Machine Learning Project Report:

Abalone Age Prediction

Hao Heng, Shuang Ma, Ziyu Huang

JUNE 27, 2020

THE GEORGE WASHINGTON UNIVERSITY
2121 I St NW, Washington, DC 20052

Table of Contents

1. Introduction

Nowadays, abalone market is extending in the past decade years in China and even get wider recently. Abalone is a shellfish considered as a delicacy not only in China but also in many parts of the world. Based on information from Wikipedia, abalone is an excellent source of pantothenic acid and iron, and a nutritious food resource, and 100 grams of abalone can offer more than twenty percent recommended daily intake of these nutrients. The abalone farms are usually in America, East Asia and Australia. The economic value of abalone is positively correlated with the age of it. Hence, the goal to detect the age of abalone is important for farmers and customers to evaluate the price. However, farmers usually determined abalone age by cutting the shell through the cone, staining it, then counting the number of rings through a microscope, and these processes are boring and time consuming. This method is very troublesome and time-consuming. However, some physical quantities are easy to measure such as gender, length, diameter, whole weight etc. We will use these simple physical quantities to predict the age of abalone using machine learning methods.

We have applied regression algorithm and classification algorism. We used Linear Regression, Random Forest for regression, and we used Multiple Layer Perceptron, Logistic Regression for classification.

2. Data Set Description

The data set that we have has totally number of 4177 instances, 8 attributes with no mission values. The characteristic for this data set is multivariate, and the characteristics for attribute are categorical, integer and real.

Name	Data Type	Measurement	Description
Sex	Categorical		M, F, I (Infant)
Length	Continuous	mm	Longest Shell Measurement
Diameter	Continuous	mm	Perpendicular to Length
Height	Continuous	mm	With meat in shell
Whole Weight	Continuous	grams	Whole abalone
Shucked Weight	Continuous	grams	Weight of meat
Viscera Weight	Continuous	grams	Gut weight (after bleeding)
Shell Weight	Continuous	grams	After being dried
Rings	Continuous		+ 1.5 gives the age in years

This data set is for predicting the age of abalone from physical measurements. Based on the information from UCI that the age of abalone is determined by cutting the shell through the cone

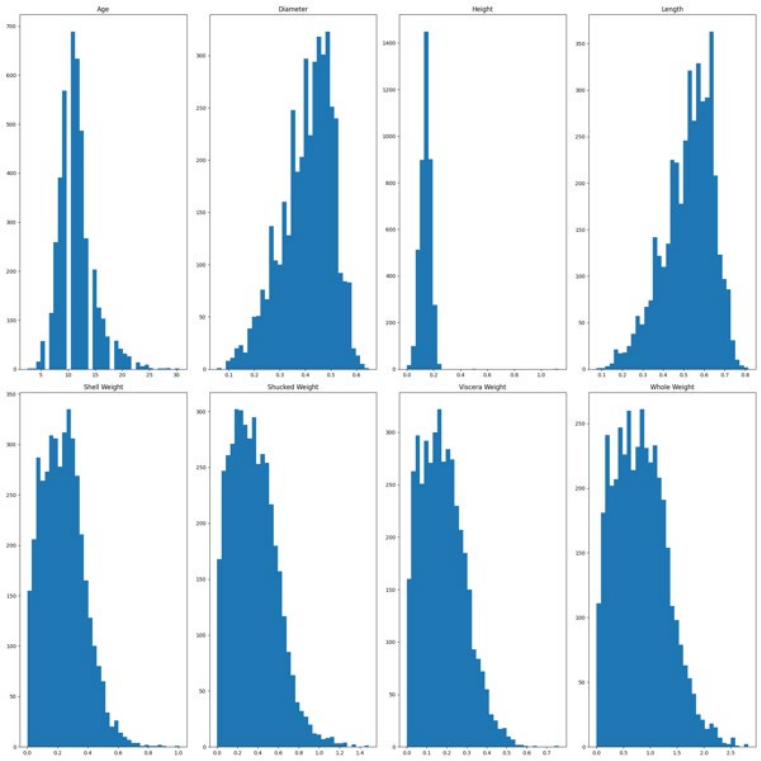
and counting the number of rings through a microscope. This task is in some way boring and time-consuming.

