PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering Degree in Computer Science and Engineering

By

MANGADUDDI KISHORE BALAJI
REG . NO : 40110722

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

OCT 2022

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **MANGADUDDI KISHORE BALAJI (40110722)** who carried out the project entitled "**ABALONE DATASET USING MACHINE LEARNING**" under my supervision from Aug 2022 to Oct 2022.

Internal Guide

Name: Ms.S.R.Srividhya

Head of the Department

Dr.L.Lakshmanan

Submitted for Viva voce Examination held on _____

InternalExaminer **ExternalExaminer**

DECLARATION

I **MANGADUDDI KISHORE BALAJI** hereby declare that the Project Report entitled **ABALONE DATASET USING MACHINE LEARNING** done by me under the guidance of Ms.**S.R.Srividhya**) at cognibot is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D.**, **Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms.S.R.Srividhya** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

ABSTRACT

Abalone is a marine snail found in the cold coastal regions. Age is a vital characteristic that is used to determine its worth. Currently, the only viable solution to determine the age of abalone is through very detailed steps in a laboratory. This paper exploits various machine learning models for determining its age. A comprehensive analysis of various machine learning algorithms for abalone age prediction is performed which include, backpropagation feed-forward machine learning(BPFFNN), K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, Random Forest, Gauss Naive Bayes, and Support Vector Machine (SVM). In addition, five different optimizers were also tested with BPFFNN to evaluate their effect on its performance. Comprehensive experiments were performed using our data set

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

ABBREVATION	EXPANSION
CSV	**Comma Separated Values**
MAE	**MEAN ABSOLUTE ERROR**
MSE	**MEAN SQUARED ERROR**

CHAPTER-1

INTRODUCTION

1.1 ABALONE DATASET

Abalone is a shellfish considered a delicacy in many parts of the world. An excellent source of iron and pantothenic acid, and a nutritious food resource and farming in Australia, America and East Asia. 100 grams of abalone yields more than 20% recommended daily intake of these nutrients. The economic value of abalone is positively correlated with its age. Therefore, to detect the age of abalone accurately is important for both farmers and customers to determine its price. However, the current technology to decide the age is quite costly and inefficient. Farmers usually cut the shells and count the rings through microscopes to estimate the abalones age. Telling the age of abalone is therefore difficult mainly because their size depends not only on their age, but on the availability of food as well. Moreover, abalone sometimes form the so-called 'stunted' populations which have their growth characteristics very different from other abalone populations This complex method increases the cost and limits its popularity. Our goal in this report is to find out the best indicators to forecast the rings, then the age of abalones.

1.2 MACHINE LEARNING

One of its own, Arthur Samuel, is credited for coining the term, “machine learning” with his research (PDF, 481 KB) (link resides outside IBM) around the game of checkers. Robert Nealey, the self-proclaimed checkers master, played the game on an IBM 7094 computer in 1962, and he lost to the computer. Compared to what can be done today, this feat seems trivial, but it’s considered a major milestone in the field of artificial intelligence

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

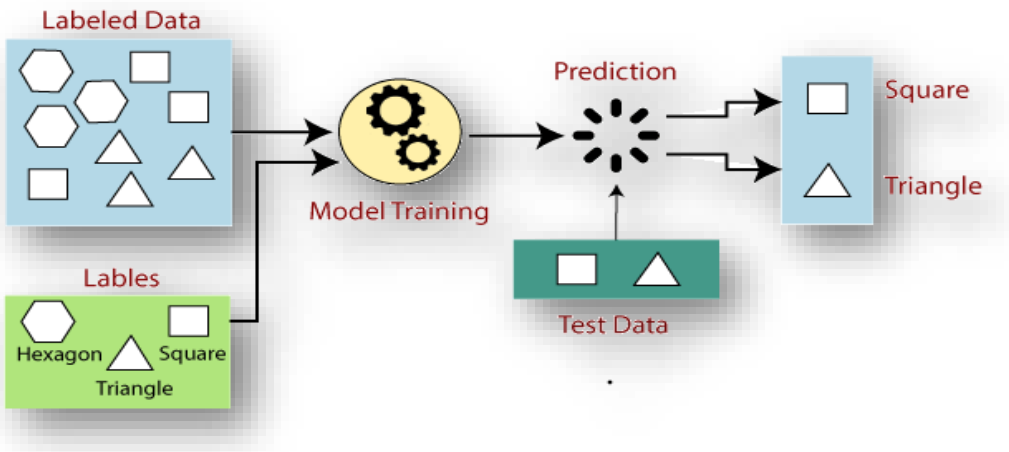
1.3 TYPES OF MACHINE LEARNING

Classical machine learning is of four basic approaches.There are…

- Supervised learning.
- Usupervised learning.

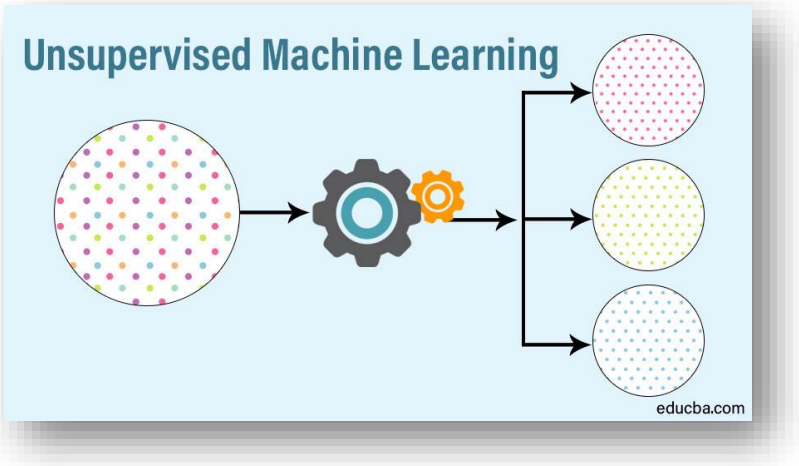
- Semi-supervised learning.
- Reinforcement learning.

