

Data Analytics Assignment - 4: Abalon Age Prediction

Team ID :

PNT2022TMID45269

Student Name: MOHAN P

Project Name :Visualizing and Predicting Heart Diseases with an Interactive Dash Board

Student Roll No: 812119104020

Dataset :

<https://drive.google.com/file/d/1mOWrMc8b-ODshkEfyHB1UFwO5V5s3fcW/view>

Import Necesssary packages

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
sns.set_style("darkgrid")
```

```
import train_test_split
from sklearn.preprocessing import LabelEncoder
```

```
from sklearn.linear_model import LinearRegression
```

```
from sklearn import metrics
```

Download and Load the dataset

```
df=pd.read_csv('/content/abalone.csv')
```

Perform descriptive statistics on the dataset

```
df.head()
```

```
df.tail()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
4172	F	0.565	0.450	0.165	0.8870	0.3700	0.2390	0.2490	11
4173	M	0.590	0.440	0.135	0.9660	0.4390	0.2145	0.2605	10
4174	M	0.600	0.475	0.205	1.1760	0.5255	0.2875	0.3080	9
4175	F	0.625	0.485	0.150	1.0945	0.5310	0.2610	0.2960	10
4176	M	0.710	0.555	0.195	1.9485	0.9455	0.3765	0.4950	12

```
df.shape
```

```
(4177, 9)
```

```
df.describe()
```

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000
mean	0.523992	0.407881	0.139516	0.828742	0.359367	0.180594	0.180594
std	0.120093	0.099240	0.041827	0.490389	0.221963	0.109614	0.109614
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	0.000500
25%	0.450000	0.350000	0.115000	0.441500	0.186000	0.093500	0.093500
50%	0.545000	0.425000	0.140000	0.799500	0.336000	0.171000	0.171000
75%	0.615000	0.480000	0.165000	1.153000	0.502000	0.253000	0.253000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4177 entries, 0 to 4176
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Sex              4177 non-null   object
1   Length           4177 non-null   float64
2   Diameter         4177 non-null   float64
3   Height           4177 non-null   float64
4   Whole weight     4177 non-null   float64
5   Shucked weight   4177 non-null   float64
6   Viscera weight   4177 non-null   float64
```

```

7  Shell weight    4177 non-null    float64
8  Rings          4177 non-null    int64
dtypes: float64(7), int64(1), object(1)
memory usage: 293.8+ KB

```

```

df['age']=df['Rings']+1.5
df=df.drop('Rings', axis = 1)

```

```
df.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	age
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	16.5
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	8.5
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	10.5
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	11.5
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	8.5

Check for Missing values and deal with them

```
df.isnull().sum()
```

```

Sex                0
Length             0
Diameter           0
Height             0
Whole weight       0
Shucked weight     0
Viscera weight     0
Shell weight       0
age                0
dtype: int64

```

```
df.columns
```

```

Index(['Sex', 'Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight',
       'Viscera weight', 'Shell weight', 'age'],
      dtype='object')

