abalone_project-build_rings(age)_predictor

K Somachandra Senerath de Silva

03/01/2020

1. Introduction / Overview

The purpose of this project is to apply machine learning techniques such as linear and polynomial regression to the abalone dataset, from the UCI repository, to enable predicting the age (number of shell rings) of an abalone from its physical characteristics.

The dataset provides 8 physical characteristics (sex, length, diameter, height, whole weight, shucked weight, viscera weight, shell weight) as possible predictors in addition to the the number of shell rings (henceforth referred to as rings for convenience). This equates to a total of 9 attributes. The number of rings (integer), which varies from 1 to 29, +1.5 is thought to be a reasonable estimate of the age in years. There are a total of 4177 samples or records. For this project, the rings outcome variable is treatd as a continuous variable as it ranges from 1 to 29 with intervals of 1.

Repository citation: Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
Dataset source: Abalone Data Set from UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/Abalone

The dataset is first split into a 90% training set (train) and 10% test set (validation) where the train set is used for analysing and training models to be built for rings (age) prediction while the validation set is used to test the prediction accuracy of these models. The models trained using train set are used to predict the ratings in the validation set and then compared against actual values in the set. The accuracy is established using the RMSE (root mean square error) between the two values with a lower value indicating a smaller error and hence higher accuracy.

This overview is followed by 3 more sections which contain the key steps performed:
Section 2. Methods and Analysis - where the dataset is split into the training and test sets and cleaned (if required) to the required form and prepared for building models of the desired prediction model. After a basic exploration to aid in the visualization of the dataset, the modeling approach is explained and several prediction models are built. The models are then run and their accuracies are determined.
Section 3. Results - the accuracies of the various models are compared and insights gained from running the models are discussed.
Section 4. Conclusion - recommendations are made, and their limitations with the potential for future work are discussed.

2. Methods and Analysis

Predicting the age of abalone from physical measurements: the age of an abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope – the number of rings (integer) +1.5 is thought to be a reasonable estimate of the age in years. As the other measurements are easier to obtain, the purpose of this section is to use them to predict the number of rings (age).

2A. Data Manipulation

Here the dataset is downloaded and split into the train and test sets. The original data has been preprocessed and cleaned prior to download where examples with missing values were removed from the abalone dataset (the majority having the predicted value missing), and the ranges of the continuous values have been scaled (by dividing by 200) for use with an ANN (abalone dataset readme file - abalone.names). As such no further cleaning is done and the dataset is deemed suitable for analysis.

Predicting the age of abalone from physical measurements: The age of an abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope – Rings (integer) +1.5 is thought to be a reasonable estimate of the age in years. As the other measurements are easier to obtain, the purpose of this section is to use them to predict the number of rings (age).

2A. Data Manipulation

Here the dataset is downloaded and split into the train and test sets. The original data has been preprocessed and cleaned prior to download where examples with missing values were removed from the abalone dataset (the majority having the predicted value missing), and the ranges of the continuous values have been scaled (by dividing by 200) for use with an ANN (abalone dataset readme file - abalone.names). AS such no further cleaning is done and the dataset is deemed suitable for analysis.

Install packages if required:

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(stringr)) install.packages("stringr", repos = "http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot", repos = "http://cran.us.r-project.org")
if(!require(GGally)) install.packages("GGally", repos = "http://cran.us.r-project.org")
```

Libraries used:

```
library(tidyverse)
library(caret)
library(data.table)
library(stringr)
library(dplyr)
library(ggplot2)
library(GGally)
```

Download dataset

```
# Abalone dataset:
# https://archive.ics.uci.edu/ml/datasets/Abalone

### Auto-download and read the dataset into a data frame
abalone <- read.csv("http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data",
                    header=FALSE)

# add column names
```

```
names(abalone) <- c("sex", "length", "diameter", "height", "weight.whole",
                    "weight.shucked", "weight.viscera", "weight.shell", "rings")

# confirm that there are 4177 observations
nrow(abalone)
```

```
## [1] 4177
```

```
# confirm that there are no missing values
sum(is.na(abalone))
```

```
## [1] 0
```

```
# Confirm column names
names(abalone)
```

```
## [1] "sex"            "length"         "diameter"       "height"
## [5] "weight.whole"   "weight.shucked" "weight.viscera" "weight.shell"
## [9] "rings"
```

