tsf

July 20, 2021

1 Author : Afraz Muneer

2 Task 1 : Prediction using Supervised Machine Learning

3 The Spark Foundation

This is a simple linear regression task which involves just two variables

4 Importing Libraries

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

5 Reading DataSet

[2]:
```
df = pd.read_csv("https://raw.githubusercontent.com/AdiPersonalWorks/Random/
↪master/student_scores%20-%20student_scores.csv")
```

checking head of dataset

[3]:
```
df.head()
```

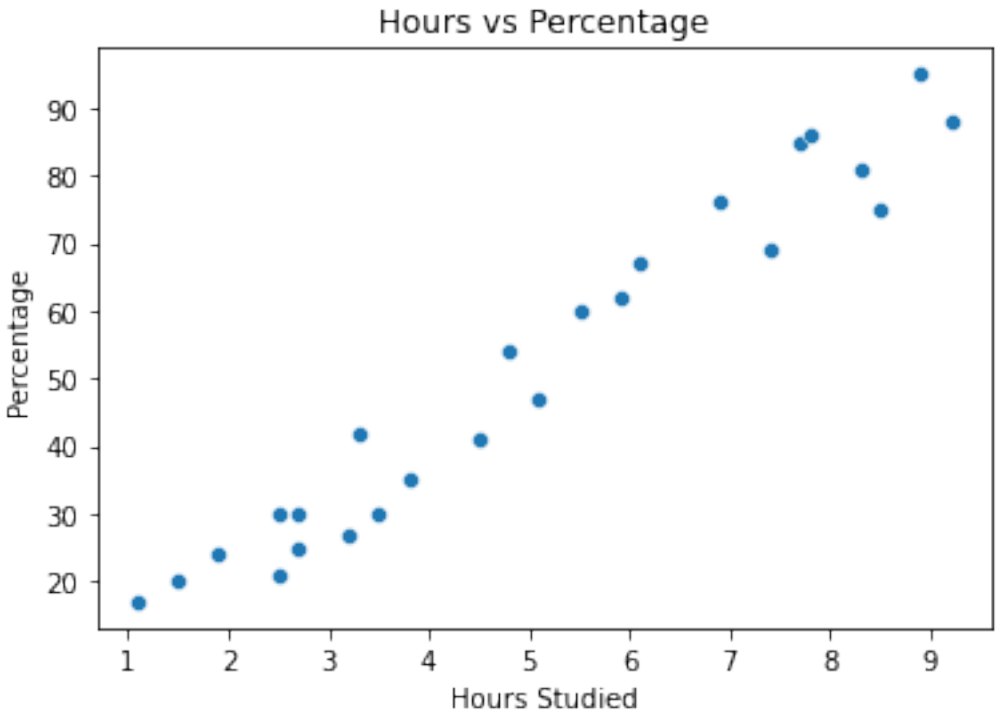
[3]:

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

6 Data Visualization

Scatter plot of Hours vs Percentage

[4]:
```
fig1 = sns.scatterplot(data=df,x="Hours", y="Scores")
fig1.set(xlabel='Hours Studied', ylabel='Percentage', title='Hours vs␣
 ↪Percentage')
plt.show()
```



[5]:
```
df.columns
```

[5]:
```
Index(['Hours', 'Scores'], dtype='object')
```

7 Data Preperation

Dividing attributes and labels

[6]:
```
X = df.iloc[:, :-1].values
y = df.iloc[:, 1].values
```

importing train_test_split method

[7]:
```
from sklearn.model_selection import train_test_split
```

splitting Testing and Training data

[8]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.
 ↪33,random_state=0)
```

8 Training Algorithm

Importing Linear Regression Module

[9]:
```
from sklearn.linear_model import LinearRegression
```

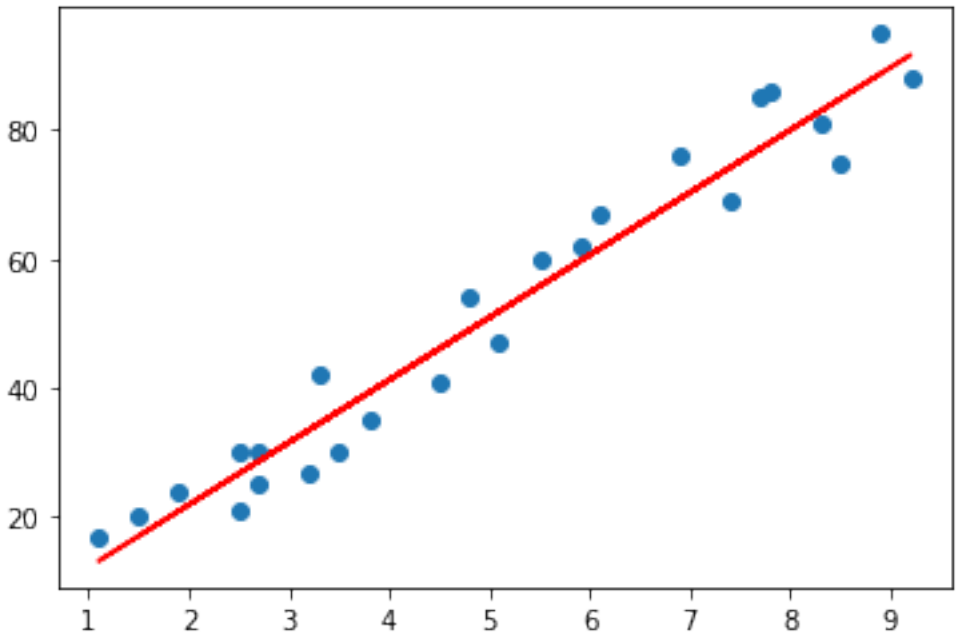
[10]:
```
model = LinearRegression()
```

[11]:
```
model.fit(X_train, y_train)
```

[11]:
```
LinearRegression()
```

[12]:
```
line = model.coef_ * X + model.intercept_
```

[13]:
```
plt.scatter(X, y)
plt.plot(X, line, color='red');
plt.show()
```



9 Making Predictions

[14]:
```
print(X_test)
y_pred = model.predict(X_test)
```

```
[[1.5]
 [3.2]
 [7.4]
```

```
[2.5]
[5.9]
[3.8]
[1.9]
[7.8]
[6.9]]
```

Comparing Actual vs Predicted

[15]:
```
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

[15]:

	Actual	Predicted
0	20	17.042892
1	27	33.516954
2	69	74.217577
3	30	26.733516
4	62	59.681640
5	35	39.331329
6	24	20.919142
7	86	78.093827
8	76	69.372265

Testing the model with our own data

[16]:
```
hours = float(input("Enter Hour of Studies :"))
test = np.array([hours])
test = test.reshape(-1, 1)
own_pred = model.predict(test)
print("No of Hours = {}".format(hours))
print("Predicted Score = {}".format(own_pred[0]))
```

```
Enter Hour of Studies :9.25
No of Hours = 9.25
Predicted Score = 92.14523314523314
```

10 Evaluating Model

[17]:
```
from sklearn import metrics
```

[18]:
```
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('Root Mean Squared Error:',
      np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
Mean Absolute Error: 4.691397441397438
Mean Squared Error: 25.463280738222547
Root Mean Squared Error: 5.046115410711743
```

11 Conclusion

I was successfully able to carry-out Prediction using Supervised ML task and was able to evaluate the model's performance on various parameters.

Thank you!

[ ]: