Abalone dataset analysis using linear regression and decision tree regression models

Joanna Wojcik

R00169918
Cork Institute of Technology

Contents

Abstract

The purpose of this document is to critically analyze linear regression and decision tree regression models and their applicability towards predicting the age of abalone mollusks - a process which is otherwise very time consuming, as well as specifying their advantages and disadvantages. Additionally the document will feature a number of data data cleaning and transformation operations. Furthermore Ridge and Lasso regression models have been discussed and implemented. From all the models Lasso regression, for the linear side of the analysis, and decision tree regression, for the non-linear side, provided the best predictions with the tuned decision tree analysis providing the best prediction accuracy among all models.

The tools used include iPython 3 using Jupyter Notebook, visualisation libraries: Matplotlib and Seaborn, and Scikit learn module for machine learning providing functionality for implementing the regression models programmatically as well as simulating prediction of new value, and the evaluating accuracy of the model developed.

1 Models

1.1 Linear Regression

1.1.1 Definition

In statistics Linear regression is considered a simple predictive algorithm that can be used to build prediction models on quantitative data. In machine learning, it's encompassing supervised learning algorithms such as least squares technique, ridge regression, lasso regression, etc. and their role is the same - building predictive models based on quantitative data [1]. In the simplest form linear regression fits a prediction line on top of the feature data of a given dataset. As the name suggests it can be represented by a linear function and it its simplest form is:

$$f(x) = ax + b \tag{1}$$

f(x) represents the target variable,'x' is the independent variable to base our prediction on, 'a' represents the slope of the line and 'b' represents the y-axis intercept. Formula 1 assumes , however, that there's only a single independent feature, 'x', to base the prediction on. In highly simplistic terms linear regression modeling can be expressed as a question: How to chose best 'a' and 'b' values allowing most accurate prediction of f(x)? One of the methods that can be applied is to define an error function, also called a loss function, defining the amount of data not residing on a given linear function - f(x). Such an error is called a 'residual' and is defined as the distance of a given data point from the proposed regression line.

1.1.2 Measuring the Error

To define the complete error of any given regression line, denoted by a Greek letter ε(epsilon), one could simply sum up all of the residuals.

$$\epsilon = \sum_{i=1}^{n}(y_i - \hat{y_i})^2 \tag{2}$$

However, such method would be highly inaccurate as any positive residual would be immediately cancelled out, partially or fully, depending on the magnitude, by a negative residual. For that reason a number of methods to measure model errors are defined for linear regression:

1. mean absolute error (MAE)
2. root mean square error (RMSE)
3. least squares technique (RSS)

to counter the effects of the bias of the model by taking into the account the likes of the variance of the error.

Mean absolute error is defined as the average of the sum of absolute value difference between predicted and actual values:

$$MAE = \frac{\sum\limits_{i=1}^{n} |y_i - \hat{y_i}|}{n} \tag{3}$$

Its main advantage is can be considered giving all the errors the same weight meaning that the totality of errors is averaged out without any penalty applied on account of the magnitude of the error. Depending on the context, however, the same can be considered a disadvantage and should be taken into the account when performing the analysis.

Similarly to MAE, the root mean square error can be used to measure the error of the prediction model and some [2] recommend computing both measurements to asses the error of the model.

$$RMSE = \sqrt{\frac{\sum\limits_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{4}$$

The main advantage, again, depending on the context of the model under assessment, is applying weight to the magnitude of the errors detected by squaring the difference of y_i from $\hat{y}_i$.

1.1.3 Least Squares Technique

Least squares technique bases itself on summing up squares of any given residuals as unlike in the previous example where residuals could be both positive and negative thus cancelling each other out, as any given number, in the set of real numbers, will be a positive number. It also from that definition that another name for the technique, residual sum of squares or RSS, comes from.

$$RSS = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{5}$$

From equation 2 we know that the error, epsilon, for a single value can be expressed as:

$$\epsilon = (\hat{y}_i - y_i)^2 \tag{6}$$

therefore RSS can also be defined as the sum of all εvalues:

$$RSS = \sum_{i=1}^{n}\epsilon_i^2 \tag{7}$$

From there we can see another definition of a root mean square error from equation 4 that RSS can be expressed in relation to RMSE:

$$RSS = nRMSE^2 \tag{8}$$

1.1.4 Summary of Advantages and Disadvantages

Advantages:

- simplicity of calculation involved
- easy to interpret visually
- by comparison with decision tree model, quicker to execute
- can be visualized even for large datasets

Disadvantages:

- only applicable to data with linear relationship between attributes
- appropriate only for quantitative data
- often requires substantial amount of data preprocessing
  - for non-quantitative data requires attribute encoding, such as one hot encoding that in turn generates high correlation between attributes
  - multi-collinearity affects the outcome and as such correlated features have to be transformed or removed before proceeding
- only suited to regression problems

1.2 Decision Trees

1.2.1 Introduction

The decision tree is a rule-based technique for data mining that can be used, unlike linear regression described in section 1.1, for either regression or classification problems, for which different kind of tree would be implemented. If the model is expected to produce continuous quantitative data, such as prices of properties in the coming year, a regression tree would be required. For qualitative data, such as predicting diabetes in humans based on a number of symptoms, would require a classification tree. The focus of this section will be on decision tree regression.

1.2.2 Decision Tree Regression Implementation

The method implements prediction by breaking a given dataset into subsets, where subsets are homogeneous and where homogeneity is calculated via standard deviation of a given instance.

Subsequently, a number of rules, depending on the features contained in the dataset, are applied to each subset to generate tree nodes iteratively [3] in each step while at the same time keeping track of the end structure of the tree. To an extent the method could be compared to a flowchart where no feedback loops are allowed, therefore making the model easy to explain as it is relatively easy to visualize the end structure of the tree, subject to the number of prediction-contributing features.

The tree is built top-down, starting from a root node corresponding to the best predictor for the dataset, which branches would extend downwards from. Going down the tree, depending on the position, the nodes would carry a different meaning. Terminal nodes, where no more decisions can be made, are called leaf nodes while nodes from which branches, or choices, extend from are called internal nodes or decision nodes, as presence of branches extending from them implicates that a decision will have to be made as to which node the execution will progress to next.

Given the mode of operation, the function implemented is stepwise, not linear enabling the model to accommodate more complex relationships between the data. Due to the method by which subsets are determined the algorithm Additionally, it makes it less susceptible to the effects of feature collinearity, meaning that the dataset needs less preparation prior to running the prediction model.

1.2.3 Summary of Advantages and Disadvantages

Advantages:

- applicable to both qualitative and quantitative data
- easy to imagine end structure for the dataset with a small number of features
- can handle complex, non-linear relationships

Disadvantages:

- Doesn't predict a single answer, rather a number of possible answers
- prone to overfitting when model learns from the noise
- finding the best parameters to apply to regression algorithm can be a very time-consuming process - the more branches present the more time may be required to tweak the model

2 Abalone dataset analysis

Starting out with the analysis of the abalone dataset there are 4177 records, contained in rows numbered from 0 to 4176 and separated into 9 columns with each column corresponding to an attribute, or as called later during the analysis phase, feature.

The purpose of this analysis is to predict the age of an abalone, calculated as the number of rings +1.5, based on the linear regression and decision tree models and it will begin by analyzing the data provided accounting for any missing data, data errors, such as measurements that are not in line with domain rules, etc. The analysis will be carried out using Jupyter Notebook interface using iPython 3 language and any graphs, figures, tables, etc., unless explicitly stated, should be assumed as a result of a programmatic analysis carried out in same.

Name	Units	Description	Imported Datatype
Sex	M, F, I	M (male) F (female) I (infant)	object
Length	mm	Longest shell measurement	floating point number
Diameter	mm	Perpendicular to length	floating point number
Height	mm	With meat in shell	floating point number
Whole weight	grams	Whole abalone	floating point number
Shucked weight	grams	Weight of meat	floating point number
Viscera weight	grams	Gut weight (after bleeding)	floating point number
Shell weight	grams	After being dried	floating point number
Rings	-	+1.5 gives the age in years	object

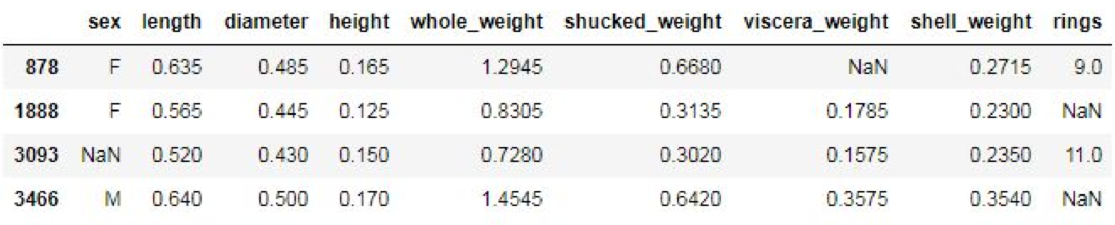
Table 1: Attributes Found in the Dataset

2.1 Identifying and removing erroneous data

2.1.1 Identifying non-numerical values

Starting off with the analysis the first thing to check is the state of the data, including its data types - any calculations can only be carried out on numerical data after all. One of the methods that can be used to enumerate non-numerical data, one hot encoding, will be presented and discussed in the subsequent sections.

First thing to notice when checking types of data imported from abalone dataset is that neither ***sex*** nor ***rings*** attributes are considered numerical. Both have been imported as an "object" type, as per Table 1, and while it is expected for the sex attribute, rings should have been treated as numerical data. The fact that it is treated as otherwise could indicate an error in the data and needs to be investigated. Instead of checking each record individually and verifying if the rings value could be converted to a number instead decided to force conversion of rings column into numbers, while ignoring any errors, knowing that any data that did not get converted to a legitimate number will be turned into a "NaN" (not a number) value which in turn can be detected easily along with non-numerical values in other columns. After checking for the presence of non-numerical values as presented in figure 1 it has been discovered that following rows hold erroneous data:



	sex	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
878	F	0.635	0.485	0.165	1.2945	0.6680	NaN	0.2715	9.0
1888	F	0.565	0.445	0.125	0.8305	0.3135	0.1785	0.2300	NaN
3093	NaN	0.520	0.430	0.150	0.7280	0.3020	0.1575	0.2350	11.0
3466	M	0.640	0.500	0.170	1.4545	0.6420	0.3575	0.3540	NaN

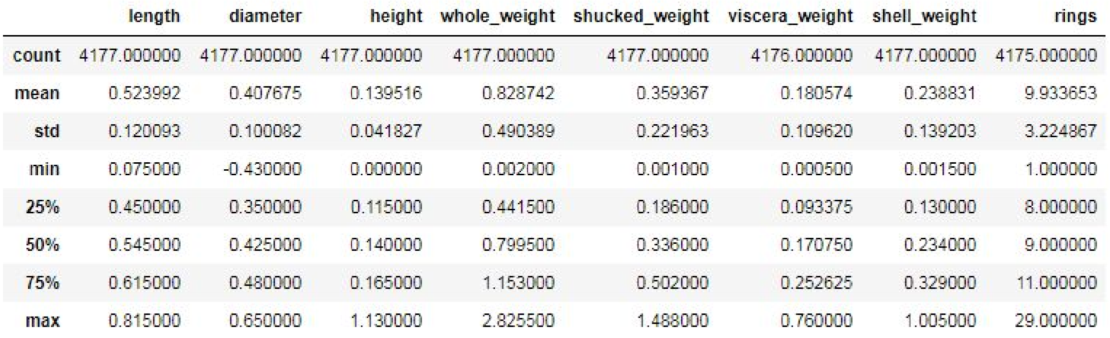
Figure 1: Non-numerical values per column

- row 878 is missing ***viscera weight*** value
- row 1888 is missing ***rings*** value
- row 3093 is missing ***sex*** value
- row 3466 is missing ***rings*** value

Currently, only the analysis of the magnitude of the erroneous data is relevant and until such time that all errors within reason are identified no data will be replaced or removed.

2.1.2 Identifying values equal to, or below, 0

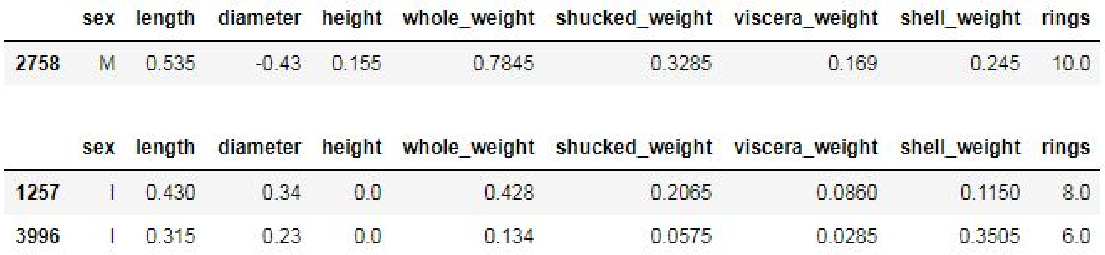
For a living organism such as an abalone, it's unfeasible for any of its attributes to be equal to or below 0 value therefore, the data provided should be checked for same. Starting with the simple method to describe the data, with especially its minimums per column we can determine if there are any negative or 0 values present.



