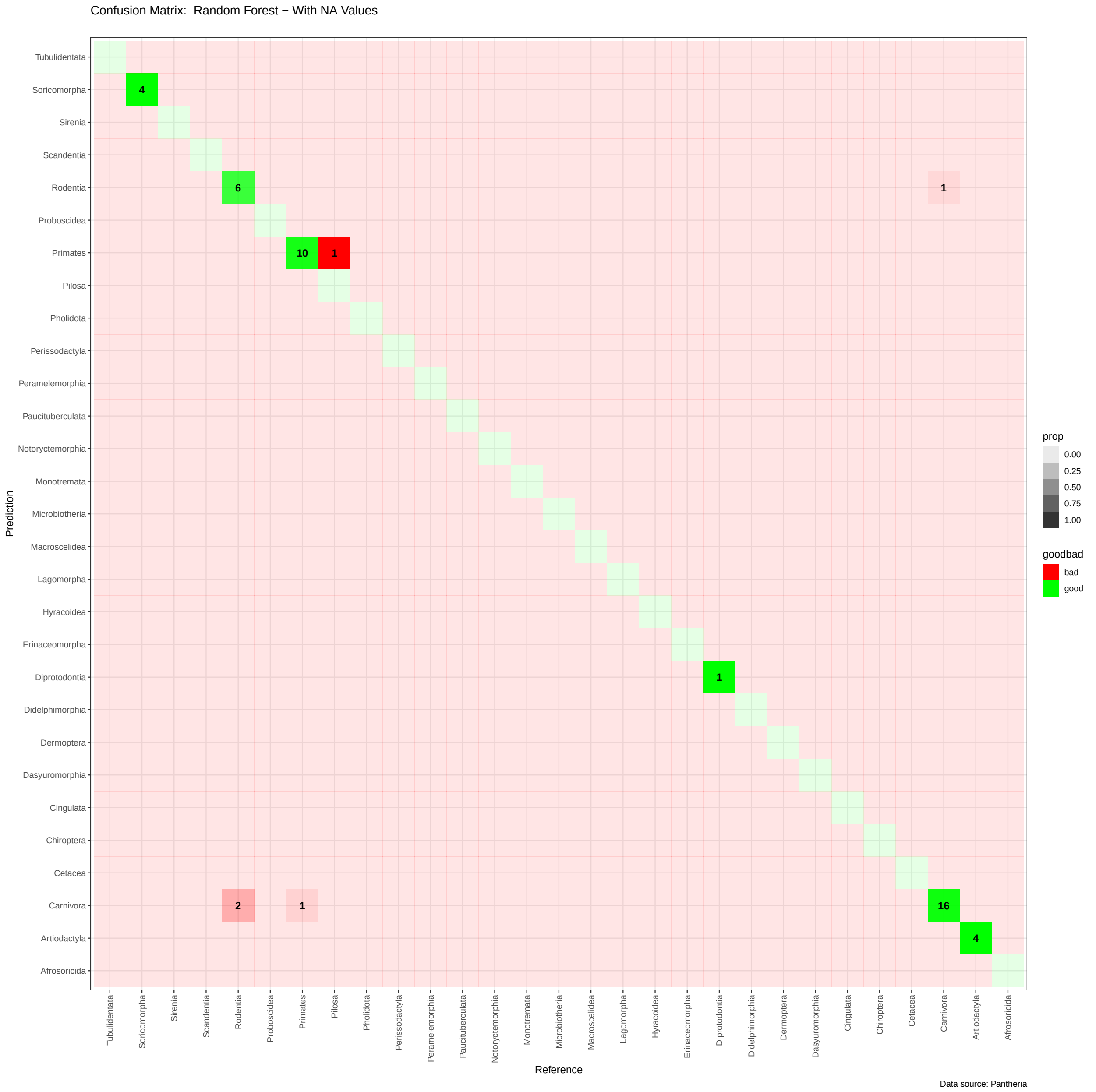
As noted previously - there are some non-random patterns in the missingness of data - some of the columns are applicable only to land based, flying, or marine mammals. During model construction we will focus on common columns for land based mammals. In the final model construction, we will experiment with adding available case analysis to enable use of more specialized columns like forearm length and geo coordinates that are not as universally recorded or where special handling will be required.

We have set the training set and test set to use. This is the data set with NA values.

The model is now trained. We will now apply a prediction on the test data with NA's. This is expected to produce a limited number of actual predictions since it will only be able to train using complete case analysis.

model	dataset	Predicted	Accuracy	F1_Mean
1- Baseline	With 0's	1	0.3844156	0.5553471
2- RPART Classification Tree	With 0's	13	0.7220779	0.6191113
3- RPART Classification Tree - Pruned	With 0's	12	0.7155844	0.5797862
4- Random Forest	With NA's	6	0.8913043	0.9329966

This model can only produce 46 predictions out of 770 challenges.



Data source: Pantheria

The random forest model with NA's results in very high accuracy, but there are many NA's in the predictions. This is not a very practical approach if you want to produce a prediction for every row of test data. To improve the number of outcomes predicted, we will experiment with various strategies to impute data, in order to "fill in blanks", and attempt to balance the classes in order encourage the model to predict more minority classes.

Imputing 0's

There are missing values on many rows, relative to the overall data set. (NA's were originally denoted in the dataset as -999, these values have been converted to an R standard *NA*) The sparseness of this data has the potential to skew any statistical analysis and presents a challenge to development of a classification tree model. There are several options available to handle the missing data.

Imputing 0 is the simplest possible approach - to fill in imputed values for any values that are missing from the data with a 0. Imputing 0's is a crude way of assigning a value that represents missing data. The fact that data is missing may be predictive in the Pantheria data, but this is not necessarily the case for data collected through other outside sources.

Imputing the mean, mode, or median, or a predicted value may also be a way to fill in the gaps in the training data. Every approach to imputation has some advantages and drawbacks. Some methods are easier to implement, but will reduce accuracy by introducing uncertain data. It is ultimately a choice in the final implementation, how to apply imputation to the data, depending on the desired model performance.

In this dataset - missingness is sometimes a predictor - certain fields are set only on certain orders as previously discussed. Chiroptera are the only order with arm lengths measured. Cetacea and Sirenia do not have geocodes and values for terrestriality. Missingness can be a valuable hint as to the correct class, but it is also potentially overfitting the model by assuming that real-world challenge data is going to have the same pattern of missingness.

We will start by imputing 0 for all missing values. This is similar to using an available case analysis approach - by adding 0 the RF will be able to use all the rows in the training data, since it will not encounter any columns for which there are NA values.

