31-day31-linear-regression

October 28, 2023

Simple Linear Regression By: Loga Aswin

[43]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[44]:
```
data = pd.read_csv("/content/student_scores.csv")
print(data.head())
```

```
   Hours  Scores
0    2.5      21
1    5.1      47
2    3.2      27
3    8.5      75
4    3.5      30
```

[26]:
```
data.shape
```

[26]:
```
(500, 4)
```

[27]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Gender  500 non-null    object
 1   Height  500 non-null    int64
 2   Weight  500 non-null    int64
 3   Index   500 non-null    int64
dtypes: int64(3), object(1)
memory usage: 15.8+ KB
```

[28]:
```
#checking Null Values
data.isna().sum()
```

[28]:
```
Gender    0
Height    0
```

```
Weight    0
Index     0
dtype: int64
```

[29]:
```
#Drop duplicate values

data.duplicated()
```

[29]:
```
0      False
1      False
2      False
3      False
4      False
       …
495    False
496    False
497    False
498    False
499    False
Length: 500, dtype: bool
```

Calculate Summary Statistics:

[30]:
```
data.describe()
```

[30]:

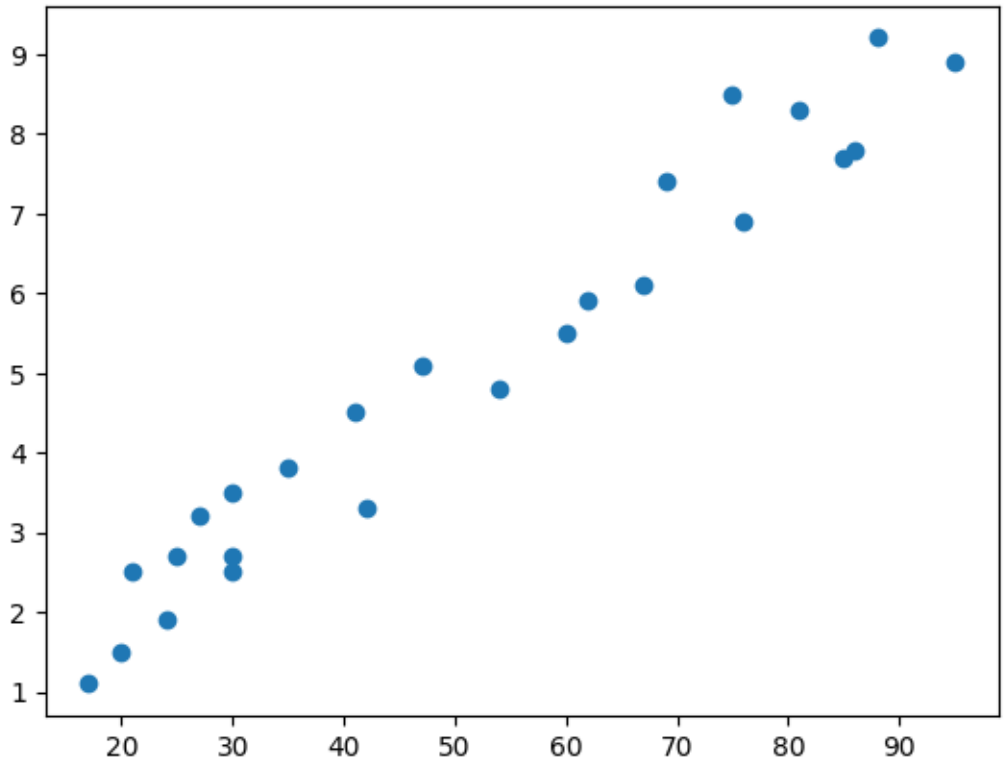
	Height	Weight	Index
count	500.000000	500.000000	500.000000
mean	169.944000	106.000000	3.748000
std	16.375261	32.382607	1.355053
min	140.000000	50.000000	0.000000
25%	156.000000	80.000000	3.000000
50%	170.500000	106.000000	4.000000
75%	184.000000	136.000000	5.000000
max	199.000000	160.000000	5.000000

[45]:
```
data.dropna(inplace = True)
```

Visualization- (Scatter Plot)

[46]:
```
plt.scatter(data["Scores"], data["Hours"])
```

[46]:
```
<matplotlib.collections.PathCollection at 0x7ff8e2615210>
```



[49]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
```