	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4176.000000	4177.000000	4175.000000
mean	0.523992	0.407675	0.139516	0.828742	0.359367	0.180574	0.238831	9.933653
std	0.120093	0.100082	0.041827	0.490389	0.221963	0.109620	0.139203	3.224867
min	0.075000	-0.430000	0.000000	0.002000	0.001000	0.000500	0.001500	1.000000
25%	0.450000	0.350000	0.115000	0.441500	0.186000	0.093375	0.130000	8.000000
50%	0.545000	0.425000	0.140000	0.799500	0.336000	0.170750	0.234000	9.000000
75%	0.615000	0.480000	0.165000	1.153000	0.502000	0.252625	0.329000	11.000000
max	0.815000	0.650000	1.130000	2.825500	1.488000	0.760000	1.005000	29.000000

Figure 2: Dataset statistics per column

The statistics computed and presented in figure 2 clearly state that ***diameter*** and ***height*** attributes both contain rows with illegal values that need to be identified and addressed. Using simple Python commands it can be identified, as shown on figure 3 that row indexed as 2758 contains negative ***diameter*** while rows 1257 and 3996 contain 0 value for ***height***.



	sex	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
2758	M	0.535	-0.43	0.155	0.7845	0.3285	0.169	0.245	10.0

	sex	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
1257	I	0.430	0.34	0.0	0.428	0.2065	0.0860	0.1150	8.0
3996	I	0.315	0.23	0.0	0.134	0.0575	0.0285	0.3505	6.0

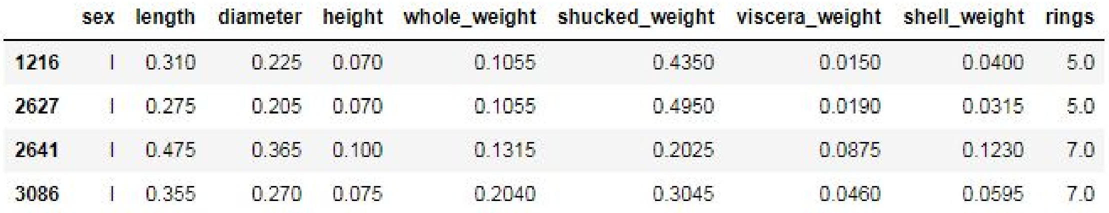
Figure 3: Rows with negative or 0 values

2.1.3 Analysing if shucked or shell weights are greater than whole abalone

Endemic to abalone itself it is invalid for any specimen to have:

1. higher weight when shucked (stripped of its shell) than whole weight
2. higher viscera weight than the whole weight
3. higher shell weight than the whole weight

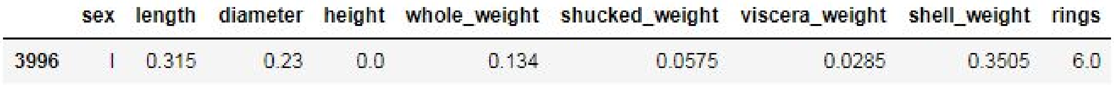
and such records can be identified either by a computation method or by drawing a scatter plot dedicated to specific parameters listed above. While it is possible to identify errors using a visual method by narrowing down the graph field to the precise location of the erroneous data it is much faster to compute it.



	sex	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
1216	I	0.310	0.225	0.070	0.1055	0.4350	0.0150	0.0400	5.0
2627	I	0.275	0.205	0.070	0.1055	0.4950	0.0190	0.0315	5.0
2641	I	0.475	0.365	0.100	0.1315	0.2025	0.0875	0.1230	7.0
3086	I	0.355	0.270	0.075	0.2040	0.3045	0.0460	0.0595	7.0

Figure 4: Rows with shucked weight greater that whole weight

Figures 4 and 5 present a total of 5 records that do not meet the criteria stated in list 2.1.3. Additionally, a check to determine if for any record viscera weight is greater than that of a whole specimen has been carried out with no records showing that particular discrepancy.



	sex	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
3996	I	0.315	0.23	0.0	0.134	0.0575	0.0285	0.3505	6.0

Figure 5: Rows with shell weight greater that whole weight

2.1.4 Summary of erroneous data records

During the analysis described in section 2.1.1 through 2.1.3 a total of 11 records have been identified as erroneous with row 3996 being flagged by 2 separate checks - height having a 0 value and shell weight being higher than that of whole abalone specimen - it was only counted once towards the total number of errors. 11 rows constitute less than 0.26% of the dataset and as such removal of the erroneous rows will not affect further analysis significantly. However, had the number been more significant the missing data would have had to have been substituted with either mean or median values for a given attribute.

2.2 Identifying outliers in the dataset

The next logical step in performing the analysis is to identify any data outliers, also referred to residuals, that may affect future calculations. For the purpose of this analysis, the outliers are defined as a data point lying either $1.5 * IQR$ below first quartile ($Q1$) or $1.5 * IQR$ above third quartile ($Q3$). Given figures 6 and 7 it is easy to observe that outliers fitting definition above are present in the dataset as the $1.5 * IQR$ range is defined by the whiskers on box plot for each attribute. The graphs have been split into 2 for clarity as scores for rings attribute are significantly larger that that of other attributes, thus obfuscating their details.

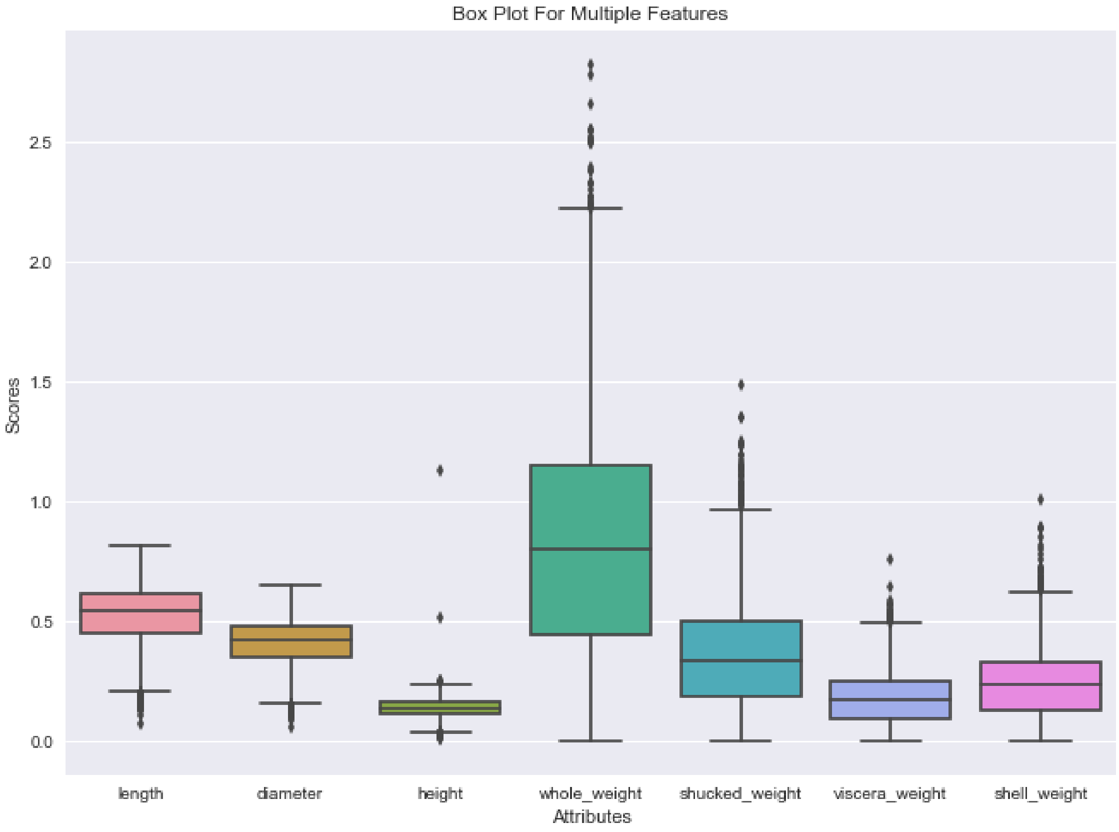


Figure 6: Box Plot of Dataset Attributes

While it is easy to determine using a visual method that outliers are present in the data, it is insufficient a method for determining how many data points are truly affected. Therefore, a

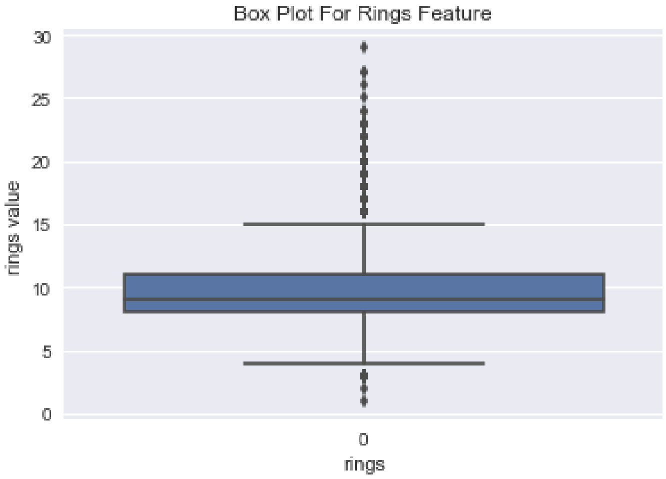


Figure 7: Box Plot of Rings Attribute

simple calculation executed in Jupyter Notebook using Python was used to determine the following outliers' counts for each attribute, as per Table 2

	Outliers count
Rings	278
Length	30
Diameter	7
Height	3
Whole weight	31
Shucked weight	20
Viscera weight	16
Shell weight	11

Table 2: Summary of outliers

The total number of outliers found is 396 which constitutes of the 9.5% of the current dataset, which now comprises of 4166 rows ($4177 - 11$ where was the initial row count and 11 is the number of erroneous data points removed as per conclusion of section 2.1.4).

It has already been alluded to in section 2.1.4 that the volume of erroneous, or as in this case, outlying, data affects the method of rectifying the affected data points. In the case presented removing even 396 data points still leaves 3770 points to carry analysis on, therefore for simplicity's sake, the data points were removed.

2.3 Checking the Distribution of the Cleaned Data

Since the data has been cleaned the distribution of the data can now be checked. The operation has been carried out via calculation for precise measurements with results collected in Table 3 as well via generating histograms for each of the 8 numerical attributes present: height, length, diameter, whole weight, shucked weight, shell weight, viscera weight and rings with results presented on Figures 8 through 15. Additionally, similar observations could be made from boxplot graphs from Figures 6 and 7 which clearly display the IQR and median, however, it has been deemed prudent to carry out additional analysis in order to make definite observations.

Starting with length and diameter attributes, both appear approximately symmetrical. Despite gathered metrics showing signs of mean value being smaller than the median, which would define a left-skew of data, the magnitude of the difference is very small. Subsequently, height attribute has been checked and with its mean and median almost matching it is decidedly the most symmetrically distributed attribute in the dataset.

From all sources, it can be surmised that the distribution of all of the weight attributes follows the same pattern of even distribution. While it may appear that data is skewed right it is only because the graphs terminate at 0 value, where due to domain reasons none of the attributes can hold negative values. Visual cues are backed up by the numerical data from Table 3 where the difference between mean and median, while mean is indeed greater of the two numbers, is too small to conclude skewness of the data. The final attribute, rings, also appears to be symmetrically distributed, Figure 15, with the mean and median values collected, do support that argument.

To conclude the features of the dataset are distributed in approximately symmetrical fashion allowing for reliable prediction of typical results for a given value of a given attribute - which will be the focus of sections 3, 4 and 5.