SUPERVISED LEARNING: In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.



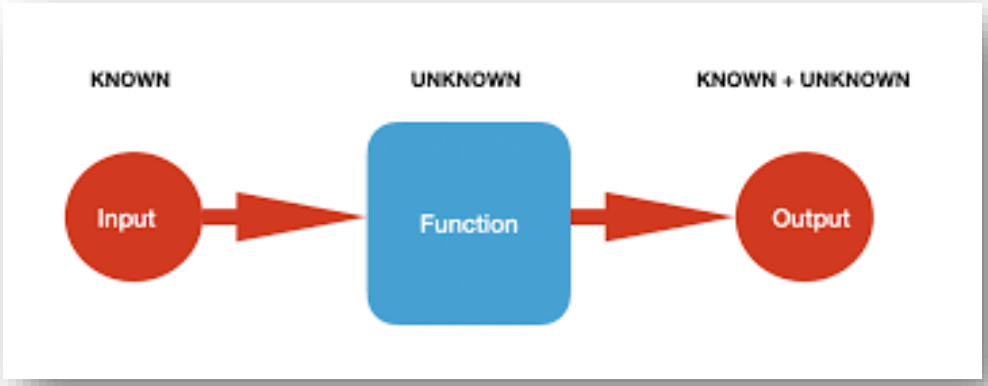
SUPERVISED LEARNING (fig.1.1)

UNSUPERVISED LEARNING: This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.



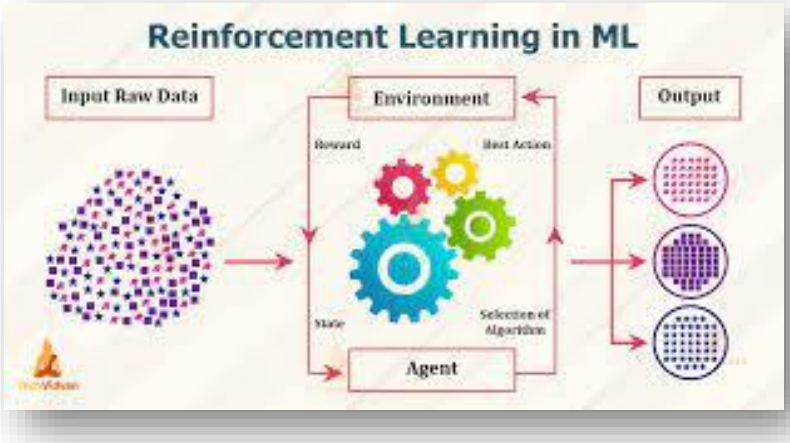
UNSUPERVISED LEARNING (fig.1.2)

SEMI SUPERVISED LEARNING: This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.



SEMI-SUPERVISED LEARNING fig.1.3

REINFORCEMENT LEARNING: Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. But for the most part, the algorithm decides on its own what steps to take along the way.



REINFORCEMENT LEARNING fig 1.4

CHAPTER-2

AIM AND SCOPE OF THE PRESENT INVESTIGATION

2.1 ABALONE DATASET:

Predicting the age of abalone from physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope -- a boring and time-consuming task**.**

2.2 AIM :

Predicting the age of abalone from physical measurements by machine learning

2.3 SCOPE :

Abalone is a variety of mollusks or sea snails with feet and tentacles. Its ear-shaped shells are made up of microscopic pieces of calcium carbonate, stacked one on top of the other, almost like tiny Lego blocks.

Abalone can form pearls, and these can be made into jewelry. They are exceptionally rare, so it is very lucky to find abalone pearls.They form in the mother-of-pearl shell, at the edge of the shell or in the gut. Only one in several hundred abalone mollusks will form pearls.

2.4 REQUIREMENTS

- Dataset
- Jupyter Notebook
- Programming skills Python
- Access to a laptop or computer with internet connectivity

2.5 OUT COMES

▪ The dataset was analysed using histograms, bar chart, correlation matrices, confusion matrix.

▪ Logistic Regression model was used to obtain the good accuracy rate

CHAPTER-3

EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED

3.1 PLATFORM USED

Jupyter is a free, open-source, interactive web tool known as a computational notebook, which researchers can use to combine software code, computational output, explanatory text and multimedia resources in a single document. The notebook extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results. The Jupyter notebook combines two components:

Fig 3.1

3.1.1 WEB APPLICATION:

It is a browser based tool whose main purpose is to author documents interactively by combining explanatory texts, mathematics, computations and representations of objects. Main features of the web application

- In-browser editing for code, with automatic syntax highlighting, indentation,nd tab completion/introspection.

- The ability to execute code from the browser, with the results of computations attached to the code which generated them.

- Displaying the result of computation using rich media representations, such as HTML, LaTeX, PNG, SVG, etc. For example, publication quality figures rendered by the matplotlib library, can be included inline

- In-browser editing for rich text using the Markdown markup language, which can provide commentary for the code, is not limited to plain text.

- The ability to easily include mathematical notation within markdown cells using LaTeX, and rendered natively by MathJax.

3.1.2 NOTEBOOK DOCUMENTS :

A representation of all content visible in the we application including the inputs and outputs of computations, explanatory texts, mathematics, and representations of objects.

- Notebooks documents contain inputs and outputs of interactive session as well as additional text that accompanies the code it is meant for execution.

- Notebook files can serves as a complete computational record of a session, interleaving executable code with explanatory text, mathematics and rich representations of resulting object.

- Notebooks may be exported to a range of static formats, including HTML, restructredText, LaTex, PDF, and slide shows, via the nbconvert command.

- .ipynb notebook documents available from a public URL can be shared via the Jupyter notebook viewer . This service loads the notebook document from the URL renders it as a static web page.

- The results may thus be shared as a public blog post, without other users needing to install the jupyter notebook themselves, in effect, nb viewer is simply nb convert as a web service, so you can do your own static conversions with nb convert without relying on nb viewer

3.2 LANGUAGE USED :

Python is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting-edge technology in Software Industry. Python Programming Language is very well 10 suited for Beginners, also for experienced programmers with other programming languages like C++ and Java. some facts about Python Programming Language:

Fig 3.2

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms.
- Python programs generally are smaller than other programming languages like Java. Programmers must type relatively less and indentation requirement of thelanguage, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like Google,Amazon, Facebook, Instagram, Dropbox, Uber… etc

3.3 DATASET FILE TYPE USED

CSV file

- A Comma Separated Values (CSV) file is a plain text file that contains a list of data.
- These files are often used for exchanging data between different applications. For example,databases and contact managers often support CSV files
- These files may sometimes be called Character Separated Values or Comma Delimited files.
- A CSV file has a fairly simple structure. It's a list of data separated by commas.
- A CSV file (comma separated values) is a special type of file that you can create or edit in Excel. Instead of storing information in columns, CSV files store data separated by commas.

3.4 LIBRARIES/MODULES

PANDAS :

Pandas is an open-source, BSD-licensed Python library providing high performance, easy -to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Pandas makes it simple to do many of the time consuming, repetitive tasks associated with working with data, including: Data cleansing, Data fill, Data normalization, Merges and joins, Data visualization, Statistical analysis, Data inspection, Loading and saving data, and much more.

SKLEARN :

For machine learning, Python and R are the most widely used programming languages today. Scikit-learn (sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the

Python numerical and scientific libraries NumPy and SciPy.

MATPLOTLIB:

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc. ▪ NumPy NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

SEABORN :

Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library. The graphs created can also be customized easily. seaborn provides a variety of visualization patterns. It uses fewer syntax and has easily interesting default themes.

3.5 ABOUT ABALONE SHELL

Abalone is a shellfish to be precise that lives in cold coastal waters around the world. Biologically, abalone is a mollusk belonging to the Gas tropoda class. In plain English, this means that abalone is technically a type of marine snail.

ABALONE SHELL fig(3.3)

How to determine the age of an abalone?

Scientific studies on abalones require knowing the age of an abalone, but the process of determining age is complicated. It involves measuring the number of layers of shell (“rings”) that make up the abalone’s shell. This is done by taking a sample of shell, staining it and counting the number of rings under the microscope.This process is tedious and time-consuming

How machine learning can solve this problem?

As we know, Age is a number and we have data that contains the physical measurements of abalones and their ages. We can build a machine learning model that can predict the age of abalone given its physical measurements like weight, height, etc.

About the data set

Data comes from an original (non-machine-learning) study : The Population Biology of Abalone (_Haliotis_ species) in Tasmania.
From the original data examples with missing values were removed, and the ranges of the continuous values have been scaled by dividing with 200.

3.6 Attribute Information:

Given is the attribute name, attribute type, the measurement unit and a brief description. The number of rings is the value to predict: either as a continuous value or as a classification problem.

Name / Data Type / Measurement Unit / Description

Sex / nominal / -- / M, F, and 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

driedRings / integer / -- / +1.5 gives the age in years

ALGORITHMS USED

3.7 ALGORITHMS IN MACHINE LERNING

In Machine Learning there are many classification algorithms to solve this problem. Examples of Classification algorithms are Logistic Regression, Knearest neighbours, decision tree, Random forest , etc.

The algorithms used to predict the age of abalone in machine learning

→Linear regression

→Decision tree

REGRESSION :

Regression is a technique for investigating the relationship between independent variables or features and a dependent variable or outcome It's used as a method for predictive modelling in machine learning, in which an algorithm is used to predict continuous outcomes

Solving regression problems is one of the most common applications for machine learning models, especially in supervised machine learning Algorithms are trained to understand the relationship between independent variables and an outcome or

dependent variable. The model can then be leveraged to predict the outcome of new and unseen input data, or to fill a gap in missing data

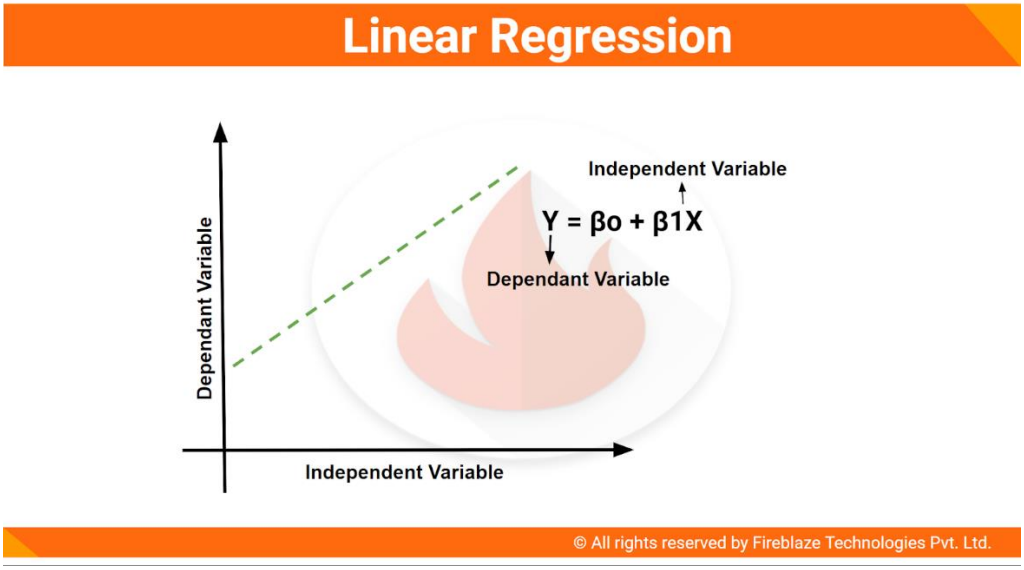
→Linear regression

→Decision tree

3.7.1 LINEAR REGRESSION :

Linear is a machine learning algorithm based on supervised learning. It performs **a** regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output).Hence,the name is Linear Regression.
In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

