SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

ABALONE DATASET USING MACHINE LEARNING

PROJECT SUPERVISOR : Ms.S.R.SRIVIDHYA

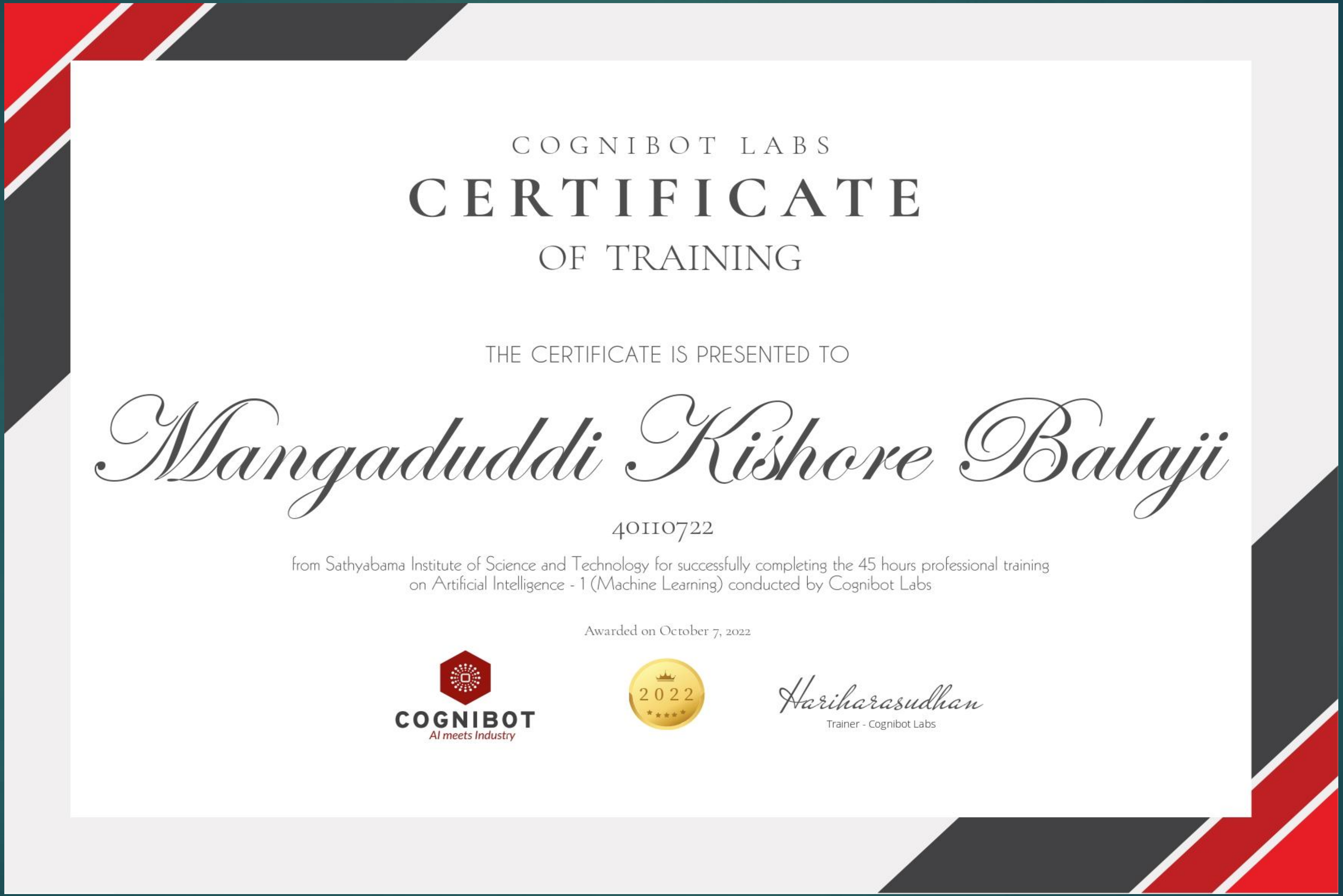
NAME OF THE STUDENT : MANGADUDDI KISHORE BALAJI

REGISTER NUMBER : 40110722

PRESENTATION OUTLINE

- COURSE CERTIFICATE
- INTRODUCTION
- OBJECTIVES
- SYSTEM ARCHITECTURE / IDEATION MAP
- MODULE IMPLEMENTATION
- APPLICATION SNAPSHOTS
- RESULT AND DISCUSSIONS
- CONCLUSION & FUTURE WORK
- REFERENCES