	Length	Diameter	Height	Whole Weight	Shucked Weight
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000
mean	0.523992	0.407881	0.139516	0.828742	0.359367
std	0.120093	0.099240	0.041827	0.490389	0.221963
min	0.075000	0.055000	0.000000	0.002000	0.001000
25%	0.450000	0.350000	0.115000	0.441500	0.186000
50%	0.545000	0.425000	0.140000	0.799500	0.336000
75%	0.615000	0.480000	0.165000	1.153000	0.502000
max	0.815000	0.650000	1.130000	2.825500	1.488000

	Viscera Weight	Shell Weight	Age
count	4177.000000	4177.000000	4177.000000
mean	0.180594	0.238831	11.433684
std	0.109614	0.139203	3.224169
min	0.000500	0.001500	2.500000
25%	0.093500	0.130000	9.500000
50%	0.171000	0.234000	10.500000
75%	0.253000	0.329000	12.500000
max	0.760000	1.005000	30.500000

According to the described information, we can get some basic information for all numerical features such as mean, standard deviation, minimum, maximum and median. No numerical features have minimum value 0 except for Height. Based on the median and mean for each numerical feature, all of them could be normally distributed but we also need to check the distribution by looking at the histogram for each feature.

So, with the following diagrams, we can conclude that all features are not normally distributed but close to normality



For our dataset, there are no missing values as you can see from the bottom left list. On the bottom right, it gave us the data type for each feature.

```
(4177, 9)
Sex               0
Length            0
Diameter          0
Height            0
Whole Weight      0
Shucked Weight    0
Viscera Weight    0
Shell Weight      0
Age               0
dtype: int64
```

```
Sex               object
Length           float64
Diameter         float64
Height           float64
Whole Weight     float64
Shucked Weight   float64
Viscera Weight   float64
Shell Weight     float64
Age              float64
dtype: object
```

3. Models

3.1: Linear Regression

Given a data set $\{y_i,\ x_{i1}, \ldots,\ x_{ip}\}_{i=1}^{n}$ of n statistical units. A linear regression model assumes that the relationship between the dependent variable y and the p-vector of regressor x is linear. The error variable ε is an unobserved random variable that add "noise" to the linear relationship between y and xi. The model is shown as below:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = x_i^T \beta + \varepsilon_i, \text{ i = 1, …, n,}$$

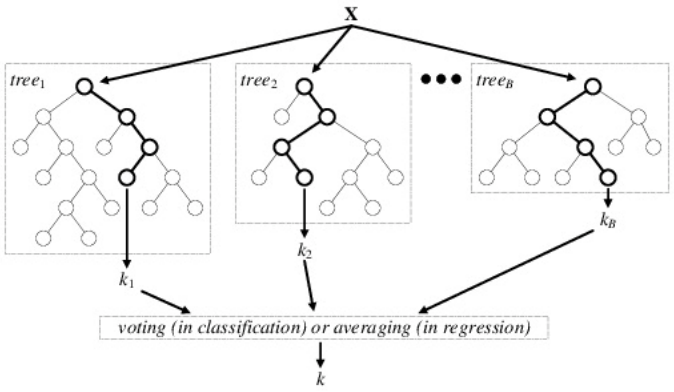
where T denotes the transpose, so that $x_i^T \beta$ is the inner product between vectors x_i and β. It can be also written in matrix notation as $y = X\beta + \varepsilon$, where

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_1 \end{pmatrix} \qquad X = \begin{pmatrix} x_1^{\mathrm{T}} \\ x_2^{\mathrm{T}} \\ \vdots \\ x_n^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}$$

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \qquad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