$$(a1x_i+a_0)= \text{Predicted value}$$



LINEAR REGRESSION fig 3.4

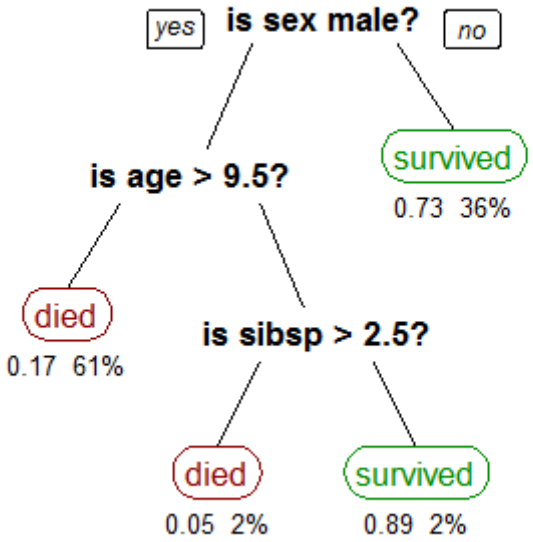
3.7.2 DECISION TREE

Decision Tree A decision tree is a simple supervised learning algorithm that can be employed for both classification and regression tasks. It continuously split the data into smaller subset based on some criteria. Then a voting mechanism is followed to make the final decision. There are two main types of decision trees: classification trees and regression trees. The classification trees are the ones where the output variable is discrete, while in the case of regression trees, the output variable is continuous. To construct the decision tree, entropy and information gain are generally employed . The process iteratively continues splitting the data at each node until the leaves are pure. To avoid the overfitting problem, a limit on the depth of the decision tree is also introduced.

The information gain for each attribute is calculated using the following equations:

$$Gini\ index = 1 - \sum pi\ 2\ c\ i{=}1$$

$$Gain\ ratio = Information\ gain/\ Split\ information$$



DECISION TREE 3.5

CHAPTER-4

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

4.1 RESULT

A number of experiments were performed to obtain the optimal parameters for each model. All the experiments were performed on the standard Intel (R) Core (TM) İ5-7200U CPU @ 2.50GHz computer in an Anaconda environment with Python as the programming language. The training dataset consists of 4176 samples. These samples were divided into training consisting of 2923 samples (70%) and testing 1253 samples (30%) subsets.

The Training accuracy with different optimizers is shown BPFFNN model obtained high accuracy for both training and testing. Moreover, Adadelta optimizer scored better compared to other optimizers with BPFFNN (89% training and 88% testing). The figure shows that all optimizers produced similar results except Sgd optimizer.

the convergence of five different optimization algorithms is illustrated in terms of training loss over the epochs. BPFFNN model had a lower training loss with Adagrad optimizer. Sgd starts with a rapid descent, but after 150 epoch stops improving. Rmsprop, Adadelta and Adam optimizers seem to perform almost the same.

Table 1 shows the confusion matrix for a multiclass classification problem with three classes (1, 2 and 3). As seen in the table, TP1 is the number of true positive samples in the class 1, that is, the number of samples that are correctly classified from class 1. E12 is misclassified samples, i.e., the samples from class 1 that were incorrectly classified as class 2. Accordingly, the false negative in the 1 class (FN1) is the sum of all class 1 samples that were incorrectly classified as class 2 or 3, i.e., is the sum of E12 and E13.

Briefly, FN of any class is equal to the sum of a row except value TP. The false positive (FP) of any class is equal to the sum of a column except the value TP. The true negative (TN) of any class is equal to the sum of values except row of true class and the column of predicted class.

FN1=E21 + E31 (10)

FP1 = E12 + E13 (11)

6 TN1 = TP2 +E32 +E23 + TP3 (12)

summarizes the accuracy of all the classifiers. Generally, all classifiers performed equally well, except the Gauss Naive Bayes, which obtained relatively lesser accuracy (60.88%). Moreover, the Random Forest classifier produced the highest performance on our dataset (87%) followed by SVM, which achieved an accuracy of 86.76 %. Furthermore, KNN and Decision tree classifiers reach almost equal accuracy 86.28%, 86.44%, respectively. Compared with the other classifiers, the performance of the proposed model was relatively better. From the obtained results, we can conclude that the BPFFNN reached the best accuracy in the abalone age prediction task.

.

TRUE/PRED	**G1**	**G2**	**G3**
G1	TP1	E21	E31
G2	E12	TP2	E32
G3	E32	E23	TP3

confusion matrix for a multiclass classification problem with three classes.

TABLE 4.1

G1: age < 7, G2: 7 ≤ age ≤, and G3: age > 16

4.2 RESULT FOR DECISION

matrix of the decision tree algorithm is presented in. Obtained results demonstrate that 1018 data in class 2 are classified correctly, i.e. TP2. Only 31 data in class 2(between 7 and 16 age of abalone) are misclassified, that is, FN2. The decision tree algorithm gave the best results after the random forest algorithm for class 2. Only 24 data classified correctly in the Group 3. The decision tree algorithm gave the worst results after the random forest algorithm for class 3. While for 3 class, it does not perform well

TRUE/PRED	G1	G2	G3
G1	41	25	0
G2	10	1018	21
G3	0	115	24

result for Decision Tree

TABLE 4.2

The overall results obtained for abalone classification using the six conventional classifiers were satisfactory except Gauss Naive Bayes classifier. The proposed BPFFNN outperformed all other classifiers in terms of classification accuracy.

