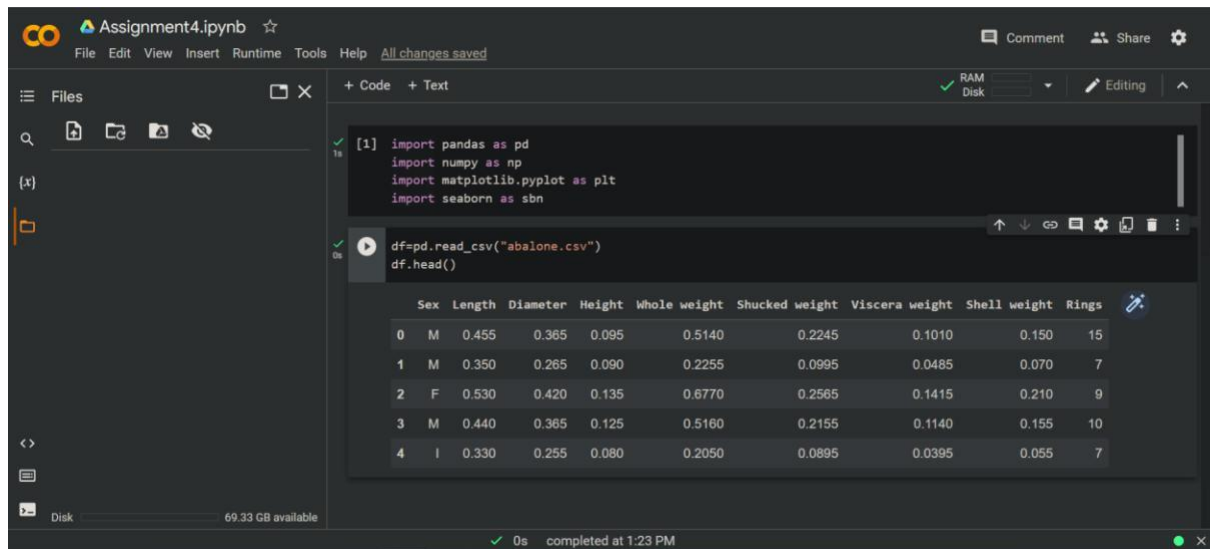


## Assignment - 4

Team ID : PNT2022TMID23536

Loading the dataset:



The screenshot shows a Jupyter Notebook titled 'Assignment4.ipynb'. The code cell contains the following Python code:

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

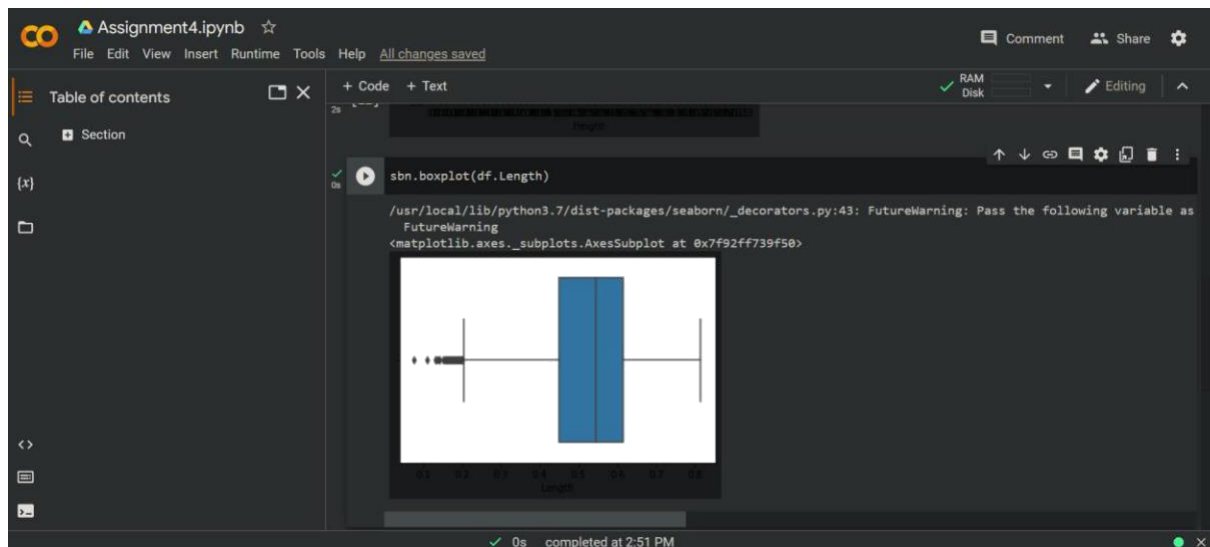
df=pd.read_csv("abalone.csv")
df.head()
```

The output displays the first five rows of the 'abalone' dataset:

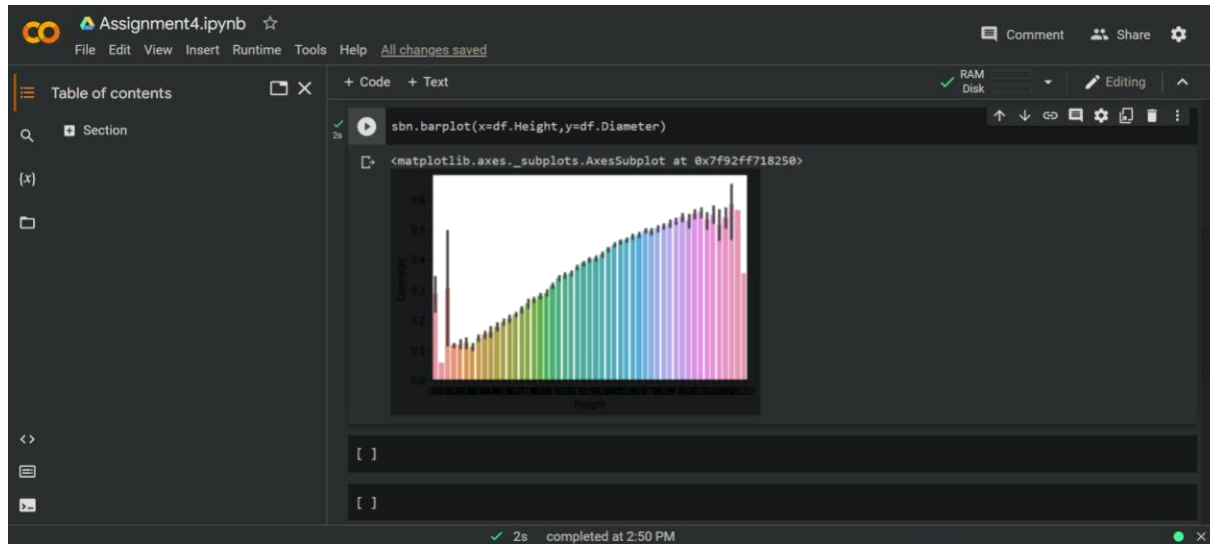
	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
4	I	0.330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

Perform Below Visualizations:

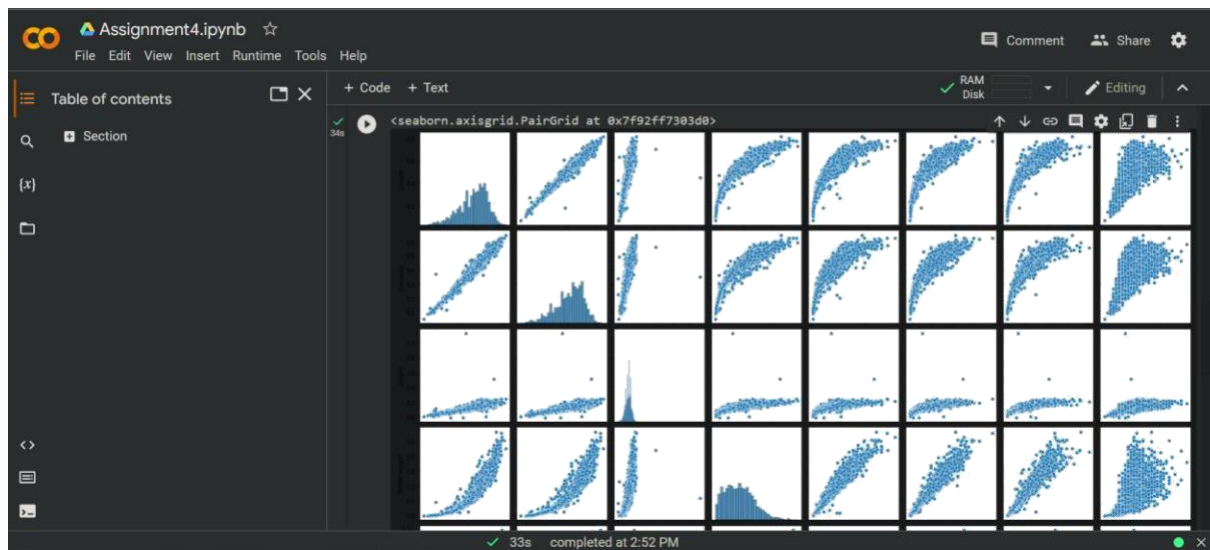
Univariate Analysis



## Bi-Variate Analysis



## Multi-Variate Analysis



Perform descriptive analytics on the dataset

Assignment4.ipynb

File Edit View Insert Runtime Tools Help Saving...

Table of contents

Section

(x)

<>

+

Code

+ Text

RAM

Disk

Editing

[15] df['Length'].mode()