	Mean	Std Deviation	Minimum	Q1	Median	Q3	Maximum
Length	0.5199	0.1121	0.2000	0.4463	0.5350	0.6100	0.7600
Diameter	0.4040	0.0927	0.1500	0.3450	0.4200	0.4750	0.6000
Height	0.1370	0.0354	0.0300	0.1100	0.1400	0.1650	0.2400
Whole Weight	0.7877	0.4424	0.0385	0.4296	0.7640	1.1150	2.1275
Shucked Weight	0.3455	0.2032	0.0115	0.1800	0.3250	0.4910	0.9600
Viscera Weight	0.1727	0.1004	0.0005	0.0900	0.1635	0.2435	0.4690
Shell Weight	0.2247	0.1220	0.0065	0.1250	0.2200	0.3110	0.5965
Rings	9.4125	2.3368	4.0000	8.0000	9.0000	11.0000	15.0000

Table 3: Distribution Metrics for Cleaned Dataset

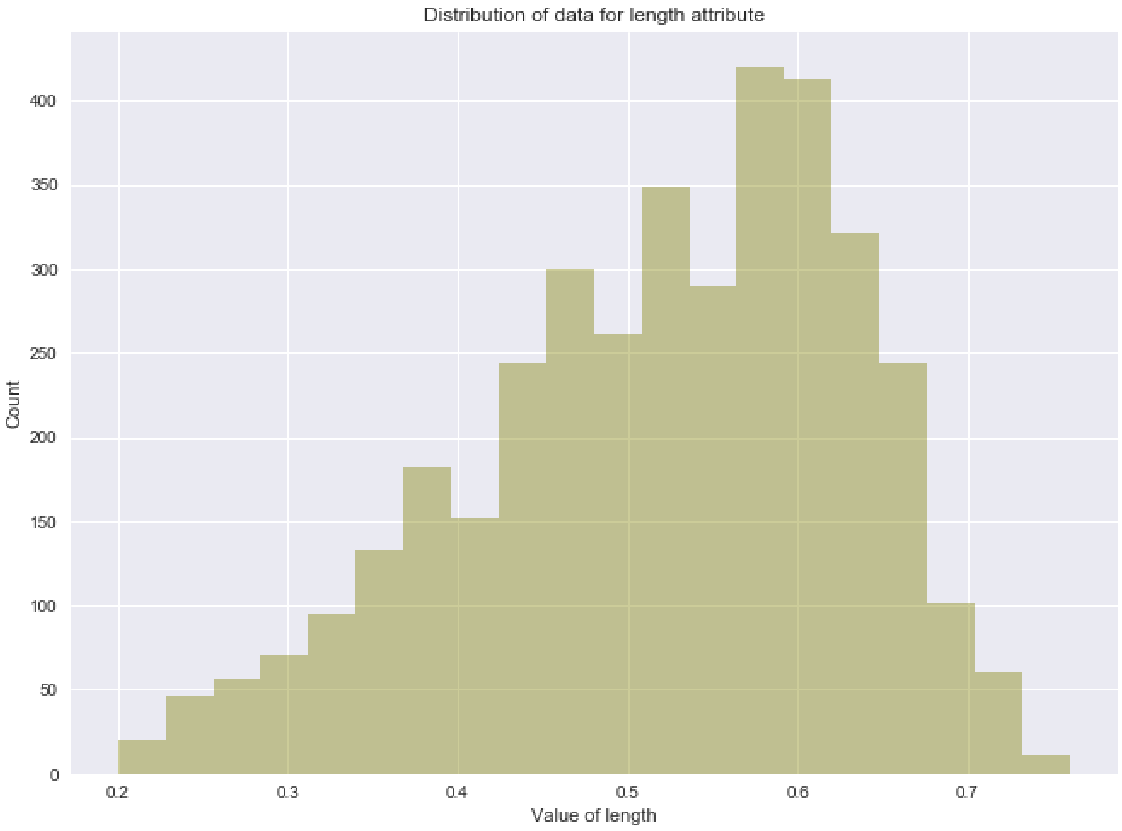


Figure 8: Histogram of Length Attribute

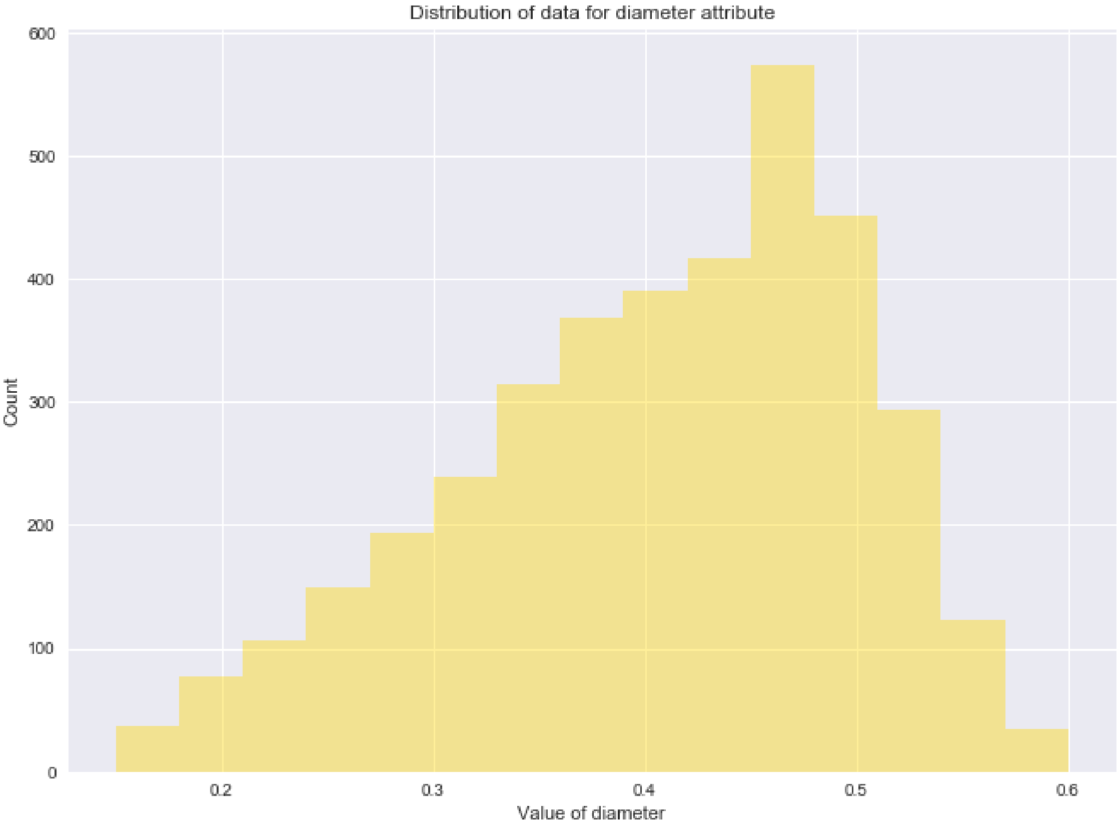


Figure 9: Histogram of Diameter Attribute

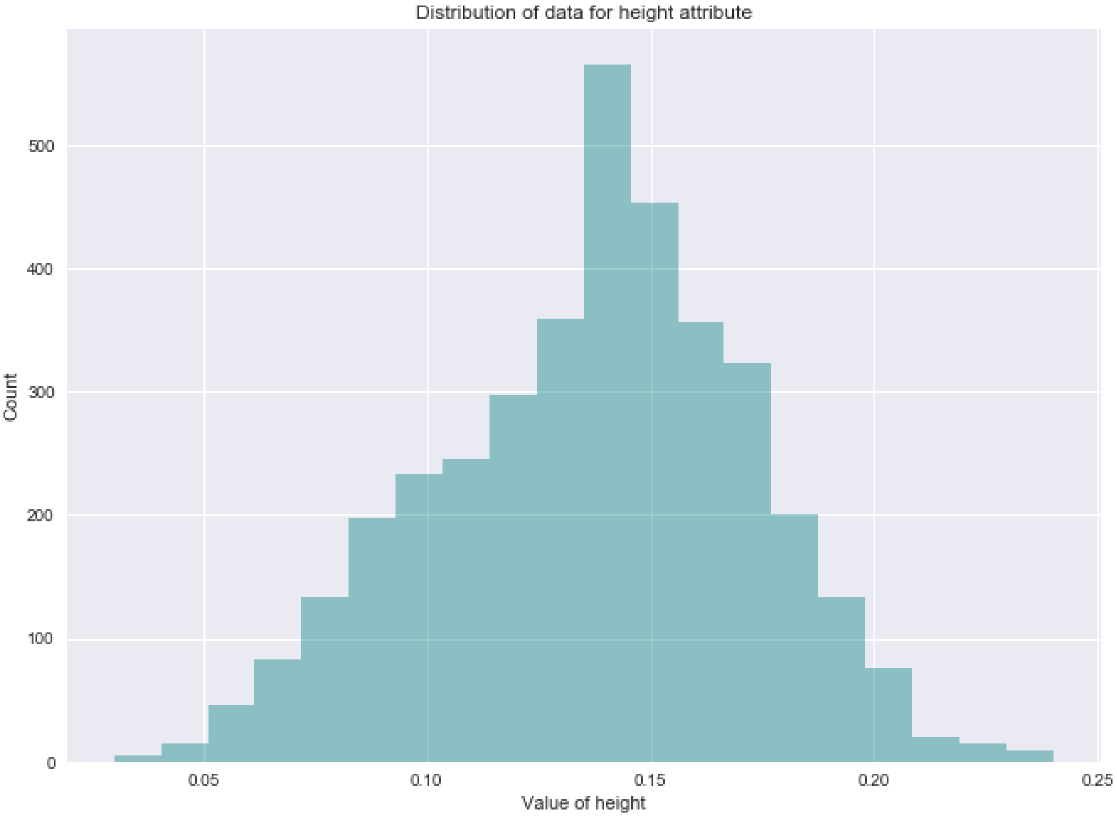


Figure 10: Histogram of Height Attribute

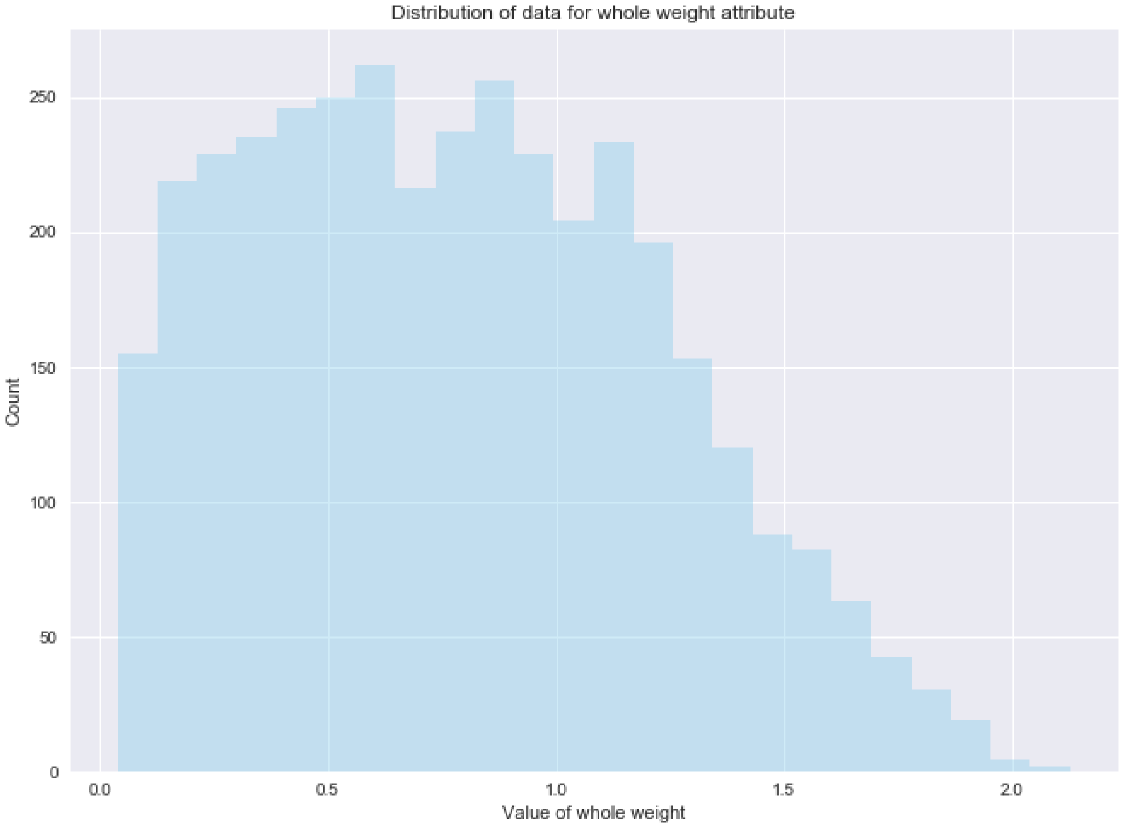


Figure 11: Histogram of Whole Weight Attribute

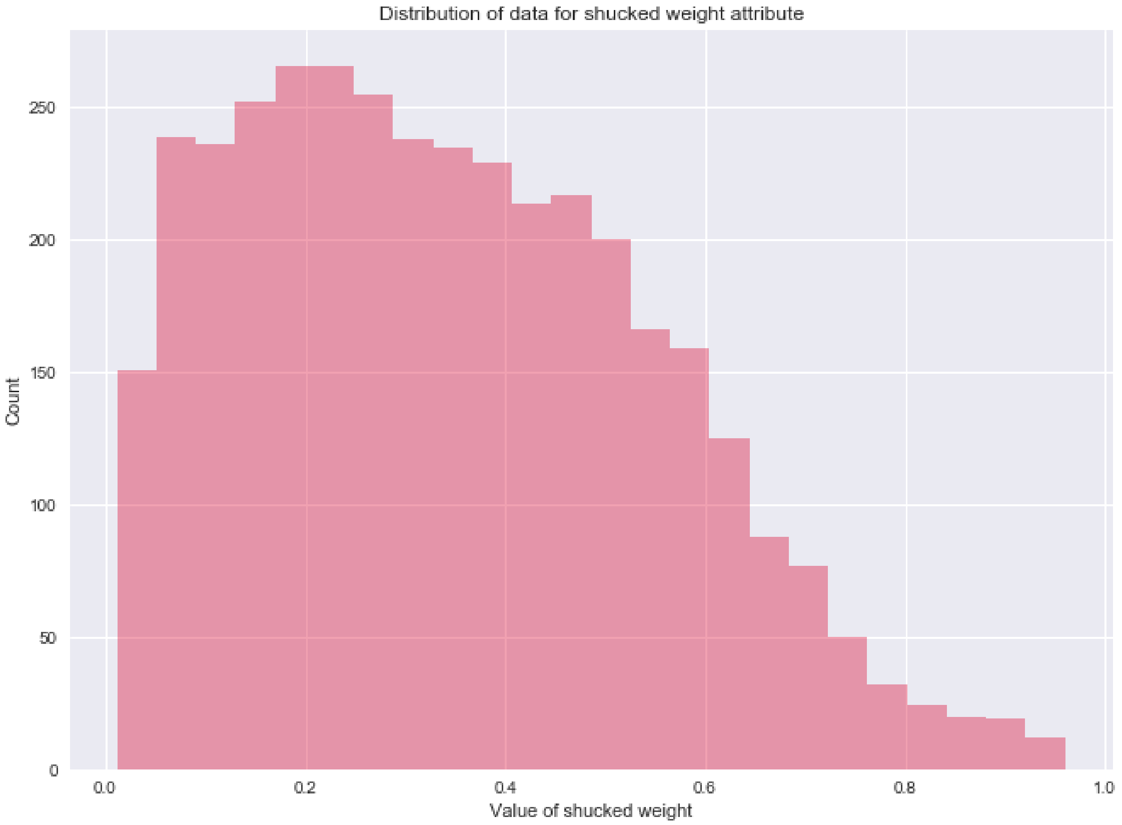


Figure 12: Histogram of Shucked Weight Attribute

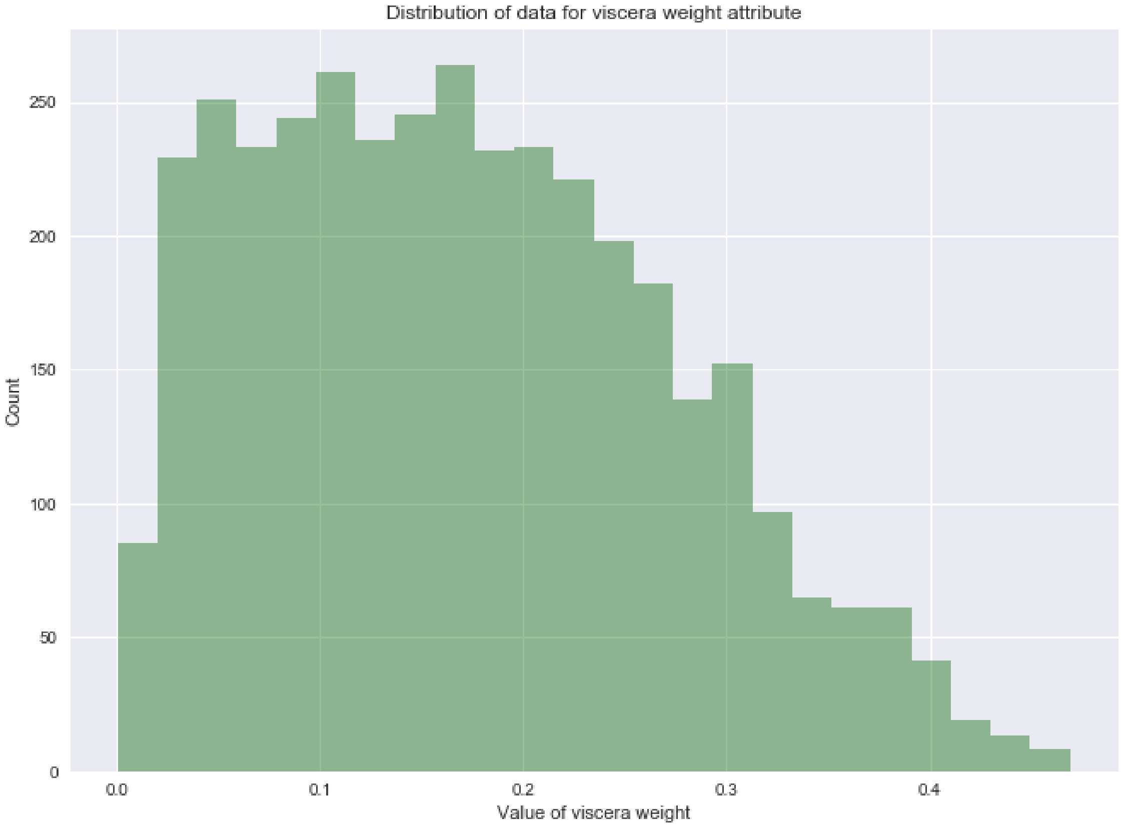


Figure 13: Histogram of Viscera Weight Attribute

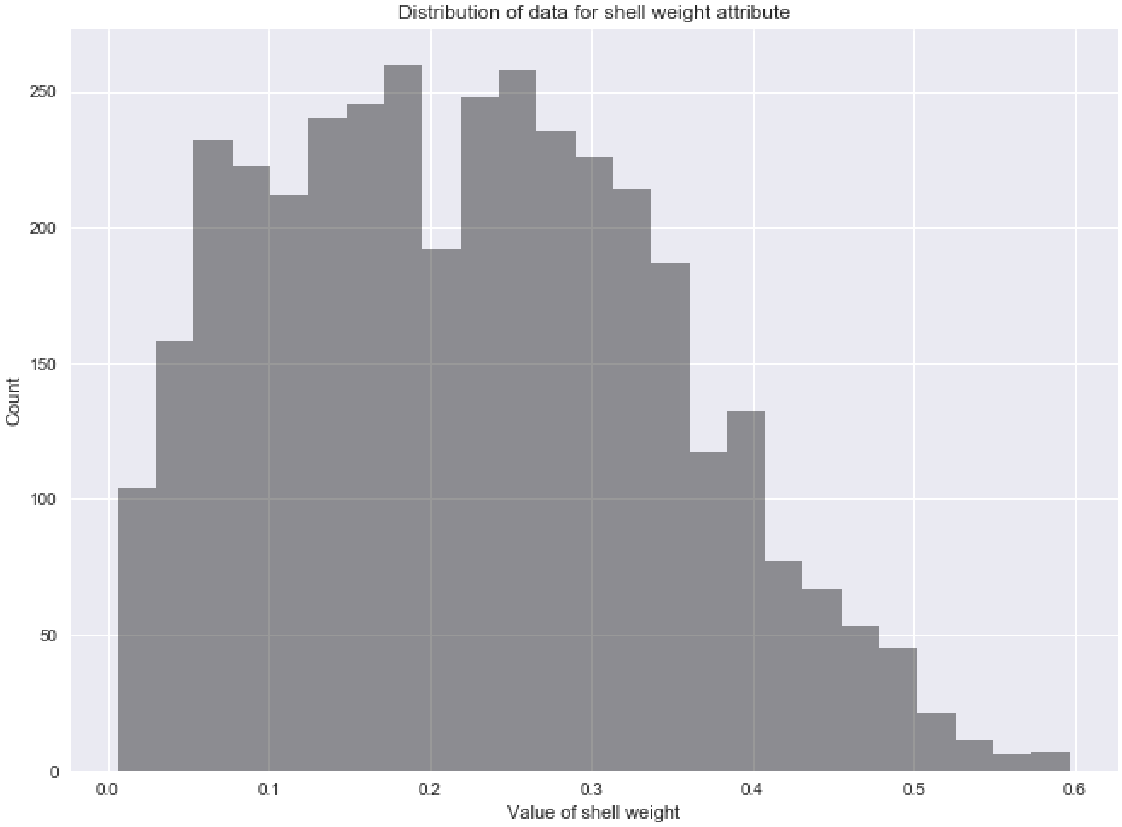


Figure 14: Histogram of Shell Weight Attribute

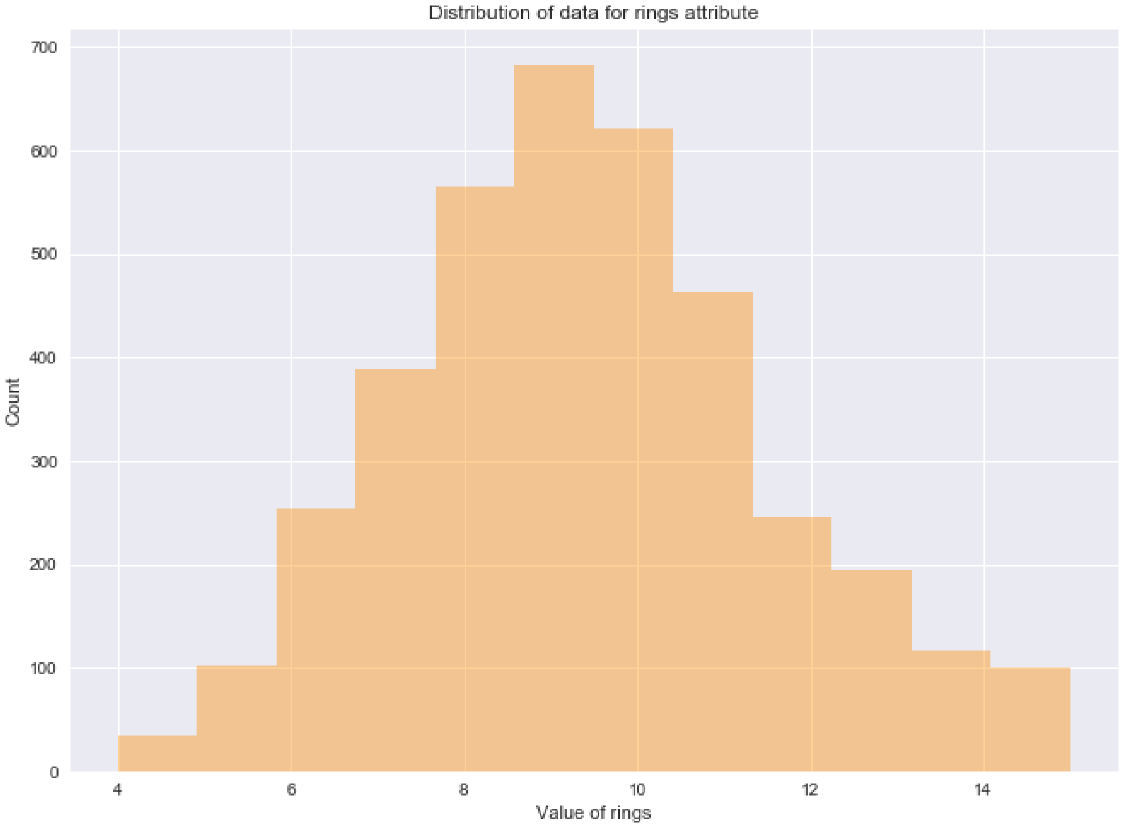


Figure 15: Histogram of Rings Attribute

2.4 Identifying relationships between features

2.4.1 Detecting feature correlation

Following the identification and removal of any data outliers in order to build the prediction model features most relevant to the ability to predict abalone's age via predicting its number of rings need to established. One of the methods of identifying such features is via determining the correlation between the attributes. A high correlation between features needs to be investigated as it will skew the prediction model. Creating a heat map, as presented in Figure 16 is an easy way to visualize feature correlation and it was chosen over correlation matrix table precisely because it allows for a quick interpretation of the results. From the heat map it is evident that there is a strong correlation between height, length and diameter features indicating that they should either be transformed into a new feature or removed, which is also considered a feature-transformation method. In the case of this specific dataset aforementioned attributes can be used to calculate the volume of an abalone specimen, thus, they can be transformed into a single feature. The calculation used was $height * length * diameter$ and while not mathematically correct for calculating the volume of a cylinder, it is deemed sufficient as a feature transformation technique. Additionally, while transforming Figure 17 and Figure 18 detail the post-transformation feature correlation factors and from them it is evident that the lowest correlation exists between rings and shucked weight where rings are the target variable. To ensure that all features are taken into the account when investigating the most significant feature contributing towards the prediction model if any, all of the features must be represented as a quantitative data, sex feature included. The challenge with encoding abalone's sex comes from the fact that there are 3 possible values for it male, female and infant, meaning it doesn't translate straight into a boolean value. The easiest solution to this problem is to utilize one-hot encoding where non-numerical, non-boolean values are encoded into quantitative data. In abalone's case, sex attribute will generate 3 new, mutually exclusive, features, Female (F), Male (F) and Infant (I), where only 1 of the listed can hold 1, or true value, per record. At the same time, sex feature is removed from the dataset. Unfortunately, one hot encoding comes at a significant disadvantage to the state of the data as newly created F, M and I features are highly correlated with one another.