COURSE CERTIFICATE



COGNIBOT LABS

CERTIFICATE

OF TRAINING

THE CERTIFICATE IS PRESENTED TO

Mangaduddi Kishore Balaji

40110722

from Sathyabama Institute of Science and Technology for successfully completing the 45 hours professional training on Artificial Intelligence - 1 (Machine Learning) conducted by Cognibot Labs

Awarded on October 7, 2022

COGNIBOT
AI meets Industry

2022

Hariharasudhan
Trainer - Cognibot Labs

INTRODUCTION

- THE MAIN AIM IS TO PREDICT THE AGE OF THE ABALONE BASED ON THE MACHINE LEARNING

- LINEAR REGRESSION AND DECISION TREE ALGORITHMS

- ABALONE ARE A VERY COMMON TYPE OF SHELLFISH. THERE FLESH IS CONSIDER TO BE A DELICACYAND THEIR SHELLS ARE POPULAR IN JEWELLARY

- IN THIS WORK I CONSIDER THE PROBLEM OF ESTIMATING THE AGE ABALONE GIVEN ITS PHYSICAL CHARACTERISTICS

OBJECTIVES

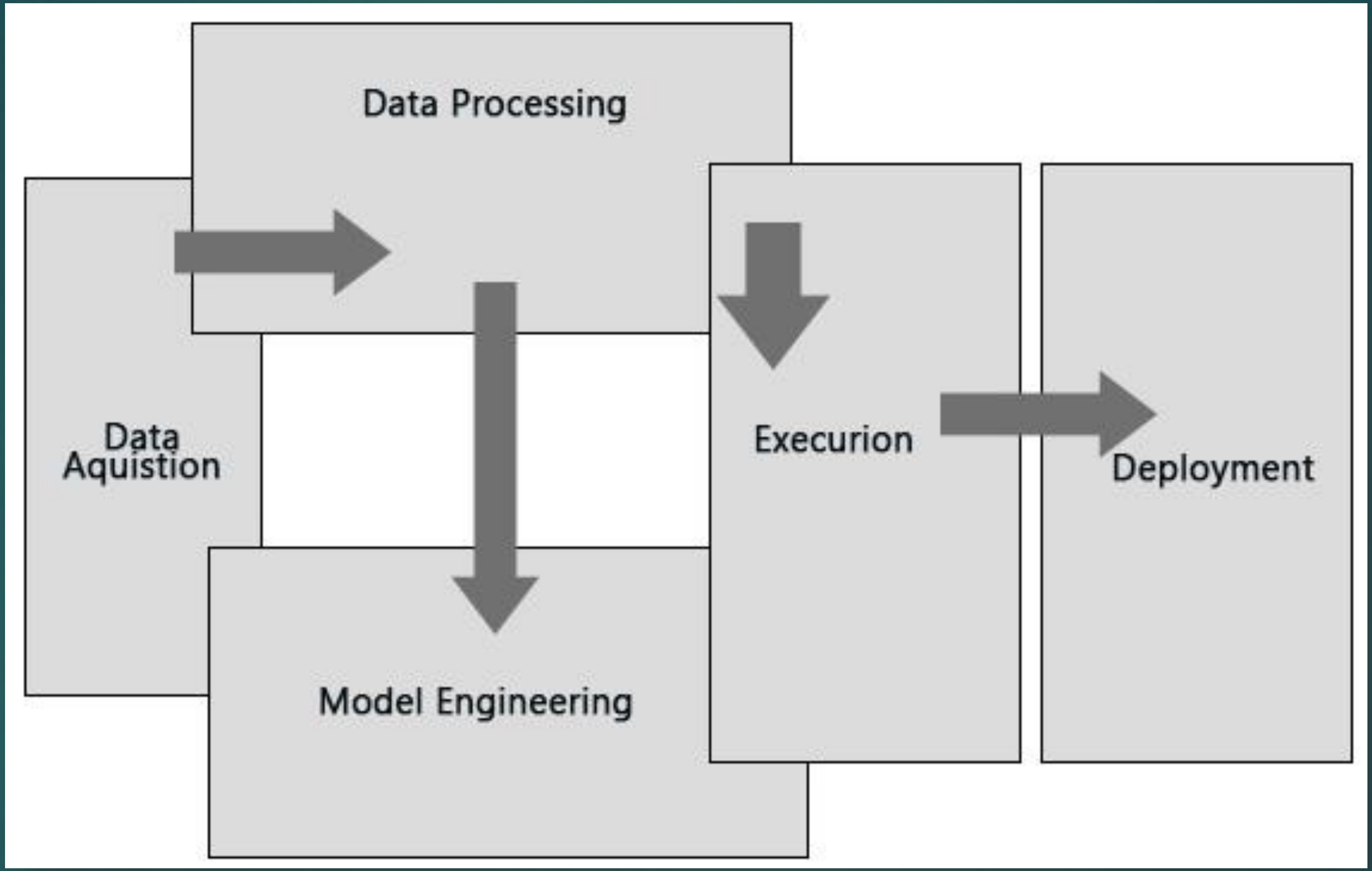
- ❑ The objective of this project is to predicting the age of abalone from physical measurements using the 1994 abalone data "The Population Biology of Abalone (Haliotis species) in Tasmania. I. Blacklip Abalone (H. rubra) from the North Coast and Islands of Bass Strait".

