Animal max-longevity project report

Tang Li

Mar 09, 2022

Contents

Note: All the links are colored in* violet *(without underline), feel free to click and navigate around the pdf!

Introduction

About this project

In this machine learning project, I'll solve a regression problem of predicting animal max-longevity, which is the *recorded* highest lifespan for each species (e.g. Bowhead whale: 211 years). Since this is a regression problem, the goal is to achieve a low rmse (root mean squared error). The requirement is to develop at least 2 models with 1 model more advanced than linear regression.

My approach is to create ensembles: combining results of different methods into one that hopefully improves the result. I developed two ensemble models. The first ensemble is linear, combining cubist and lm. The second ensemble is non-linear, combining rf and rpart. The results of both ensembles are similar: linear achieved a rmse of 8.31 years, and non-linear achieved a rmse of 8.91 years. I think my results are good considering animals max-longevity could range from about 2 to 200 years.

Dataset

I picked the curated animal max-longevity dataset from AnAge, and from here on we'll call it the **age** dataset. AnAge recorded many species max-longevity and their features. The age dataset has 4219 rows and 31 columns. Each row represents a species, and each column is a feature.

Due to some rows and columns have large number of NAs, I'll clean the dataset, **only using a subset of the data**. The cleaned dataset has 996 rows and 7 columns. All details about original dataset and data cleaning is in the Methods data cleaning section. Here we show the final list of variables in the cleaned dataset, and preview 1 row to better visualize.

Class	Common name	Maximum longevity (yrs)	Adult weight (g)	Female maturity (days)	Gestation/Incubation (days)	Litter/Clutch size
Mammal	Bowhead whale	211	1e+08	8212	396	1

info

Variable	Description
Class	taxonomy class, bird or mammal
Common name	name we are familiar with, e.g. Golden eagle

outcome (usually in a list, referred to as y)

Variable	Description
Maximum longevity (yrs)	the *recorded* highest lifespan for each species

Note: emphasis on recorded because animals could live above its species max-longevity, but its age is just not recorded in dataset yet

predictors (usually in a matrix, referred to as x)

Variable	Description
Adult weight (g)	adult weight in grams
Female maturity (days)	female age of sexual maturity
Gestation/Incubation (days)	pregnancy time mammals gestate litters, birds incubate eggs
Litter/Clutch size	number of young (litters or eggs) in 1 birth

Key steps

The objective is to develop the 2 final ensemble models, and see how well the final rmses are. A model is characterized by its tuning parameters and training data. During tuning stage we find parameters that minimize rmse. During training stage we specify which training data to use, and the more data the better.

1. Data cleaning
   - find good predictors and ensure our data has no missing data (NAs)
2. Data splitting
   - split our data into 3 sets: training, test and validation
   - training set (available): tuning and training
   - test set (available): evaluate intermediate models
   - validation set (hold out): evaluate final 2 models
3. Model development
   - Tune and train with different methods using training set
   - Combine methods and create 2 ensembles, 1 linear and 1 non-linear
   - Evaluate all methods using test set
4. Final test
   - Since more data is better, put train and test sets back together to form the available set
   - Train the 2 final ensembles using available set
   - Evaluate 2 final ensembles using validation set

Methods

Data cleaning

> The original 31 variables specification is in appendix original dataset section. There I describe what each variable represents, show the 5-number summary and number of NAs.

To use the caret package, **my cleaned dataset must have no NAs** because otherwise the train function would fail. A not-so-good alternative is to let caret omit the rows with NAs on required variables, but I won't know which rows have been omitted when computing rmse.

Which rows to pick

Even though AnAge included many classes of animals, I'll only analyze **birds and mammals**, due to other animals classes (amphibian, reptile, fish) have too many NAs in their features. Also I've kept only rows that meet quality standard. To summarize:

- Class: keep bird and mammal (renamed from Aves, Mammalia), remove all other classes
- Specimen origin: keep captivity and wild animals, remove ones with unknown origin
- Sample size: keep sample size $n \geq 10$ (small, medium, huge categories), remove rows with $n < 10$ (tiny category)
- Data quality: keep rows that have acceptable and high confidence of max-longevity, remove rows with low or questionable confidence

Which predictors to pick

Now our cleaned set only has rows of birds and mammals, but it's still not NA free. The final cleaned dataset must have no NAs. The challenge is that our two goals conflict each other:

1. Goal: Keep as many predictors as possible, since more predictors generally yield better result
2. Goal: Keep as many rows as possible, since more training data also yield better result
3. Conflict: the more predictors we **require** (i.e. a row must have no missing data for any predictor column), the less rows we have for training data

Step 1 is to decide on a set of candidate predictors that guarantee enough animals for training. Step 2 is to assess the candidate predictors for correlation. Finally we filter rows that have no NAs for the required predictors.

Step 1: find candidates At this stage we don't require rows to have no NAs, since if a candidate is dropped at the end, we would have required more than we need, which leads to fewer rows left.

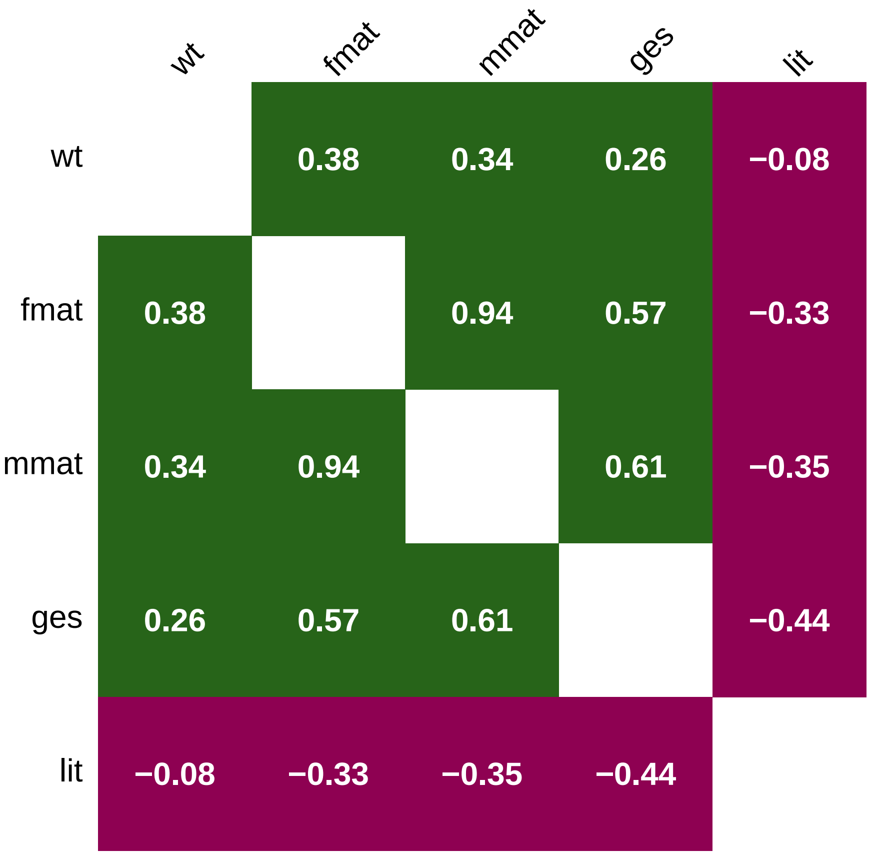
I decided to *initially* include these 5 life history predictors that guarantee enough animals and good result:

- **Adult weight (g)**: A heavy weight/larger body size could protect animal from predators, may have greater chance to survive and might increase longevity.
- **Female maturity (days), Male maturity (days)**: If an animal lives long, then it might not need to reach (sexual) maturity soon and its pre-adulthood is longer, and vice versa.
- **Gestation/Incubation (days)**: Just like maturity, an animal's pregnancy duration might also be a biological indicator to its longevity. Birds often take less time to incubate eggs than mammals gestate litters.
- **Litter/Clutch size**: Animal produce more litters/eggs if not all young can survive. Less chance to survive might cause shorter longevity.

Step 2: assess candidates Caret is a powerful machine learning library which we will use through out this project. The first tip from caret is to **eliminate predictors with near-zero variance**, because otherwise models may crash. The table below shows nzv column has all FALSE, which means that none of our predictors have near-zero variance.

	freqRatio	percentUnique	zeroVar	nzv
Adult weight (g)	1.10	67.85	FALSE	FALSE
Female maturity (days)	3.20	17.80	FALSE	FALSE
Male maturity (days)	2.94	12.66	FALSE	FALSE
Gestation/Incubation (days)	1.32	13.88	FALSE	FALSE
Litter/Clutch size	2.17	5.03	FALSE	FALSE

The second tip is to **remove highly correlated predictors** to improve model performance, and I set the correlation cutoff to be ≤ 0.75. We can visualize the correlation matrix in a plot (*variable names are abbreviated to save space*). Male maturity (mmat) is highly correlated with female maturity (fmat) with correlation = 0.94. In this case, caret removes the variable with the largest mean absolute correlation out of the pair, and outputs male maturity.



	wt	fmat	mmat	ges	lit
wt		0.38	0.34	0.26	−0.08
fmat	0.38		0.94	0.57	−0.33
mmat	0.34	0.94		0.61	−0.35
ges	0.26	0.57	0.61		−0.44
lit	−0.08	−0.33	−0.35	−0.44	

```
## Compare row 3  and column  2 with corr  0.94
##   Means:  0.56 vs 0.398 so flagging column 3
## All correlations <= 0.75
```

```
## [1] "Male maturity (days)"
```

Step 3: final dataset The final predictors left are **Adult weight (g), Female maturity (days), Gestation/Incubation (days), and Litter/Clutch size**. We filter rows with data for all 4 predictors. The final age dataset has 996 rows and 7 columns. The table below shows how many birds and mammals we have.

Class	n
Bird	382
Mammal	614

Data splitting

The age dataset is split into 3 sets: training set, test set, and validation set. First we split age into available set (80%) and validation set (20%). Then we split available set into training (80%) set and test set (20%). Training set is used to develop models, and test set is used to evaluate model performance. Validation set is only used to evaluate the final 2 models. Both splits are 80%-20%, because our age dataset is small (only ~1000 rows) I want to leave plenty of validation data. We have 641 rows for training set, 158 rows for test set, and 197 rows for validation set.