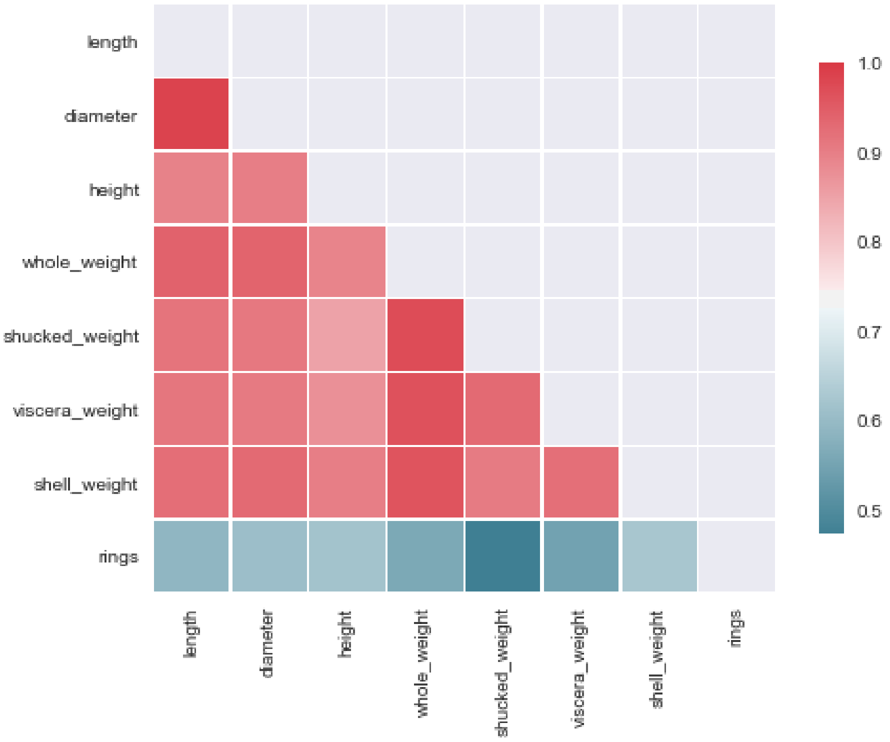


Figure 16: Feature Correlation Heat Map

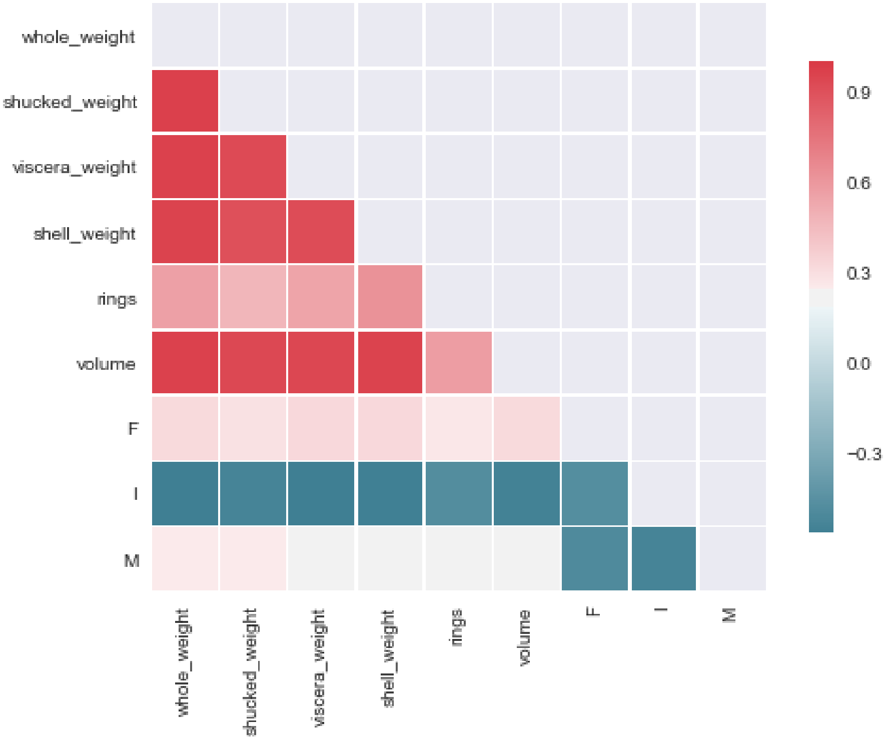
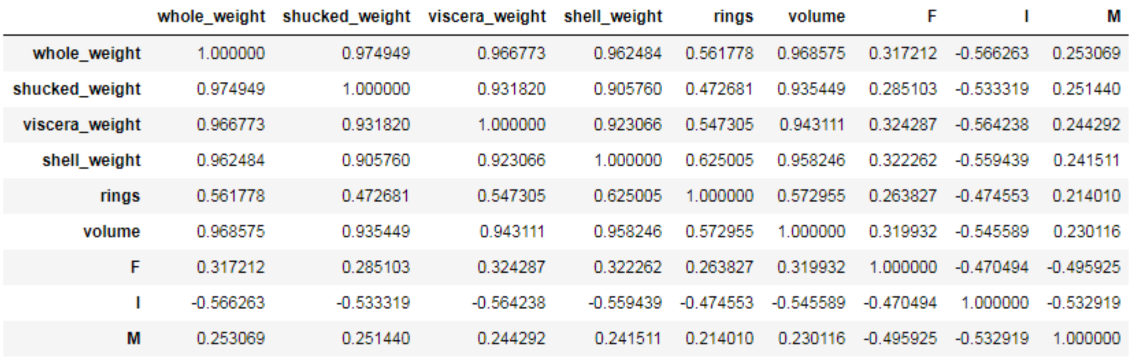


Figure 17: Transformed Feature Correlation Heatmap

The last question that remains to be answered is that of whether the relationship between the features is linear, as it is the only type of relationship that linear regression can accommodate. Examining the relationship of shucked weight to the number of rings on Figure 19 it is observed that the relationship is indeed somewhat linear, however, it does not appear to follow the same



	whole_weight	shucked_weight	viscera_weight	shell_weight	rings	volume	F	I	M
whole_weight	1.000000	0.974949	0.966773	0.962484	0.561778	0.968575	0.317212	-0.566263	0.253069
shucked_weight	0.974949	1.000000	0.931820	0.905760	0.472681	0.935449	0.285103	-0.533319	0.251440
viscera_weight	0.966773	0.931820	1.000000	0.923066	0.547305	0.943111	0.324287	-0.564238	0.244292
shell_weight	0.962484	0.905760	0.923066	1.000000	0.625005	0.958246	0.322262	-0.559439	0.241511
rings	0.561778	0.472681	0.547305	0.625005	1.000000	0.572955	0.263827	-0.474553	0.214010
volume	0.968575	0.935449	0.943111	0.958246	0.572955	1.000000	0.319932	-0.545589	0.230116
F	0.317212	0.285103	0.324287	0.322262	0.263827	0.319932	1.000000	-0.470494	-0.495925
I	-0.566263	-0.533319	-0.564238	-0.559439	-0.474553	-0.545589	-0.470494	1.000000	-0.532919
M	0.253069	0.251440	0.244292	0.241511	0.214010	0.230116	-0.495925	-0.532919	1.000000

Figure 18: Transformed Feature Correlation Data

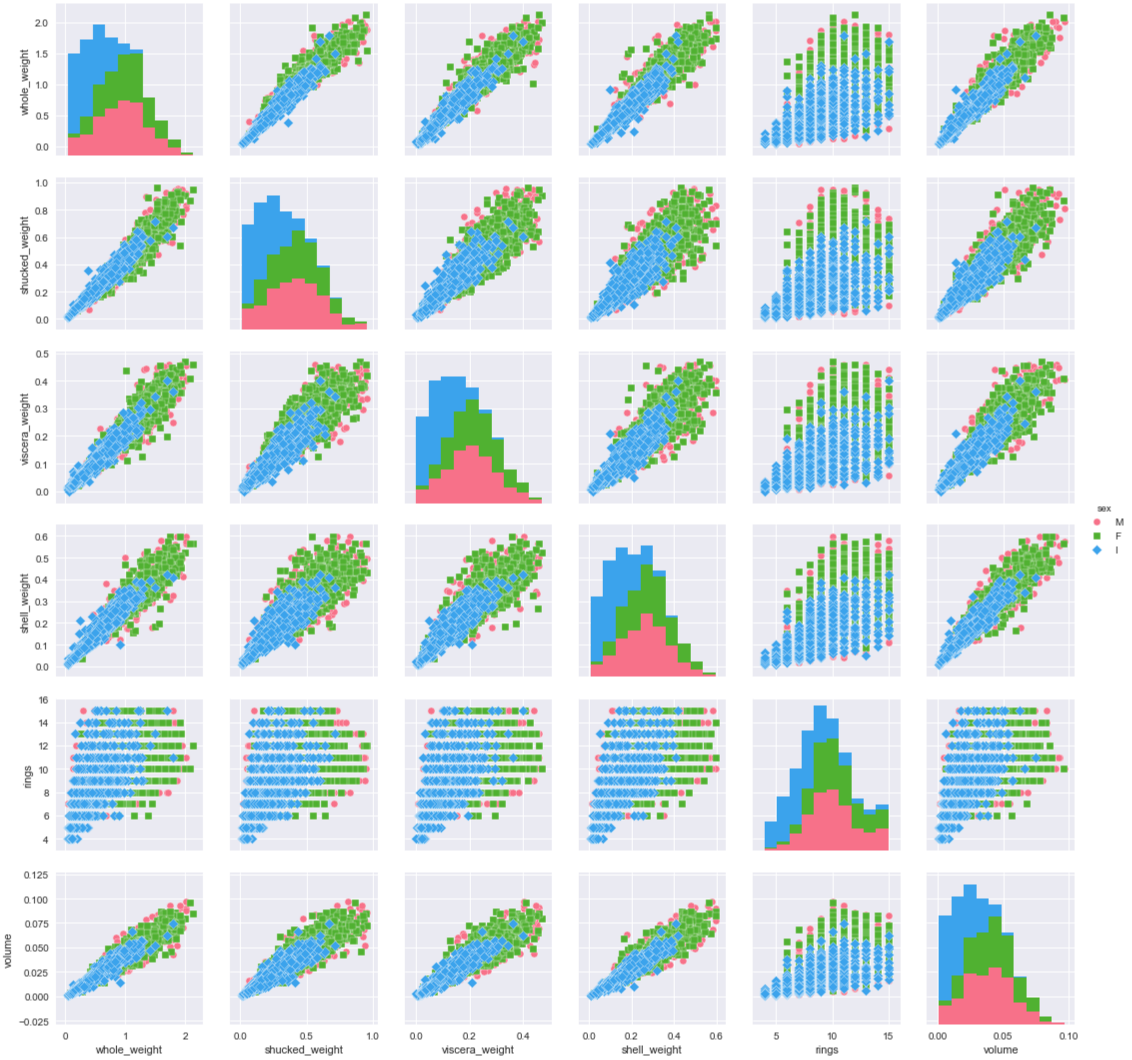


Figure 19: Feature scatterplot

linear function per abalone's gender. It can be observed from Figures 20 and 21 that infant abalones follow a different linear function, which is expected to impact the analysis, however, at this point in time the only feasible solutions are to either exclude infant abalones from the complete analysis or try to build a model accommodating such a significant discrepancies. The shucked weight feature was chose for illustrating this point due to its low level of correlation with target variable. Volume feature was chosen for the second graph for the exact opposite reason, its high correlation to target variable, to illustrate that level of correlation between features does not have an effect an the

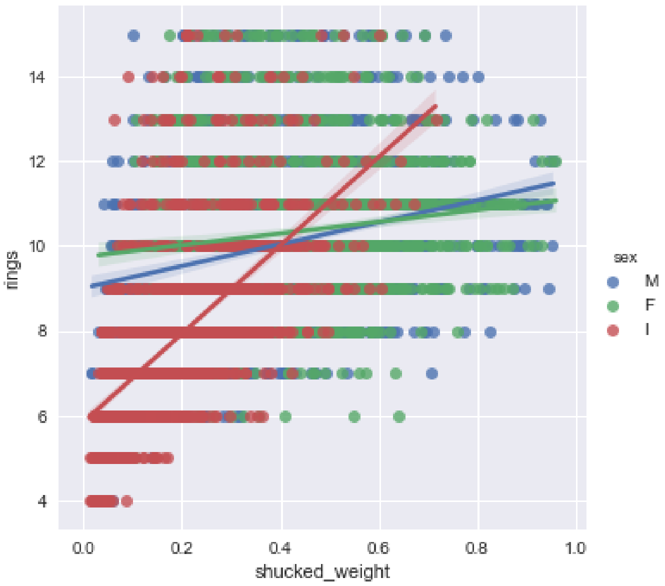


Figure 20: Shucked weight to rings relationship

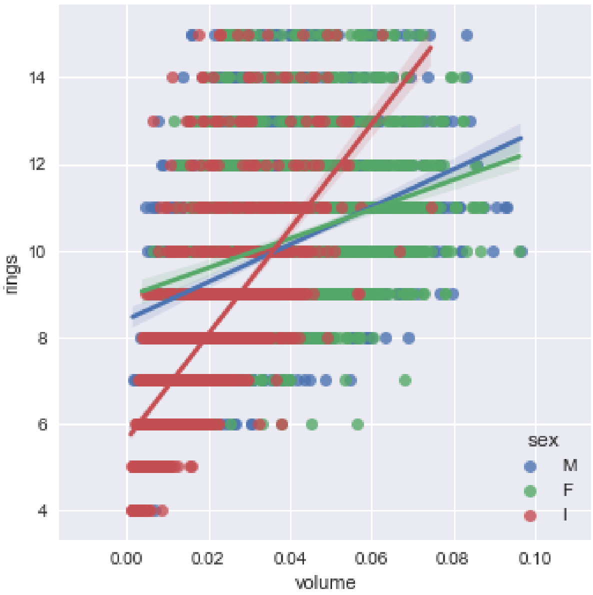


Figure 21: Volume to rings relationship

observed difference of linear function between genders.