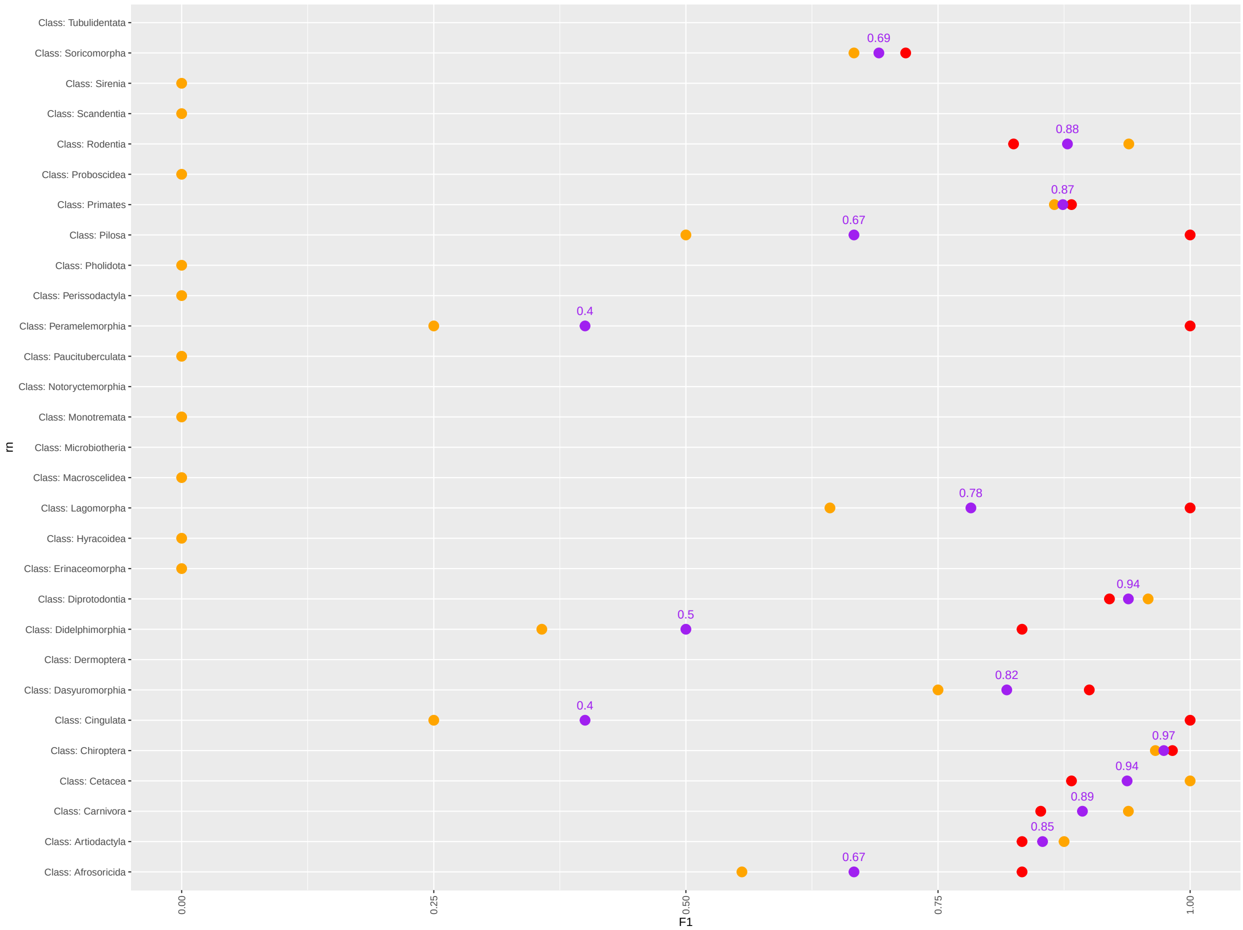
Using this method will result in many false pieces of information in the training data, like animals with 0 mass. But applying 0 to both the training and test data means that there will be many cases where a 0 in a training data column will match a 0 in a test row, because both were missing the same column.

This will cause the decision tree to think this is a good match because both animals have, for example, 0 mass. Obviously this is just a "hack" to make the models line up. But it works well if the patterns of missingness are non-random.

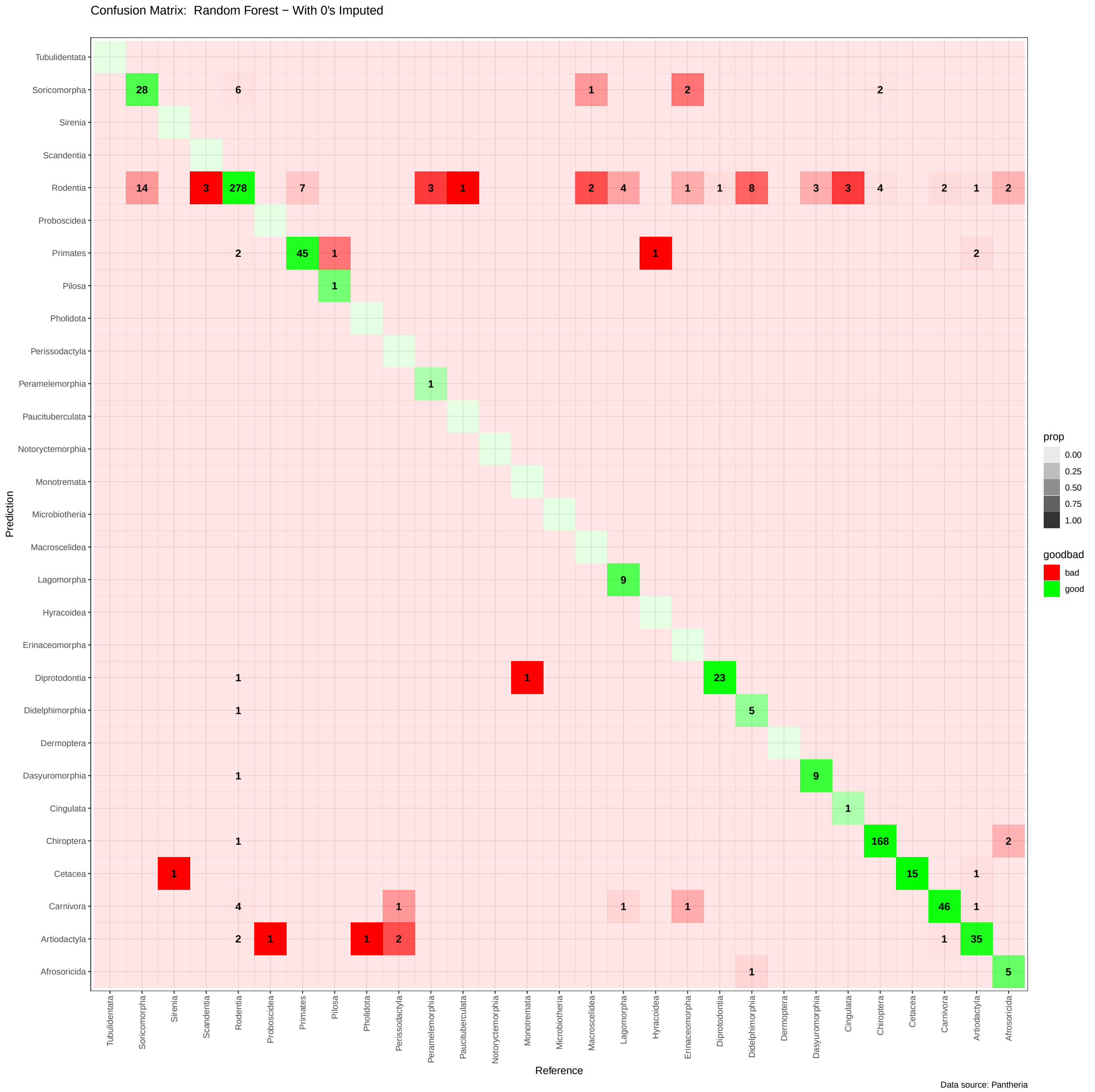
After experimenting with the imputation of 0's, We will try various other imputation strategies throughout the model building section to try to meet our stated goals and make the final model less reliant on matching these patterns of missingness.

model	dataset	Predicted	Accuracy	F1_Mean
1- Baseline	With 0's	1	0.3844156	0.5553471
2- RPART Classification Tree	With 0's	13	0.7220779	0.6191113
3- RPART Classification Tree - Pruned	With 0's	12	0.7155844	0.5797862
4- Random Forest	With NA's	6	0.8913043	0.9329966
5- Random Forest	With 0's	15	0.8688312	0.7516451

Here is another way to visualize the results. Red dots indicate precision, orange indicates recall, and the F1 score is the purple dot. The F1 Score is stamped on the chart.



