Student Score Prediction

August 14, 2024

1 Student Score Prediction based on their Study hours

1.0.1 By Jhumar Godwin C. Caraan

The objective of this project is to develop a prediction model using supervised learning, specifically linear regression, to estimate student scores based on the number of hours they study.

1.1 Metadata

Hours - The number of hours that the student studied.

Scores - The total score achieved by the student in the exam.

1.2 Import necessary libraries

[120]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

1.3 Read the data

[124]:
```
# Reading data from remote link
url = "http://bit.ly/w-data"
df = pd.read_csv(url)

# Print the first 5 rows
df.head(5)
```

[124]:

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

1.4 EDA

[122]:
```
# Checking the number of rows and columns
df.shape
```

[122]:
```
(25, 2)
```

[123]:
```
# Checking null values
df.isnull().sum()
```

[123]:
```
Hours     0
Scores    0
dtype: int64
```

[100]:
```
# Checking Duplicates
df.duplicated().sum()
```

[100]:
```
0
```

[101]:
```
# Checking datatypes and the entries in the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Hours   25 non-null     float64
 1   Scores  25 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

[102]:
```
# Getting statatistical summary of data
df.describe()
```

[102]:

	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

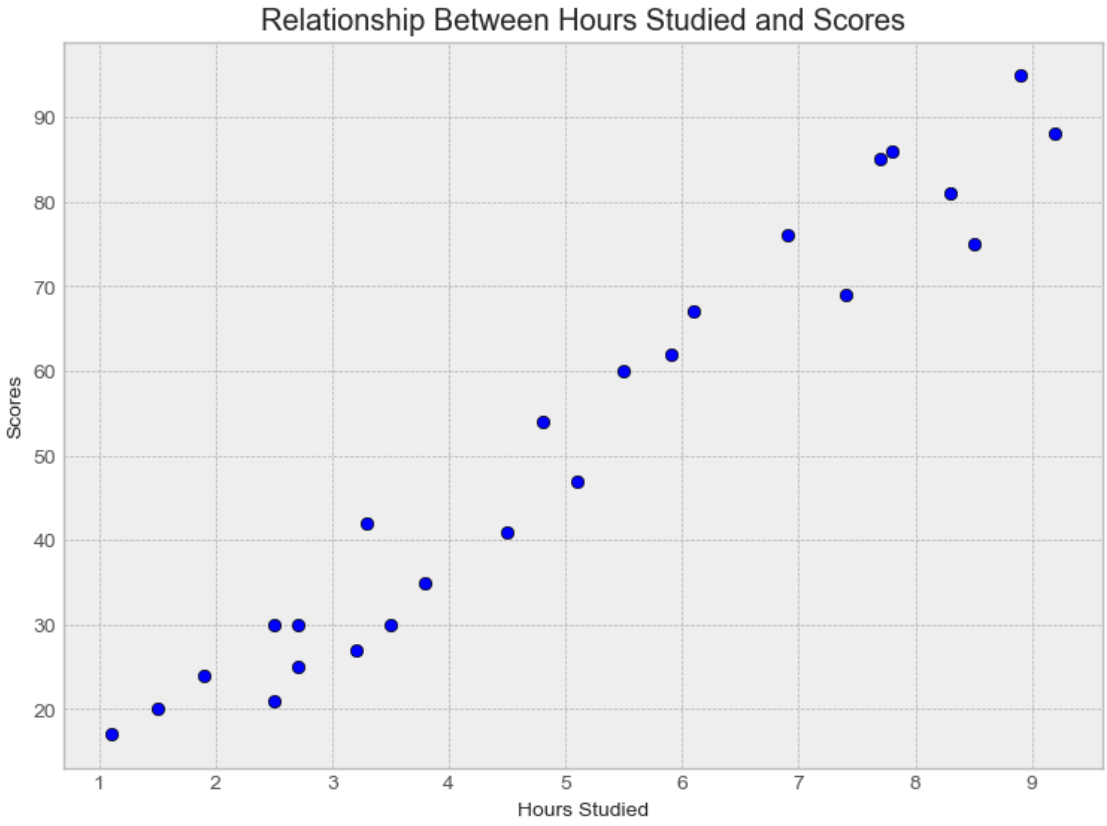
[131]:
```
import matplotlib.pyplot as plt

# Set the style for the plot
plt.style.use('bmh')
```

```
# Create a scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(df['Hours'], df['Scores'], color='blue', edgecolor='black')

# Add title and labels
plt.title('Relationship Between Hours Studied and Scores')
plt.xlabel('Hours Studied')
plt.ylabel('Scores')

# Display the plot
plt.show()
```



1.5 Training the model

[132]:
```
X=df[['Hours']].values # Making Hours 2d arry
y=df['Scores'].values
```

[133]:
```
from sklearn.model_selection import train_test_split
```