```
0    0.550
1    0.625
dtype: float64
```

[17] df['Height'].mean()

```
0.13951639932966242
```

[20] df.count()

```
Sex          4177
Length       4177
Diameter     4177
Height       4177
Whole weight 4177
Shucked weight 4177
Viscera weight 4177
Shell weight 4177
Rings        4177
dtype: int64
```

0s completed at 2:56 PM

Assignment4.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Table of contents

Section

(x)

<>

+

Code

+ Text

RAM

Disk

Editing

[23] df['Shell weight'].sum()

```
997.5964999999999
```

[24] df['Rings'].product()

```
0
```

[25] df['Whole weight'].max()

```
2.8255
```

[ ]

0s completed at 2:59 PM

Check for Missing values and deal with them, Find the outliers and replace them outliers

Assignment4.ipynb

File Edit View Insert Runtime Tools Help Saving...

Table of contents

Section

(x)

<>

+

Code

+ Text

RAM

Disk

Editing

[27] df.isna().any()

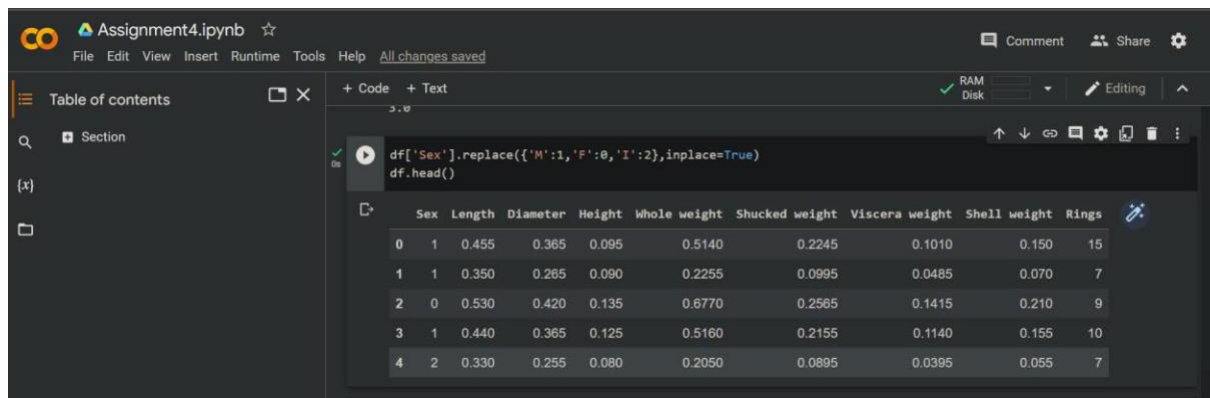
```
Sex          False
Length       False
Diameter     False
Height       False
Whole weight False
Shucked weight False
Viscera weight False
Shell weight False
Rings        False
dtype: bool
```

qu1=df.Rings.quantile(0.25)  
qu3=df.Rings.quantile(0.75)  
qr=qu3-qu1  
print(qr)