The random forest model with 0 imputed into NA's results in an overall accuracy of 0.8688312. This has increased the number of classes predicted and resulted in fairly accurate predictions across many of the classes. However the imputation of 0 in training and test data may be overfitting the model by expecting the same pattern of missingness.



Taking a few examples we can see the most common class, Rodentia has been correctly identified the most times. Some rodents were misidentified a variety of different incorrect orders.

A random forest model that imputes 0 to every NA value is shown. This model is quite accurate, with 0.8688312 overall accuracy. It correctly identifies the majority of rows across 18 of the 25 orders in the test data.

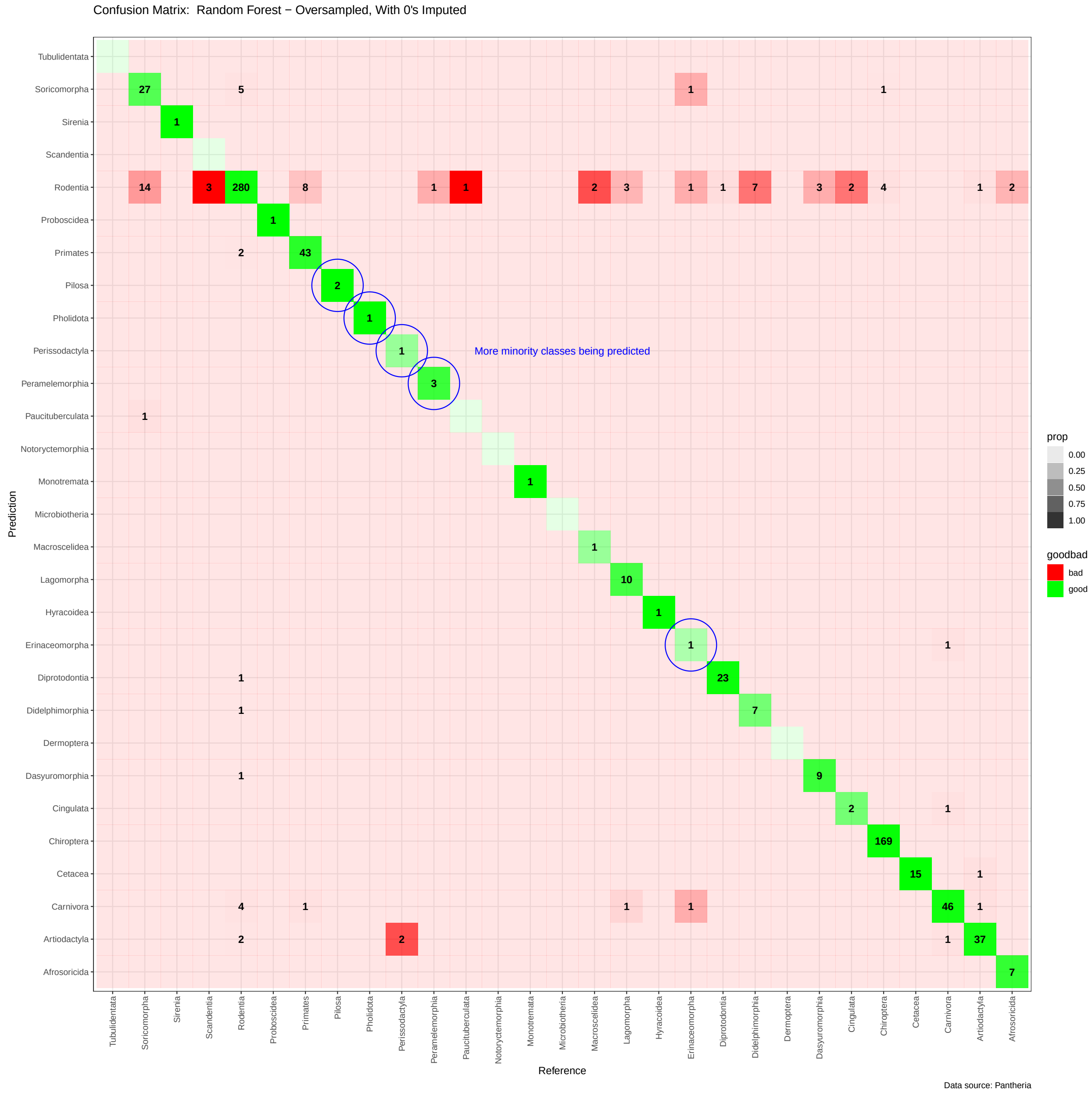
Oversampling

Not only does this data suffer from sparseness, but it also has very imbalanced classes. By oversampling the minority classes, we can provide enough observations to appear more frequently in the training data sets and increase the chance that a minority class is selected by the model.

This oversample function will draw random rows until there are at least 100 rows in each order, up to a maximum of 2000 new rows.

We will now train with the oversampled data.

model	dataset	Predicted	Accuracy	F1_Mean
1- Baseline	With 0's	1	0.3844156	0.5553471
2- RPART Classification Tree	With 0's	13	0.7220779	0.6191113
3- RPART Classification Tree - Pruned	With 0's	12	0.7155844	0.5797862
4- Random Forest	With NA's	6	0.8913043	0.9329966
5- Random Forest	With 0's	15	0.8688312	0.7516451
6- Random Forest	With 0's, Oversampled	23	0.8935065	0.8312761



After oversampling the minority classes, the model attains an accuracy of 0.8935065, and it does better at finding the less common classes. For example - Lagomorpha, Erinaceomorpha, etc. are more consistently chosen. In this model, ~ 23 of the orders are correctly identified at least once. So oversampling seems to have the expected effect of revealing more of the long-tail cases that would be missed in an imbalanced dataset.

Imputing a Grouped Mean

It should be possible to impute a more realistic value than 0 for the missing values, which could make the model more robust and able to handle data entries that don't exactly match the combination of columns recorded in Pantheria. This imputation would be expected to reduce the precision somewhat, as it is adding a lot of new "theoretical" data. The objective would be that the theoretical data could be correct, and could closely match actual challenge values passed in to the model.

The imputed value will be an estimate based on the mean of each genus, family or order. Imputation increases the homogeneity of the dataset, and will consequently underestimate the standard deviation of any statistical analysis. So the mean data shoud be considered speculative - but thats ok - we are only using it for training the model. It should make the model more robust and adaptable to different sets of input parameters which may or may not match the data provided by Pantheria.

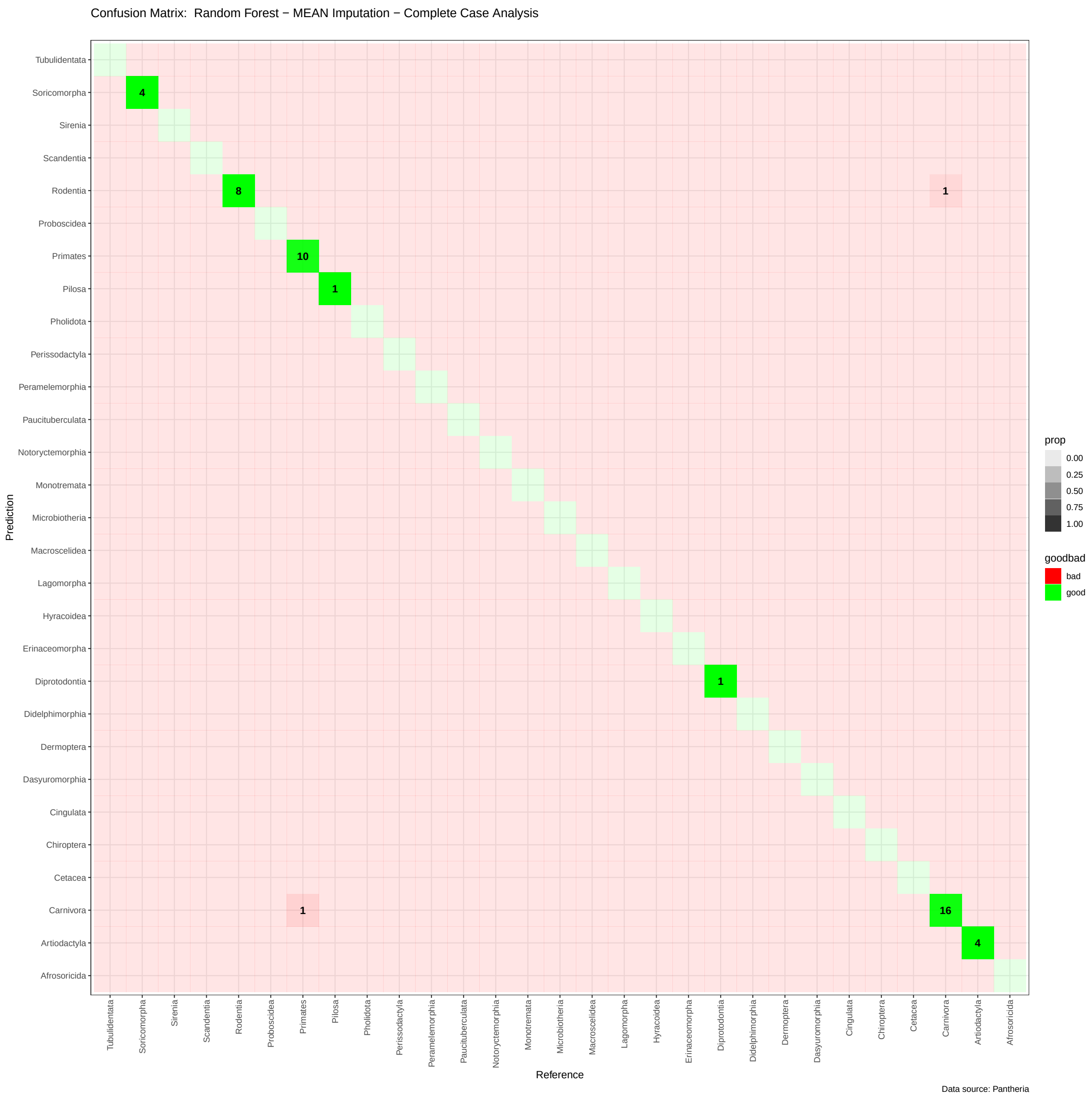
To impute mean values - we will create a copy of the training data set that replaces missing values with estimated means based on the taxonomical classes assigned. This code will first attempt to get an average for each column that's missing, by grouping first at the genus, then the family, then the order, and finally from the entire Mammalia class.

taxo_order	taxo_family	taxo_genus	taxo_species	mass_grams	headlen_mm
Afrosoricida	Chrysochloridae	Calcochloris	obtusirostris	24.05	100.84000
Afrosoricida	Tenrecidae	Micropotamogale	lamottei	69.59	160.00000
Afrosoricida	Chrysochloridae	Amblysomus	hottentotus	62.60	121.49000
Afrosoricida	Tenrecidae	Microgale	brevicaudata	8.99	94.49333
Afrosoricida	Tenrecidae	Microgale	dryas	40.00	94.49333
Afrosoricida	Chrysochloridae	Cryptochloris	zyli	19.31	84.85000
Afrosoricida	Tenrecidae	Echinops	telfairi	152.25	160.00000
Afrosoricida	Tenrecidae	Microgale	gracilis	23.30	94.49333
Afrosoricida	Tenrecidae	Hemicentetes	semispinosus	134.00	175.00000
Afrosoricida	Tenrecidae	Microgale	dobsoni	36.66	94.49333

The MEAN imputation has added estimates based on a grouped mean in every missing cell for all common values. We can now train a new random forest using this data and see how well it performs.

This version uses a complete case analysis similar to model 4. It produces very few predictions, but they are even more accurate than a simple randomforest with NA's. This implies that the MEAN data is helping (or at least, not hurting) the accuracy when it can match rows of data in the test data with imputed mean values in the training data.

model	dataset	Predicted	Accuracy	F1_Mean
1- Baseline	With 0's	1	0.3844156	0.5553471
2- RPART Classification Tree	With 0's	13	0.7220779	0.6191113
3- RPART Classification Tree - Pruned	With 0's	12	0.7155844	0.5797862
4- Random Forest	With NA's	6	0.8913043	0.9329966
5- Random Forest	With 0's	15	0.8688312	0.7516451
6- Random Forest	With 0's, Oversampled	23	0.8935065	0.8312761
7- Random Forest	MEAN Imputation with NA's	7	0.9565217	0.9763906


Confusion Matrix: Random Forest − MEAN Imputation − Complete Case Analysis

Data source: Pantheria

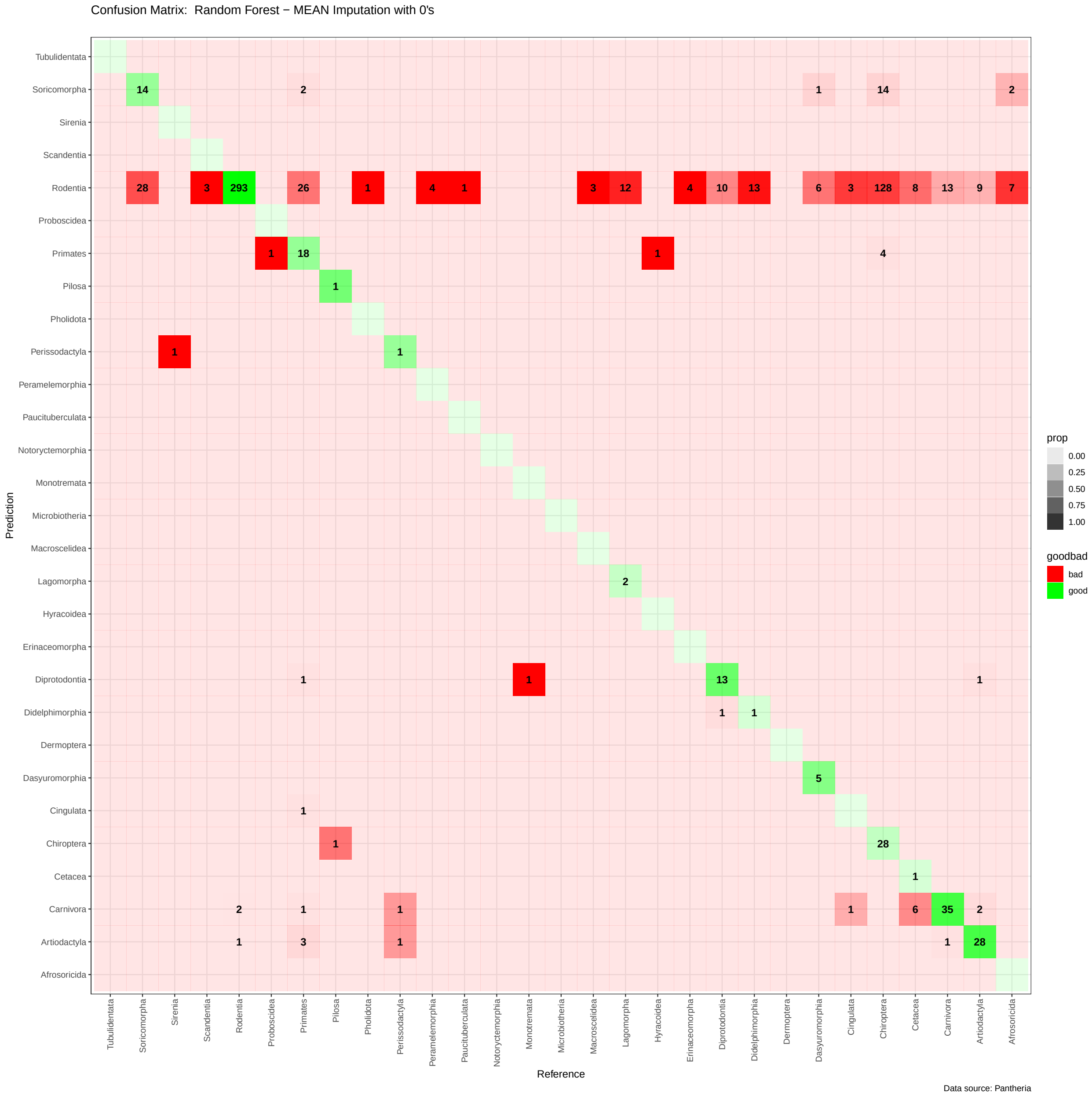
Changing the NA’s to 0 in the test data could increase outcomes - but it will probably reduce accuracy, since there will be many cases where a 0 imputed in the test data follows a decision tree rule based on a imputed mean estimate in the training data, and this will “throw off” the decision tree.