```
# Summary information
summary(abalone)
```

```
##  sex          length         diameter          height        weight.whole
##  F:1307   Min.   :0.075   Min.   :0.0550   Min.   :0.0000   Min.   :0.0020
##  I:1342   1st Qu.:0.450   1st Qu.:0.3500   1st Qu.:0.1150   1st Qu.:0.4415
##  M:1528   Median :0.545   Median :0.4250   Median :0.1400   Median :0.7995
##           Mean   :0.524   Mean   :0.4079   Mean   :0.1395   Mean   :0.8287
##           3rd Qu.:0.615   3rd Qu.:0.4800   3rd Qu.:0.1650   3rd Qu.:1.1530
##           Max.   :0.815   Max.   :0.6500   Max.   :1.1300   Max.   :2.8255
##  weight.shucked   weight.viscera    weight.shell        rings
##  Min.   :0.0010   Min.   :0.0005   Min.   :0.0015   Min.   : 1.000
##  1st Qu.:0.1860   1st Qu.:0.0935   1st Qu.:0.1300   1st Qu.: 8.000
##  Median :0.3360   Median :0.1710   Median :0.2340   Median : 9.000
##  Mean   :0.3594   Mean   :0.1806   Mean   :0.2388   Mean   : 9.934
##  3rd Qu.:0.5020   3rd Qu.:0.2530   3rd Qu.:0.3290   3rd Qu.:11.000
##  Max.   :1.4880   Max.   :0.7600   Max.   :1.0050   Max.   :29.000
```

Create training set (train) and test set (validation)

Split data into train set (90%) and validation set (10%)

```
# Create train set and validation set for testing train set predictions
# Validation set will be 10% of abalone data derived from test_index
set.seed(1, sample.kind="Rounding")

# if using R 3.5 or earlier, use `set.seed(1)` instead
test_index <- createDataPartition(y = abalone$rings, times = 1, p = 0.1, list = FALSE)
train <- abalone[-test_index,]
validation <- abalone[test_index,]

# remove unecessary data
rm(test_index, abalone)
```

2B. Data Visualization

Explores the train dataset: generally the correlation between the output variable 'rings' and any of the physical characteristics is only average at around 0.4 - 0.6. However there is better correlation between any 2 of the physical characteristics at around 0.8 - 0.9 which would affect their use as independant variables with respect to the output variable.

```
head(train)
```

```
##   sex length diameter height weight.whole weight.shucked weight.viscera
## 1   M  0.455    0.365  0.095       0.5140         0.2245         0.1010
## 2   M  0.350    0.265  0.090       0.2255         0.0995         0.0485
## 3   F  0.530    0.420  0.135       0.6770         0.2565         0.1415
## 4   M  0.440    0.365  0.125       0.5160         0.2155         0.1140
## 5   I  0.330    0.255  0.080       0.2050         0.0895         0.0395
## 6   I  0.425    0.300  0.095       0.3515         0.1410         0.0775
##   weight.shell rings
## 1        0.150    15
## 2        0.070     7
## 3        0.210     9
## 4        0.155    10
## 5        0.055     7
## 6        0.120     8
```

```
str(train)
```

```
## 'data.frame':    3758 obs. of  9 variables:
##  $ sex           : Factor w/ 3 levels "F","I","M": 3 3 1 3 2 2 1 1 3 1 ...
##  $ length        : num  0.455 0.35 0.53 0.44 0.33 0.425 0.53 0.545 0.475 0.55 ...
##  $ diameter      : num  0.365 0.265 0.42 0.365 0.255 0.3 0.415 0.425 0.37 0.44 ...
##  $ height        : num  0.095 0.09 0.135 0.125 0.08 0.095 0.15 0.125 0.125 0.15 ...
##  $ weight.whole  : num  0.514 0.226 0.677 0.516 0.205 ...
##  $ weight.shucked: num  0.2245 0.0995 0.2565 0.2155 0.0895 ...
##  $ weight.viscera: num  0.101 0.0485 0.1415 0.114 0.0395 ...
##  $ weight.shell  : num  0.15 0.07 0.21 0.155 0.055 0.12 0.33 0.26 0.165 0.32 ...
##  $ rings         : int  15 7 9 10 7 8 20 16 9 19 ...
```