In addition, we compared our approach with CNN based method proposed by authors in [8], which reported 79.09% accuracy. We believe that for simple datasets such as the one we used in this study, the conventional machine learning approaches are more effective than deep learning-based approaches. Even though deep learning-based approaches have shown high classification accuracy for many problems, yet they are data intensive.

We prefer conventional machine learning approaches over deep learning methods for both ease of implementation and classification accuracy in scenarios like this where the dataset is small.

4.3 DISCUSSION :

I prepared a report covering most of the applied part of linear regression, decision tree i.e. sklearn, statsmodels, ANOVA, and linear regression using Tensorflow and deep linear regression. The dataset was a very different one, coming from old research on abalones and how to predict their age. I have also provided a paper link with which we matched the results and reached some final conclusions as well. This notebook is appropriate for someone starting out with linear regression, or someone who is bored and would like to look into a not so technical dataset!

4.4 PERFORMANCE ANALYSIS

In this notebook we perform an exploratory data analysis over the Abalone Dataset, originally published at UCL machine learning , and which can be found at https://archive.ics.uci.edu/ml/datasets/Abalone. In this analysis we seek to understand the distribution of the dataset attributes, as well as the relationship between them.

The analysis is divided in five sections: on section

1. we briefly present what is an abalone and what is machine learning.
2. we presetnt the aim and scope of the present investigation.
3. we performe the experimental materials, algorithm used and attributes
4. we present our result and analysis
5. Finally on section 5 we present our conclusions.

CHAPTER-5

SUMMARY AND CONCLUSIONS

SUMMARY:

Based on the model results, we can see that the logistic regression model is perfor ming well on new examples of abalone, as described linear regression by an score R2 Score on Validation set : 0.5300147524184923 of and R2 score for the Decisio n Tree Regressor R2 Score on Validation set : 0.18175791293752985 We focus on these two metrics because they evaluate overall performance of model instead of weighing one class over another. Moreover, given a certain set of biological featur es of abalone, we're able to predict whether an abalone is old or young fairly accur ately while minimizing false negatives and false positives. We were able to obtain t hese results by testing different values for the model's hyperparameter, C, on vario us validation sets of the abalone training data in order to obtain an optimal logisti c regression model We also obtained the coefficients of the various biological feat ures that helped us understand how the features were influencing the prediction. T he weight features (shucked weight, whole weight, and shell weight) specifically ha d a large influence on the model's predictions. Contrasting the distributions of the se weight features between the old and young abalone helped us to investigate w hy shucked weight was having an opposite predictive effect in comparison with wh ole weight and shell weight, although consulting with domain experts may help us further understand this opposing effect. Overall, the model's ability to predict whet her an abalone is young or old based on specific biological characteristics is good but should be taken with a grain of salt given the imbalance of young and old abal one within the dataset, as well as some of the limitations of the included biological characteristics.

CONCLUSIONS

In this article, we covered the basics of Machine Learning, learnt about the model regression algorithms in action with the Abalone dataset. At the end of it, we could see that the accuracy of the model was not good. This is because the number of instances per class in the dataset is less for the model to perfectly learn the patterns between the features. Moreover, since this was an introductory article, we have not used the most appropriate algos needed specifically for this dataset.

REFERENCES

https://www.kaggle.com/datasets/rodolfomendes/abalone-dataset/code

APPENDIX

A. SCREENSHOTS

Importing Required Libraries

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style("darkgrid")

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor

from sklearn import  metrics

%matplotlib inline
```

Checking our data

In [4]:
```
dataframe = pd.read_csv('abalone.csv')
dataframe.head()
```

Out[4]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7

Descriptive Statistics

In [5]:
```
dataframe.describe().T
```

Out[5]:

	count	mean	std	min	25%	50%	75%	max
Length	4177.0	0.523992	0.120093	0.0750	0.4500	0.5450	0.615	0.8150
Diameter	4177.0	0.407881	0.099240	0.0550	0.3500	0.4250	0.480	0.6500
Height	4177.0	0.139516	0.041827	0.0000	0.1150	0.1400	0.165	1.1300
Whole weight	4177.0	0.828742	0.490389	0.0020	0.4415	0.7995	1.153	2.8255
Shucked weight	4177.0	0.359367	0.221963	0.0010	0.1860	0.3360	0.502	1.4880
Viscera weight	4177.0	0.180594	0.109614	0.0005	0.0935	0.1710	0.253	0.7600
Shell weight	4177.0	0.238831	0.139203	0.0015	0.1300	0.2340	0.329	1.0050
Rings	4177.0	9.933684	3.224169	1.0000	8.0000	9.0000	11.000	29.0000

Label Encoding of Categorical Values

In [6]:
```
le=LabelEncoder()
dataframe['Sex']=le.fit_transform(dataframe['Sex'])
```

In [7]:
```
dataframe
```

Out[7]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	2	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.1500	15
1	2	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.0700	7
2	0	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.2100	9
3	2	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.1550	10
4	1	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.0550	7
...	...	...	...	...	...	...	...	...	...
4172	0	0.565	0.450	0.165	0.8870	0.3700	0.2390	0.2490	11
4173	2	0.590	0.440	0.135	0.9660	0.4390	0.2145	0.2605	10
4174	2	0.600	0.475	0.205	1.1760	0.5255	0.2875	0.3080	9
4175	0	0.625	0.485	0.150	1.0945	0.5310	0.2610	0.2960	10
4176	2	0.710	0.555	0.195	1.9485	0.9455	0.3765	0.4950	12

4177 rows × 9 columns

Data Visualisation

In [8]:
```
!pip install missingno
```

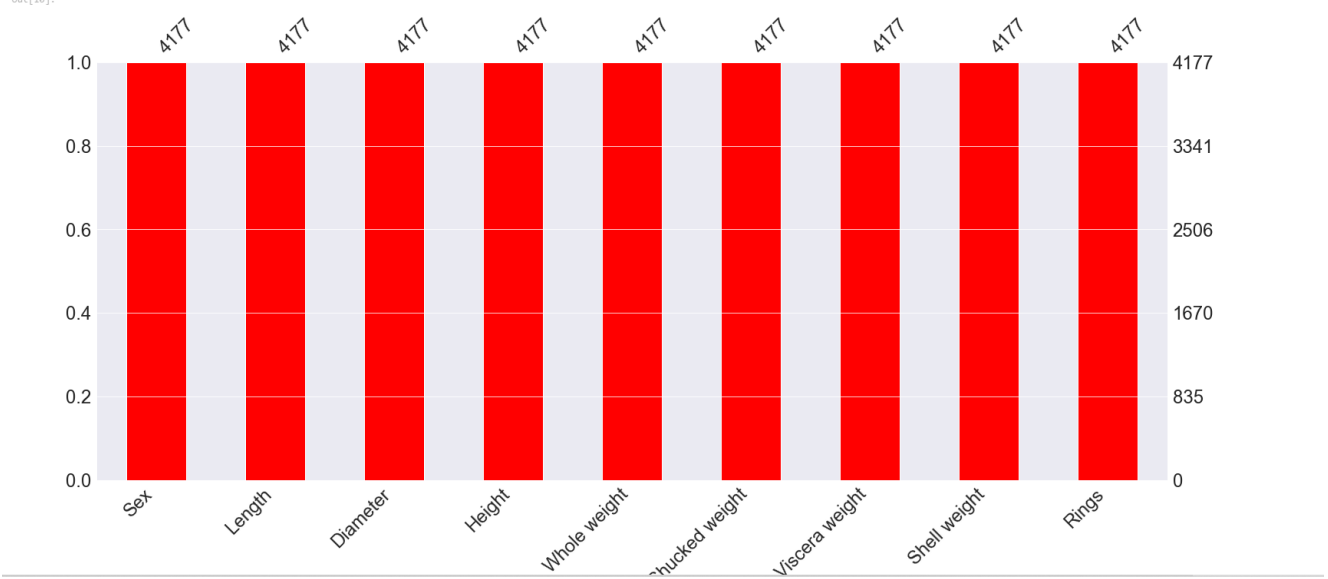
```
Requirement already satisfied: missingno in c:\users\91733\anaconda3\lib\site-packages (0.5.1)
Requirement already satisfied: seaborn in c:\users\91733\anaconda3\lib\site-packages (from missingno) (0.11.2)
Requirement already satisfied: numpy in c:\users\91733\anaconda3\lib\site-packages (from missingno) (1.21.5)
Requirement already satisfied: scipy in c:\users\91733\anaconda3\lib\site-packages (from missingno) (1.7.3)
Requirement already satisfied: matplotlib in c:\users\91733\anaconda3\lib\site-packages (from missingno) (3.5.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\91733\anaconda3\lib\site-packages (from matplotlib->missingno) (9.0.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\91733\anaconda3\lib\site-packages (from matplotlib->missingno) (4.25.0)
Requirement already satisfied: cycler>=0.10 in c:\users\91733\anaconda3\lib\site-packages (from matplotlib->missingno) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\91733\anaconda3\lib\site-packages (from matplotlib->missingno) (1.3.2)
Requirement already satisfied: packaging>=20.0 in c:\users\91733\anaconda3\lib\site-packages (from matplotlib->missingno) (21.3)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\91733\anaconda3\lib\site-packages (from matplotlib->missingno) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\91733\anaconda3\lib\site-packages (from matplotlib->missingno) (3.0.4)
Requirement already satisfied: six>=1.5 in c:\users\91733\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib->missingno) (1.16.0)
Requirement already satisfied: pandas>=0.23 in c:\users\91733\anaconda3\lib\site-packages (from seaborn->missingno) (1.4.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\91733\anaconda3\lib\site-packages (from pandas>=0.23->seaborn->missingno) (2021.3)
```

In [9]:
```
import missingno as msno
```

Plotting a Bar Chart of the Missing Values

In [10]:
```
msno.bar(dataframe,fontsize = 24, color = 'red')
```

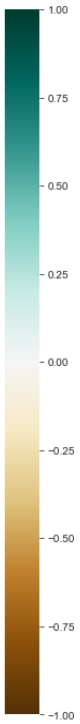
Out[10]:
```
<AxesSubplot:>
```



Heatmap of missing values

In [11]:
```
msno.heatmap(dataframe,fontsize = 24, cmap = 'BrBG')
```

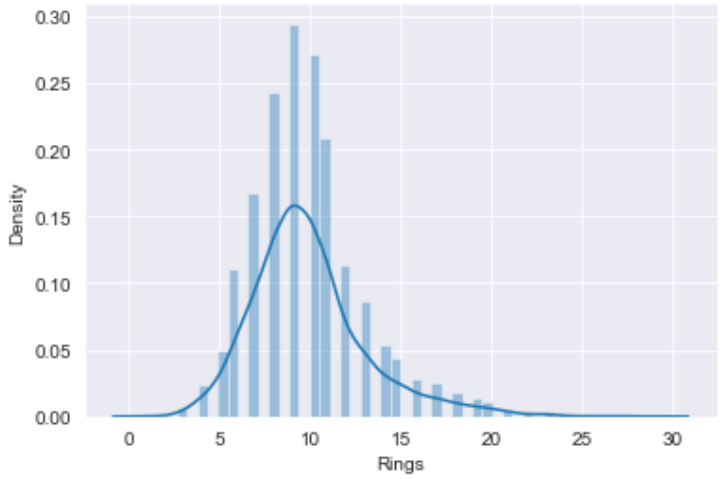
Out[11]:
```
<AxesSubplot:>
```



Displot for Target

In [12]:
```
sns.distplot(dataframe['Rings'])
```

Out[12]:
```
<AxesSubplot:xlabel='Rings', ylabel='Density'>
```



Correlation Heatmap of Variables

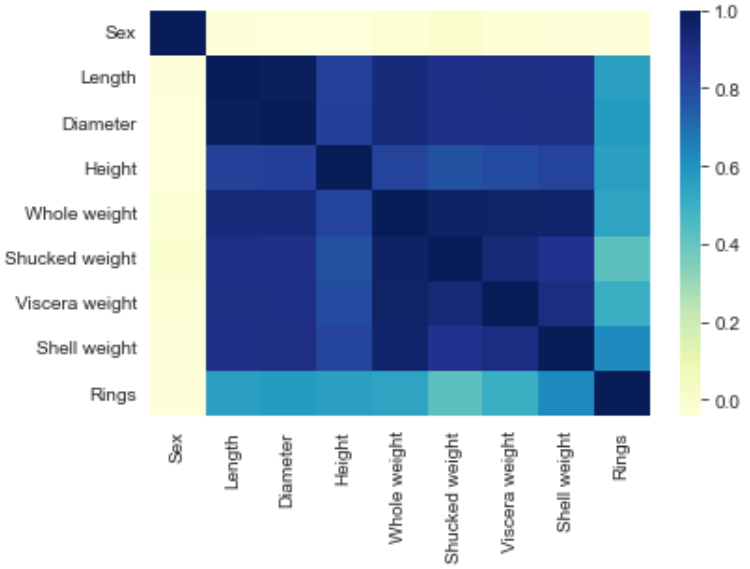
In [13]:
```
corr=dataframe.corr()
print(corr)
```