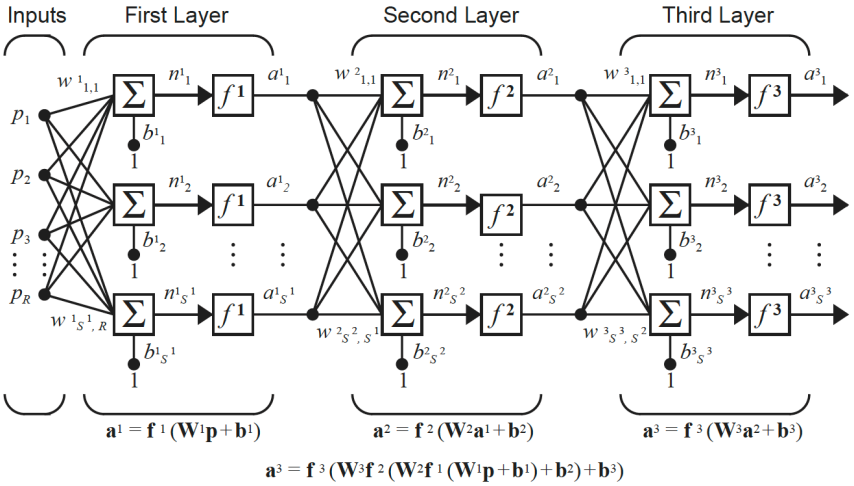
3.2: Random Forest

Random Forests randomly chooses observations and features to generate many decision trees to average the result. While a Decision tree generates rules based on the features in the data. The main issue with Random Forests is that they are slow to gather real-time predictions on but are extremely fast to train. However, RF prevents overfitting by generating trees on random subsets. We can see Random Forest as an aggregate of bunch of Decision Trees.



3.3: Multiple Layer Perceptron

The core algorithms I use are multiple layer perceptron and logistic regression. Multiple layer perceptron is connected by several single layer perceptron. Each perceptron has its own input, weight, bias and transfer function. The output of each layer becomes the input of next layer, until the network gets final result. Each layer would calculate this equation:



W is the weight, p is the input of each layer, b is the bias, f is the transfer function, and a is the output. The whole calculation order is forward propagation.

3.4: Logistic Regression

Logistic regression is another classifier that can determine a certain class such as pass/fail, is/isn't. In other words, logistic regression can only classify binary class. Logistic regression is a very fast model designed for classifying binary target. It's highly interpretable, easy to regularize, and it doesn't need to scale the feature.

- $P(y = 1|x; \theta) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$

Figure3: Logistic Regression

This equation means figuring out the possibility of y=1 given x and θ.

4. Experimental Setup

The basic package of our implementation involves sklearn, pandas, numpy, matplotlib and seaborn. First, use pd.read_csv to import data file. It can be accessed from local or through URL. Then use the function from pandas to lean the dataset about its feature columns, its target columns. During this step, we use pd.drop to drop the possible outlier in each feature column. When applying the multiple layer perceptron, first label encodes the sex column which is the only category column into numeric column. For this step, we define a map and pass that map into the abalonedata['Sex'], because abalonedata is a dataframe type instance. The column in this instance can be accessed through this way. Second, using pd.iloc to extract all 8 feature columns and one last target column, passing this two data into x and y. Third, applying PCA to x which contain all feature information to see the variance ratio, then we observe there are four columns have a much bigger variance than the rest, meanwhile, with only for components we can have

98% variance information, so we use PCA to reduce the dimension of feature from 8 into 4 for classification model. Fourth, using train_test_split function to split the x and y into training set and test set. Finally, we use the dataset on our 4 models. Two for regression and another 2 for classification. We use 70% of the data as training data and the rest 30% data as test set. After train each model, we compared the prediction result with different index for different method of model. For regression model, we compare RMSE and R2 score. For classification model we compare not only RMSE but also the accuracy score, confusion matrix and indexes in classification report.

For classification model, during first trial, we apply these two models, trying to predict the accurate value of the rings of abalone. But the performance turns out to be very bad due to too many different classes in target column. So I change the strategy, grouping the target into binary class. The instances that has a rings bigger than 10 go to group 1 with label 1. The rest goes to group 2 with label 0. The reason I choose 10 as the border is because during learning the distribution of this dataset, I notice the rings of most of instances gather around 10. But only two classes is too general. So I regroup the data. The number of group is the label of that group. This time, I equally divide the dataset into 5 group according to the value of rings.