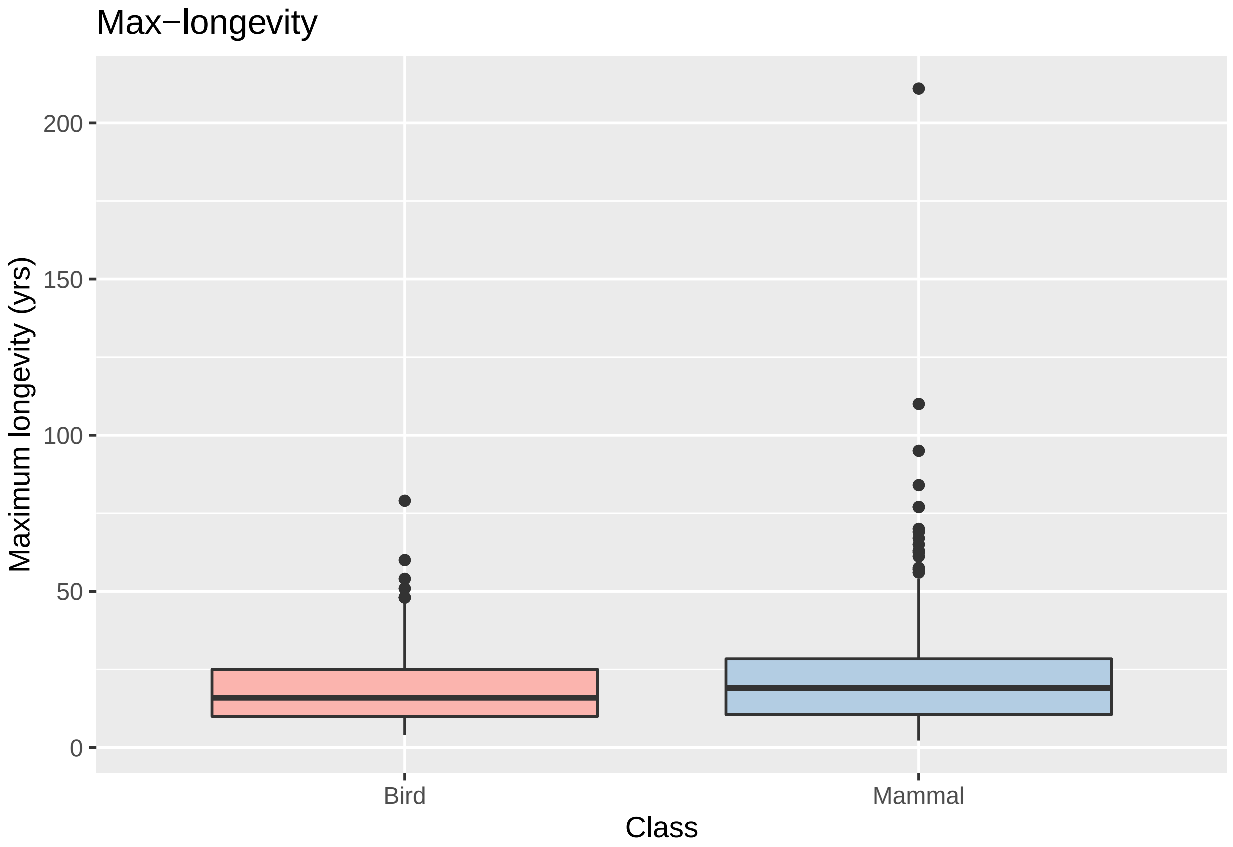
Data visualization

We'll use training set to visualize our data. The first set of plots give us some background information on birds and mammals. The second set of plots explores relationship of each predictor and max-longevity, which will give us insight on potential methods to choose.

Background

Max-longevity: The boxplot and 5-number summary below compares the max-longevity between birds and mammals. Birds have lower median and smaller range compared to mammals. Mammals also have many outliers on the higher end.

```
##    0%   25%   50%   75%  100% 
##   2.2  10.0  17.9  27.0 211.0
```



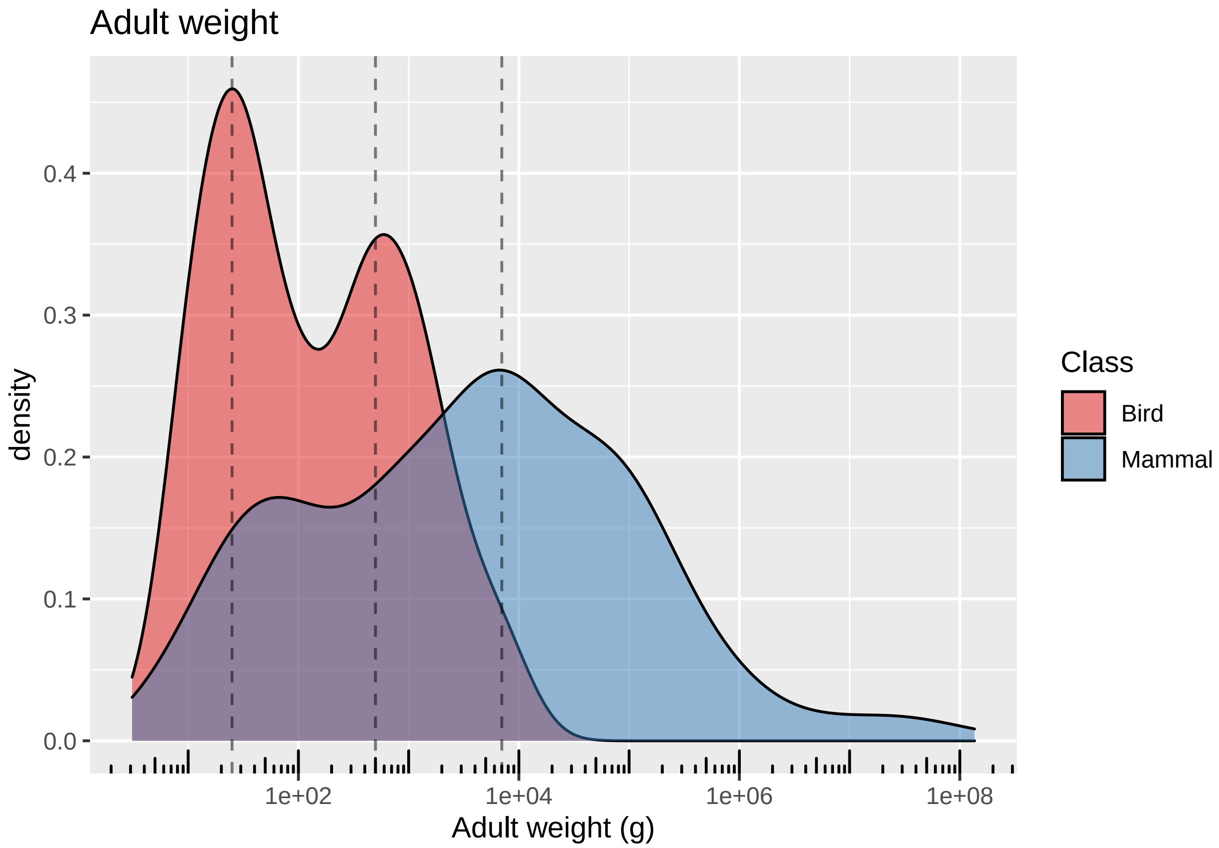
Let’s inspect birds and mammals with longest and shortest lifespan. We can see that long living species are **heavier, mature later, and gave birth to less young**. In contrast, shorter lifespan species are **lighter, mature earlier, and gave birth to more young**.

Class	Common name	Maximum longevity (yrs)	Adult weight (g)	Female maturity (days)	Gestation/Incubation (days)	Litter/Clutch size
Bird	Andean condor	79	1.05e+04	2555	56	1
Mammal	Bowhead whale	211	1.00e+08	8212	396	1

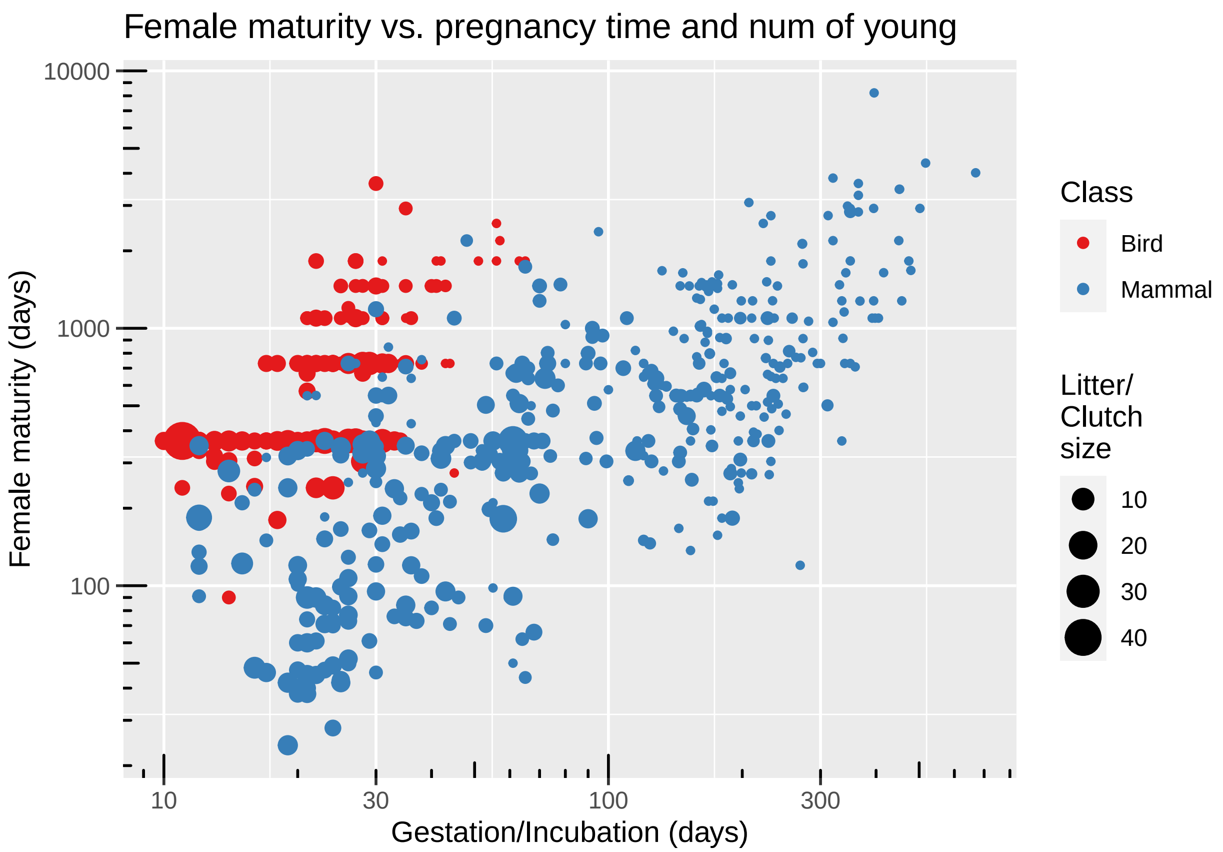
Class	Common name	Maximum longevity (yrs)	Adult weight (g)	Female maturity (days)	Gestation/Incubation (days)	Litter/Clutch size
Bird	Bachman’s sparrow	3.9	19.6	365	13	4
Mammal	Northern short-tailed shrew	2.2	21.6	46	17	6

Adult weight: Here’s a density plot showing the distribution of adult weight of birds and mammals.