```
# number of unique rings and distribution
train %>% summarise(n_rings = n_distinct(rings))
```

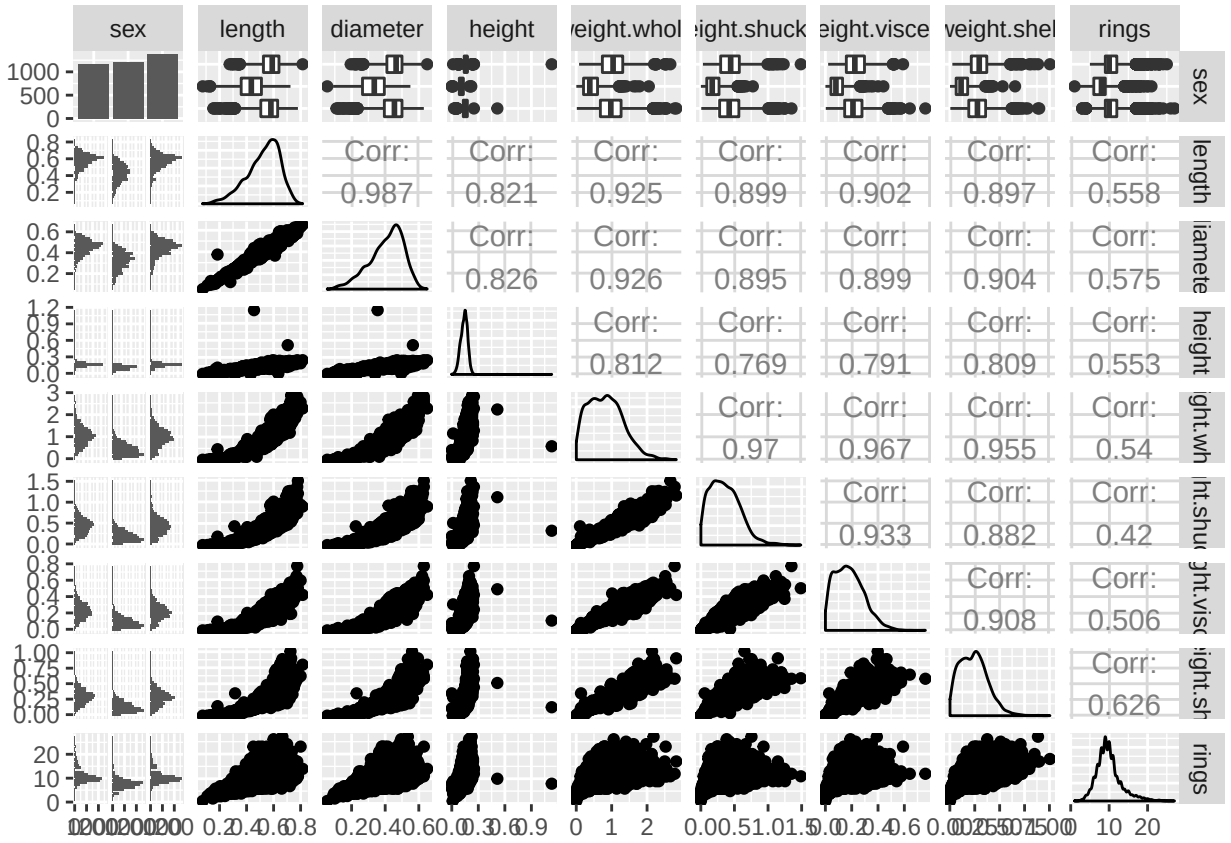
```
##   n_rings
## 1      27
```

```
table(train$rings)
```

```
##
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   1   1  15  55 107 227 349 511 620 571 437 230 186 112  98  58  54  41  30  22
##  21  22  23  24  25  26  27
##  14   5   9   2   1   1   1
```

```
# visualizing correlation between various attributes of abalone
ggpairs(train, progress = FALSE)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# It is observed that the correlation between the output variable "rings" and the predictor
# variables is average, around 0.4 - 0.6, whilst the correlation between the predictor
# variables is much higher, around 0.8 - 0.9, which would affect their usefulness as
# independant variables.
```

2C. Modeling Approach

Determining accuracy of model

To compare different models, a loss function is required to measure how far predictions deviate from actual values. RMSE will be used as the loss function in this project where a lower value indicates a higher accuracy.

Note: This is the typical error made when predicting the number of rings.