- ❑ The UCI Machine Learning Repository provides one dataset abalone.data, it contains 4177 observations, 8 descriptives features and 1 target feature

- ❑ The variable description is produced here from abalone.names file:

- o Name / Data Type / Measurement Unit / Description
- o Sex / nominal / -- / M, F, and I (infant)
- o Length / continuous / mm / Longest shell measurement
- o Diameter / continuous / mm / perpendicular to length
- o Height / continuous / mm / with meat in shell
- o Whole weight / continuous / grams / whole abalone
- o Shucked weight / continuous / grams / weight of meat
- o Viscera weight / continuous / grams / gut weight (after bleeding)
- o Shell weight / continuous / grams / after being dried
- o Rings / integer / -- / +1.5 gives the age in years

SYSTEM ARCHITECTURE / IDEATION MAP



DECISION FLOW ARCHITECTURE FOR MACHINE LEARNING SYSTEM

MODULE IMPLEMENTATION

- **PANDAS**
- **NUMPY**
- **MATPLOTLIB**
- **SEABORN**
- **SKLEARN**

PANDAS: Pandas is an open-source, BSD-licensed Python library providing high performance. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc

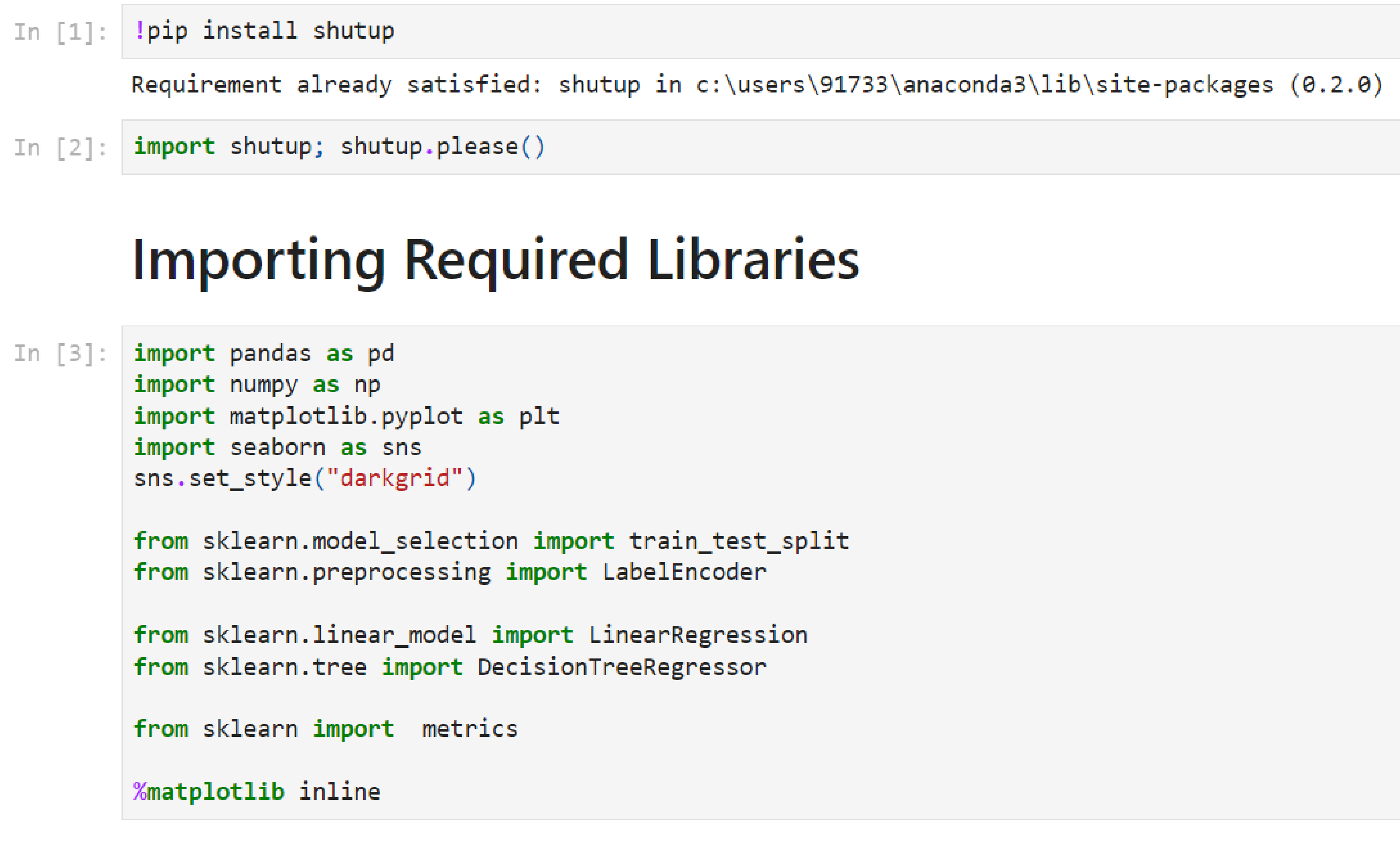
NUMPY: NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices

MATPLOTLIB: most of the matplotlib utilities lies under the pyplot submodule and are usually imported under the plt alias

SEABORN: Seaborn is **a** Python data visualization library based on matplotlib. It provides a high-level interface

SKLEARN : scikit-learn is an open-source Python library that implements a range of machine learning, pre-processing, cross-validation, and visualization algorithms using a unified interface**.**

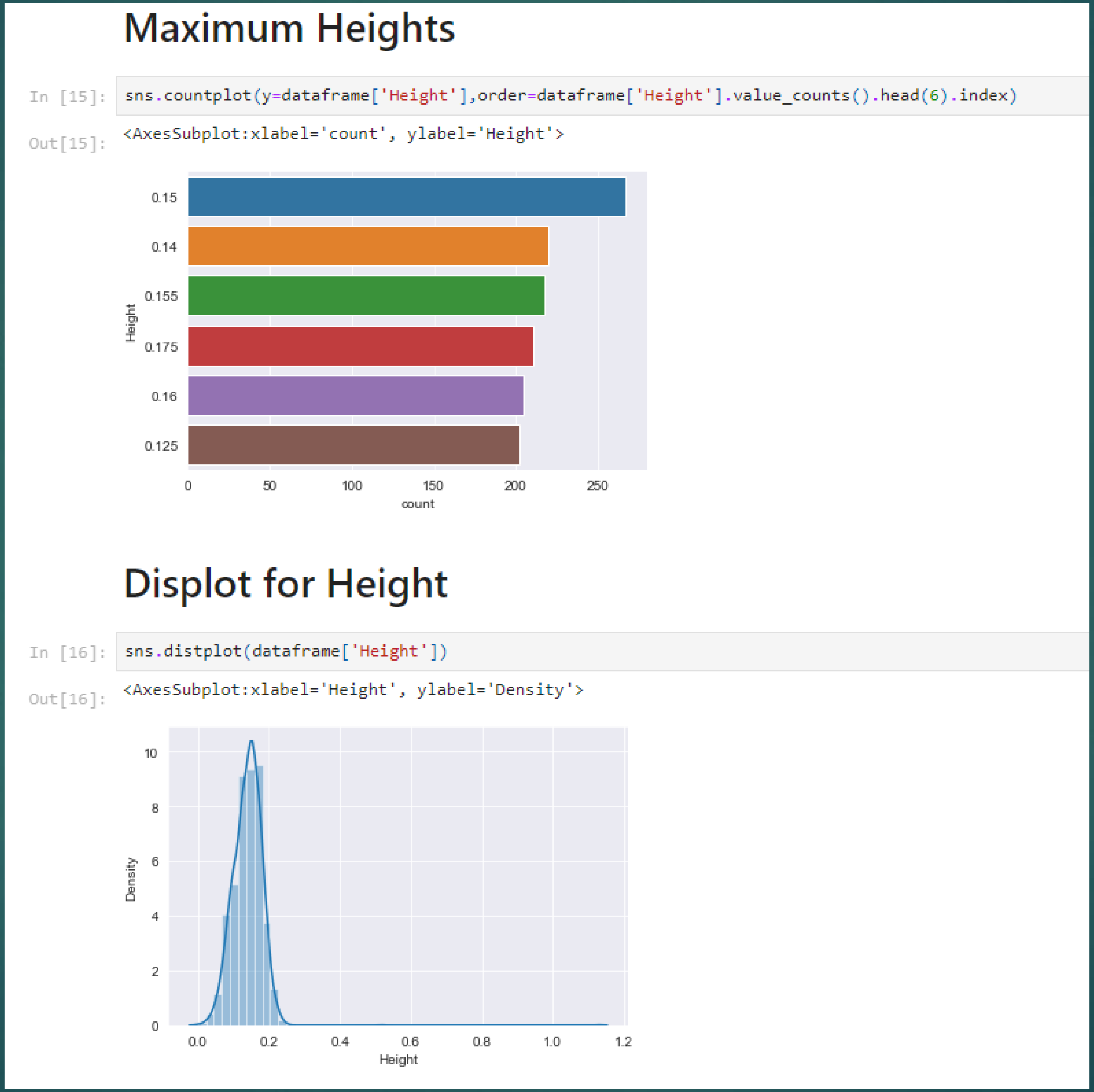
APPLICATION SNAPSHOTS



```
In [1]: !pip install shutup

        Requirement already satisfied: shutup in c:\users\91733\anaconda3\lib\site-packages (0.2.0)

In [2]: import shutup; shutup.please()
```