Overall birds are lighter than mammals. Birds density curve show two peaks around 25 g and 500 g. Mammals show a small bump around 50 g and a large peak around 7000 g (7 kg).



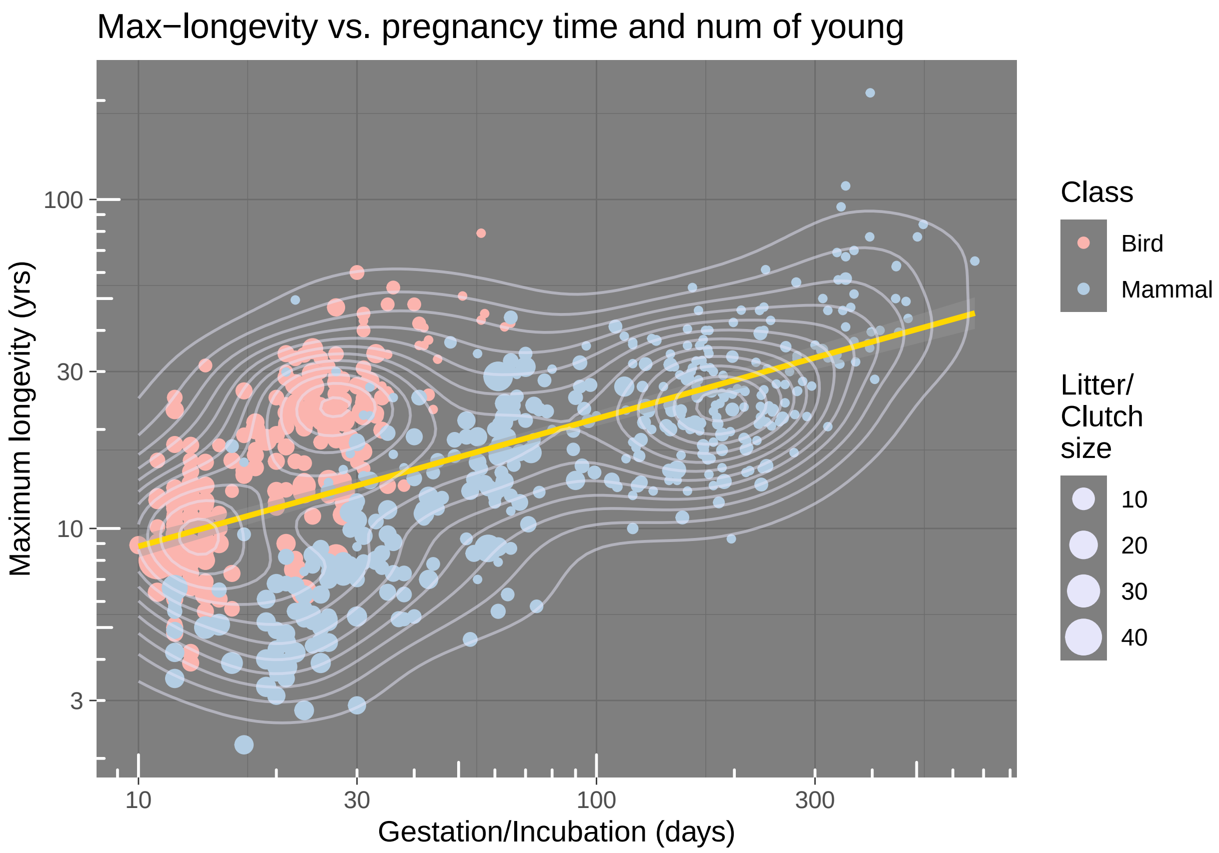
Female maturity vs. pregnancy time and num of young: We an see distinct 2 groups of points corresponds to birds and mammals. As we increase pregnancy time, the less number of young are produced. Birds generally have shorter pregnancy duration than mammals. With the same pregnancy time, birds mature later than mammals.



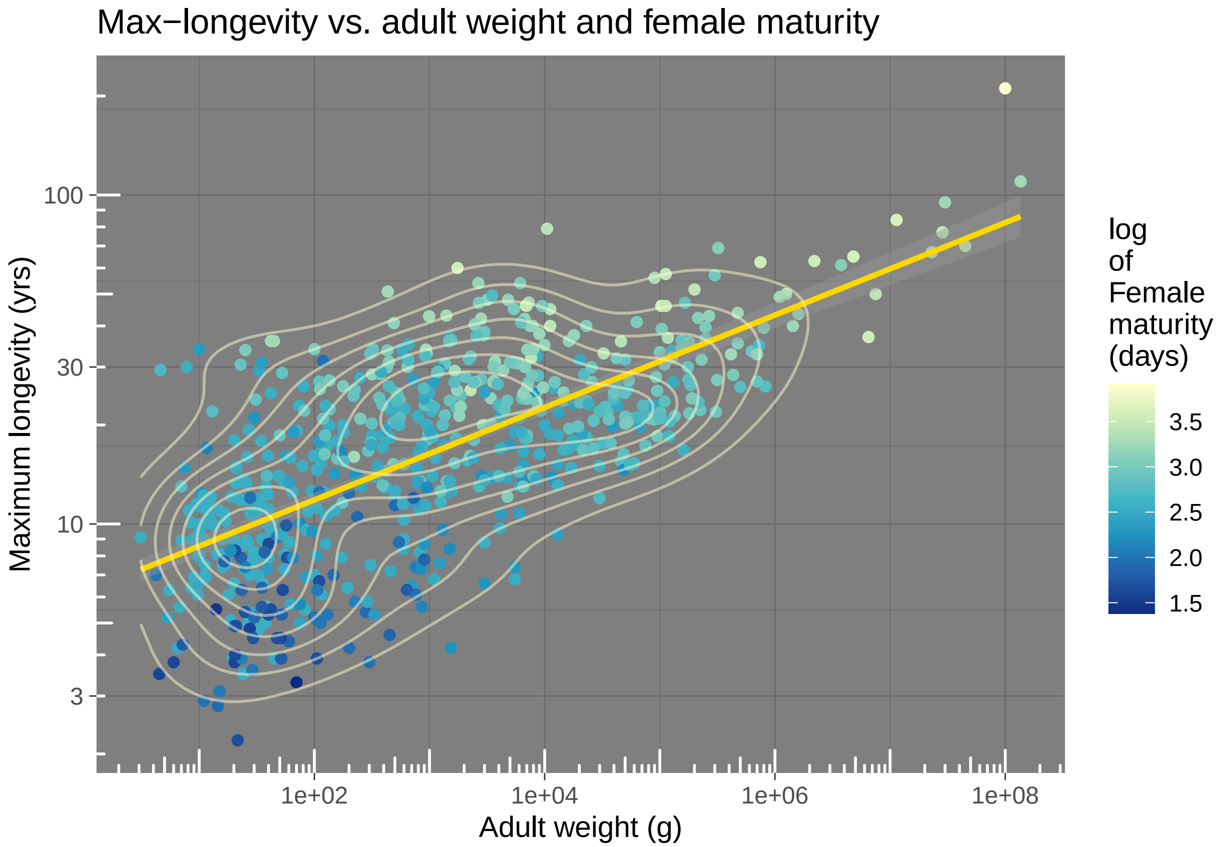
Predictors and outcome

Max-longevity vs. pregnancy time and num of young: This scatter plot shows relationships between max-longevity and two variables: positive relationship for pregnancy time, and negative relationship for number of young. The gold lm line runs nicely through the points, so linear equations are reliable. The density contours show 3 dense areas, and points group themselves into clusters. This suggests that non-linear methods' grouping techniques could work as well.

Dense area	(pregnancy time, max-longevity)
Bird	(15 days, 10 years)
Bird	(25 days, 20 years)
Mammal	(200 days, 25 years)



Max-longevity vs. adult weight and female maturity: This scatter plot shows that max-longevity has positive relationship with both adult weight and female maturity. Similar to last plot, the gold lm line shows a good fit so linear methods are suitable. The density contours show 2 dense areas, and points are tightly packed. Thus non-linear methods could be useful too.



Model development

To stick with caret's conventions, I format the datasets as follows:

- the 4 predictor columns form the **predictor matrix** x
- the max-longevity column forms the **outcome list** y

Caret also supports parallel computation to increase speed, which I have set up using doParallel library (see Rmd file).

Caret functions

Here is a brief introduction about model development using caret, and we will walk through again later when we train our first model. There are two key caret functions we will use: `train` and `predict`.

train

- Purpose: Even with the same method, we can change its behavior by supplying different values to tuning parameter(s). The objective is to find the best tuning combo that has the min rmse.
- Input:
  - training data: predictor matrix x and outcome y

  - tuning grid: supply each tuning parameter with a list of values to test, cross join generates all tuning combos
    - e.g. committees = 1 and 2, neighbors = 3 and 4
    - cross join (committees, neighbors) = (1, 3), (1, 4), (2, 3), (2, 4)
  - method: the method we want to use (e.g. cubist)

- Output:
  - result: the tuning result, all combos with their *training* rmse (*training* because it's computed from training set)
  - best tuning combo: the combo with the min *training* rmse (e.g. committees = 2, neighbors = 4, training rmse = 7.5)
  - model: the model developed using best tuning combo

predict

- Purpose: predict outcomes on test data
- Input:
  - test data: predictor matrix x
  - model: the model from train function
- Output:
  - predicted outcomes $\hat{y}$

Once we have predicted outcomes $\hat{y}$, we can compute *test* rmse by comparing with true outcomes y. The test rmse is important because it shows how well model works with new data. On contrary, the *training* rmse is not as important, because it is computed with samples from training set (also same data to train the model). Hence training rmse is usually lower compared test rmse.

Linear models

All linear methods generate linear equations. In our case, we have 4 predictors and each predictor x_i is multiplied with a coefficient c_i plus an intercept b at the very end, and $\hat{y}$ would be our predicted max-longevity.

$$\hat{y} = c_1 \cdot x_1 + c_2 \cdot x_2 + c_3 \cdot x_3 + c_4 \cdot x_4 + b$$

There are 2 linear methods we want to test: **cubist and lm**.