```r
RMSE <- function(true_rings, predicted_rings){
  sqrt(mean((true_rings - predicted_rings)^2)) }
```

Model 1: Reference model or just the average

This will represent the worst possible prediction to be modelled. Assumes all abalone regardless of the physical characteristics have the same num of rings with the differences explained by random variation.

```r
# calculates mean rings value
mu_hat <- mean(train$rings)
mu_hat
```

```
## [1] 9.936136
```

```r
# checking RMSE
reference_rmse <- RMSE(validation$rings, mu_hat)
reference_rmse
```

```
## [1] 3.126598
```

```r
# Creating a table to store and compare accuracies of different models
rmse_results <- data_frame(method = "Model 1: Just the average", RMSE = reference_rmse)
```

```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

Model 2: LR - External physical characteristics

This model uses linear regression via the lm function to fit output variable "rings" to its external characteristics only: "length", "diameter", "height", "weight.whole". This does not require the physical characteristics post cutting open an abalone. Hence this method could help abalone farmers, sellers and buyers to value an abalone without having to cut it open.

```r
# Building / Training model
fit2 <- lm(rings ~ weight.whole + diameter + length + height, data = train)
fit2
```

```
##
## Call:
## lm(formula = rings ~ weight.whole + diameter + length + height,
##     data = train)
##
## Coefficients:
##  (Intercept)  weight.whole      diameter        length        height
##      2.83771       0.04084      24.34268     -10.60004      19.30079
```

```r
summary(fit2)
```

```
##
## Call:
## lm(formula = rings ~ weight.whole + diameter + length + height,
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.4905  -1.6321  -0.6206   0.8964  13.7561
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.83771    0.32483   8.736  < 2e-16 ***
## weight.whole   0.04084    0.23901   0.171    0.864
## diameter      24.34268    2.71420   8.969  < 2e-16 ***
## length       -10.60004    2.21655  -4.782  1.8e-06 ***
## height        19.30079    1.82961  10.549  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.602 on 3753 degrees of freedom
## Multiple R-squared:  0.3535, Adjusted R-squared:  0.3528
## F-statistic:   513 on 4 and 3753 DF,  p-value: < 2.2e-16
```

```
# using model, fit2, to predict rings in validation set
p2 <- predict(fit2, newdata = validation)

str(p2)
```

```
##  Named num [1:419] 12.51 10 9.86 9.4 7.59 ...
##  - attr(*, "names")= chr [1:419] "32" "56" "58" "117" ...
```

```
class(p2)
```

```
## [1] "numeric"
```

```
# accuracy
model_2_rmse <- RMSE(validation$rings, p2)
model_2_rmse
```

```
## [1] 2.473504
```

```
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Model 2: LR - External physical characteristics",
                                     RMSE = model_2_rmse ))
```