```
3.0
```

0s completed at 3:01 PM

Check for Categorical columns and perform encoding



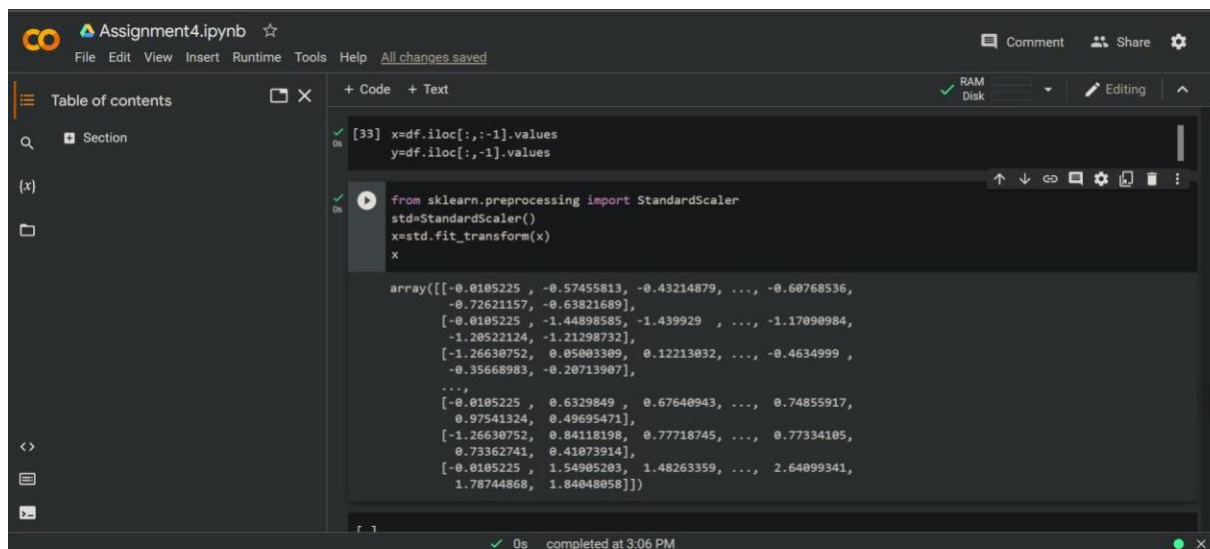
The screenshot shows a Jupyter Notebook interface with the file 'Assignment4.ipynb'. The code cell contains the following Python code:

```
df['Sex'].replace({'M':1,'F':0,'I':2},inplace=True)
df.head()
```

The output of the code is a preview of the first five rows of the DataFrame:

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

Split the data into dependent and independent variables, Scale the independent variables



The screenshot shows a Jupyter Notebook interface with the file 'Assignment4.ipynb'. The code cell contains the following Python code:

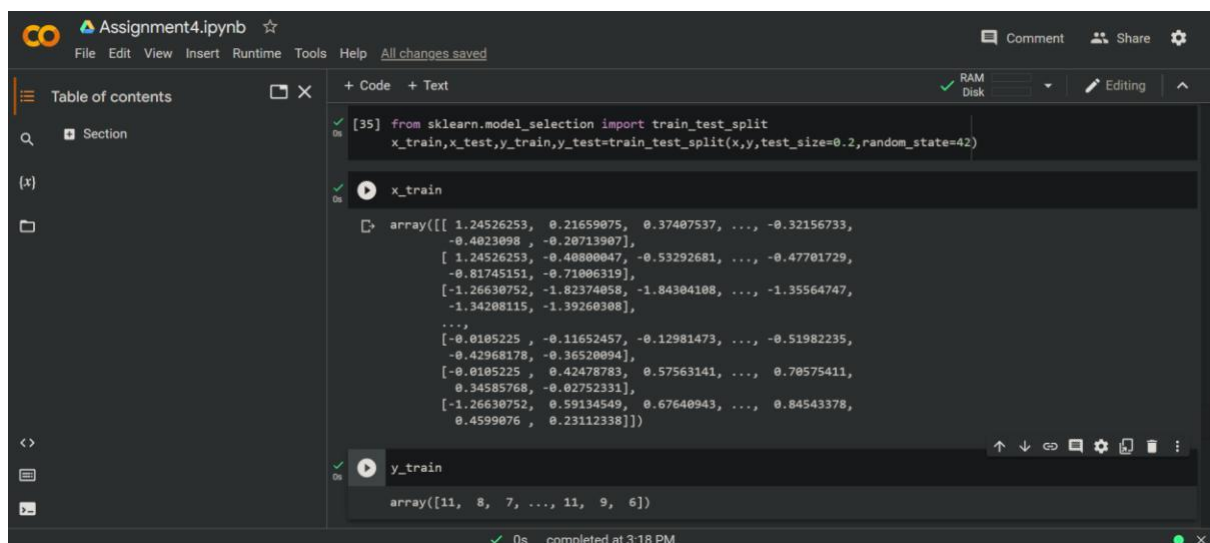
```
[33] x=df.iloc[:, :-1].values
     y=df.iloc[:, -1].values

from sklearn.preprocessing import StandardScaler
std=StandardScaler()
x=std.fit_transform(x)
x
```

The output of the code is a 2D array representing the scaled features:

```
array([[ -0.0105225,  -0.57455813,  -0.43214879, ...,  -0.60768536,
        -0.72621157,  -0.63821689],
       [ -0.0105225,  -1.44898585,  -1.439929, ...,  -1.17090984,
        -1.28522124,  -1.21298732],
       [ -1.26630752,  0.05003309,  0.12213032, ...,  -0.4634999,
        -0.35668983,  -0.20713907],
       ...,
       [ -0.0105225,  0.6329849,  0.67640943, ...,  0.74855917,
        0.97541324,  0.49695471],
       [ -1.26630752,  0.84118198,  0.77718745, ...,  0.77334105,
        0.73362741,  0.41073914],
       [ -0.0105225,  1.54905203,  1.48263359, ...,  2.64099341,
        1.78744868,  1.84048058]])
```

Split the data into training and testing



The screenshot shows a Jupyter Notebook interface with the file 'Assignment4.ipynb'. The code cell contains the following Python code:

```
[35] from sklearn.model_selection import train_test_split
     x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)