Cubist Instead of producing 1 model containing 1 linear equation, Cubist can generate many linear models containing many equations. For each model, there are rules that determine which equation to use.

> For example, a model might look like: *if* weight > 200 *then* use equation 1, *if* (some other condition) *then* use equation 2, etc.

Cubist will obtain a list of predictions from each model, and uses an internal function accounting all models to produce the final $\widehat{y}$ list.

Cubist has two tuning parameters: $committees$ (number of models) and $neighbors$ (for adjusting equations). Let's walk through caret's algorithm use cubist as an example. Once we understand how caret works, we won't need much explanations or show code for subsequent models.

```r
set.seed(1)
fit <- train(
  x = train_set$x,
  y = train_set$y,
  method = "cubist",
  tuneGrid = expand.grid(
    committees = seq(1, 9, 2),
    neighbors = seq(5, 9, 1)
    )
)
```

Tuning param	Values	Values expanded
committees	seq(1, 9, 2)	1, 3, 5, 7, 9
neighbors	seq(5, 9, 1)	5, 6, 7, 8, 9

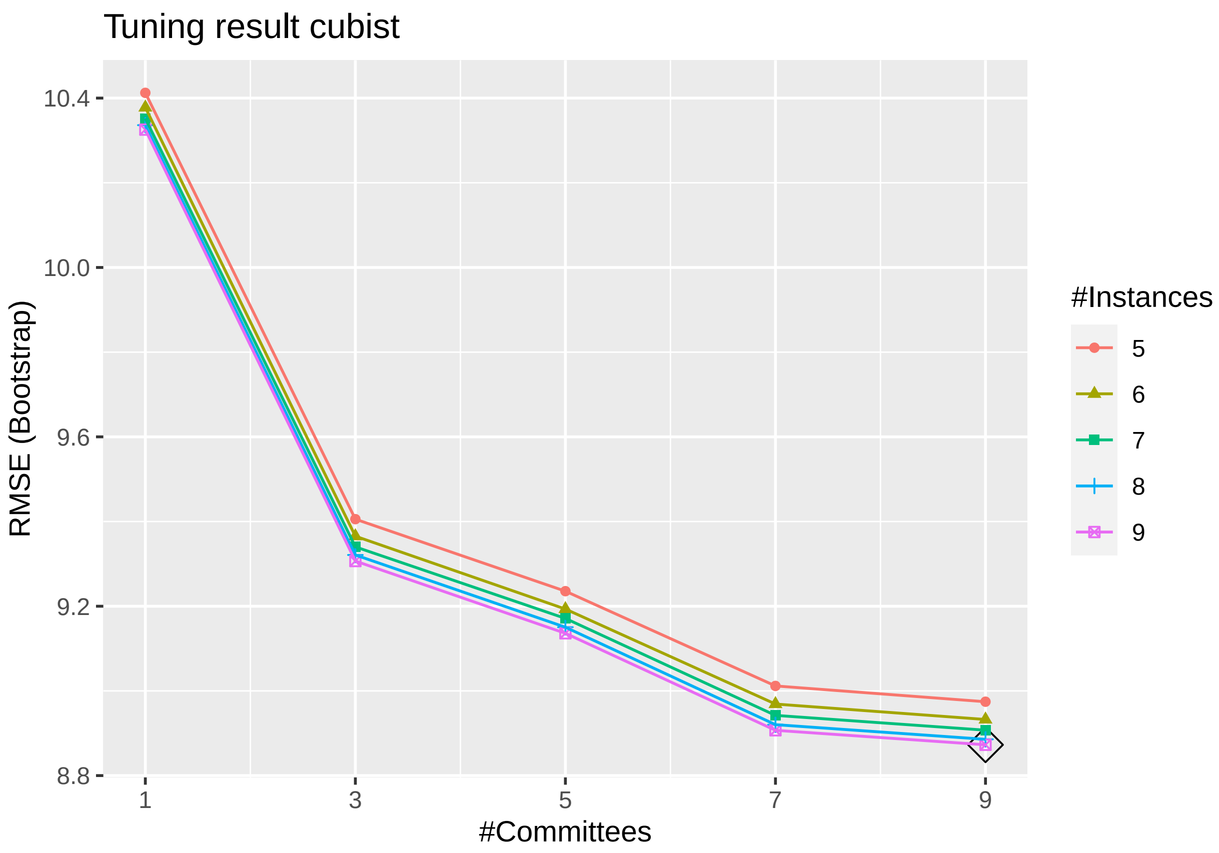
For each tuning combo (runs $5 \cdot 5 = 25$ iterations)

- Bootstrap is caret's default resampling method. Our training data is sampled with replacement to generate 25 C-train and C-test sets. (*The C prefix avoids the naming conflict with our datasets.*)
- **For each pair of C-train and C-test** (runs 25 iterations)
  - Caret fits model with C-train, and evaluates rmse with C-test.
- Evaluate *training* rmse = the average of the 25 rmses for this tuning combo

Select the tuning combo with the min *training* rmse

Here is the best tuning combo (ignore the row number on the left), and the tuning result plot.

	committees	neighbors
25	9	9



Then we predict max-longevity using predict function and evaluate test rmse.

method	rmse
cubist	7.18

lm Lm is a simple method but its performance is usually really good. There are no tuning parameters for lm. We can see the coefficients for each predictor. The rmse is a bit better than cubist.

```
##                  (Intercept)            `Adult weight (g)`
##                     9.82e+00                      5.91e-07
##      `Female maturity (days)` `Gestation/Incubation (days)`
##                     1.35e-02                      2.53e-02
##          `Litter/Clutch size`
##                    -2.60e-01
```

method	rmse
cubist	7.18
lm	7.07

Ensemble To create an ensemble we can take the mean of predicted $\hat{y}$ from our 2 methods, cubist and lm. The ensemble's rmse is better than both our models.

method	rmse
ensemble	6.91
lm	7.07
cubist	7.18

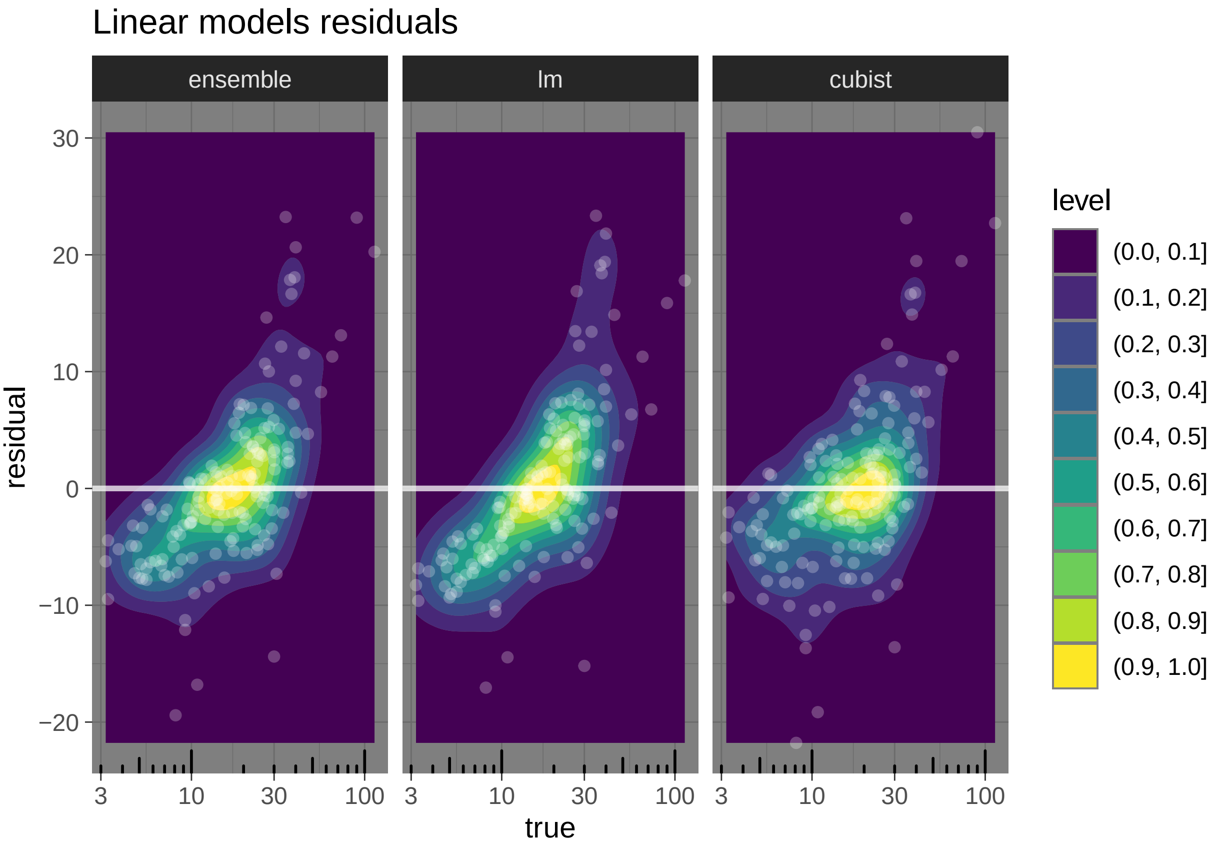
Here's a residual plot showing predicted vs. true max-longevity. The residual r is the difference between true y and predicted $\hat{y}$ outcome. We want our residuals to be as close to 0 as possible.

$$r = y - \hat{y}$$

There are 3 cases:

1. $r = 0$ means a perfect prediction. The point would be on the center line.
2. $r > 0$ means an *under*estimate. The point would be above the line.
3. $r < 0$ means an *over*estimate. The point would be below the line.

The 2d plot shows how dense the points are in a contoured area. The lighter the color, the more dense. The ensemble takes an averaged shape of cubist and lm. The super dense regions of all 3 methods are centered on the line. The plot shows that the residuals are usually within 10 years.



We ordered the residuals to get the top 3 mistakes made by the ensemble. The top mistakes are underestimates (since $r > 0$) , and all 3 residuals are around 20 years.

Class	Common name	y	residual
Bird	Rock dove	35	23.2
Mammal	Killer whale	90	23.2
Mammal	Indian flying fox	40	20.6

Non-linear

We will look at rf and rpart which are popular non-linear methods. Both methods are tree based.

rpart Rpart stands for *recursive partitioning*, and it generates a binary decision tree. Each node is a split on a variable that improves the result the most, and rpart recursively partitions to create nodes. Finally the partitioning stops, and leaf nodes are predicted outcomes $\widehat{y}$.