We have imputed 0 to the test data to try this out.

model	dataset	Predicted	Accuracy	F1_Mean
1- Baseline	With 0’s	1	0.3844156	0.5553471
2- RPART Classification Tree	With 0’s	13	0.7220779	0.6191113
3- RPART Classification Tree - Pruned	With 0’s	12	0.7155844	0.5797862
4- Random Forest	With NA’s	6	0.8913043	0.9329966
5- Random Forest	With 0’s	15	0.8688312	0.7516451
6- Random Forest	With 0’s, Oversampled	23	0.8935065	0.8312761
7- Random Forest	MEAN Imputation with NA’s	7	0.9565217	0.9763906

model	dataset	Predicted	Accuracy	F1_Mean
8- Random Forest	MEAN Imputation with 0's	13	0.5714286	0.4677925

As expected, the accuracy goes way down, because the imputed 0's in test data and imputed mean in the training data don't match well.



The challenge here, is that the test data can't be imputed in the same way as the training data- we don't know the genus, family or order of the test rows - so we can't impute based on these unknown factors. Any comparison between test data with 0's and *imputed* training data with estimates will lead to significant mismatches in the data.

However - using *imputed* data for training combined with a more complex, *available case* model may produce a "best of both worlds" result - with our informed guess for all fields in the training data being used as the basis of predictions - but ONLY training on the same combination of fields that are supplied in the test data.

The most obvious benefit of this approach, is that since we know that marine mammals never have geography data, and because only bats have forearm length - an available case model will allow use of these predictors, but ONLY when provided as part of the test data. It will also use of test data that has only a portion of the predictors provided. An available case model should allow more of an apples-to-apples comparison if a row in the test data has the same fields as a similar species in the training data- and even if some predictors are not present in the training data- the available case model will adapt by using the estimated mean values for those fields only.

Model Analysis

After all the foregoing model tests, the method that appears to give the greatest overall number of outcomes, and highest accuracy against our 20% internal test data is the oversampled data with 0 imputed.

The accuracy of this model is about the same as that of other models, but it is also producing a broader range of possible outcomes due to the oversampling of minority classes.

Other imputation strategies and use of available case analysis strategies may make this model more robust, at the cost of some accuracy.

We will now switch to the final encapsulation and testing of the model against the hold-out data.

Encapsulation of the Model

Now that the model has been developed, we will encapsulate the models into self-contained functions.

Predict Order - Basic Model

Here is a basic model that will predict the order and impute 0 if indicated and/or apply oversampling.

To explore a more sophisticated implementation of the model, that may be more robust and practical - we will now build a version that "drills down" to make predictions to the family and genus ranks, uses MEAN imputation, available case analysis, and oversampling.

Predict Genus - Available Case Model

This version of the model uses only available cases in the training data having all of the same predictors set. The MEAN data will have values set on every row for all predictors, either factual or estimated. The model will train a random forest and make a prediction for one test row at a time, using ONLY the list of columns specified in that one row of test data. It will then do the same process for the family, and the genus.

This is a bit complex to accomplish. Here's how it will work:

Round 1 - Order (least specific)

- First, we sample 1 row of data which includes, family and order and genus.
- Create a Training Set that removes family & genus columns leaving only the order
- Fit the RF Model for the Order rank
- Make a prediction for Order

Round 2 - Family (more specific)

- Filter the original training data to include only the family

- Create a Training Set that includes only the predicted order, with only the family column and predictors
- Fit the RF Model for the Family rank
- Make a prediction for Family

Round 3 - Genus (most specific)

- Filter the original training data to include only the genus
- Create a Training Set that includes only the predicted family, with only the genus column and predictors
- Fit the RF Model for the Genus rank
- Make a prediction for Genus

This approach enables us to predict across more than 53 outcome classes, which is the upper limit of the randomForest package. Since the correct *order* will be predicted about 80% of the time, this would be the maximum expected accuracy when predicting at the *family* or *genus* level. If the first guess was wrong the second must be wrong. However if it predicts the order right in the first round, then it should have a pretty good chance of getting the family right on the second step, etc. as it will use a more focused inter-class dataset to make this second and third layer of prediction.

This is the container for the actual model that runs through the test data one row at a time.

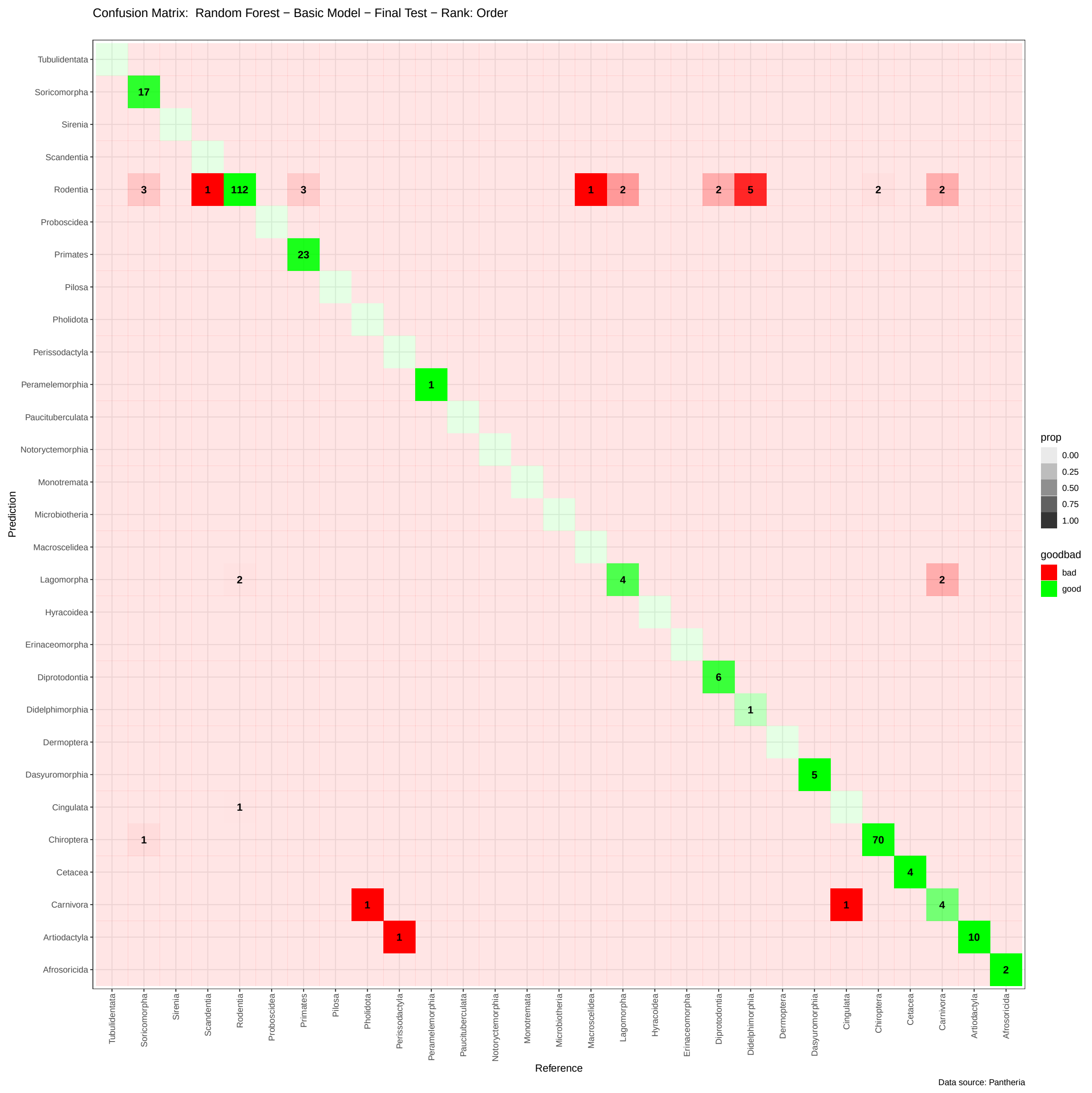
Results

Calling the Basic Model

The **predictorder** function will run the basic order-level prediction model with 0 imputed and all data columns used.

Assessing the Basic Model

We have run the basic model against our hold-out data set. Now we will examine how the basic model performed.



model	dataset	Predicted	Accuracy	F1_Mean
1- Baseline	With 0's	1	0.3844156	0.5553471
2- RPART Classification Tree	With 0's	13	0.7220779	0.6191113
3- RPART Classification Tree - Pruned	With 0's	12	0.7155844	0.5797862
4- Random Forest	With NA's	6	0.8913043	0.9329966
5- Random Forest	With 0's	15	0.8688312	0.7516451
6- Random Forest	With 0's, Oversampled	23	0.8935065	0.8312761
7- Random Forest	MEAN Imputation with NA's	7	0.9565217	0.9763906
8- Random Forest	MEAN Imputation with 0's	13	0.5714286	0.4677925
10- Random Forest	Basic Model - Order - FINAL	13	0.8961938	0.8426042

The basic model is able to predict the Order with overall accuracy of 0.8961938. This looks like very high accuracy, but as noted earlier - this may be the result of overfitting, particularly with respect to missingness

- as it would rely on the patterns of missingness being the same in a real world challenge scenario. We will now examine how the Advanced model compares on the same hold-out dataset.

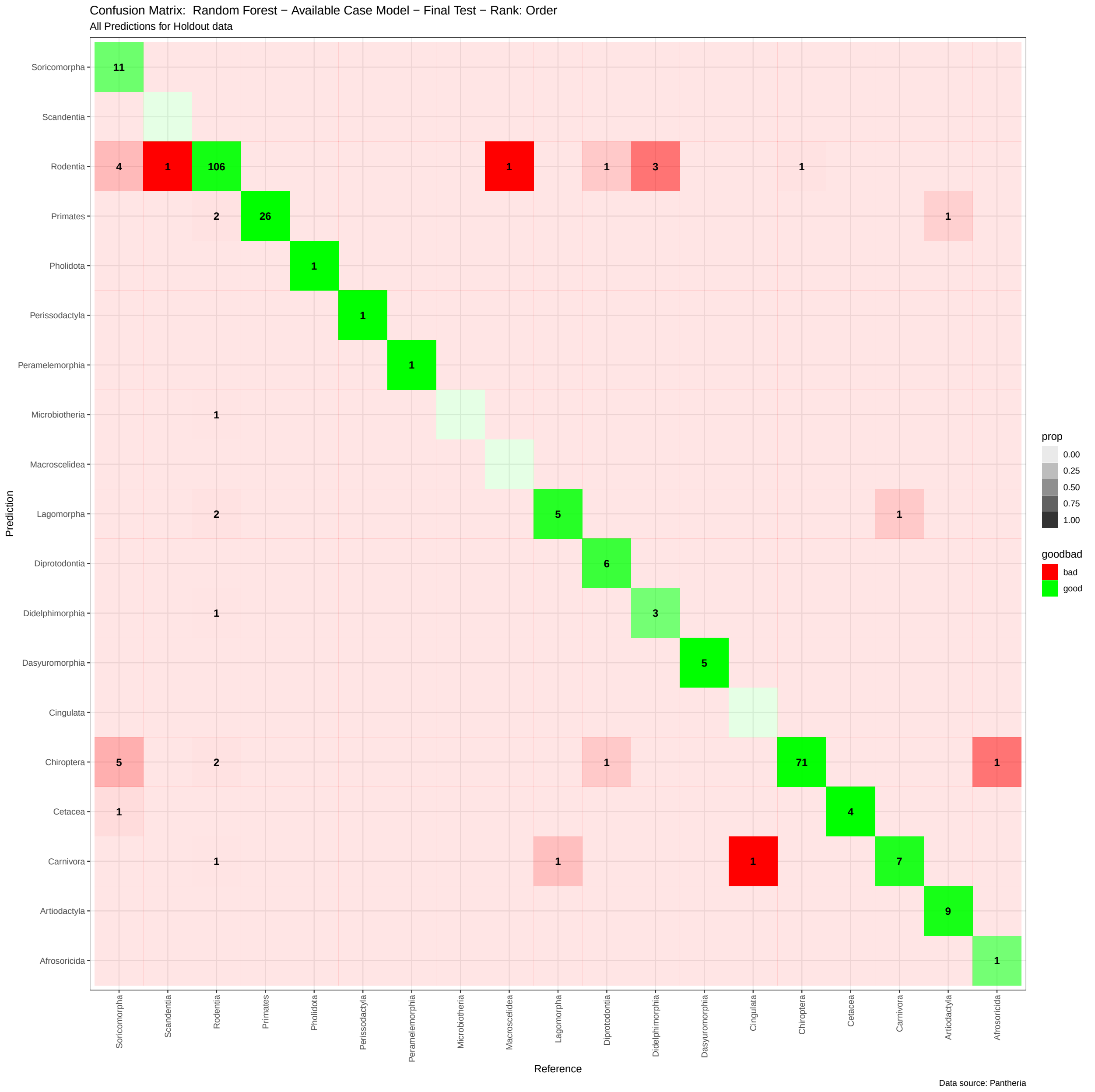
Calling the Advanced Available Case Model

The **acgenusmodel** function will run the final available case model. This model usually takes 10-15 minutes to run against the ~300 rows of hold-out testing data.