Model 3: LR - External physical characteristics less weight.whole

Model 2 is better than the average but still has a high RMSE. However it was observed in the summary of the fitted model that the p-value for the weight.whole variable was very high unlike the other variables and intercept. Hence model 3 will use model 2 without this variable, expecting a better RMSE.

```r
# Building / Training model
fit3 <- lm(rings ~ diameter + length + height, data = train)
fit3
```

```
##
## Call:
## lm(formula = rings ~ diameter + length + height, data = train)
##
## Coefficients:
## (Intercept)     diameter       length       height
##       2.793       24.421      -10.530       19.368
```

```r
summary(fit3)
```

```
##
## Call:
## lm(formula = rings ~ diameter + length + height, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.5579  -1.6286  -0.6238   0.9025  13.7858
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.793      0.196  14.251  < 2e-16 ***
## diameter      24.421      2.675   9.130  < 2e-16 ***
## length       -10.530      2.178  -4.835 1.39e-06 ***
## height        19.368      1.786  10.842  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.602 on 3754 degrees of freedom
## Multiple R-squared:  0.3535, Adjusted R-squared:  0.353
## F-statistic: 684.2 on 3 and 3754 DF,  p-value: < 2.2e-16
```

```r
# using model, fit3, to predict rings in validation set
p3 <- predict(fit3, newdata = validation)

# accuracy
model_3_rmse <- RMSE(validation$rings, p3)
model_3_rmse
```

```
## [1] 2.473651
```

```r
# comment: does not help to remove weight.whole as the RMSE became higher i.e. less accurate

rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Model 3: LR - External pc less weight.whole",
                                     RMSE = model_3_rmse ))
```

Model 4: LR - Using all predictors

Models 2 and 3 still have a high RMSEs hence the need to explore the use of all 8 physical characteristics as predictors i.e. including sex.

```
# Building / Training model
fit4 <- lm(rings ~ sex + length + diameter + height + weight.whole + weight.shucked
           + weight.viscera + weight.shell, data = train)
fit4
```

```
##
## Call:
## lm(formula = rings ~ sex + length + diameter + height + weight.whole +
##     weight.shucked + weight.viscera + weight.shell, data = train)
##
## Coefficients:
##    (Intercept)            sexI            sexM          length        diameter
##        3.72060        -0.83974         0.04305         0.15389        11.31315
##         height    weight.whole  weight.shucked  weight.viscera    weight.shell
##       10.10331         9.59004       -20.96173       -10.05230         7.39874
```

```
summary(fit4)
```

```
##
## Call:
## lm(formula = rings ~ sex + length + diameter + height + weight.whole +
##     weight.shucked + weight.viscera + weight.shell, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.782 -1.316 -0.333  0.866 11.910
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.72060    0.30542  12.182  < 2e-16 ***
## sexI            -0.83974    0.10813  -7.766 1.04e-14 ***
## sexM             0.04305    0.08790   0.490    0.624
## length           0.15389    1.89769   0.081    0.935
## diameter        11.31315    2.33519   4.845 1.32e-06 ***
## height          10.10331    1.57050   6.433 1.41e-10 ***
## weight.whole     9.59004    0.78402  12.232  < 2e-16 ***
## weight.shucked -20.96173    0.88370 -23.721  < 2e-16 ***
## weight.viscera -10.05230    1.36817  -7.347 2.47e-13 ***
## weight.shell     7.39874    1.20391   6.146 8.80e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.198 on 3748 degrees of freedom
## Multiple R-squared:  0.5396, Adjusted R-squared:  0.5385
## F-statistic: 488.1 on 9 and 3748 DF,  p-value: < 2.2e-16
```

```
# using model, fit4, to predict rings in validation set
p4 <- predict(fit4, newdata = validation)

# accuracy
model_4_rmse <- RMSE(validation$rings, p4)
model_4_rmse
```

```
## [1] 2.174076
```

```
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Model 4: LR - Using all predictors",
                                     RMSE = model_4_rmse ))
```