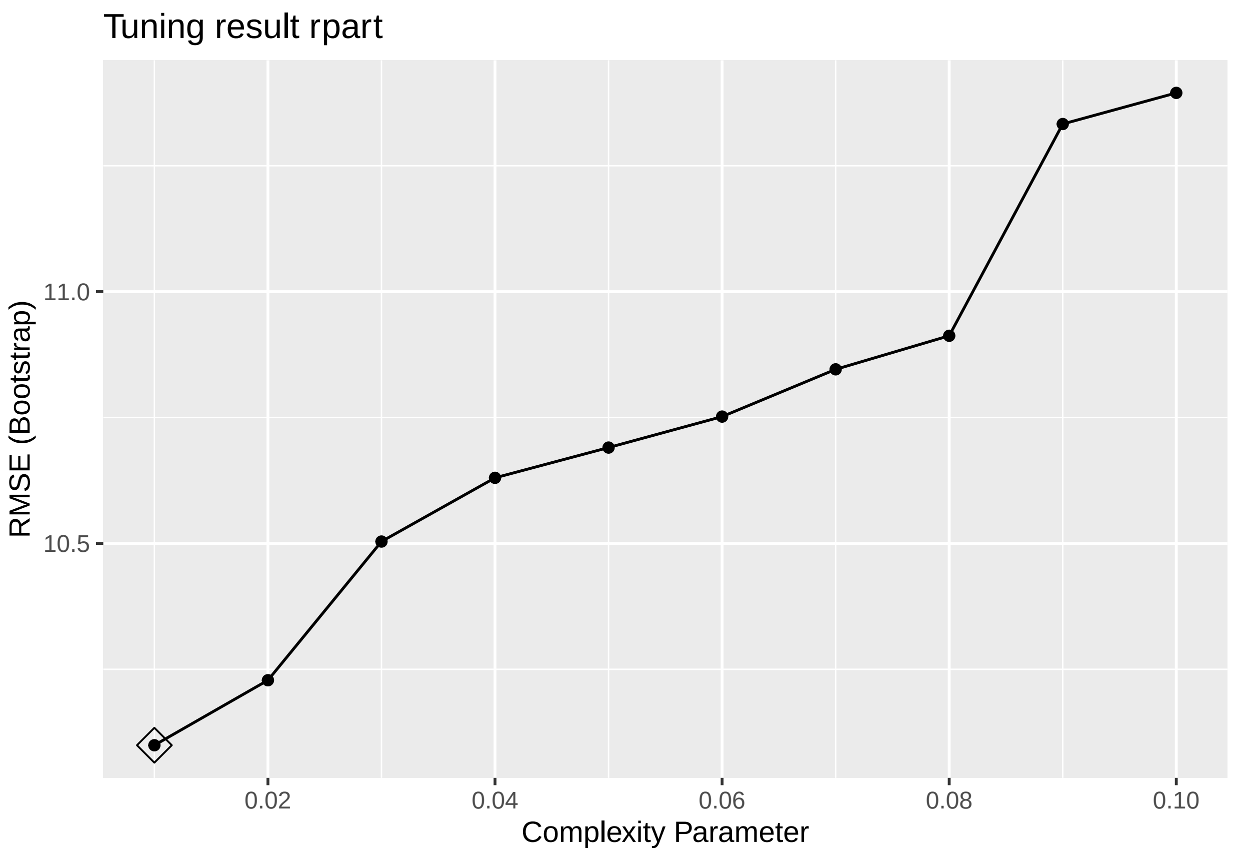
> For example, a split might look like $maturity < 800$. Data fall into either *yes* or *no* branch, then splitting continues.

Rpart stop partitioning when can't improve result by cp amount, and cp is a tuning parameter.

Tuning param	Values	Values expanded
cp	seq(0.01, 0.1, 0.01)	10 values: 0.01, 0.02, 0.03, ..., 0.1

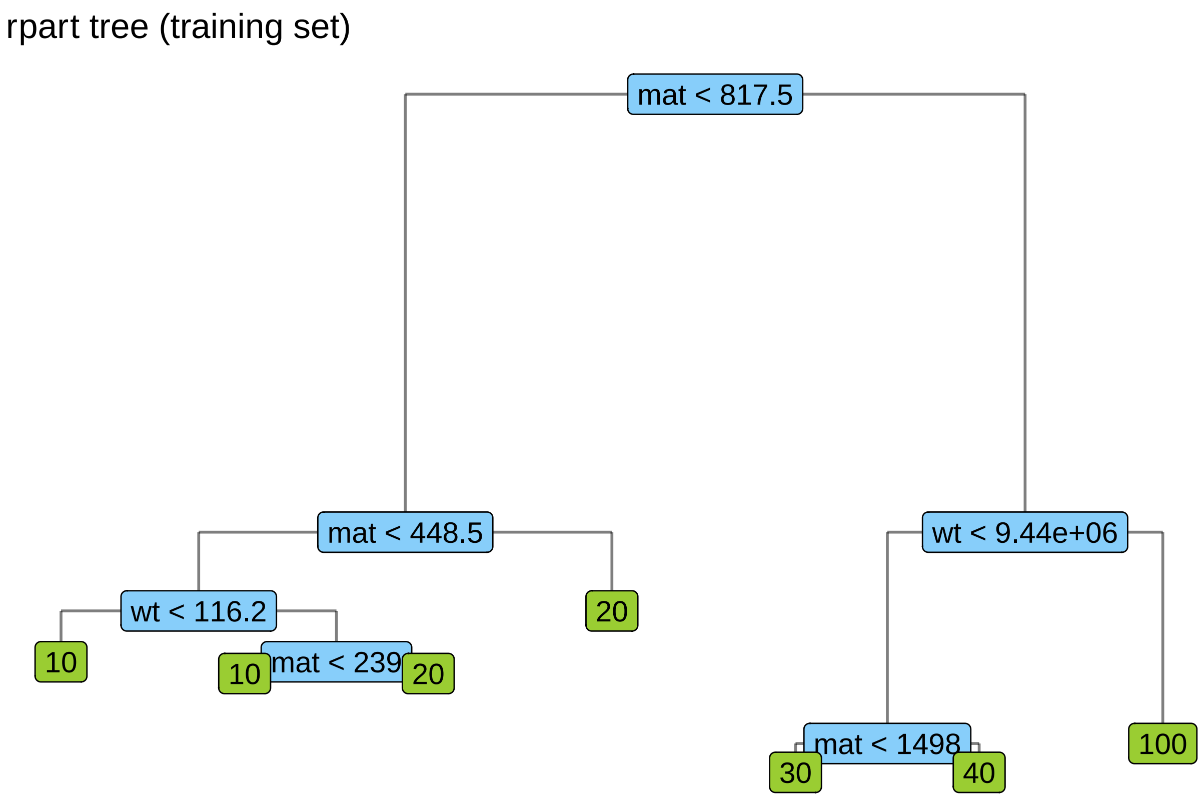
Here is the best cp.

cp
0.01



method	rmse
rpart	8.54

The rmse of rpart is higher than linear models. To see why, let's look the decision tree. It's interesting that there are only a few groups, and all predicted years are in multiples of 10.



rf Rf stands for random forest. Like its name, random forest produces not just one tree, but hundreds. The predicted $\hat{y}$ will be account for predictions from all trees. Unlike rpart, rf *randomly* chooses predictors to split. The number of predictors $mtry$ is a tuning parameter, and for regression rf suggests $mtry = \sqrt{npredictors} = \sqrt{4} = 2$. So we will plug in $mtry = 2$ and won't tune.

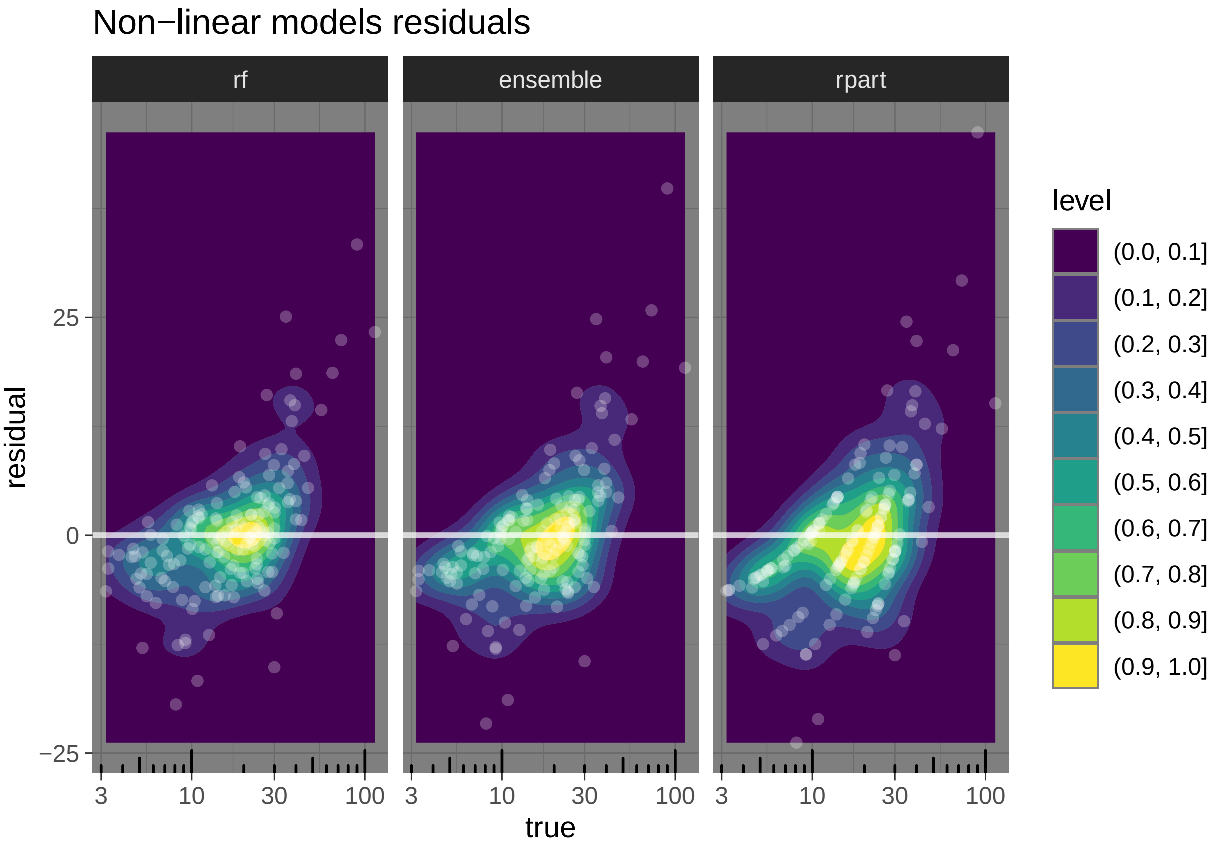
The rmse is better than rpart because rf has more decision trees.

method	rmse
rpart	8.54
rf	7.40

Ensemble We use the same approach to create ensemble, taking the mean of rpart and rf predictions. The ensemble's rmse is not ranked the top, but since it's only a bit worse than rf, we will choose the ensemble as final model. Maybe with a different train-test split the ranks would change, so using ensemble is safer.

method	rmse
rf	7.40
ensemble	7.81
rpart	8.54

Looking at the plot, all three methods have super dense region centered on the line. The residuals are usually within 10 years, and grow larger (20, 30 years) as true max-longevity increases.