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
```



Maximum Heights

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

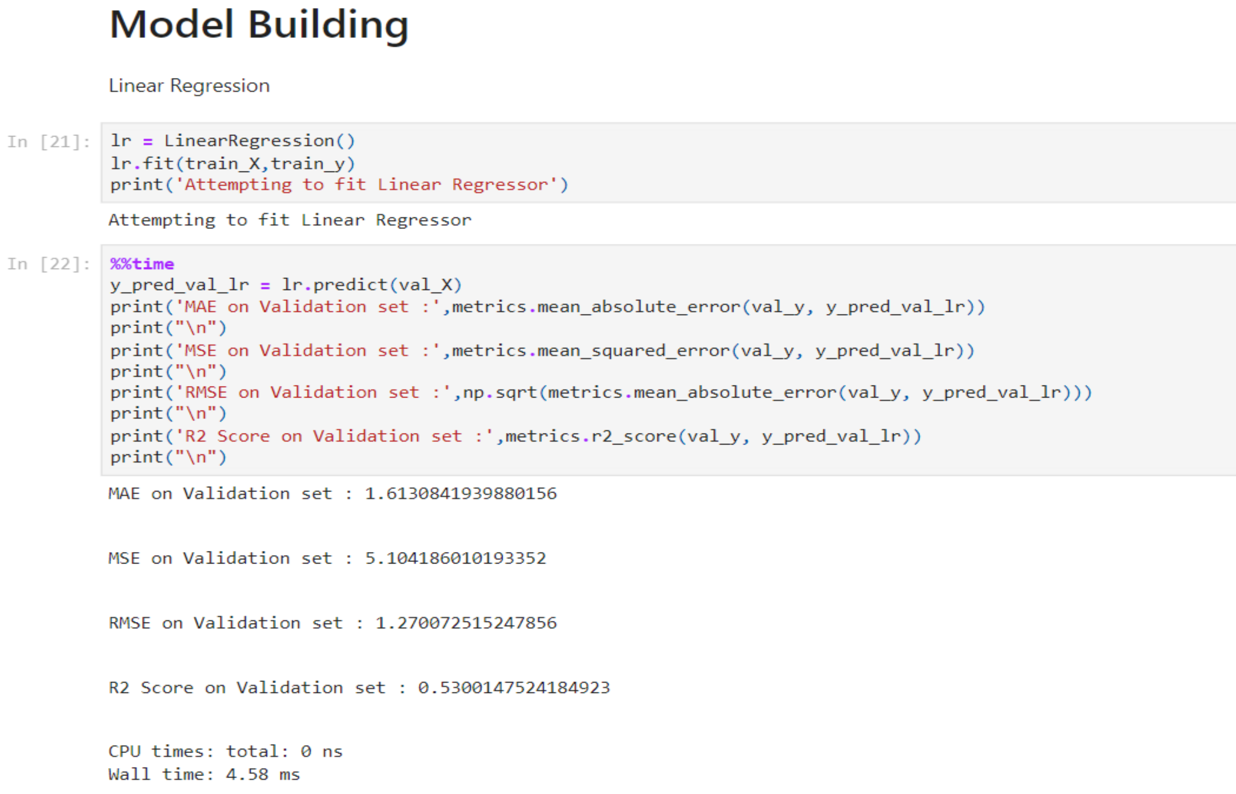
```
Out[15]: <AxesSubplot:xlabel='count', ylabel='Height'>
```

Displot for Height

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

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

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.



Model Building

Linear Regression

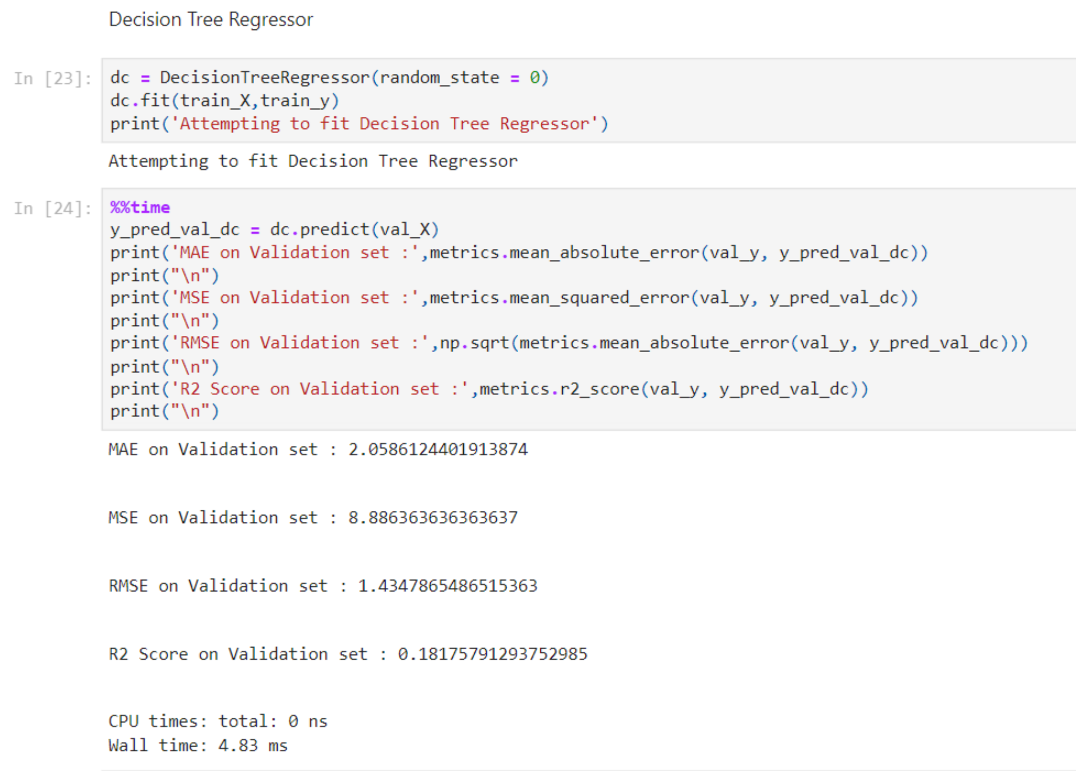
```
In [21]: lr = LinearRegression()
         lr.fit(train_X,train_y)
         print('Attempting to fit Linear Regressor')
```

```
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: 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 .



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
```

RESULT AND DISCUSSIONS

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 4177samples. These samples were divided into training consisting of 2923 samples (70%) and testing 1253 samples (30%) subsets.

CONCLUSION & FUTURE WORK

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
- Git hub

THANK YOU