Randomly selecting 10 rows reveals how the model performed on just a few of the predictions. It looks like it has done pretty well and made a reasonable guess at both the Order and Family level in most cases, with more confusion at the Genus level, as expected.

pred_order	pred_fam	pred_genus	correct_order	correct_fam	correct_genus
Soricomorpha	Soricidae	Sorex	Soricomorpha	Soricidae	Sorex
Chiroptera	Vespertilionidae	Chalinolobus	Chiroptera	Vespertilionidae	Chalinolobus
Rodentia	Muridae	Pseudomys	Rodentia	Muridae	Melomys
Chiroptera	Hipposideridae	Hipposideros	Chiroptera	Molossidae	Otomops
Artiodactyla	Bovidae	Tetracerus	Artiodactyla	Bovidae	Gazella
Rodentia	Muridae	Gerbillus	Rodentia	Cricetidae	Microtus
Soricomorpha	Soricidae	Sorex	Soricomorpha	Soricidae	Sorex
Rodentia	Caviidae	Microcavia	Rodentia	Cricetidae	Tylomys
Artiodactyla	Bovidae	Ovis	Artiodactyla	Bovidae	Ovis
Rodentia	Cricetidae	Abrothrix	Rodentia	Cricetidae	Akodon

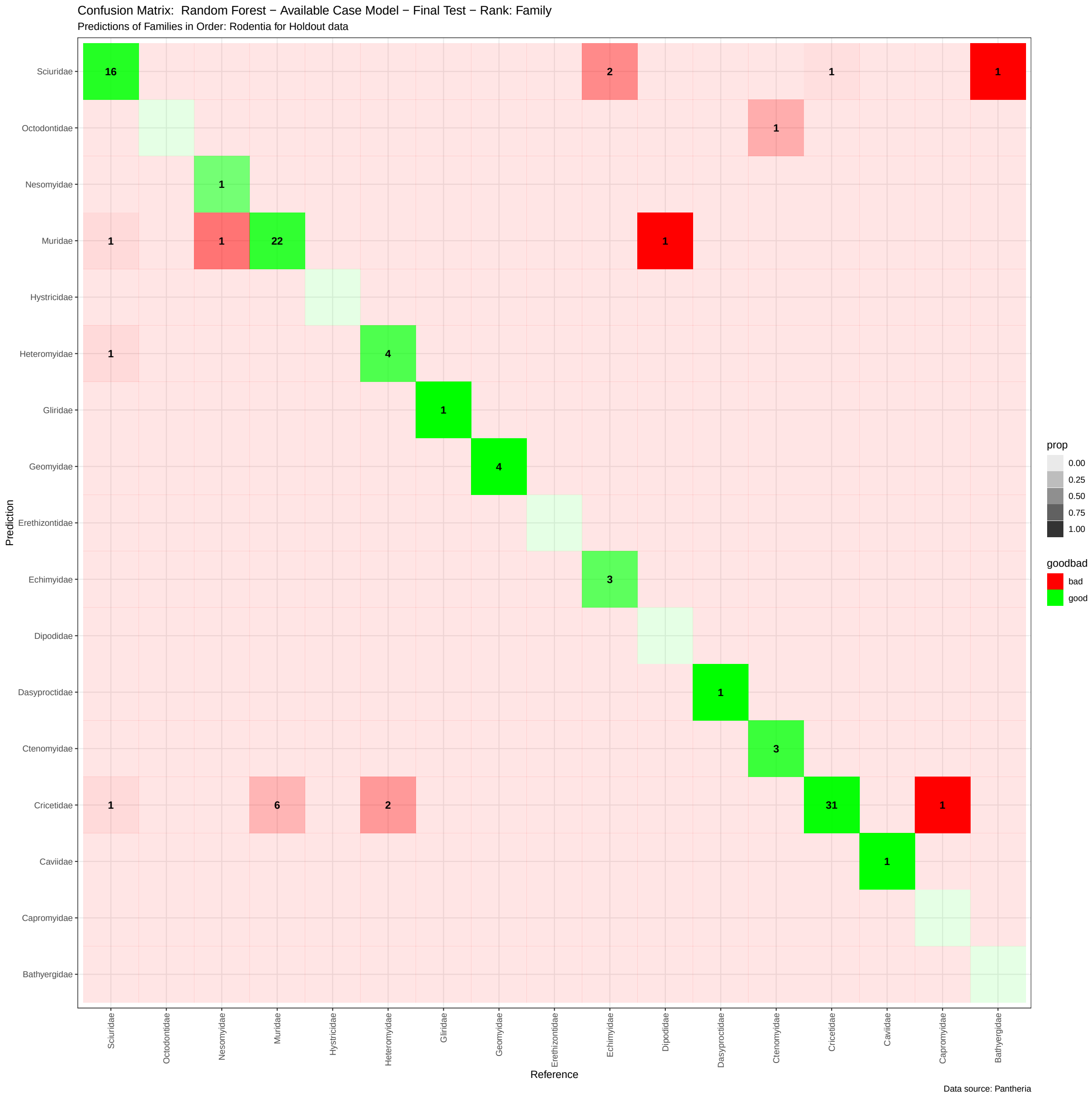
Having examined a few of the raw predictions, we will now use our familiar confusion matrix visualization charts to examine the outcome of the final advanced available case model. Since this model is predicting three different sets of factors, three confusion matrices will be necessary to examine the various ranks of Order, Family and Genus.


Confusion Matrix: Random Forest − Available Case Model − Final Test − Rank: Order

All Predictions for Holdout data

Data source: Pantheria

The plot above explores the prediction across all orders. We can drill down to the family rank and observe how the model has performed within one order. We will use Rodentia since it is the order with the most predictions.