Group 0: Rings of instance <5

Group 1: 5<=Rings of instance<10

Group 2: 10<=Rings of instance<15

Group 3: 15<=Rings of instance<20

Group 4: 20<=Rings of instance<=29

(There is no instance with rings ‘28’)

5. Evaluation Index

5.1: Regression Model

Root Mean Square Error (RMSE):

Root Mean Square Error is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are, and RMSE is a measure of how spread out these residuals are. Normally speaking, the smaller RMSE, the better the model.

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

R-squared:

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. The closer R-squared to 1, the better performance the model has.

$$R - squared = \frac{Explained\ variation}{Total\ variation}$$

5.2: Classification Model

Confusion matrix:

Confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). In binary classification, confusion matrix always looks like this:

		Predicted class	
		Positive	Negative
Actual class	Positive	True positive	False negative
	Negative	False positive	True negative

The dimension of confusion matrix is decided by the number of class the dataset has. Each row and corresponding column means one class. As the figure showed above, if the prediction mainly falls in the main diagonal, the performance of that model is good.

Classification report:

This report shows precision, recall, f1-score and support. It also shows the accuracy of model, macro average (averaging the unweighted mean per label), weighted average (averaging the support-weighted mean per label), and sample average (only for multilabel classification).

6. Results

6.1: Linear Regression

```
print result of lr:
score:  0.3019867549668874
mse:  2.6494631577077326
rms:  1.6277171614588735
r^2:  0.5483131246344208
```

The result after we standardize the abalone data and using linear regression method to predict the age is shown as above. The score is 0.302, RMSE is 1.628, and R-squared is 0.548.

6.2: Random Forest

Tree numbers for our model which is 1380 with RMSE 2.1049581 and R2 score 0.535383635270687

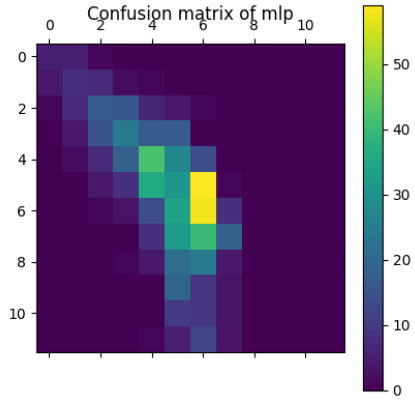
6.3: Multi-Layer Perceptron

6.3.1: On original tarter column

The confusion matrix:

```
print result of mlp:
[11  7 12 10  5  9  8  6 13 14 15  4] target
[[ 5  5  1  0  0  0  0  0  0  0  0  0]
 [ 4  8  7  2  1  0  0  0  0  0  0  0]
 [ 1  7 17 16  6  4  1  0  0  0  0  0]
 [ 0  4 15 24 17 17  0  0  0  0  0  0]
 [ 0  2  7 18 42 27 14  0  0  0  0  0]
 [ 0  0  4  8 36 31 59  1  0  0  0  0]
 [ 0  0  1  3 14 34 58  8  0  0  0  0]
 [ 0  0  0  0  8 32 40 18  0  0  0  0]
 [ 0  0  0  1  4 21 24  4  0  0  0  0]
 [ 0  0  0  0  0 19  9  3  0  0  0  0]
 [ 0  0  0  0  0 10  9  3  0  0  0  0]
 [ 0  0  0  0  1  5 12  3  0  0  0  0]]
```

The plot of confusion matrix:



The classification report:

	precision	recall	f1-score	support
4	0.50	0.45	0.48	11
5	0.31	0.36	0.33	22
6	0.33	0.33	0.33	52
7	0.33	0.31	0.32	77
8	0.33	0.38	0.35	110
9	0.15	0.22	0.18	139
10	0.26	0.49	0.34	118
11	0.45	0.18	0.26	98
12	0.00	0.00	0.00	54
13	0.00	0.00	0.00	31
14	0.00	0.00	0.00	22
15	0.00	0.00	0.00	21
accuracy			0.27	755
macro avg	0.22	0.23	0.22	755
weighted avg	0.25	0.27	0.24	755

The score and error:

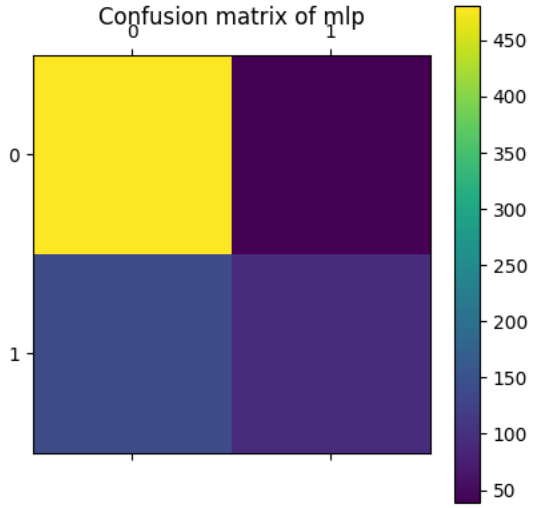
```
accuracy score:  0.26887417218543047
mse:  3.6079470198675496
rms:  1.8994596652383935
r^2:  0.35332027786267606
```

6.3.2: On binary target column

The confusion matrix:

```
print result of mlp:
[0 1] target
[[481  39]
 [140  95]]
```

The plot of confusion matrix:



The classification report:

```
 [140  95]]
              precision    recall  f1-score   support

           0       0.77      0.93      0.84       520
           1       0.71      0.40      0.51       235

    accuracy                           0.76       755
   macro avg       0.74      0.66      0.68       755
weighted avg       0.75      0.76      0.74       755
```

The score and error:

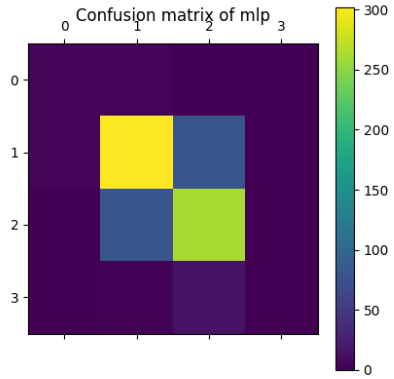
```
accuracy score:  0.7629139072847683
mse:  0.2370860927152318
rms:  0.48691487214422996
r^2:  -0.10593289689034369
cross validation:  0.7562807017543859
```

6.3.3: On 5 group target column:

The confusion matrix:

```
print result of mlp:
[1 2 0 3] target
[[  4   5   0   0]
 [  4 302  79   0]
 [  0  81 263   0]
 [  0   2  15   0]]
```

The plot of confusion matrix:



The classification report:

	precision	recall	f1-score	support
0	0.50	0.44	0.47	9
1	0.77	0.78	0.78	385
2	0.74	0.76	0.75	344
3	0.00	0.00	0.00	17
accuracy			0.75	755
macro avg	0.50	0.50	0.50	755
weighted avg	0.74	0.75	0.74	755

The score and error:

```
accuracy score:  0.7536423841059603
mse:  0.2543046357615894
rms:  0.5042862637050403
r^2:  0.20217508558345354

cross validation:  0.7550526315789473
```

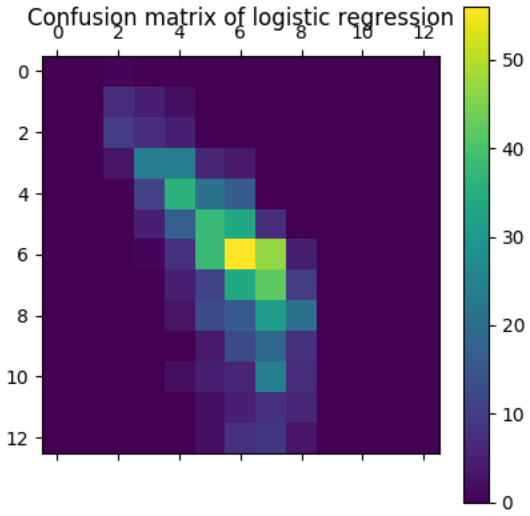
6.4: Logistic Regression

6.4.1: On original target column:

The confusion matrix:

```
show the result of logistic regression:
[[ 0  0  1  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  7  5  2  0  0  0  0  0  0  0  0]
 [ 0  0 10  7  5  0  0  0  0  0  0  0  0]
 [ 0  0  3 24 24  6  4  0  0  0  0  0  0]
 [ 0  0  0 11 36 21 16  0  0  0  0  0  0]
 [ 0  0  0  5 17 38 34  7  0  0  0  0  0]
 [ 0  0  0  1  8 38 56 47  5  0  0  0  0]
 [ 0  0  0  0  5 11 34 42 10  0  0  0  0]
 [ 0  0  0  0  3 13 16 31 21  0  0  0  0]
 [ 0  0  0  0  0  4 13 19  8  0  0  0  0]
 [ 0  0  0  0  2  5  6 24  7  0  0  0  0]
 [ 0  0  0  0  0  2  5  8  6  0  0  0  0]
 [ 0  0  0  0  0  2  8  9  3  0  0  0  0]]
```

The plot of confusion matrix:



The classification report:

	precision	recall	f1-score	support
0	0.77	0.93	0.84	520
1	0.71	0.40	0.51	235
accuracy			0.76	755
macro avg	0.74	0.66	0.68	755
weighted avg	0.75	0.76	0.74	755

The score and error:

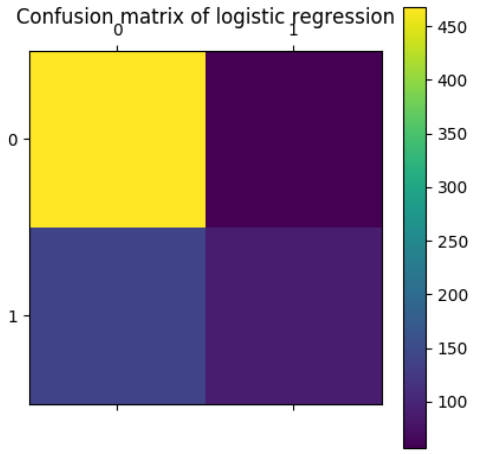
```
mse:  3.6251655629139075
rms:  1.903986754920818
r^2:  0.39534501969835656
accuracy_score:  0.30066225165562915

cross validation:  0.7442280701754387
```

6.4.2: On binary target column:

The confusion matrix:

```
show the result of logistic regression:
[[468  57]
 [140  90]]
```

The plot of confusion matrix:



The classification report:

	precision	recall	f1-score	support
4	0.50	0.45	0.48	11
5	0.31	0.36	0.33	22
6	0.33	0.33	0.33	52
7	0.33	0.31	0.32	77
8	0.33	0.38	0.35	110
9	0.15	0.22	0.18	139
10	0.26	0.49	0.34	118
11	0.45	0.18	0.26	98
12	0.00	0.00	0.00	54
13	0.00	0.00	0.00	31
14	0.00	0.00	0.00	22
15	0.00	0.00	0.00	21
accuracy			0.27	755
macro avg	0.22	0.23	0.22	755
weighted avg	0.25	0.27	0.24	755

The score and error:

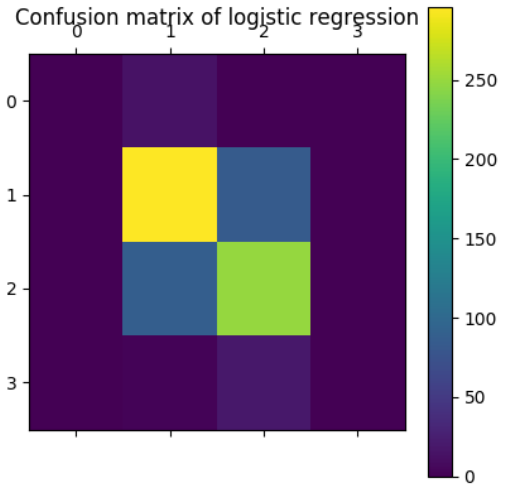
```
mse:  0.2609271523178808
rms:  0.5108102899490973
r^2:  -0.23175983436852965
accuracy_score:  0.7390728476821192
```

