THE SPARKS FOUNDATION

GRIPMARCH22

TASK 1 - Prediction using Supervised ML

To Predict the percentage of marks of the students based on the number of hours they studied

AUTHOR: Shantanu Muppidwar

```
# importing the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
```

```
# Reading the Data
data = pd.read_csv('http://bit.ly/w-data')
data.head(5)
```

```
# Check if there any null value in the Dataset
data.isnull == True
```

```
     False
```

There is no null value in the Dataset so, we can now visualize our Data.

```
sns.set_style('darkgrid')
sns.scatterplot(y= data['Scores'], x= data['Hours'])
plt.title('Marks Vs Study Hours',size=20)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
```

From the above scatter plot there looks to be correlation between the 'Marks Percentage' and 'Hours Studied', Lets plot a regression line to confirm the correlation.

```
sns.regplot(x= data['Hours'], y= data['Scores'])
plt.title('Regression Plot',size=20)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
print(data.corr())
```

It is confirmed that the variables are positively correlated.

Training the Model

1) Splitting the Data

```
# Defining X and y from the Data
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values
# Spliting the Data in two
train_X, val_X, train_y, val_y = train_test_split(X, y, random_state = 0)
```

2) Fitting the Data into the model

```
regression = LinearRegression()
regression.fit(train_X, train_y)
print("---------Model Trained---------")
```

```
     ---------Model Trained---------
```

Predicting the Percentage of Marks

```
pred_y = regression.predict(val_X)
prediction = pd.DataFrame({'Hours': [i[0] for i in val_X], 'Predicted Marks': [k for k in pre
prediction
```

Comparing the Predicted Marks with the Actual Marks

```
compare_scores = pd.DataFrame({'Actual Marks': val_y, 'Predicted Marks': pred_y})
```

```
compare_scores
```

Visually Comparing the Predicted Marks with the Actual Marks

```
plt.scatter(x=val_X, y=val_y, color='blue')
plt.plot(val_X, pred_y, color='Black')
plt.title('Actual vs Predicted', size=20)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
```

Evaluating the Model

```
# Calculating the accuracy of the model
print('Mean absolute error: ',mean_absolute_error(val_y,pred_y))
```

```
     Mean absolute error:  4.130879918502482
```

Small value of Mean absolute error states that the chances of error or wrong forecasting through the model are very less.

What will be the predicted score of a student if he/she studies for 9.25 hrs/ day?

```
hours = [9.25]
answer = regression.predict([hours])
print("Score = {}".format(round(answer[0],3)))
```

```
     Score = 93.893
```

According to the regression model if a student studies for 9.25 hours a day he/she is likely to score 93.89 marks.

✓ 0s completed at 10:55 PM ● ×