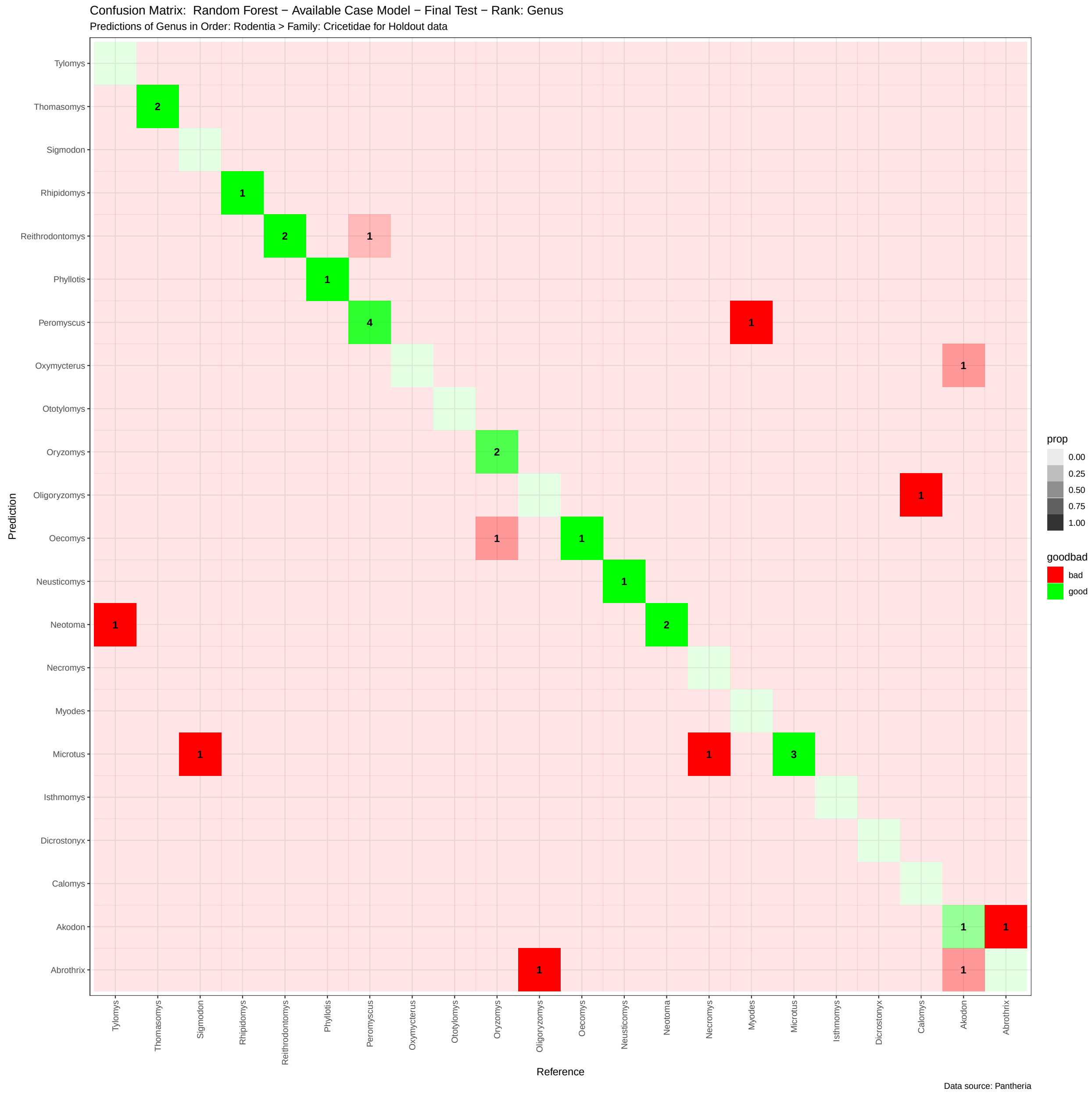


Confusion Matrix: Random Forest − Available Case Model − Final Test − Rank: Family

Predictions of Families in Order: Rodentia for Holdout data

Data source: Pantheria

And finally we can drill down to the Genus rank and observe how the model has performed within one family. We will use Rodentia > Cricetidae since it is the Family with the most predictions.



Assessing the Advanced Model

Now we will look at how the advanced model has performed across all predictions at each rank.

model	dataset	Predicted	Accuracy	F1_Mean
1- Baseline	With 0's	1	0.3844156	0.5553471
2- RPART Classification Tree	With 0's	13	0.7220779	0.6191113
3- RPART Classification Tree - Pruned	With 0's	12	0.7155844	0.5797862
4- Random Forest	With NA's	6	0.8913043	0.9329966
5- Random Forest	With 0's	15	0.8688312	0.7516451
6- Random Forest	With 0's, Oversampled	23	0.8935065	0.8312761
7- Random Forest	MEAN Imputation with NA's	7	0.9565217	0.9763906

model	dataset	Predicted	Accuracy	F1_Mean
8- Random Forest	MEAN Imputation with 0's	13	0.5714286	0.4677925
10- Random Forest	Basic Model - FINAL	13	0.8961938	0.8426042
11- Random Forest	Advanced Model - Order - FINAL	15	0.8892734	0.8622059
12- Random Forest	Advanced Model - Family - FINAL	50	0.7370242	0.8215850
13- Random Forest	Advanced Model - Genus - FINAL	102	0.4705882	0.8485562

The final accuracy on the hold-out data using available case analysis, mean imputation, and oversampling applied to the training data, is around 0.8961938 at the Order rank; 0.7439446 at the Family rank; and 0.4705882 at the Genus rank when making predictions against our final hold-out validation data. The model is predicting correctly 15 different orders out of the 18 distinct orders contained in the holdout test data. As shown in the table above, the mean F1 scores are also in the same neighborhood as the models we used in testing - so that is a good sign that we are getting a reasonable balance of precision and recall in the final model.

Conclusion

It was challenging to construct a model that provided good overall accuracy and made a diversity of predictions. The final model selected produces fairly high accuracy at the Order level, with expected decreases in accuracy as the model predicts at a more specific rank of taxonomy.

One of the challenges of this model was that we did not have very abundant data to begin with. Since each row in Pantheria represents one species, there were a limited number of rows to use, and it was necessary to use a strategic approach to partitioning the data to ensure a good cross section of rows would be randomly selected for training, test, and hold-out data sets.

The basic model provides a fast and simple way to predict to the order level. The accuracy is very high, but it is not very useful, and would rely on having test data with similar patterns of missingness. So that model is not very robust and may be somewhat overfitted, especially if it were used with real-world challenge data.

The available case model solves some of these problems, but at a steep cost in performance and run time. Because it must construct 3 different RF models per prediction, and because the RF model varies depending on the number of data points provided, it can only make one prediction at a time.

Performance Tuning

There may be room for further improvement through implementation of more complex imputation strategies on the training data to estimate missing values more precisely. The grouped mean assumes that taxonomical groups are homogeneous - but imputation based on linear regression may sometimes be more accurate. However the data already contains some extrapolated values - so one must be careful not to over-extrapolate from inferred data.

Implementation of a more sophisticated imputation strategy was beyond the scope of this report - but may be a good subject for further exploration. Addition of more data from other sources could also help fortify the training data available from Pantheria. Model tuning through optimization of the Random Forest parameters, or adjustment to the amount of oversampling may also enhance model performance.

Interactive Demonstration

In order to demonstrate an application of the model, a "Mammal Predictor" Shiny App was developed that utilizes the final, available case model to make a prediction for a user-entered challenge. This interactive shiny app takes various inputs values and returns the predicted order, family and genus based on any combination of provided inputs.

https://ryancooper.shinyapps.io/ShinyZoo/

Commentary

I hope this report has demonstrated a viable approach for applying random forests to hierarchical taxonomy data, while addressing some of the issues of sparseness and imbalanced classes. It was my goal to demonstrate through this capstone project how I have come to understand and appreciate the R language, and how I have learned to implement lessons from the various courses throughout the HarvardX Data Science Professional Certificate program.

This was a fascinating and fun experiment to try to solve several challenges in this dataset. The remarkable diversity of life on Earth cataloged through Pantheria and GBIF provides an extensive source of information - though not always a complete picture. It is quite awe-inspiring to consider that the Pantheria data was derived from over 3000 independent research sources - this project owes a huge debt of gratitude to the many dedicated researchers and data entry technicians who painstakingly compiled this collection of data.

This dataset shows off the diversity of nature, and provides ample opportunities to explore the incredible variety of traits observed for the thousands of unique, individual species in the class of Mammalia. The biodiversity of Earth is a precious resource, and it is our responsibility to study, protect and preserve the many unique species with which we share the planet.

References

Rafael Irizzary, Introduction to Data Science (2020), github page, https://rafalab.github.io/dsbook/

Leo Breiman, Random Forests (2001), Statistics Dept. University of California, Berkeley https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

Chao Chen, Using Random Forest to Learn Imbalanced Data, Statistics Dept. University of California, Berkeley https://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf

Andrew Gelman; Jennifer Hill, Missing-data imputation, Data Analysis Using Regression and Multilevel/Hierarchical Models, (2006) Cambridge University Press

E. Jones, et. al., PanTHERIA: a species-level database of life history, ecology, and geography of extant and recently extinct mammals., (2016): Ecological Society of AMerica http://esapubs.org/archive/ecol/E090/184/metadata.htm

PanTHERIA is Made avaiable under Creative Commons 0. To the extent possible under law, the authors have waived all copyright and related or neighboring rights to this data.

Global Biodiversity Information Facility: Free and open access to biodiversity data - https://www.gbif.org/

Key Packages:

DPLYR package https://cran.r-project.org/web/packages/dplyr/index.html

GGPlot2 package https://cran.r-project.org/web/packages/ggplot2/index.html

MICE package https://cran.r-project.org/web/packages/mice/index.html

RGBIF Package https://cran.r-project.org/web/packages/rgbif/index.html

RPart Package https://cran.r-project.org/web/packages/rpart/index.html

Using Plot CP in RPart https://www.rdocumentation.org/packages/rpart/versions/4.1-15/topics/plotcp

R Documentation https://www.rdocumentation.org