x_train
```

The output of the code is a 2D array representing the training features:

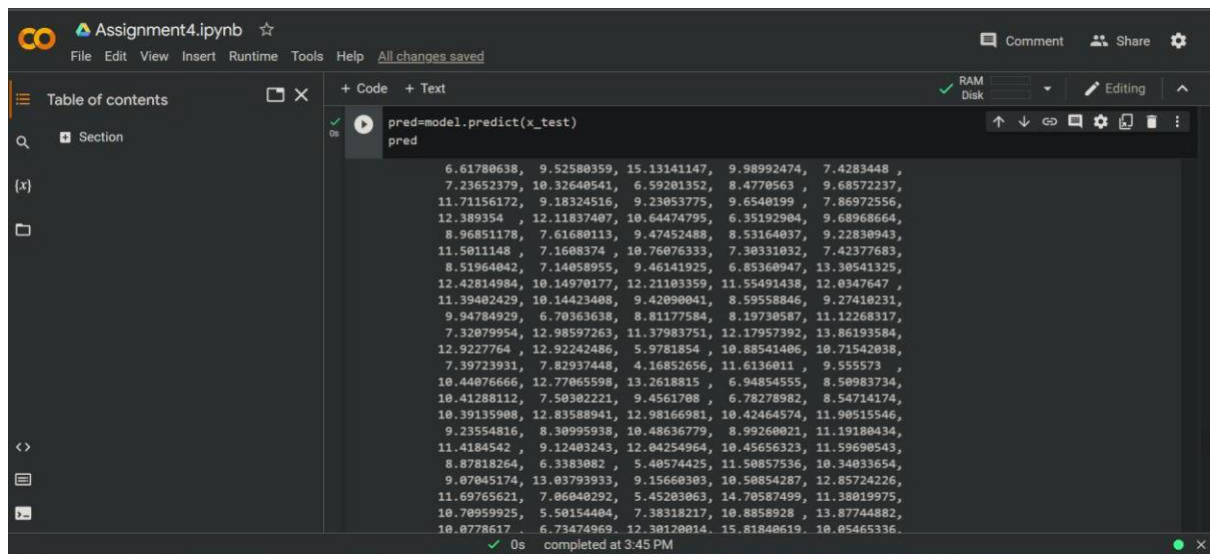
```
array([[ 1.24526253,  0.21659075,  0.37407537, ..., -0.32156733,
        -0.4023098,  -0.20713907],
       [ 1.24526253, -0.40800047, -0.53292681, ..., -0.47701729,
        -0.81745151, -0.71006319],
       [ -1.26630752, -1.82374058, -1.84304108, ..., -1.35564747,
        -1.34208115, -1.39260308],
       ...,
       [ -0.0105225, -0.11652457, -0.12981473, ..., -0.51982235,
        -0.42968178, -0.36520094],
       [ -0.0105225,  0.42478783,  0.57563141, ...,  0.70575411,
        0.34585768, -0.02752331],
       [ -1.26630752,  0.59134549,  0.67640943, ...,  0.84543378,
        0.4599076,  0.23112338]])
```

The next code cell shows the output of the training labels:

```
y_train
array([11,  8,  7, ..., 11,  9,  6])
```



## Test the Model

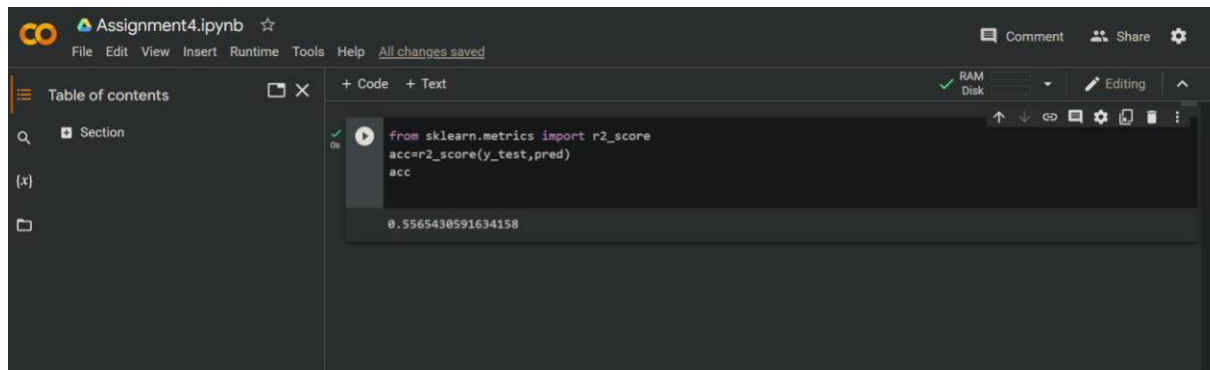


A screenshot of a Jupyter Notebook titled "Assignment4.ipynb". The interface includes a top menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A "Table of contents" sidebar is on the left. The main code cell contains the following Python code:

```
pred=model.predict(x_test)
pred
```

The output of the code cell is a large array of 50 numerical values, representing the model's predictions for the test set. The values are displayed in a single line, separated by commas. The status bar at the bottom indicates "0s completed at 3:45 PM".

## Measure the performance using Metrics



A screenshot of a Jupyter Notebook titled "Assignment4.ipynb". The interface is similar to the previous one. The main code cell contains the following Python code:

```
from sklearn.metrics import r2_score
acc=r2_score(y_test,pred)
acc
```

The output of the code cell is a single numerical value, 0.5565430591634158, representing the R-squared score. The status bar at the bottom indicates "0s completed at 3:45 PM".