6.4.3: On 5 group target column:

The confusion matrix:

```
show the result of logistic regression:
[[  0  14   0   0]
 [  0 296  85   0]
 [  0  89 249   0]
 [  0   3  19   0]]
```

The plot of confusion matrix:



The classification report:

	precision	recall	f1-score	support
0	0.50	0.44	0.47	9
1	0.77	0.78	0.78	385
2	0.74	0.76	0.75	344
3	0.00	0.00	0.00	17
accuracy			0.75	755
macro avg	0.50	0.50	0.50	755
weighted avg	0.74	0.75	0.74	755

The score and error:

```
mse:  0.2900662251655629
rms:  0.538577965726006
r^2:  0.15972984510306132
accuracy_score:  0.7218543046357616
cross validation:  0.7417192982456141
```

7. Conclusion

Regression Model:

In our report, regression algorithms and classification algorithms are used to predict the target. For the regression algorithm, we use linear regression and random forest. Root of Mean Squared Error (RMSE) and R-Squared are used to compare the model. As for the classification algorithm, multiple layer perceptron and logistic regression, accuracy score is used to compare the models. The detailed information is shown as follows

Regression algorithms		
Model	RMSE	R-Squared
Linear regression	1.881	0.392
Random Forest	2.104	0.535

From the table above, R-Squared of random forest is bigger than that of linear regression, it turns out that the effect of random forest is better. However, RMSE of random forest is bigger than that of linear regression. Since we expect that the better model has smaller RMSE, another model is needed to predict the target.

Classification Model:

When applying the model to the original target column, the confusion matrix of two model both shows there are certain class that model would predict incorrectly. The accuracy from

classification report is also very low. This is probably because of the number of class in target column is too big. On the contrast, in other regrouped target column, the performance improve greatly. They both reach a high accuracy. But the confusion matrix still suggests that the model is not good at predicting some specific classes.

Classification algorithms			
Model	RMSE	R-Squared	Accuracy score
Multiple layer perceptron	0.55	-0.39	0.70
Logistic regression	0.46	-0.004	0.72

As we can see, accuracy score of multiple layer perceptron is 0.70, and accuracy score of logistic regression is 0.72, so logistic regression has the better performance than another. However, it is not significantly superior to the other one, so further analysis is needed to be done in the future.

8. Limitation and Improvement

The performance of our model is not as good as what we have expected, probably due to the imbalanced data set. The solution to this problem is reducing the number of class into binary

class. The model can predict general range of age of abalone. Other solutions involve increasing more instance and features by provide some feature engineering.

References

[1] Wikipedia: Random forest. (2020, June 22). Retrieved June 30, 2020, from https://en.wikipedia.org/wiki/Random_forest

[2] Patel, Savan. "Chapter 3 : Decision Tree Classifier - Theory." *Medium*, Machine Learning 101, 14 May 2017, medium.com/machine-learning-101/chapter-3-decision-trees-theory-e7398adac567.

[3] "3.2.4.3.2. Sklearn.Ensemble. RandomForestRegressor — Scikit-Learn 0.20.3 Documentation." *Scikit-Learn.Org*, 2018, scikitlearn.org /stable /modules/generated /sklearn.ensemble.RandomForestRegressor.html.

[4] "Abalone Dataset." *Kaggle*, 19 July 2018, www.kaggle.com/rodolfomendes/abalone-dataset.

[5] Ragnisah. "EDA- Abalone Age Prediction." Kaggle, 28 Aug. 2019, www.kaggle.com/ragnisah/eda-abalone-age-prediction.

[6] Anthonypino. "Price Analysis and Linear Regression." *Kaggle*, 16 Aug. 2017, www.kaggle.com/anthonypino/price-analysis-and-linear-regression.

[7] Ragnisah. "EDA- Abalone Age Prediction." Kaggle, 28 Aug. 2019, www.kaggle.com/ragnisah/eda-abalone-age-prediction.

[6] Anthonypino. "Price Analysis and Linear Regression." *Kaggle*, 16 Aug. 2017, www.kaggle.com/anthonypino/price-analysis-and-linear-regression.