```
                     Sex    Length  Diameter    Height  Whole weight  \
Sex             1.000000 -0.036066 -0.038874 -0.042077     -0.021391
Length         -0.036066  1.000000  0.986812  0.827554      0.925261
Diameter       -0.038874  0.986812  1.000000  0.833684      0.925452
Height         -0.042077  0.827554  0.833684  1.000000      0.819221
Whole weight   -0.021391  0.925261  0.925452  0.819221      1.000000
Shucked weight -0.001373  0.897914  0.893162  0.774972      0.969405
Viscera weight -0.032067  0.903018  0.899724  0.798319      0.966375
Shell weight   -0.034854  0.897706  0.905330  0.817338      0.955355
Rings          -0.034627  0.556720  0.574660  0.557467      0.540390

                Shucked weight  Viscera weight  Shell weight     Rings
Sex                 -0.001373       -0.032067     -0.034854 -0.034627
Length               0.897914        0.903018      0.897706  0.556720
Diameter             0.893162        0.899724      0.905330  0.574660
Height               0.774972        0.798319      0.817338  0.557467
Whole weight         0.969405        0.966375      0.955355  0.540390
Shucked weight       1.000000        0.931961      0.882617  0.420884
Viscera weight       0.931961        1.000000      0.907656  0.503819
Shell weight         0.882617        0.907656      1.000000  0.627574
Rings                0.420884        0.503819      0.627574  1.000000
```

```
In [14]: sns.heatmap(corr,cmap = 'YlGnBu')
```

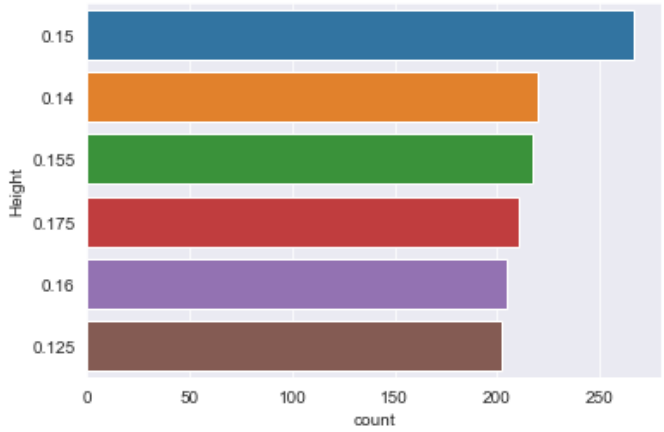
```
Out[14]: <AxesSubplot:>
```



Maximum Heights

```
In [15]: sns.countplot(y=dataframe['Height'],order=dataframe['Height'].value_counts().head(6).index)
```

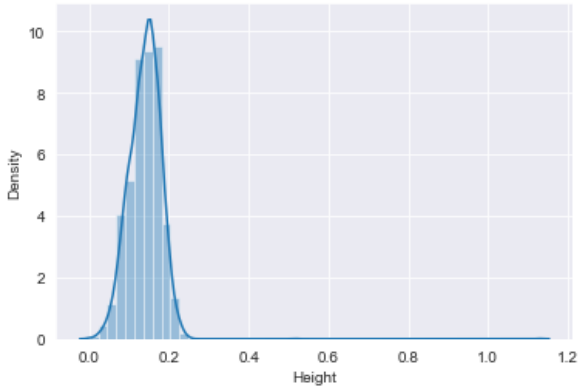
```
Out[15]: <AxesSubplot:xlabel='count', ylabel='Height'>
```



Displot for Height

In [16]:
```
sns.distplot(dataframe['Height'])
```

Out[16]:
```
<AxesSubplot:xlabel='Height', ylabel='Density'>
```



Breaking down into X and y

In [17]:
```
X = dataframe.iloc[:, :-1].values
y = dataframe.iloc[:, -1].values
```

Creating Training and Validation sets

In [18]:
```
train_X,val_X,train_y,val_y = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