```

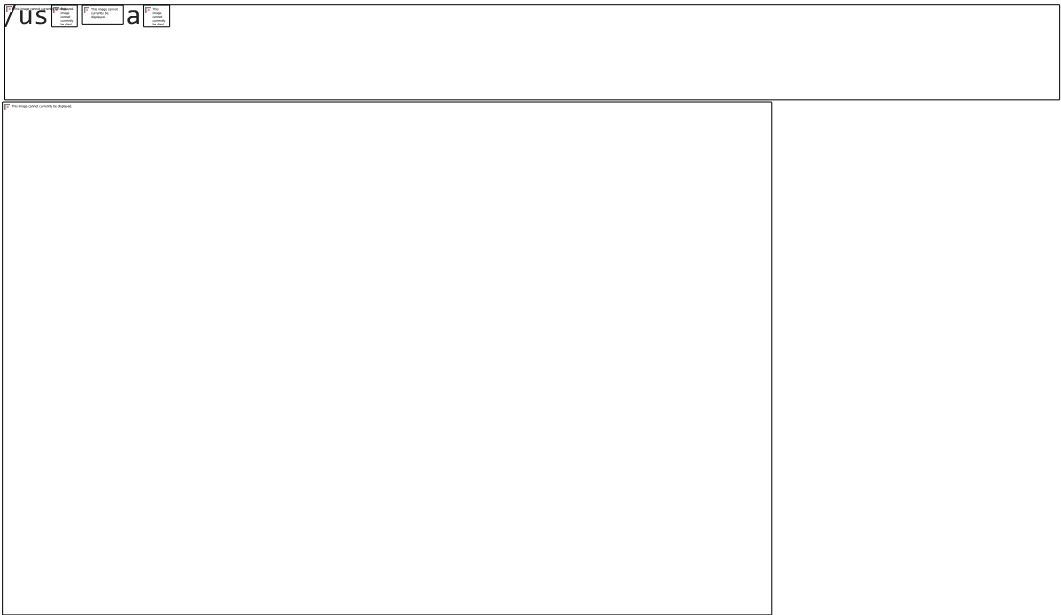
Perform Below Visualizations

· Univariate Analysis

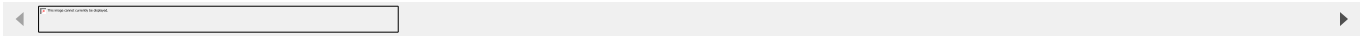
Bi-Variate Analysis

· Multi-Variate Analysis

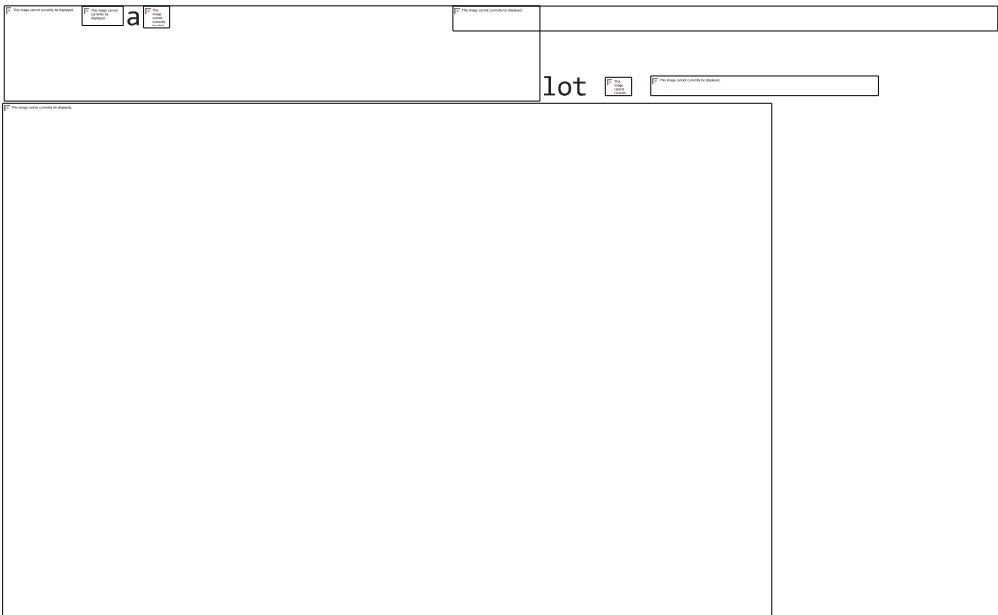
```
#univariate analysis
sns.distplot(df['Length'])
```



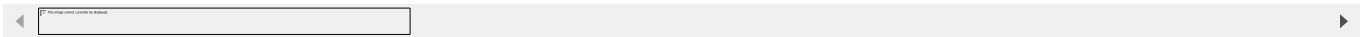
FutureWarning: `d:



```
sns.boxp
```



FutureWarning: `ss th



```
i- a is
lot()
```

The page content is empty.

The page content is empty.

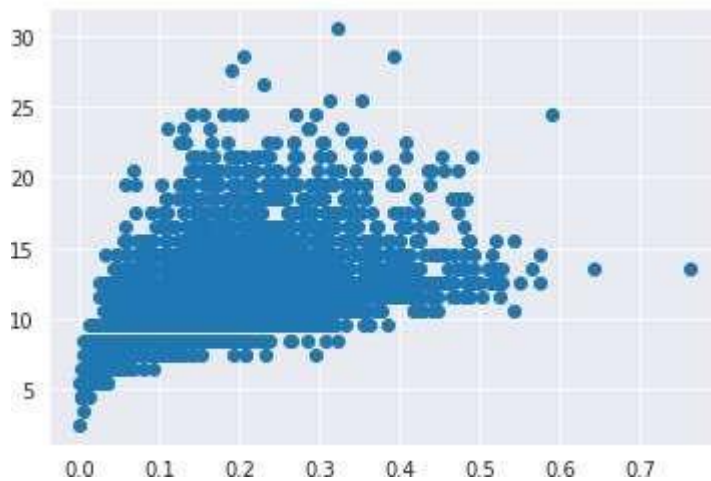
The page content is empty.