Satisfied that linear relationship between data does indeed exist linear regression can be performed on the dataset.

3 Performing Linear Regression

Dummy regression has been selected to act as a baseline for detecting model improvements. Using DummyRegressor package available from sci-kit learn library as well as using k-fold cross-validation technique a baseline for RSS, Mean Absolute Error(MAE), Mean Squared Error(MSE) and Standard deviation (σor sigma) of the error has been established. K-fold cross-validation is a technique where a given dataset is divided into an X amount of equally sized chunks called folds and then the model is trained iteratively X amount of times. During each iteration 1 fold is reserved for validation set while remaining ones are used as training set.

For the Dummy Regressor a cross-validation analysis using K-fold of 5, 10, 15 and 20 has been performed, with the RSS, MAE, MSE and σvalues not improving between the 2 models, on the contrary as the number of folds increased so did all of the metrics mentioned indicating dropping performance of the model. After training the dummy model a new target prediction has been made using cross-validation method again using 5 and 10 K-fold splits. From the baseline distribution of errors diagram in Figure 22 a slight left skew of data can be observed with most of errors centered left of 0 value. However, the distribution of the error is almost normal (Gaussian).

	MAE	MSE	RMSE	Std dev	RSS
Dummy 5 fold	1.870800	5.46082	2.33684	2.33715	0.0
Dummy 10 fold	1.870950	5.46259	2.33722	2.337529	0.0
Linear Regression Base Model	1.247948	2.60970	1.61546	1.615670	0.52459
Linear Regression Transformed Model	1.306114	2.86274	1.69196	1.615670	0.47729

Table 4: Summary of collected statistics for linear regression

Subsequently, linear regression, using 5 K-fold cross-validation was performed on both base model and the transformed one, where volume feature has been generated. A number of analyses have been run to determine that 5 k-fold yielded to the lowest MAE, MSE, and RMSE metrics, while at the same time marginally, in the range of thousandth floating point, higher RSS score. Details and comparison of all metrics obtained can be seen in Table 4. It can be observed that

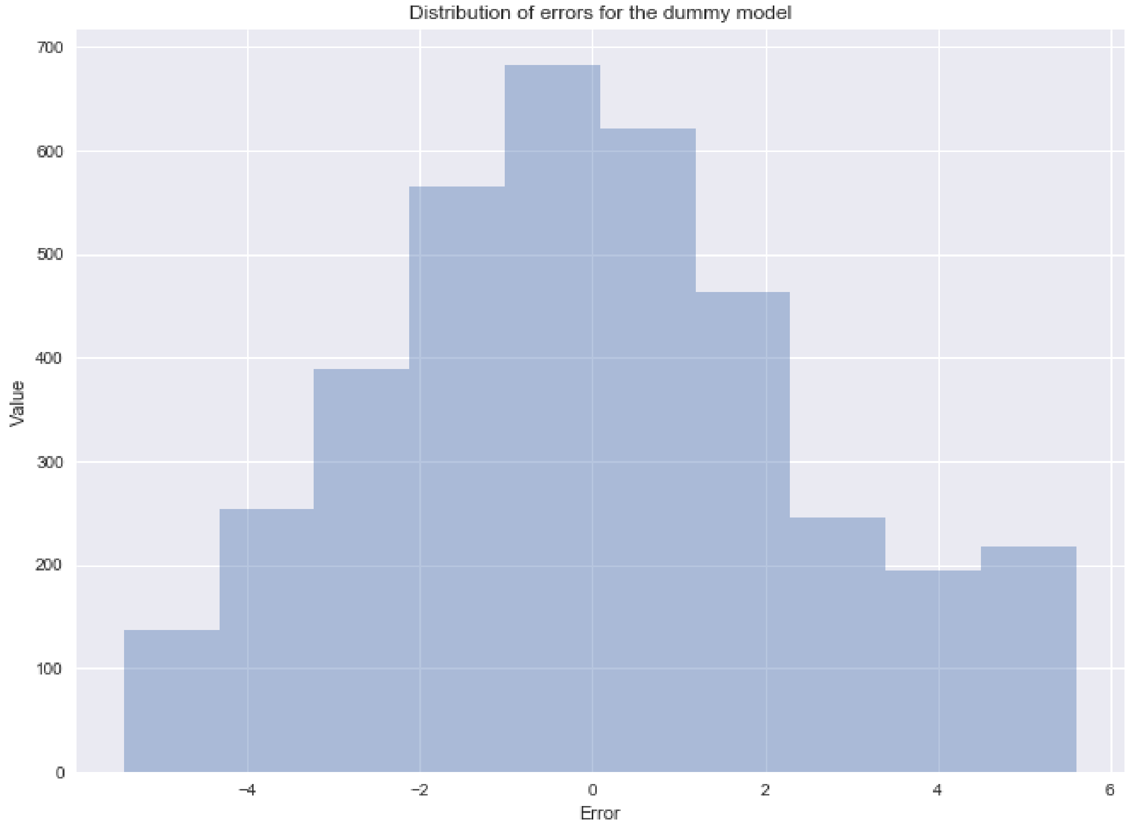


Figure 22: Distribution of errors for the dummy model using transformed dataset

linear regression executed on the base dataset, without transformed features, has achieved better metrics on almost all accounts, with standard deviation being the only one where they are equal.

From Figures 22 and 23 it can be observed that the distribution of error count changes significantly between the 2, with linear regression achieving higher number of smaller errors centered slightly left of 0 value, while the baseline implementation presenting more of a normally distributed figure. It's an indicator that the linear regression model, while containing errors, contains less number of large errors, therefore achieving greater accuracy.

Linear regression, while performing better than the baseline, its accuracy still leaves much to be desired as the error counts are quite significant. One possible improvement that can be done is attempting to predict the number of rings using a model that is not as susceptible to feature collinearity as linear regression is with possible linear analysis candidates being Ridge or Lasso Regression. Alternatively, it could be attempted to further transform the features, however, the danger is, as verified with transforming of height, length and diameter features into a volume one, the end result may actually not achieve greater accuracy.

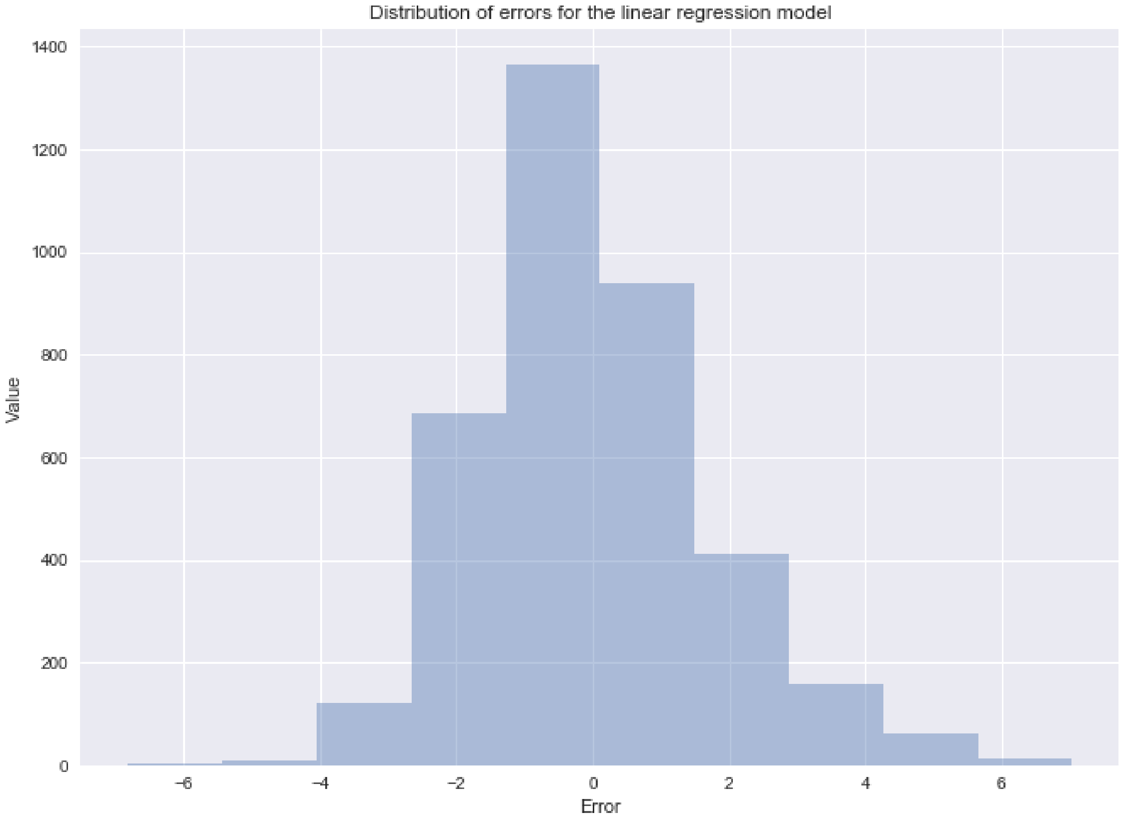


Figure 23: Distribution of errors for the linear regression model using transformed dataset

4 Performing Decision Tree Regression

Accuracy of all prediction iterations has been calculated using Mean Absolute Percentage Error (MAPE) as part of the calculation, as there are no features with 0 values, which if present would immediately disqualify the measurement. MAPE is defined as the sum of the absolute value of the difference between actual and predicted values multiplied by 100% and subsequently divided by the number of occurrences, as per equation 9.

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y_i}}{y_i} \right| \tag{9}$$

As a measure of the percentage of errors in order to get a semblance of the accuracy of the model, it is assumed that $Accuracy = 100\% - MAPE$. The analysis consisted of 3 decision tree models:

1. baseline, with no parameters set
2. unoptimized decision tree with randomly assigned values for:
   - `min_samples_split=30`
   - `min_samples_leaf=10`
   - `random_state=0`
3. model with parameters obtained by a number of iterations of executing `GridSearchCV` to obtain best parameters. Different ranges for the following parameters have been specified for each iteration with Table 5 summarizing what ranges have been checked for each of the parameters listed

The parameter ranges have been adjusted iteratively through executing multiple grid search tryouts as well as checking which attempt yielded the best accuracy when fitted to the model. The range for
texttttmax_depth parameter has been purposefully decreased as the increase of it yielded no visible

Parameter name	Range Checked using Grid Search
criterion	mse, mae, friedman_mse
max_depth	1-10 for mae & 1-20 for mse
min_samples_leaf	1-21
min_samples_split	2-11
min_weight_fraction_leaf	0-0.5

Table 5: Best Parameter Search Specification

improvement. It has been predicted, however, that due to the number of correlated features a criterion minimizing the L1 loss will possibly yield the best outcome - with more on L1 and L2 regularization discussed in section ??. The results of each iteration specifying best results for each criterion have been summarized in Table 6.

	Accuracy (%)	MAE	Std dev	MSE	RMSE
Baseline	81.91679	1.67692	2.25067	5.06631	2.25085
Unoptimized	85.60937	1.33001	1.73331	3.00358	1.73308
Best with MAE	86.84227	1.24271	1.71844	2.97759	1.72557
Best with MSE	86.35139	1.26292	1.66292	2.76458	1.66270
Best with Friedman MSE	86.27439	1.26561	1.65593	2.74137	1.65571

Table 6: Summary of the Metrics Identified Between Model Iterations

As for the distribution of the errors it can be observed from Figure 24 depicting the baseline for further model optimization with no additional parameters specified that the range of errors is greater than that of unoptimized or optimized decision trees, shown on Figures 25 and 26 respectively. The distribution is mostly symmetrical therefore it can be assumed that mean and median are approximately at the same value with little to no skew observed. From the same figures, it can be observed that even minimal optimization improves the distribution of the errors, for starters by reducing the range of errors, meaning that less significant misses are present in the prediction.

The results of the analysis, listed in Table 6, indicate that, as predicted, the best accuracy score has been obtained by using mean absolute error criterion for generating subsets of the dataset, with additional parameters listed in Table 7.