Model 5: LR - Using all predictors less length

Models 4 summary statistics show that length is not significant at p = 0.05 and hence this model runs model 4 without lenght as a predictor.

```
# Building / Training model
fit5 <- lm(rings ~ sex + diameter + height + weight.whole + weight.shucked
           + weight.viscera + weight.shell, data = train)
fit5
```

```
##
## Call:
## lm(formula = rings ~ sex + diameter + height + weight.whole +
##     weight.shucked + weight.viscera + weight.shell, data = train)
##
## Coefficients:
##    (Intercept)            sexI            sexM        diameter          height
##        3.72876        -0.83925         0.04308        11.48286        10.10769
##   weight.whole  weight.shucked  weight.viscera    weight.shell
##        9.58936       -20.95670       -10.04205         7.39616
```

```
summary(fit5)
```

```
##
## Call:
## lm(formula = rings ~ sex + diameter + height + weight.whole +
##     weight.shucked + weight.viscera + weight.shell, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7878 -1.3178 -0.3322  0.8677 11.9130
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.72876    0.28833  12.932  < 2e-16 ***
## sexI            -0.83925    0.10795  -7.774 9.74e-15 ***
## sexM             0.04308    0.08788   0.490    0.624
## diameter        11.48286    1.03603  11.084  < 2e-16 ***
```

```
## height          10.10769    1.56937   6.441 1.34e-10 ***
## weight.whole     9.58936    0.78387  12.233  < 2e-16 ***
## weight.shucked -20.95670    0.88140 -23.777  < 2e-16 ***
## weight.viscera -10.04205    1.36214  -7.372 2.05e-13 ***
## weight.shell     7.39616    1.20333   6.146 8.75e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.197 on 3749 degrees of freedom
## Multiple R-squared:  0.5396, Adjusted R-squared:  0.5386
## F-statistic: 549.3 on 8 and 3749 DF,  p-value: < 2.2e-16
```

```
# using model, fit, to predict rings in validation set
p5 <- predict(fit5, newdata = validation)

# accuracy
model_5_rmse <- RMSE(validation$rings, p5)
model_5_rmse
```

```
## [1] 2.173926
```

```
# model 5 produces a very slight improvement in RMSE

rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Model 5: LR - Using all predictors less length",
                                     RMSE = model_5_rmse ))
```

Model 6: Poly2Regression - all predictors

The lm function with selective polynomial regression is applied to all predictors. After testing several 2nd and 3rd degree polynomial regression fits to various attributes, the 2nd degree fit is applied to all predictors except sex which is kept at a linear fit.

```
poly2model_all_pc <- lm(rings ~ sex + poly(diameter,2) + poly(height,2) +
                          poly(weight.whole,2) + poly(weight.shucked,2) +
                          poly(weight.viscera,2) + poly(weight.shell, 2), data = train)
poly2model_all_pc
```

```
##
## Call:
## lm(formula = rings ~ sex + poly(diameter, 2) + poly(height, 2) +
##     poly(weight.whole, 2) + poly(weight.shucked, 2) + poly(weight.viscera,
##     2) + poly(weight.shell, 2), data = train)
##
## Coefficients:
##               (Intercept)                       sexI                       sexM
##                 1.014e+01                 -6.543e-01                  3.128e-03
##       poly(diameter, 2)1        poly(diameter, 2)2          poly(height, 2)1
##                 1.221e+01                 -2.541e+01                  2.899e+01
##         poly(height, 2)2    poly(weight.whole, 2)1    poly(weight.whole, 2)2
##                -1.226e+01                  4.142e+02                 -8.802e+01
## poly(weight.shucked, 2)1  poly(weight.shucked, 2)2  poly(weight.viscera, 2)1
```

```
##               -3.413e+02                 7.480e+01                -8.850e+01
## poly(weight.viscera, 2)2    poly(weight.shell, 2)1    poly(weight.shell, 2)2
##                2.073e+01                 7.594e+01                -9.905e-01
```

```
summary(poly2model_all_pc)
```

```
##
## Call:
## lm(formula = rings ~ sex + poly(diameter, 2) + poly(height, 2) +
##     poly(weight.whole, 2) + poly(weight.shucked, 2) + poly(weight.viscera,
##     2) + poly(weight.shell, 2), data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1575  -1.2702  -0.2879   0.8829  11.5144
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.014e+01  6.589e-02 153.961  < 2e-16 ***
## sexI                       -6.543e-01  1.072e-01  -6.101 1.16e-09 ***
## sexM                        3.128e-03  8.480e-02   0.037 0.970578
## poly(diameter, 2)1          1.221e+01  9.371e+00   1.304 0.192473
## poly(diameter, 2)2         -2.541e+01  3.429e+00  -7.410 1.56e-13 ***
## poly(height, 2)1            2.899e+01  5.414e+00   5.355 9.05e-08 ***
## poly(height, 2)2           -1.226e+01  3.106e+00  -3.948 8.04e-05 ***
## poly(weight.whole, 2)1      4.142e+02  2.691e+01  15.393  < 2e-16 ***
## poly(weight.whole, 2)2     -8.802e+01  1.230e+01  -7.155 1.00e-12 ***
## poly(weight.shucked, 2)1 -3.413e+02  1.337e+01 -25.540  < 2e-16 ***
## poly(weight.shucked, 2)2  7.480e+01  7.167e+00  10.436  < 2e-16 ***
## poly(weight.viscera, 2)1 -8.850e+01  1.055e+01  -8.392  < 2e-16 ***
## poly(weight.viscera, 2)2  2.073e+01  5.867e+00   3.533 0.000416 ***
## poly(weight.shell, 2)1      7.594e+01  1.230e+01   6.175 7.31e-10 ***
## poly(weight.shell, 2)2     -9.905e-01  5.564e+00  -0.178 0.858716
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.118 on 3743 degrees of freedom
## Multiple R-squared:  0.5731, Adjusted R-squared:  0.5715
## F-statistic: 358.9 on 14 and 3743 DF,  p-value: < 2.2e-16
```

```
# using model, poly2model_all_pc, to predict rings in validation set
p6 <- predict(poly2model_all_pc, newdata = validation)

# accuracy
model_6_rmse <- RMSE(validation$rings, p6)
model_6_rmse
```

```
## [1] 2.152143
```

```
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Model 6: Poly2R - all predictors excl sex",
                                     RMSE = model_6_rmse ))
rmse_results %>% knitr::kable()
```

