Sygnific

Project 3

Perdition of Student Marks with Linear Regression

OVERVIEW

A project to understand and implement the concepts of Linear Regression that will outline how the regression concept works . the prediction will be determined on the number of hours a student will study and the scores he will receive accordingly.

Software Requirements

1. Programming Language : Python
2. Environemnt: Jupyter Notebooks / Google Collab
3. Database: CSV(export type)
4. Operation System: Windows XP or above
5. Librarires Used: Pandas,Folium, Seaborn, Scikit, SKLEARN
6.Datasets used: Student Dataset

1. Open a New Notebook and import the required libraires and read the csv file

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from sklearn.model_selection import train_test_split
```

Description : Here we are importing all the required libraries for our Project. Pandas is used for data manipulation and analysis. We are using NumPy library for scientific computing applications and matplotlib, seaborn for data visualization. Train_test_split to Split the data. All of the statistics functions are located in the package scipy.stats

2. Importing the Student Dataset

```
df = pd.read_csv('/data.csv')
```

Description : With the help of pandas read_csv method we are Reading the data and storing the data in dataframe df.

3. Viewing and Exploring the Data

```
4. print("Now our data is loaded")
5. df
```

Output:

0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17
15	8.9	95
16	2.5	30
17	1.9	24
18	6.1	67
19	7.4	69
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

6.

Description : Here we find that our data has two columns named hours and scores.

```
df.shape
```

```
(25, 2)
```

```
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
```

Description :with .info() we get the basic info of our data like name of columns, no of rows, data types and memory usage.

```
df.describe()
```

	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

```
df.corr()
```

Out[23]:

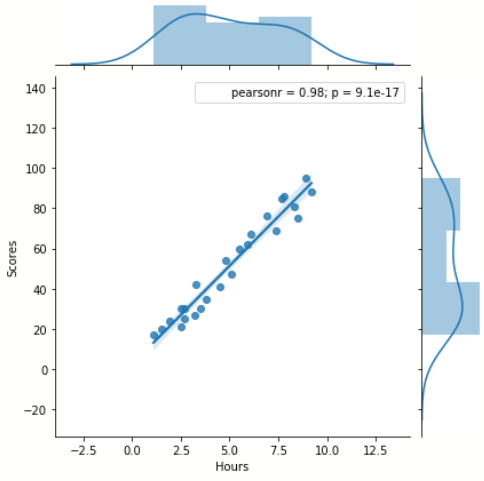
	Hours	Scores
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False
6	False	False
7	False	False
8	False	False
9	False	False
10	False	False
11	False	False
12	False	False
13	False	False
14	False	False
15	False	False
16	False	False
17	False	False
18	False	False
19	False	False
20	False	False
21	False	False
22	False	False
23	False	False
24	False	False

Description : df.describe() will give us certain values in return. It gives us count of data entries, mean of a data, standard deviation of data, min value, max value etc. Next we are checking for null values in our data. But we dont have any null values so it retruns false.

7. Visualizing the Linear Relation between Hours & Scores (Drawing a joint Plot

```
sns.jointplot(df['Hours'], df['Scores'], kind = "reg").annotate(stats.pearsonr) plt.show()
```

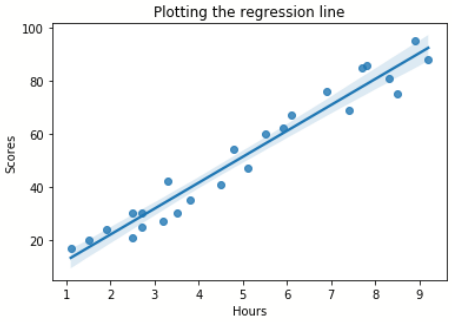
8.



Description : With the help of this graph we can conclude that there is a linear relation between no of hours studied and marks obtained.

9. Visualizing the Correlation

```
sns.regplot(x="Hours", y="Scores", data=df) plt.title("Plotting the regression line")
```



Description : here we are usng regplot of seaborn library and ploting a graph for hours vs scores. We find that all the points are close to the line of regression showing that it is a strong linear regression.

Using Simple linear regression to predict the data as we only have two columns.

Dividing Our Dataset into training and testing

```
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 0)

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
Out[27]:
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Description :Here we are stroing score and hours columns in X and result in y. then we are using train_test_split to divide the data in 70:30 ratio. 70% of data is used to train the model and 30% is used to test the model. Next we are importing the linearRegression and then assigning to variable regressor. Now this regressor has all the functions and methods of this linearregression. With .fit() method our model wil get trained on training data.

After Training now performing Prediction

```
y_pred = regressor.predict(X_test)

y_pred
```

```
Out[28]:
```

```
array([17.05366541, 33.69422878, 74.80620886, 26.8422321 , 60.12335883,
       39.56736879, 20.96909209, 78.72163554])
```

Description : .predict()is used to predict the dependent feature.and we are stroing those predicted values in y_pred.

Comparing Actual vs Predicted Value

```
df1 = pd.DataFrame({'Actual': y_test, 'Predicted_Score': y_pred})
df1
```

	Actual	Predicted_Score
0	20	17.053665
1	27	33.694229
2	69	74.806209
3	30	26.842232
4	62	60.123359
5	35	39.567369
6	24	20.969092
7	86	78.721636

Description : Here we are comparing the actual test values and predicted values. We find that our model predicted close to the orginal values.

Conclusion

In this Project we are able to predict the student marks using linear regression. In this process we did some data collection, data visualization. Then we splited our data into training data and testing data. We trained our model on traing data and tested our models performance with Testing data. Finally we were able to predict the students score with this linear regression model.