Parameter name	Parameter Value Chosen
criterion	mae
max_depth	8
min_samples_leaf	13
min_samples_split	7
min_weight_fraction_leaf	0

Table 7: Chosen Parameters Yielding Best Accuracy Results

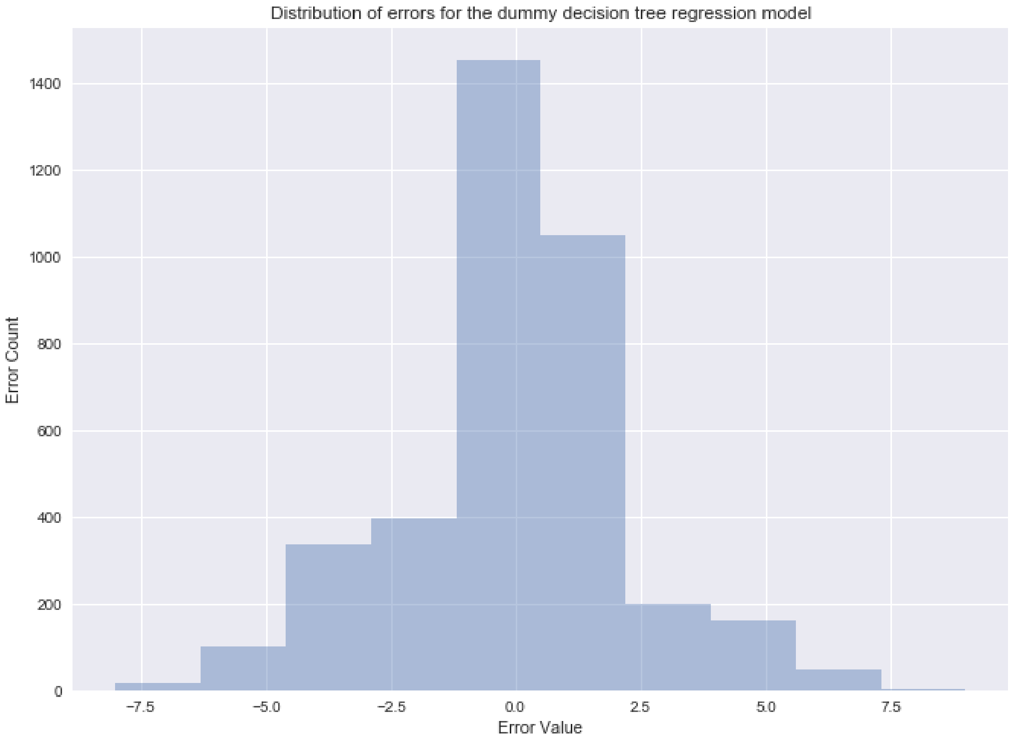


Figure 24: Distribution of Errors in Dummy Decision Tree

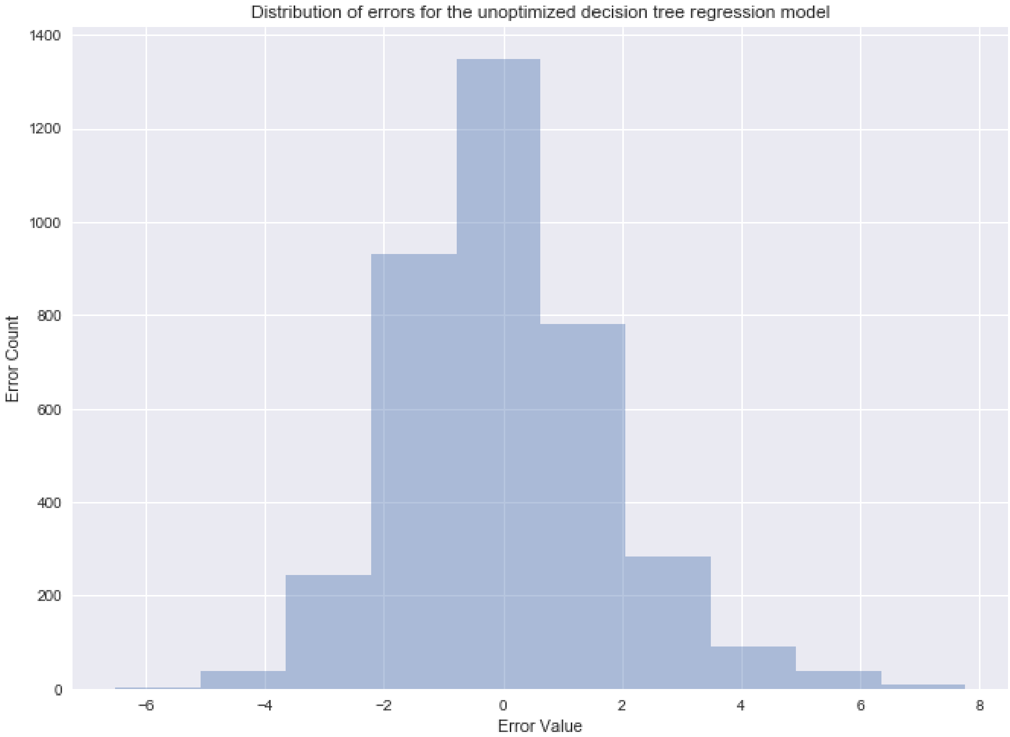


Figure 25: Distribution of Errors in Unoptimized Decision Tree

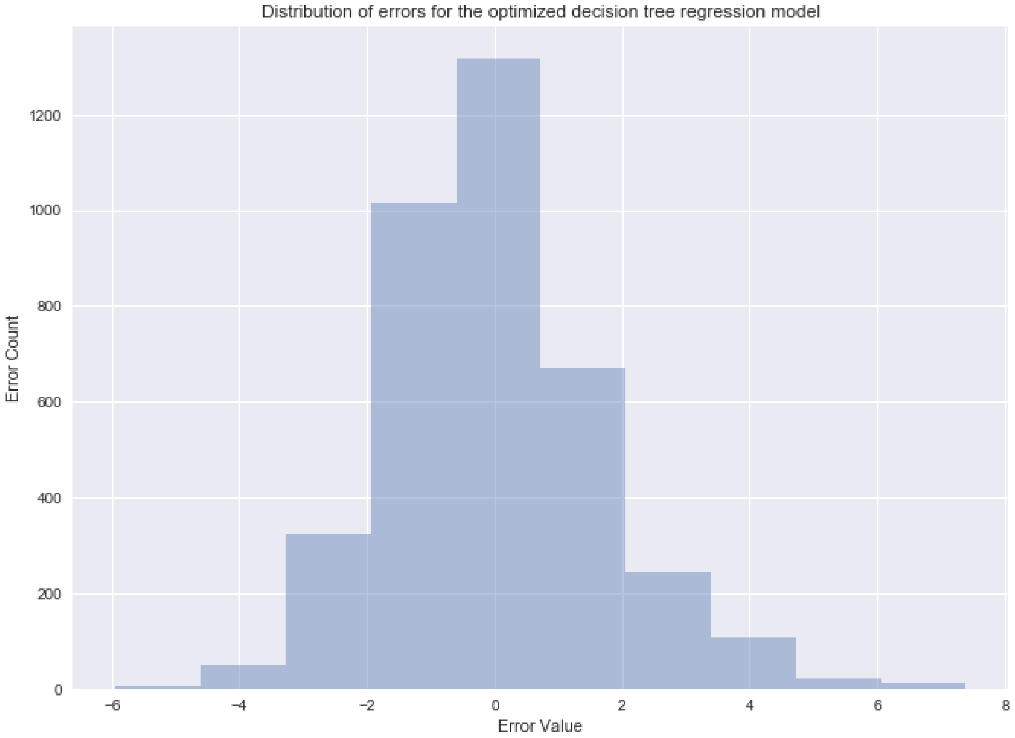


Figure 26: Distribution of Errors in Optimized Decision Tree

5 Ridge and Lasso Regression

The feature correlation analysis using heatmap as well as the one carried out using Pandas internal correlation factor calculation(Figure 18) was considered insufficient to identify the feature contributing towards building age-prediction model best. Therefore, it was decided to use regularization methods, Ridge and Lasso Regression, in order to help improve the prediction model.

Regularization is defined as introducing additional term to the loss function, the prediction model, in order to prevent overfitting of same. The 2 methods, Lasso and Ridge discussed in this section are also are L1 and L2 regularization respectively. L1 and L2 techniques both focus on coefficient shrinking, however, Table 8 outlines some key differences between the 2 methods starting with the metrics measured by each.

L1	L2
Sum of weight	Sum of square of weights
Sparse outputs	Non-sparse output
Built-in feature selection	No feature selection
High sparsity for highly correlated features	Even coefficient distribution for highly correlated features
Interpretability of models with large feature sets	Main use case is preventing overfitting

Table 8: Summary of L1 vs L2 Regularization Techniques

From the Table 8 it is yet unclear what are the advantages or disadvantages of one model over the other, however, the context of the dataset under analysis as well as undertaken objective are the key. For models with a significant number of features, especially prominent in the healthcare industry, it may be difficult to identify significant features for the prediction model, therefore L2 regularization methods would be at a disadvantage. On the other hand, if the objective was to prevent overfitting of the model, where features have been identified as of equal importance then the L2 method will be advantageous.

Both methods do perform well despite the presence of correlated features, L1 by simply picking the most significant feature and zeroing the coefficient of the related features, thus excluding them from the prediction model, while L2 ensures even distribution of the coefficients of the correlated features.

5.1 Lasso Regression

5.1.1 Model definition

Lasso stands for Least Absolute Shrinkage and Selection Operator and it is a L1 regularization technique, meaning that a factor of sum of absolute value of coefficients[1] is added in the optimization objective. It is commonly used to computationally identify the significant features as the method applies weights to all the features and as part of its operation penalizes the features that are the least contributing toward target variable $\hat{y_i}$. It is defined as RSS + α ∗ (sum of absolute value of weights) or:

$$\sum_{i=1}^{n}(\hat{y_i} - y_i)^2 + \alpha \sum_{j=0}^{n} |w_j| \tag{10}$$

where α refers to the factor of the penalty applied to a feature and w_i refers to the weight of the feature. It can be expected that the values of α will have the following impact:

1. $\alpha = 0$, is the same as linear regression,
2. $\alpha = \infty$, the coefficients will be zero due to the infinite weighting on square coefficients anything less than 0 will make the objective infinite
3. $0 < \alpha < \infty$, the coefficients will be found between 0 and the ones obtained from simple linear regression

5.1.2 Implementing the model

The first step is to obtain the best value of the α factor in order to fit the model best. In order to do that a number of alpha parameters have been generated programmatically. The numbers generated were evenly spaced over an interval between 10^{-6} and 10^{10} with such robust interval being chosen to cover a large number of scenarios. Following on the alphas were fitted to the model iteratively with Figure 27 illustrating the result of the parameter fitting. From the graph, it can be determined that such value of α exists for which weight applied is not zero and coefficients are not 0. In order to identify it, a Lasso cross-validation has been performed with 10 K-folds and a maximum number of iterations set to 10^6 or 1 million. Best value of α has been identified as $2.9231135368408643 * 10^{-5}$ or approximately 0.0000292.

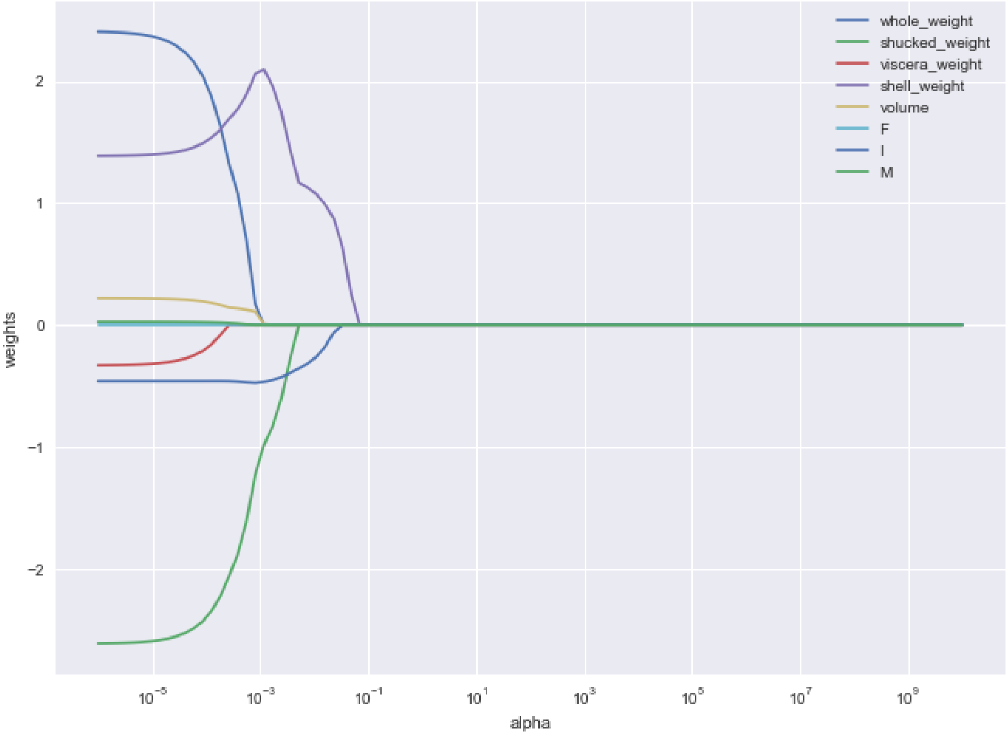


Figure 27: Plot of Changes of Weights for Given Alpha Factor

Subsequently the Lasso model was refitted with the calculated α and the Lasso score of approximately 0.1074069 has been obtained and Figure 28 illustrates that many of the coefficients are zero even for very small value of alpha specified while Table 9 states MSE and MAE metrics obtained from Lasso regression with starting α of 0.1 and best-fitted model with α set to 0.0000292. From there significant improvement of MAE and MSE metrics can be observed of the best-fitted model over one where no parameter fitting has been performed. However, the score metric shows increase with the optimized model. This is, however, expected as according to the documentation of the calculation implemented the best score metric that can be obtained is 1, therefore its increase in the fitted model is a sign of improvement.