```
# Splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                test_size=0.2, random_state=0)
```

[134]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

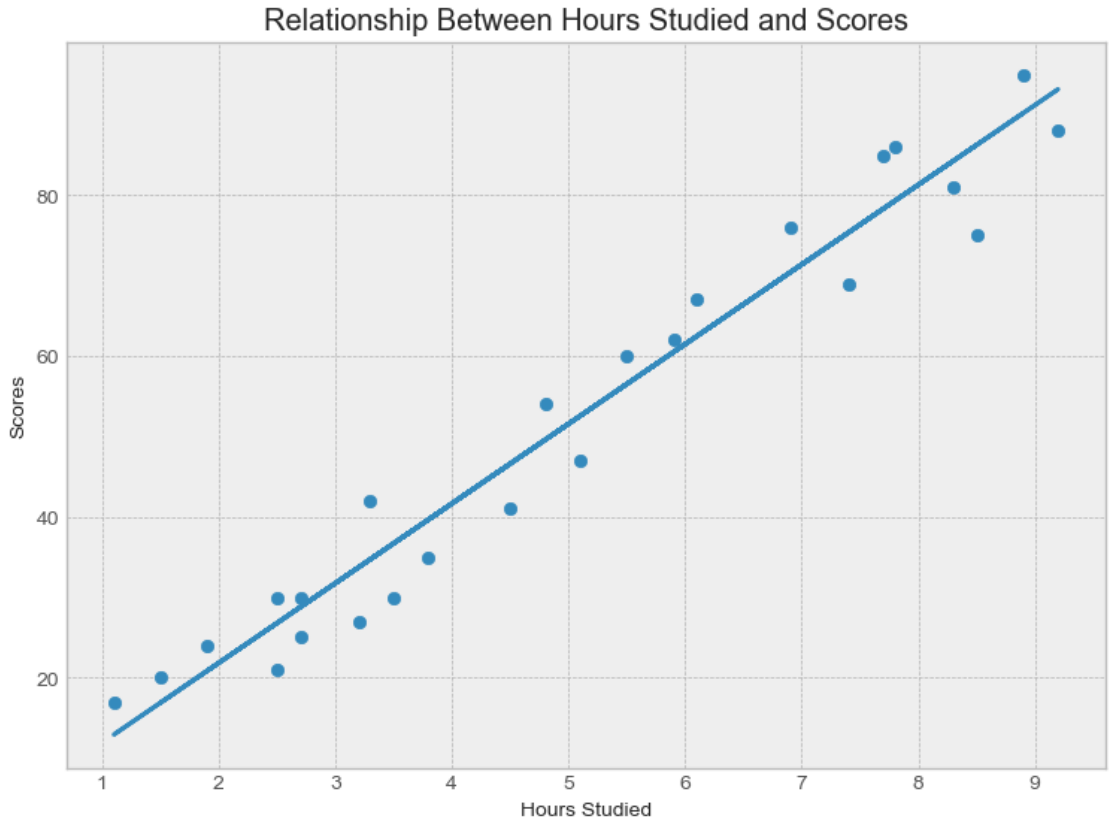
[134]:
```
LinearRegression()
```

[145]:
```
# Plotting the regression line
lr = regressor.coef_*X+regressor.intercept_

# Plotting for the test data
plt.figure(figsize=(8, 6))
plt.scatter(X, y)
plt.plot(X, lr);

# Add title and labels
plt.title('Relationship Between Hours Studied and Scores')
plt.xlabel('Hours Studied')
plt.ylabel('Scores')

plt.show()
```



[109]:
```
print(X_test) # Testing data - In Hours
y_pred = regressor.predict(X_test) # Predicting the scores
```

```
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]]
```

[110]:
```
# Comparing Actual vs Predicted
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

[110]:

	Actual	Predicted
0	20	16.884145
1	27	33.732261
2	69	75.357018
3	30	26.794801
4	62	60.491033

1.6 Evaluate the model

[111]:
```
from sklearn import metrics
#using R-squared(r2_score) as an metric to evaluate our model which returns a␣
 ↪value
#It ranges from 0 to 1, and a higher value indicates a better fit.
round(metrics.r2_score(y_test, y_pred),2)
```

[111]:
```
0.95
```

1.7 Save and Load the model

[112]:
```
import pickle


with open('student_score_prediction.pkl', 'wb') as f:
    pickle.dump(regressor, f)
```

[113]:
```
# Load the model
with open("student_score_prediction.pkl", "rb") as file:
    model = pickle.load(file)
```

[142]:
```
# Sample DataFrame with study hours
df = pd.DataFrame({
    'Hours': [1.5, 3.0, 4.5, 5.0, 6.5, 8.0, 9.25]
})

df['Predicted_Scores'] = model.predict(df[['Hours']])

# Display the DataFrame with predicted scores
print(df)
```

```
   Hours  Predicted_Scores
0   1.50         16.884145
1   3.00         31.750129
2   4.50         46.616114
3   5.00         51.571442
4   6.50         66.437427
5   8.00         81.303412
6   9.25         93.691732
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\base.py:413: UserWarning: X
has feature names, but LinearRegression was fitted without feature names
  warnings.warn(
```

So the score of the student that studied for 9.25 hours is 93.69