Similar to linear ensemble, non-linear ensemble's top mistakes are underestimates (since $r > 0$), but the mistakes are larger compared to linear.

Class	Common name	y	residual
Mammal	Killer whale	90	39.8
Mammal	Dugong	73	25.8
Bird	Rock dove	35	24.8

Grouping vs. individual

Non-linear's partitioning means that all species in the same group get the same estimate (tree's leaf node), but for outliers like Killer whale, their lifespans might be too short/long to fit with the group, which then contribute to large residuals. On the other hand, even though linear also made mistake on Killer whale, the residual is lower because the linear equation gives the species its own estimate.

Final 2 models

Now it's time to create the final 2 ensemble models, linear and non-linear.

1. **tune**: No more tuning! Previously we already tuned and stored best tuning combo, so this time we just plug it in.

2. **train**: Using *all available data* we can get which is training set + test set together, because more data is better.
3. **predict**: On validation set, which will give us final 2 rmses.

To save space, I've put all linear equations of lm and cubist, and decision tree plot of rpart in appendix methods details section. It's quite interesting to see that with more training data, the tree and equations are different now. *Be sure to check those out!*

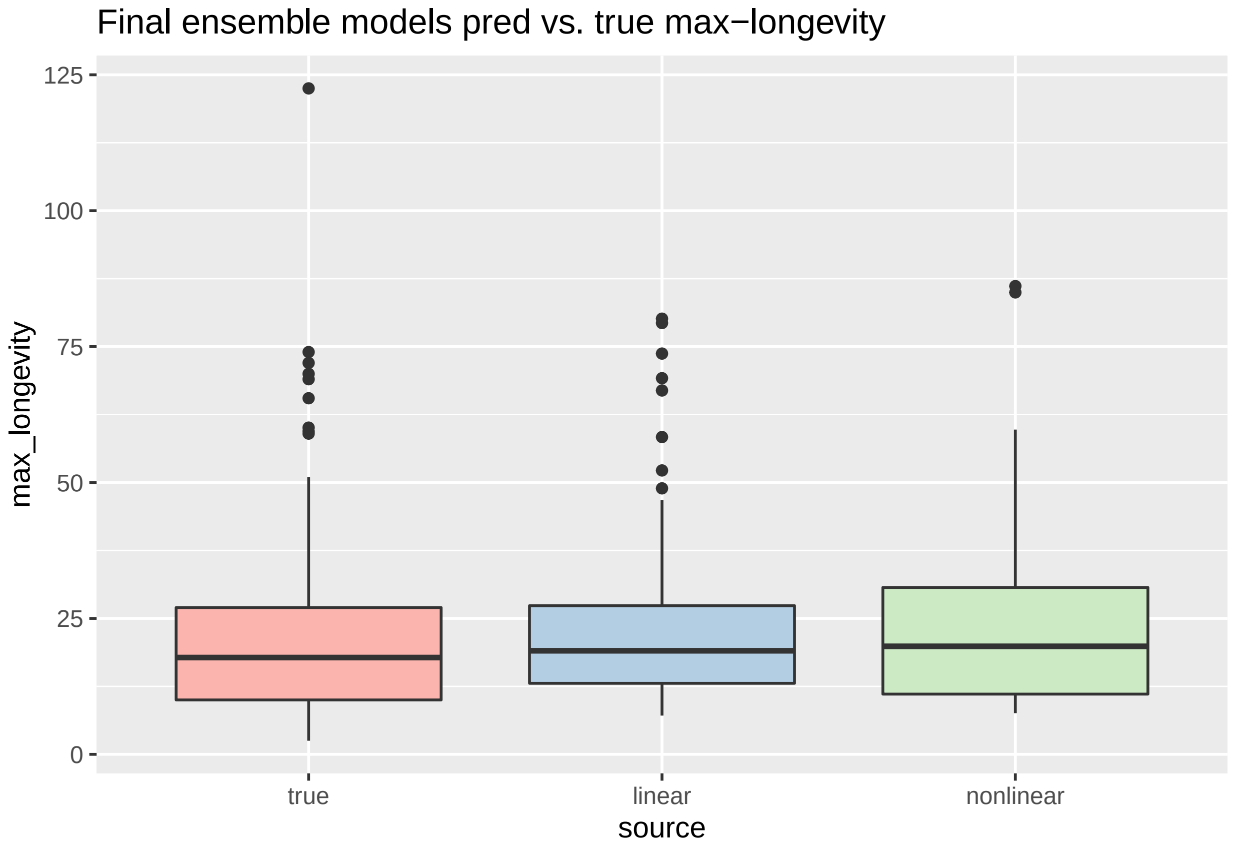
Results

method	rmse
linear	8.31
nonlinear	8.89

Linear and non-linear perform almost equally well, with linear taking a lead of about 0.5 years better than non-linear. Linear ensemble achieved a rmse of years and non-linear ensemble achieved a rmse of 8.311, 8.893 years.

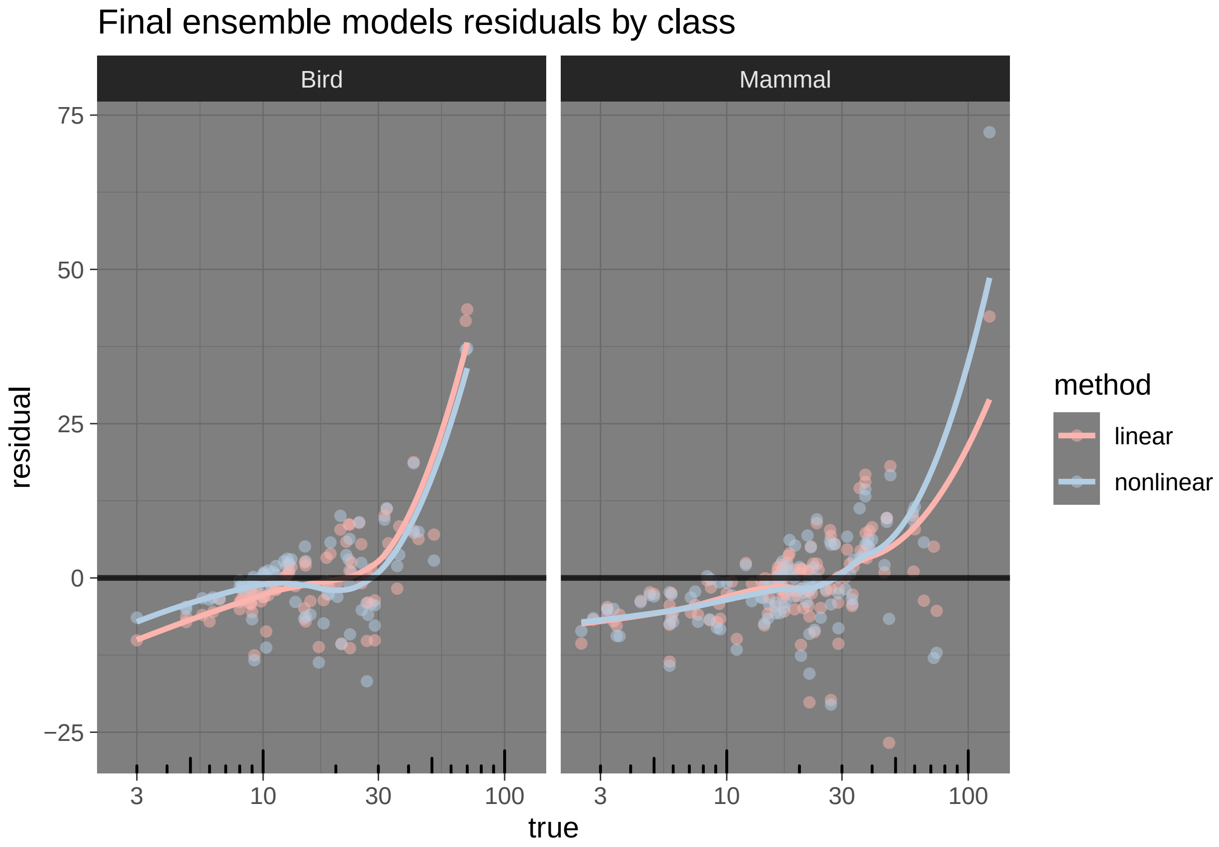
The 5-number summary and boxplot below shows the distributions of true outcome y, linear $\hat{y}$ and non-linear $\hat{y}$. Both ensembles' lowest predictions start around 7 years, which is 5 years higher than true outcome of 2.5. The highest predictions end around 80 years, but true outcome has outliers up to 120 years. However, both ensembles did excellently between Q1 and Q3, which is 50% of our data.

Source	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
true	2.50	10.0	17.8	21.4	27.0	122.5
linear	7.12	13.1	19.0	21.8	27.3	80.1
nonlinear	7.57	11.1	19.9	21.7	30.7	86.1



The two methods residuals smooth lines are very close to each other. For birds, non-linear ensemble gives a slightly better prediction. For mammals, linear ensemble gives better predictions to long lifespan outliers compared to non-linear.

Non-linear models partitions species into groups, and assign a $\hat{y}$ for the group. If an outlier is too different compared to its group members, it will have a large residual. Since linear models uses equations, outliers get their individual $\hat{y}$ which hopefully would be a better fit. (The reasoning is also discussed previously in grouping vs. individual section.)



Let’s see the top 5 mistakes made by two ensembles. Interestingly linear ensemble predicts human to live a max around 80 years, while non-linear predicts only 50 years. If we ignore human, non-linear ensemble’s top mistakes are actually better than linear’s. We see positive residual more frequently than negative, which means both models top mistakes are often underestimating max-longevity.