The page content is empty.

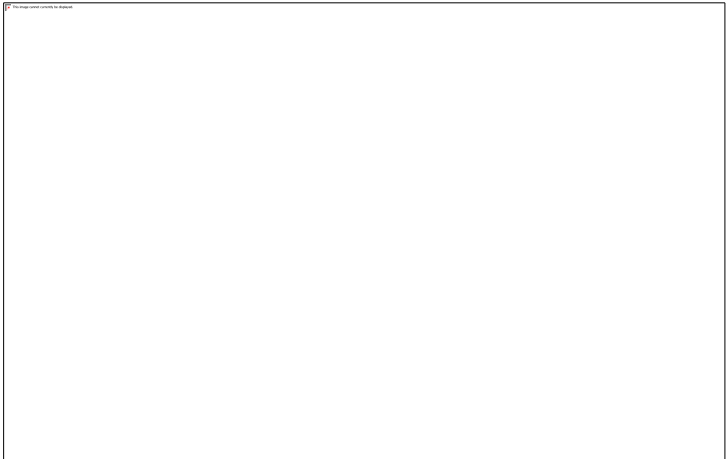
Find the outliers and replace them outliers

[illegible]

`p` `r = 's'` `s` `1` `2` `3` `4` `5` `6` `7` `8` `9` `10` `11` `12` `13` `14` `15` `16` `17` `18` `19` `20` `21` `22` `23` `24` `25` `26` `27` `28` `29` `30` `31` `32` `33` `34` `35` `36` `37` `38` `39` `40` `41` `42` `43` `44` `45` `46` `47` `48` `49` `50` `51` `52` `53` `54` `55` `56` `57` `58` `59` `60` `61` `62` `63` `64` `65` `66` `67` `68` `69` `70` `71` `72` `73` `74` `75` `76` `77` `78` `79` `80` `81` `82` `83` `84` `85` `86` `87` `88` `89` `90` `91` `92` `93` `94` `95` `96` `97` `98` `99` `100` `101` `102` `103` `104` `105` `106` `107` `108` `109` `110` `111` `112` `113` `114` `115` `116` `117` `118` `119` `120` `121` `122` `123` `124` `125` `126` `127` `128` `129` `130` `131` `132` `133` `134` `135` `136` `137` `138` `139` `140` `141` `142` `143` `144` `145` `146` `147` `148` `149` `150` `151` `152` `153` `154` `155` `156` `157` `158` `159` `160` `161` `162` `163` `164` `165` `166` `167` `168` `169` `170` `171` `172` `173` `174` `175` `176` `177` `178` `179` `180` `181` `182` `183` `184` `185` `186` `187` `188` `189` `190` `191` `192` `193` `194` `195` `196` `197` `198` `199` `200` `201` `202` `203` `204` `205` `206` `207` `208` `209` `210` `211` `212` `213` `214` `215` `216` `217` `218` `219` `220` `221` `222` `223` `224` `225` `226` `227` `228` `229` `230` `231` `232` `233` `234` `235` `236` `237` `238` `239` `240` `241` `242` `243` `244` `245` `246` `247` `248` `249` `250` `251` `252` `253` `254` `255` `256` `257` `258` `259` `260` `261` `262` `263` `264` `265` `266` `267` `268` `269` `270` `271` `272` `273` `274` `275` `276` `277` `278` `279` `280` `281` `282` `283` `284` `285` `286` `287` `288` `289` `290` `291` `292` `293` `294` `295` `296` `297` `298` `299` `300` `301` `302` `303` `304` `305` `306` `307` `308` `309` `310` `311` `312` `313` `314` `315` `316` `317` `318` `319` `320` `321` `322` `323` `324` `325` `326` `327` `328` `329` `330` `331` `332` `333` `334` `335` `336` `337` `338` `339` `340` `341` `342` `343` `344` `345` `346` `347` `348` `349` `350` `351` `352` `353` `354` `355` `356` `357` `358` `359` `360` `361` `362` `363` `364` `365` `366` `367` `368` `369` `370` `371` `372` `373` `374` `375` `376` `377` `378` `379` `380` `381` `382` `383` `384` `385` `386` `387` `388` `389` `390` `391` `392` `393` `394` `395` `396` `397` `398` `399` `400` `401` `402` `403` `404` `405` `406` `407` `408` `409` `410` `411` `412` `413` `414` `415` `416` `417` `418` `419` `420` `421` `422` `423` `424` `425` `426` `427` `428` `429` `430` `431` `432` `433` `434` `435` `436` `437` `438` `439` `440` `441` `442` `443` `444` `445` `446` `447` `448` `449` `450` `451` `452` `453` `454` `455` `456` `457` `458` `459` `460` `461` `462` `463` `464` `465`