In [19]:
```
print("Shape of Training X :",train_X.shape)
print("Shape of Validation X :",val_X.shape)
```

```
Shape of Training X : (3341, 8)
Shape of Validation X : (836, 8)
```

In [20]:
```
print("Shape of Training y :",train_y.shape)
print("Shape of Validation y :",val_y.shape)
```

```
Shape of Training y : (3341,)
Shape of Validation y : (836,)
```

Model Building

Linear Regression

In [21]:
```
lr = LinearRegression()
lr.fit(train_X,train_y)
print('Attempting to fit Linear Regressor')
```

```
Attempting to fit Linear Regressor
```

In [22]:
```
%%time
y_pred_val_lr = lr.predict(val_X)
print('MAE on Validation set :',metrics.mean_absolute_error(val_y, y_pred_val_lr))
print("\n")
print('MSE on Validation set :',metrics.mean_squared_error(val_y, y_pred_val_lr))
print("\n")
print('RMSE on Validation set :',np.sqrt(metrics.mean_absolute_error(val_y, y_pred_val_lr)))
print("\n")
print('R2 Score on Validation set :',metrics.r2_score(val_y, y_pred_val_lr))
print("\n")
```

```
MAE on Validation set : 1.6130841939880156


MSE on Validation set : 5.104186010193352


RMSE on Validation set : 1.270072515247856


R2 Score on Validation set : 0.5300147524184923


CPU times: total: 0 ns
Wall time: 4.58 ms
```

Decision Tree Regressor

In [23]:
```
dc = DecisionTreeRegressor(random_state = 0)
dc.fit(train_X,train_y)
print('Attempting to fit Decision Tree Regressor')
```

```
Attempting to fit Decision Tree Regressor
```

In [24]:
```
%%time
y_pred_val_dc = dc.predict(val_X)
print('MAE on Validation set :',metrics.mean_absolute_error(val_y, y_pred_val_dc))
print("\n")
print('MSE on Validation set :',metrics.mean_squared_error(val_y, y_pred_val_dc))
print("\n")
print('RMSE on Validation set :',np.sqrt(metrics.mean_absolute_error(val_y, y_pred_val_dc)))
print("\n")
print('R2 Score on Validation set :',metrics.r2_score(val_y, y_pred_val_dc))
print("\n")
```

```
MAE on Validation set : 2.0586124401913874


MSE on Validation set : 8.886363636363637


RMSE on Validation set : 1.4347865486515363


R2 Score on Validation set : 0.18175791293752985


CPU times: total: 0 ns
Wall time: 4.83 ms
```

B. SOURCE CODE

```
!pip install shutup
Requirement already satisfied: shutup in c:\users\91733\anaconda3\lib\site-packa
ges (0.2.0)

import shutup; shutup.please()
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style("darkgrid")

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor

from sklearn import  metrics

%matplotlib inline
dataframe = pd.read_csv('abalone.csv')
dataframe.head()
dataframe.describe().T
le=LabelEncoder()
dataframe['Sex']=le.fit_transform(dataframe['Sex'])
dataframe
!pip install missingno
import missingno as msno
msno.bar(dataframe,fontsize = 24, color = 'red')
msno.heatmap(dataframe,fontsize = 24, cmap = 'BrBG')
sns.distplot(dataframe['Rings'])
corr=dataframe.corr()
print(corr)
sns.heatmap(corr,cmap = 'YlGnBu')
sns.countplot(y=dataframe['Height'],order=dataframe['Height'].value_counts().hea
d(6).index)
sns.distplot(dataframe['Height'])
X = dataframe.iloc[:, :-1].values
y = dataframe.iloc[:, -1].values
train_X,val_X,train_y,val_y = train_test_split(X, y, test_size = 0.2, random_state =
0)
print("Shape of Training X :",train_X.shape)
print("Shape of Validation X :",val_X.shape)

print("Shape of Training y :",train_y.shape)
print("Shape of Validation y :",val_y.shape)
lr = LinearRegression()
lr.fit(train_X,train_y)
```

```
print('Attempting to fit Linear Regressor')
%%time
y_pred_val_lr = lr.predict(val_X)
print('MAE on Validation set :',metrics.mean_absolute_error(val_y, y_pred_val_lr))
print("\n")
print('MSE on Validation set :',metrics.mean_squared_error(val_y, y_pred_val_lr))
print("\n")
print('RMSE on Validation set :',np.sqrt(metrics.mean_absolute_error(val_y,
y_pred_val_lr)))
print("\n")
print('R2 Score on Validation set :',metrics.r2_score(val_y, y_pred_val_lr))
print("\n")
dc = DecisionTreeRegressor(random_state = 0)
dc.fit(train_X,train_y)
print('Attempting to fit Decision Tree Regressor')
%%time
y_pred_val_dc = dc.predict(val_X)
print('MAE on Validation set :',metrics.mean_absolute_error(val_y,
y_pred_val_dc))
print("\n")
print('MSE on Validation set :',metrics.mean_squared_error(val_y,
y_pred_val_dc))
print("\n")
print('RMSE on Validation set :',np.sqrt(metrics.mean_absolute_error(val_y,
y_pred_val_dc)))
print("\n")
print('R2 Score on Validation set :',metrics.r2_score(val_y, y_pred_val_dc))
print("\n")
```