Class	Common name	y	linear	residual
Bird	Mute swan	70	26.5	43.5
Mammal	Human	122	80.1	42.4
Bird	Common raven	69	27.3	41.7
Mammal	Melon-headed whale	47	73.7	-26.7
Mammal	Dall’s porpoise	22	42.2	-20.2

Class	Common name	y	nonlinear	residual
Mammal	Human	122	50.3	72.2
Bird	Mute swan	70	32.8	37.2
Bird	Common raven	69	32.0	37.0
Mammal	Atlantic white-sided dolphin	27	47.5	-20.5
Bird	Hawaiian goose	42	23.4	18.6

Conclusion

In this project we solved the regression problem of predicting animals max-longevity. The solution was to create 2 ensemble models, 1 linear and 1 non-linear, which are robust and proved to work very well. We dived deep into caret's predictor selection, tuning, training, and predicting process. We combined 2 linear methods cubist and lm to create the linear ensemble, and 2 non-linear methods rpart and rf to create the non-linear ensemble. We explored how linear and non-linear methods work with equations and decision trees, and compared their performance by looking at residual plots and top mistakes made.

Potential impact: With a good machine learning model, scientists only need a few animal attributes in order to predict max-longevity, and don't need to wait decades to find out. Even for scientists who don't code, the models provide linear equations and decision trees which are easy to interpret.

Limitations: As we have discussed extensively before, outliers are contributing to residuals and we need ways to make sure the fit is good. Perhaps with more data on species like the outliers we would be able to predict better. We only predicted birds and mammals because there are enough of them with no NAs. Even though our predictor matrix have no NAs, a few methods on caret can actually take a matrix with NAs. So we could use those methods to predict other classes with many NAs like amphibians, fish, reptiles, etc.

Future work: I only used numerical predictors for predicting max-longevity and didn't use animal's class (bird, mammal) which has the type factor. As we have seen in the results residuals plot, linear models predicts mammals better, and non-linear models predicts birds better. Maybe each animal class needs a different ensemble, so the algorithm to predict animal max-longevity from many classes would be a mega ensemble.

Appendix

When you are done, click the *jump back* link at the beginning/end to take you back!

Original dataset

Jump back to data cleaning

There are 31 variables in the original dataset. First I'll describe what each variable means, then I'll show the summary code output.

biology taxonomy variables

Var	Example	Var	Example
Kingdom	Animalia	Family	Hominidae
Phylum	Chordata	Genus	Homo
Class	Mammalia	Species	sapiens
Order	Primates	Common name	Human

life history variables

Var	Desc
Maximum longevity (yrs)	the maximum recorded years this species are recorded to live
Female maturity (days), Male maturity (days)	female/male age of sexual maturity
Gestation/Incubation (days)	pregnancy time: mammals gestate litters, birds incubate eggs
Weaning (days)	*mammals only*: gradually introduce infant to adult diet, fully weaned means no more milk
Litter/Clutch size	number of young: litters or eggs
Litters/Clutches per year, Inter-litter/Interbirth interval	litters per year = number of litters / inter-litter interval
Birth weight (g), Weaning weight (g), Adult weight (g)	weight at birth, weaning, adult stage in grams
Growth rate (1/days)	postnatal growth rate
IMR (per yr)	infant mortality rate
MRDT (yrs)	mortality rate doubling time
Metabolic rate (W)	rate of metabolism in watts
Body mass (g)	body mass in grams
Temperature (k)	temperature in kelvin

quality variables

Var	Desc
Specimen origin	captivity, wild, or unknown
Sample size	tiny ($<$ 10), small (10 to 1000), medium ($>$ 1000), or huge (only humans has huge sample size)
Data quality	confidence in longevity data: low, questionable, acceptable, or high

reference variables

Var	Desc
HAGRID	id of the record
Source	id of the source that can verify record's authenticity
References	id of all references for this record

This summary contains 5-number summary for all numeric variables, as well as number of NAs. Variables with many NAs are not used.

Var	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
Female maturity (days)	3.000	365.000	548.000	1.01e+03	1095.00	5.69e+04	2142
Male maturity (days)	3.000	365.000	608.000	9.16e+02	1095.00	9.86e+03	2515
Gestation/Incubation (days)	1.000	21.000	34.000	8.32e+01	122.00	6.70e+02	2515
Weaning (days)	7.000	36.000	77.000	1.44e+02	182.75	1.11e+03	3409
Litter/Clutch size	1.000	1.000	3.000	1.46e+05	5.00	3.00e+08	2140
Litters/Clutches per year	0.100	1.000	1.000	1.46e+00	2.00	1.00e+01	2927
Inter-litter/Interbirth interval	20.000	189.500	365.000	3.87e+02	374.00	4.56e+03	3456

Var	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA’s
Birth weight (g)	0.004	3.295	24.750	1.15e+04	251.38	2.00e+06	2995
Weaning weight (g)	2.065	21.700	214.000	5.47e+04	2170.00	1.70e+07	3814
Adult weight (g)	0.500	58.850	572.000	2.18e+05	4476.35	1.36e+08	1268
Growth rate (1/days)	0.000	0.021	0.081	1.52e-01	0.22	6.80e-01	3660
Maximum longevity (yrs)	0.040	9.000	15.200	2.52e+01	24.10	1.50e+04	442
IMR (per yr)	0.000	0.019	0.060	1.83e-01	0.25	2.00e+00	4176
MRDT (yrs)	0.040	1.600	5.500	1.79e+02	14.25	9.99e+02	4179
Metabolic rate (W)	0.000	0.266	0.705	1.18e+01	3.14	2.34e+03	3592
Body mass (g)	0.760	25.900	131.300	1.32e+04	1111.15	3.67e+06	3592
Temperature (K)	278.500	308.250	309.650	3.09e+02	311.05	3.14e+02	3725

Jump back to data cleaning

Methods details

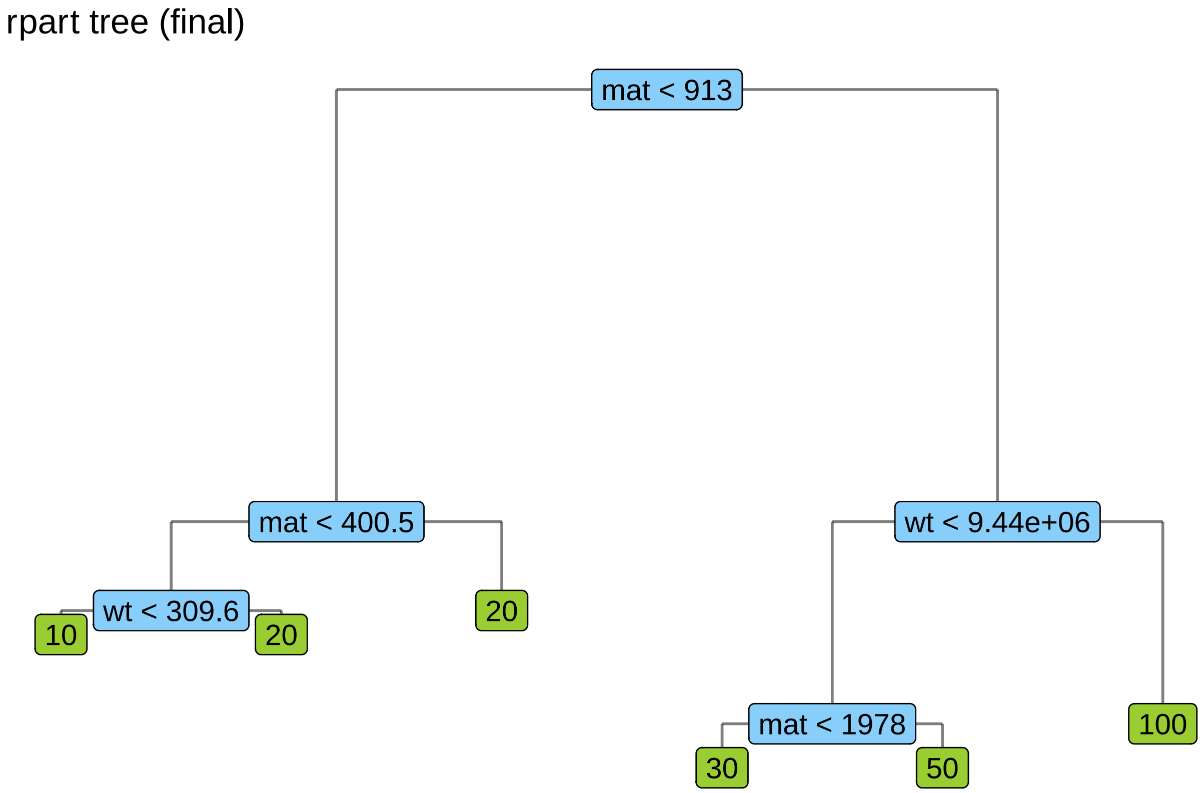
Jump back to final models

Here are the linear equations of lm and cubist, and the decision tree from rpart. There is no tree visualization for random forest because there could be hundreds of trees. I used all data that is available (train + test sets) to train the final methods.

lm

```
##                   (Intercept)            `Adult weight (g)`
##                      9.80e+00                      6.11e-07
##      `Female maturity (days)` `Gestation/Incubation (days)`
##                      1.39e-02                      2.46e-02
##          `Litter/Clutch size`
##                     -3.07e-01
```

rpart



rpart tree (final)