Split Dataset:

[51]:
```
x = data["Hours"].values.reshape(-1, 1)
y = data['Scores'].values.reshape(-1,1)
```

[52]:
```
# Split the data into train and test sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
 ↪random_state=0)
```

Model Fitting:

[55]:
```
regressor = LinearRegression()
regressor.fit(x_train, y_train)
```

[55]:
```
LinearRegression()
```

predict output for the x_test dataset

[56]:
```
y_pred = regressor.predict(x_test)
```

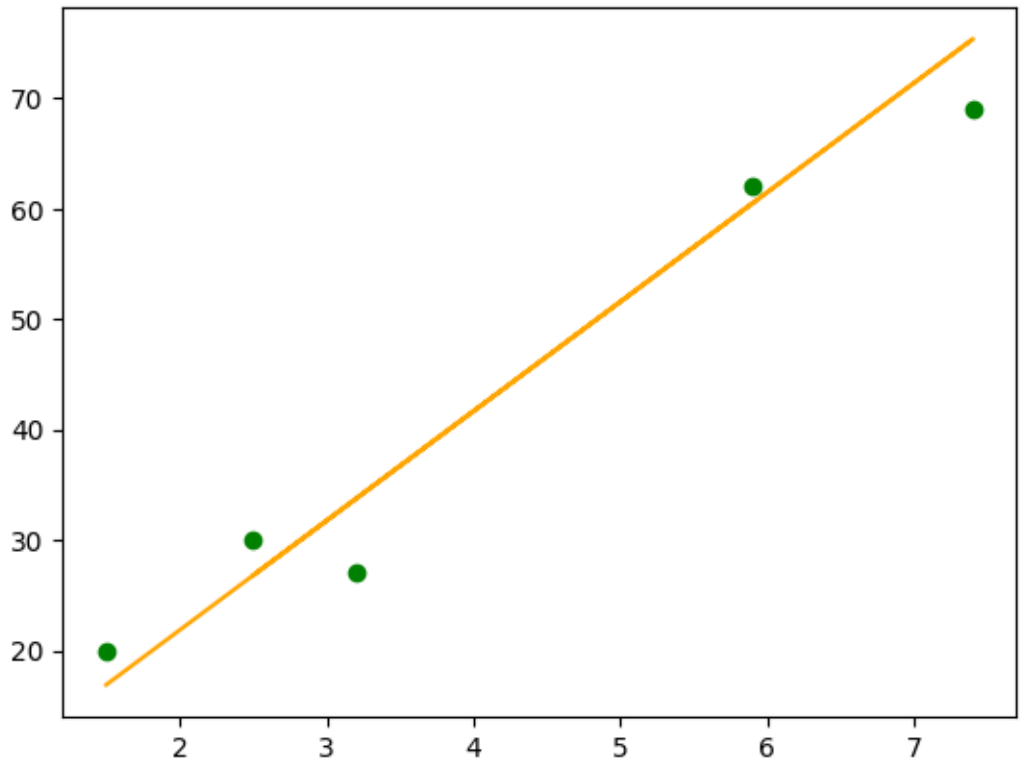
[58]:
```
print(regressor.intercept_)

print(regressor.coef_)
```

```
[2.01816004]
[[9.91065648]]
```

[61]:
```
#scatter plotplt.scatter(x_test, y_test, color="green")

#orange line shows the prediction line
plt.plot(x_test,y_pred, color="orange")
plt.show()
```



Checking Accuracy Score

[68]:
```
print("Mean Absolute Error: ", metrics.mean_absolute_error(y_test, y_pred))
print("Mean Squared Error: ", metrics.mean_squared_error(y_test, y_pred))
print("R2 Score: ", metrics.r2_score(y_test, y_pred))
```

```
Mean Absolute Error:  4.183859899002982
Mean Squared Error:  21.598769307217456
R2 Score:  0.9454906892105354
```