[illegible]

```
r = 11
p = 1
p = 1
= ], y = [' '])
```

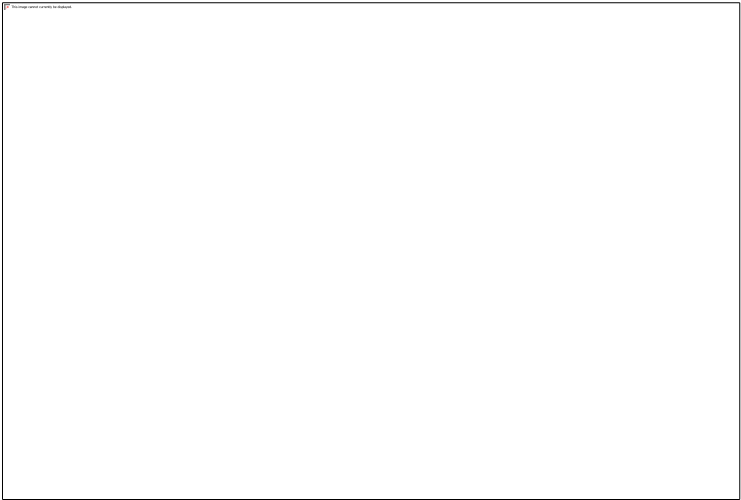


```

r =
p
p

'11'
'11'

' ] > '
' ( [' ' ] < ' in ce =
' ( [' ' ] > ' la =
], y = [' ' ])
```



```

r =
p
p

'luc'
'sce'

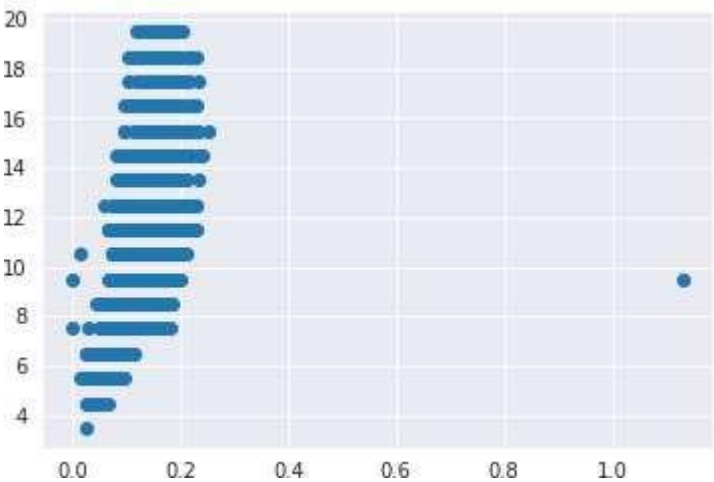
' ] > = ' [ ' ' ] < ' in ce =
' ] < 1 ) ' ' ] > ' lace =
], y = [' ' ])
```

The image shows a code editor with a single line of code: `in ce =`

```
in ce =  
la =  
r =  
p  
p
```



```
in ce = T  
in ce = T  
in ce = T  
r =  
p  
p
```



```
in ce = T  
in ce = T  
in ce = T  
r =  
p  
p
```



```
lib.s_sublot
```

```
H
```

KEY INSIGHT All but 'sex'

->

-> None of the features have minimum = 0 except Height (requires re-check)

-> Each feature has difference scale range

s

```
I('r', '1', 'uc', 'F', ''],
```

Feature Selection and Standardization

```
X = s = 1)
```

LI REGRESSION

```
sklectr
```

```
s = ()
```

```
X_new = se
```

```
in, in, in, = tri, , i = 0.25)
```

```
ir =
```

```
LinearRegression()
```

```
i =  
=
```

```
from sklearn.metrics import mean_squared_error  
s =  
p('n
```

```
p =  
p('n :')
```

n Squared Error of :3.569916
n Squared Error of

The Lower the Mean Squared Error, better the forecast.

```
from sklearn.metrics import r2  
s = r2  
,
```

```
p = r2  
y
```

R2 Score of
R2 Score of

The .

.

Colab paid products - [Cancel](#)

✓ 0s completed at 12:48 PM