cubist

```
##
## Call:
## cubist.default(x = x, y = y, committees = param$committees)
##
##
## Cubist [Release 2.07 GPL Edition]  Wed Mar  9 17:41:05 2022
## ---------------------------------
##
##     Target attribute `outcome'
##
## Read 799 cases (5 attributes) from undefined.data
##
## Model 1:
##
##   Rule 1/1: [319 cases, mean 11.76, range 2.2 to 37.1, est err 3.79]
##
##     if
##  Adult weight (g) <= 317
##     then
##  outcome = 4.29 + 0.0088505 Adult weight (g)
##            + 0.0168 Female maturity (days)
##            + 0.041 Gestation/Incubation (days) - 0.28 Litter/Clutch size
##
##   Rule 1/2: [429 cases, mean 23.14, range 4.2 to 69, est err 5.51]
##
##     if
##  Adult weight (g) > 317
##  Female maturity (days) <= 1642
##     then
##  outcome = 10.5 + 6.52e-06 Adult weight (g)
##            + 0.0149 Female maturity (days)
##            + 0.013 Gestation/Incubation (days)
##
##   Rule 1/3: [54 cases, mean 55.40, range 16 to 211, est err 10.73]
##
##     if
##  Female maturity (days) > 1642
##     then
##  outcome = 12.52 + 0.0139 Female maturity (days)
##            + 6.6e-07 Adult weight (g) + 0.006 Gestation/Incubation (days)
##            - 0.06 Litter/Clutch size
##
## Model 2:
##
##   Rule 2/1: [633 cases, mean 15.78, range 2.2 to 49.5, est err 4.67]
##
##     if
##  Female maturity (days) <= 912
##     then
##  outcome = 3.32 - 9.43e-06 Adult weight (g)
##            + 0.0209 Female maturity (days)
##            + 0.042 Gestation/Incubation (days)
##
##   Rule 2/2: [166 cases, mean 39.71, range 8.1 to 211, est err 8.10]
##
##     if
##  Female maturity (days) > 912
##     then
##  outcome = 16.11 + 0.0122 Female maturity (days)
##            + 6.8e-07 Adult weight (g) + 0.008 Gestation/Incubation (days)
##            - 0.09 Litter/Clutch size
##
## Model 3:
##
##   Rule 3/1: [303 cases, mean 11.06, range 2.2 to 37.1, est err 3.67]
##
##     if
##  Adult weight (g) <= 317
##  Female maturity (days) <= 912
##     then
##  outcome = 3.78 + 0.02319509 Adult weight (g)
##            + 0.0172 Female maturity (days)
##            + 0.052 Gestation/Incubation (days) - 0.59 Litter/Clutch size
##
```

```
##   Rule 3/2: [330 cases, mean 20.11, range 4.2 to 49.5, est err 5.46]
##
##     if
##  Adult weight (g) > 317
##  Female maturity (days) <= 912
##     then
##  outcome = 14.62 + 0.0131 Female maturity (days)
##            + 0.002 Gestation/Incubation (days)
##
##   Rule 3/3: [166 cases, mean 39.71, range 8.1 to 211, est err 8.14]
##
##     if
##  Female maturity (days) > 912
##     then
##  outcome = 12.13 + 0.0131 Female maturity (days)
##            + 6.4e-07 Adult weight (g) + 0.016 Gestation/Incubation (days)
##            - 0.18 Litter/Clutch size
##
## Model 4:
##
##   Rule 4/1: [633 cases, mean 15.78, range 2.2 to 49.5, est err 4.67]
##
##     if
##  Female maturity (days) <= 912
##     then
##  outcome = 3.53 + 0.0197 Female maturity (days)
##            + 0.039 Gestation/Incubation (days) + 9e-08 Adult weight (g)
##
##   Rule 4/2: [166 cases, mean 39.71, range 8.1 to 211, est err 8.11]
##
##     if
##  Female maturity (days) > 912
##     then
##  outcome = 16.45 + 0.0122 Female maturity (days)
##            + 7.1e-07 Adult weight (g) + 0.006 Gestation/Incubation (days)
##            - 0.07 Litter/Clutch size
##
## Model 5:
##
##   Rule 5/1: [288 cases, mean 10.87, range 2.2 to 37.1, est err 3.59]
##
##     if
##  Adult weight (g) <= 236
##  Female maturity (days) <= 912
##     then
##  outcome = 3.15 + 0.03417179 Adult weight (g)
##            + 0.0179 Female maturity (days)
##            + 0.06 Gestation/Incubation (days) - 0.58 Litter/Clutch size
##
##   Rule 5/2: [345 cases, mean 19.88, range 3.8 to 49.5, est err 5.55]
##
##     if
##  Adult weight (g) > 236
##  Female maturity (days) <= 912
##     then
##  outcome = 14.23 + 0.0142 Female maturity (days) - 8e-08 Adult weight (g)
##
##   Rule 5/3: [166 cases, mean 39.71, range 8.1 to 211, est err 8.18]
##
##     if
##  Female maturity (days) > 912
##     then
##  outcome = 11.79 + 0.0131 Female maturity (days) + 6e-07 Adult weight (g)
##            + 0.018 Gestation/Incubation (days) - 0.2 Litter/Clutch size
##
## Model 6:
##
##   Rule 6/1: [633 cases, mean 15.78, range 2.2 to 49.5, est err 4.67]
##
##     if
##  Female maturity (days) <= 912
##     then
##  outcome = 3.54 + 0.0193 Female maturity (days)
##            + 0.041 Gestation/Incubation (days) + 1e-07 Adult weight (g)
##
##   Rule 6/2: [166 cases, mean 39.71, range 8.1 to 211, est err 8.12]
##
##     if
##  Female maturity (days) > 912
##     then
##  outcome = 16.84 + 0.012 Female maturity (days)
##            + 7.5e-07 Adult weight (g) + 0.005 Gestation/Incubation (days)
##            - 0.06 Litter/Clutch size
##
## Model 7:
##
##   Rule 7/1: [288 cases, mean 10.87, range 2.2 to 37.1, est err 3.57]
##
##     if
##  Adult weight (g) <= 236
##  Female maturity (days) <= 912
##     then
##  outcome = 3.14 + 0.03417178 Adult weight (g)
##            + 0.0183 Female maturity (days)
##            + 0.058 Gestation/Incubation (days) - 0.58 Litter/Clutch size
##
##   Rule 7/2: [345 cases, mean 19.88, range 3.8 to 49.5, est err 5.54]
##
##     if
##  Adult weight (g) > 236
##  Female maturity (days) <= 912
##     then
##  outcome = 14.1 + 0.0144 Female maturity (days)
##
##   Rule 7/3: [166 cases, mean 39.71, range 8.1 to 211, est err 8.21]
##
##     if
##  Female maturity (days) > 912
##     then
##  outcome = 11.4 + 0.0133 Female maturity (days)
##            + 5.6e-07 Adult weight (g) + 0.019 Gestation/Incubation (days)
##            - 0.21 Litter/Clutch size
##
## Model 8:
##
##   Rule 8/1: [633 cases, mean 15.78, range 2.2 to 49.5, est err 4.68]
##
##     if
##  Female maturity (days) <= 912
##     then
##  outcome = 3.57 + 0.0191 Female maturity (days)
##            + 0.041 Gestation/Incubation (days) + 1.1e-07 Adult weight (g)
##
##   Rule 8/2: [166 cases, mean 39.71, range 8.1 to 211, est err 8.14]
##
##     if
##  Female maturity (days) > 912
##     then
##  outcome = 17.29 + 0.0117 Female maturity (days)
##            + 7.9e-07 Adult weight (g) + 0.004 Gestation/Incubation (days)
##
## Model 9:
##
##   Rule 9/1: [288 cases, mean 10.87, range 2.2 to 37.1, est err 3.56]
##
##     if
##  Adult weight (g) <= 236
##  Female maturity (days) <= 912
##     then
##  outcome = 3.11 + 0.03417177 Adult weight (g)
##            + 0.0185 Female maturity (days)
##            + 0.058 Gestation/Incubation (days) - 0.58 Litter/Clutch size
##
##   Rule 9/2: [345 cases, mean 19.88, range 3.8 to 49.5, est err 5.55]
##
##     if
##  Adult weight (g) > 236
##  Female maturity (days) <= 912
##     then
##  outcome = 14.07 + 0.0146 Female maturity (days)
##
##   Rule 9/3: [166 cases, mean 39.71, range 8.1 to 211, est err 8.27]
##
##     if
##  Female maturity (days) > 912
##     then
##  outcome = 10.82 + 0.0135 Female maturity (days)
##            + 5.2e-07 Adult weight (g) + 0.021 Gestation/Incubation (days)
##            - 0.21 Litter/Clutch size
##
##
## Evaluation on training data (799 cases):
```

```
##
##     Average  |error|               5.84
##     Relative |error|               0.55
##     Correlation coefficient        0.85
##
##
##  Attribute usage:
##    Conds  Model
##
##     96%   100%    Female maturity (days)
##     46%    86%    Adult weight (g)
##            86%    Gestation/Incubation (days)
##            38%    Litter/Clutch size
##
##
## Time: 0.1 secs
```

Jump back to final models

References

Dataset

Tacutu, R., Craig, T., Budovsky, A., Wuttke, D., Lehmann, G., Taranukha, D., Costa, J., Fraifeld, V. E., de Magalhaes, J. P. (2013) "Human Ageing Genomic Resources: Integrated databases and tools for the biology and genetics of ageing." *Nucleic Acids Research* **41**(D1):D1027-D1033. PubMed

- The dataset is under *Downloading AnAge* section in AnAge home page.
- AnAge home page: <https://genomics.senescence.info/species/index.html>
- Dataset (zipped tab-delimited): <https://genomics.senescence.info/species/dataset.zip>

Libraries

caret

Kuhn, M. (2008), "Building predictive models in R using the caret package," *Journal of Statistical Software*, (doi: 10.18637/jss.v028.i05).

- Book: https://topepo.github.io/caret/
- CRAN Short introduction: https://cran.r-project.org/web/packages/caret/vignettes/caret.html

cubist

Kuhn M, Quinlan R (2022). *Cubist: Rule- And Instance-Based Regression Modeling*. https://topepo.github.io/Cubist//, https://topepo.github.io/Cubist/.

- CRAN Short introduction: https://cran.r-project.org/web/packages/Cubist/vignettes/cubist.html

rpart

Terry Therneau and Beth Atkinson (2022). rpart: Recursive Partitioning and Regression Trees. R package version 4.1.16. https://CRAN.R-project.org/package=rpart

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

random forest

Liaw A, Wiener M (2002). “Classification and Regression by randomForest.” *R News*, **2**(3), 18-22. <https://CRAN.R-project.org/doc/Rnews/>.

- CRAN: https://cran.r-project.org/web/packages/randomForest/index.html

more citations

Andrie de Vries and Brian D. Ripley (2022). ggdendro: Create Dendrograms and Tree Diagrams Using ‘ggplot2’. R package version 0.1.23. https://CRAN.R-project.org/package=ggdendro

Microsoft Corporation and Steve Weston (2022). doParallel: Foreach Parallel Adaptor for the ‘parallel’ Package. R package version 1.0.17. https://CRAN.R-project.org/package=doParallel

Taiyun Wei and Viliam Simko (2021). R package ‘corrplot’: Visualization of a Correlation Matrix (Version 0.92). Available from https://github.com/taiyun/corrplot

Victor Perrier and Fanny Meyer (2021). gfonts: Offline ‘Google’ Fonts for ‘Markdown’ and ‘Shiny’. R package version 0.1.3. https://CRAN.R-project.org/package=gfonts

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

Yihui Xie and J.J. Allaire and Garrett Grolemund (2018). R Markdown: The Definitive Guide. Chapman and Hall/CRC. ISBN 9781138359338. URL https://bookdown.org/yihui/rmarkdown.

Yihui Xie (2021). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.37.