	MAE	MSE	Lasso Score
Alpha 0.1	1.83627	4.65526	0.07938
Best alpha	1.28960	2.78739	0.44877
Difference (0.1 - best alpha)	0.54667	1.86787	-0.36939

Table 9: Lasso Metrics per Alpha Factor

By definition lasso regression performs shrinking of the coefficients as well as feature selection and any feature with a score of 0 is effectively excluded from the model. In the context of the discussed dataset, it is equivalent to lasso regression operating only whole weight feature to predict the number of rings and thus the age of an abalone specimen.

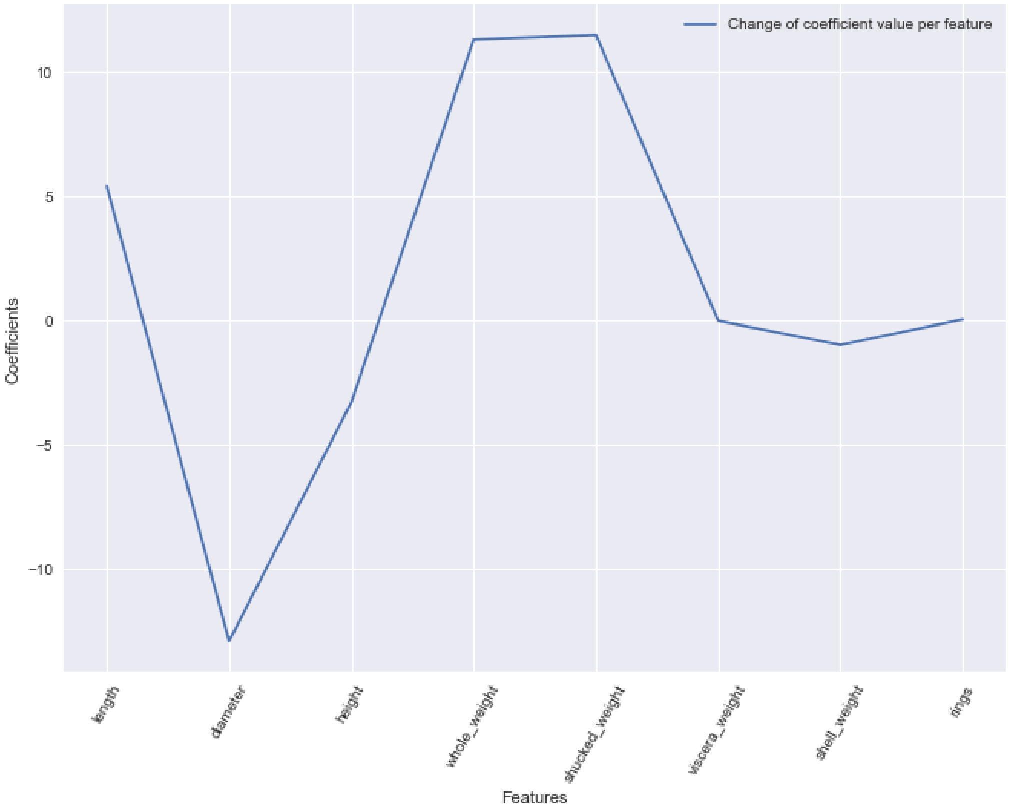


Figure 28: Lasso Coefficients Plot

5.2 Ridge Regression

5.2.1 Model definition

Ridge regression is a L2 regularization method by adding a penalty factor to square of the magnitude of coefficients[1]. It can be represented as:

$$\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \alpha \sum_{j=0}^{n} w_j^2 \tag{11}$$

From the same definition it can be deducted that impact of α will be the same as in 5.1 namely:

- $\alpha = 0$, same as linear regression
- $\alpha = \infty$, all coefficients = 0
- $0 < \alpha < \infty$, coefficients are between 0 and the ones obtained from simple linear regression

5.2.2 Implementing the model

The baseline metrics for improving the model, MSE and MAE, have been obtained from performing least square technique (RSS), which by its definition:

$$RSS = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{12}$$

is the same as calculating ridge regression scores without the α, which is precisely how the calculation has been defined as in Jupyter Notebook. Subsequently, as with the Lasso regression analysis in section 5.1 a number of α parameters have been generated programatically over the same interval of 10^{-6} to 10^{10}. Again the interval was chosen for it's robustness, ensuring that all scenarios, small and large are taken into the account with α parameters subsequently being fitted to the model iteratively. Figure 29 illustrates the change of weights along the changes of α parameter.

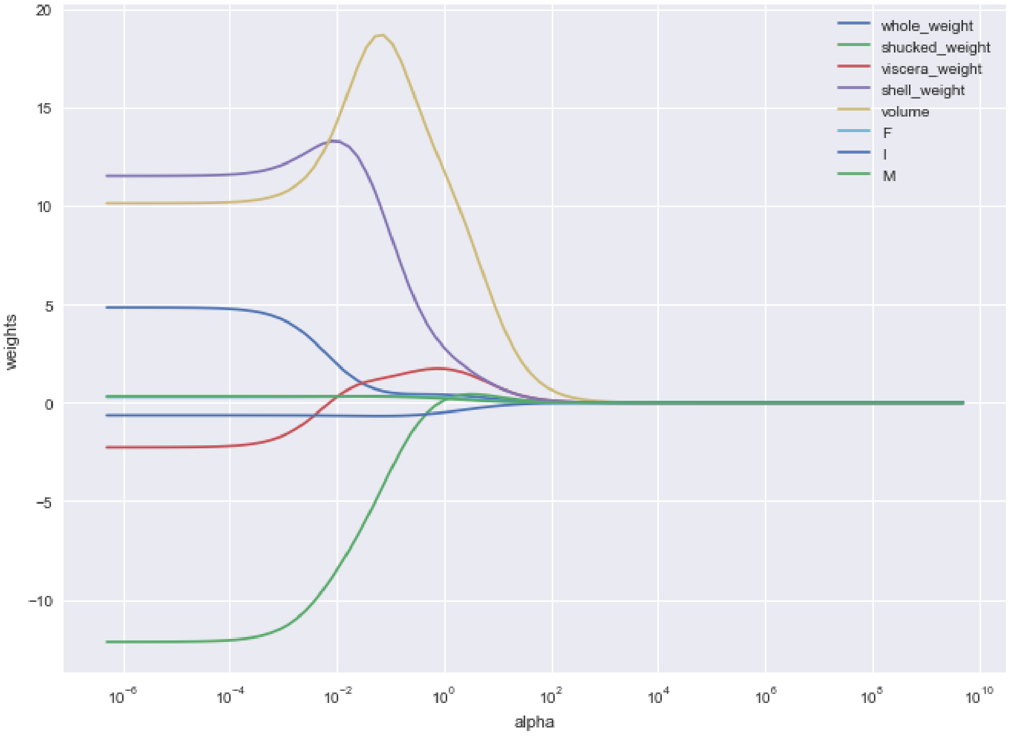


Figure 29: Plot of Changes of Weights for Given Alpha Factor

Furthermore, as the graph is not sufficient to deduct the appropriate α value ridge cross-validation method was employed instead. Ridge cross-validation calculation resulted in the α value of approximately 0.0002795 which was then used to fit the regression model and obtain MSE, MAE and score metrics for comparison with those obtained from RSS baseline. Observed values as per Table 10 show improvement of all metrics in Ridge regression over those obtained via least square technique, again bearing in mind that the increased value of score metric in the fitted model, same as in section 5.1 is a sign of improvement, with the best score possible being equal to 1.

	MAE	MSE	Score
RSS	1.34010	2.98284	0.46530
Ridge	1.33994	2.98190	0.46707
Difference (RSS-Ridge)	0.00016	0.00094	-0.00177

Table 10: Comparison of metrics obtained

5.3 Summary

To summarize the finding from sections 5.1 and 5.2 the MAE, MSE and score metrics obtained from the tuned models, using the best identified parameters have been put together in Table 11. Additionally, the difference between metrics has been calculated to determine which model in itself has obtained better parameters. From the summary table it is clear that Lasso regression succeeded in obtaining better MAE and MSE scores, however, it is using Ridge method that better RSS score is obtained.

	MAE	MSE	Score
Ridge	1.33994	2.98190	0.46707
Lasso	1.28960	2.78739	0.44877
Difference (Ridge - Lasso)	0.05034	0.19451	0.0183

Table 11: Comparison of Best Metrics from Ridge and Lasso Regression

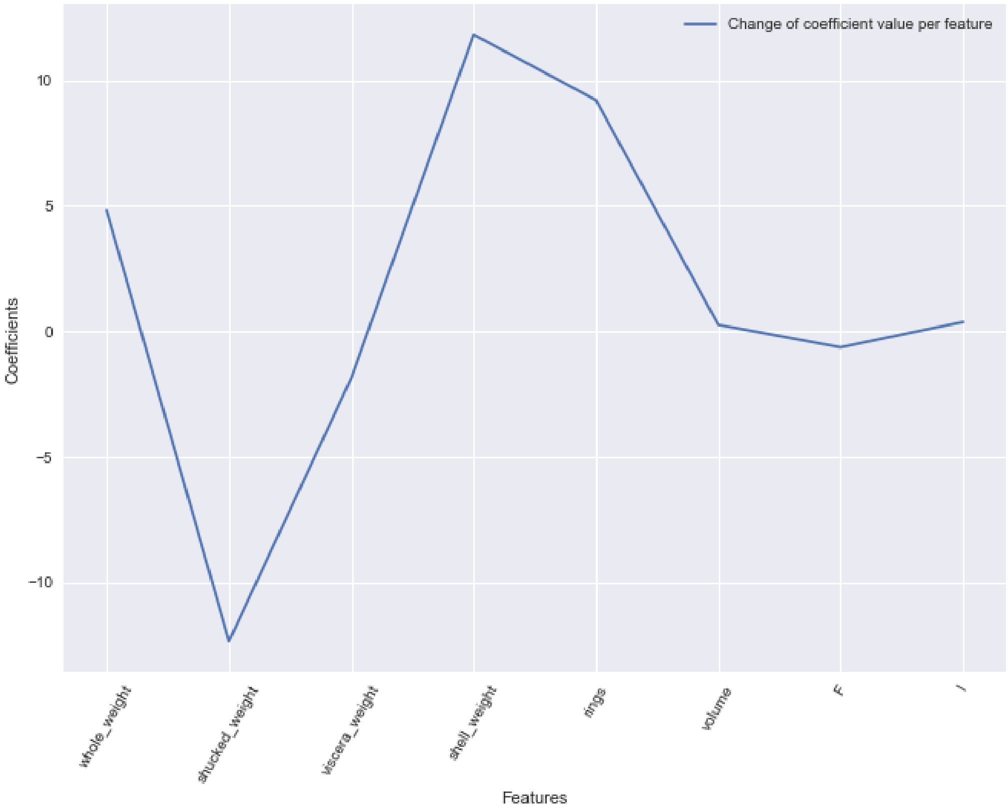


Figure 30: Ridge Coefficients Plot

6 Conclusions

In conclusion, it has been found that the non-regularized linear regression approach was not sufficient to accurately predict the number of rings of an abalone specimen, which are necessary to predict its age. However, implementing regularized approaches, lasso and ridge improved upon that result with the accuracy of the model with both model iterations scoring almost as high as that of the decision tree regression. Ultimately the optimized decision tree model yielded the best prediction accuracy.

7 Appendix

List of Tables

List of Figures

References

[1] T. Hastie, R. Tibsharani, and J. Friedman, "Springer Series in Statistics The Elements of," *The Mathematical Intelligencer*, vol. 27, no. 2, p. 83–85, 2009.

[2] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature," *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.

[3] D. Bertsimas, J. Dunn, and A. Paschalidis, "Regression and Classification using Optimal Decision Trees," pp. 1–4, 2017.