method	RMSE
Model 1: Just the average	3.126598
Model 2: LR - External physical characteristics	2.473504
Model 3: LR - External pc less weight.whole	2.473651
Model 4: LR - Using all predictors	2.174076
Model 5: LR - Using all predictors less length	2.173926
Model 6: Poly2R - all predictors excl sex	2.152143

3. Results

The table above, at the end of Model 5, summarizes the accuracies of the 5 models built to predict the number of rings (age) in section 2C above. The main insights gained are:
1. Linear regression using the external physical characteristics reduces the RMSE from 3.1266 to 2.4735 or approximately 21% while removing a high p-value variable, weight.whole, slightly increases the RMSE (less accurate).
2. Using all 8 predictors further reduces the RMSE from 2.4735 to 2.1741 or approximately a further 12% (approximately 30% more accurate than the worst case of just using the average). Dropping the length due to its high p-value only improved the RMSE very slightly.
3. Using the lm function and 2nd degree polynomial regression on all predictors (except sex), the RMSE further reduces to 2.1521 from 2.1741 (approximately 1%) or about 31% more accurate than the worst case.
4. The correlation between the output variable and the predictor variables is lower that the correlations between the predictor variables which would affect the performance of the fitted models.

Hence, overall it is seen that the best prediction comes using the lm function with 2nd degree polynomial regression on all the attributes (except sex) with an RMSE of about 2.152 whilst just using linear regression on all external characteristics as a more practical measure due not having to cut open the abalone loses about 15% accuracy (from 2.152 to 2.474).

4. Conclusion

From the results summarized in sction 3 and reasons given therein, Model 6 is recommended to provide the algorithm for predicting the age (number of rings) of an abalone with an RMSE of 2.152 although this accuracy is not very high.

The investigation has been limited to the use of linear and polynomial regressions as earlier work suggests that these methods achieve better results than from other methods including random forest, decision tree and SVR (Srishtee Kriti, 2019). The reasons for high RMSEs include using predictors which have a high correlation with each other and lack of environmental information, such as weather patterns and location (hence food availability) which may be required to solve the problem with greater accuracy.

Future work on building new models could include the investigation of mitigating the use of highly correlated predictors, looking into affect of environmental conditions, use other models such as neural networks and looking at the problem as a classification problem.

References

abalone.names, https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.names, <accessed Dec 26, 2019>.

Kevin Markham https://github.com/justmarkham work on Abalone Dataset at GitHub https://github.com/ajschumacher/gadsdc1/blob/master/dataset_research/kevin_abalone_dataset.md, <accessed Dec 29, 2019>.

Srishtee Kriti, 2019, Machine Learning with Abalone, https://medium.com/@srishtee.kriti/machine-learning-with-abalone-c752a8237e85?source=rss-6f0922344582------2, <accessed Feb 03 2020>.