The Sparks Foundation GRIP

Author - AYUSH CHHOKER

DATA SCIENCE AND BUSINESS ANALYTICS INTERN

TASK -1 - Prediction using Supervised ML

Predict the percentage of an student based on the no. of study hours.

IMPORT LIBRARIES

In [57]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import statsmodels.formula.api as smf
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
```

Importing Data set

In [58]:

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

In [59]:

```
df.head()
```

Out[59]:

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

EXPLORATORY DATA ANALYSIS

In [60]:

```
df.columns
```

Out[60]:

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

In [61]:

```
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
```

In [62]:

```
df.dtypes
```

Out[62]:

```
Hours     float64
Scores      int64
dtype: object
```

In [63]:

```
df.isnull().sum()
```

Out[63]:

```
Hours     0
Scores    0
dtype: int64
```

In [64]:

```
df.describe()
```

Out[64]:

	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

In [65]:

```
df.corr()
```

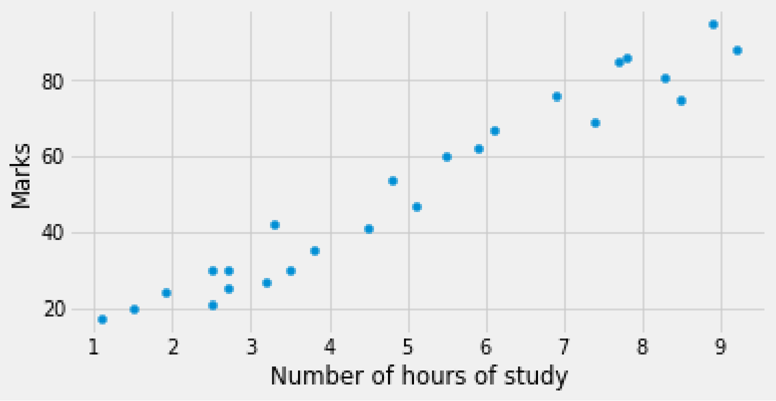
Out[65]:

	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

Simple Analysis

In [66]:

```
# data analysis
plt.style.use('fivethirtyeight')
plt.figure(figsize=(8,4),dpi=50)
plt.scatter(df["Hours"],df['Scores'])
plt.xlabel('Number of hours of study')
plt.ylabel('Marks')
plt.show()
```



We can see that the number of hours of study is highly correlated with the marks of the student

Data Pre-Processing

Splitting the data into training and test set so as to see if the model fits well on the general data

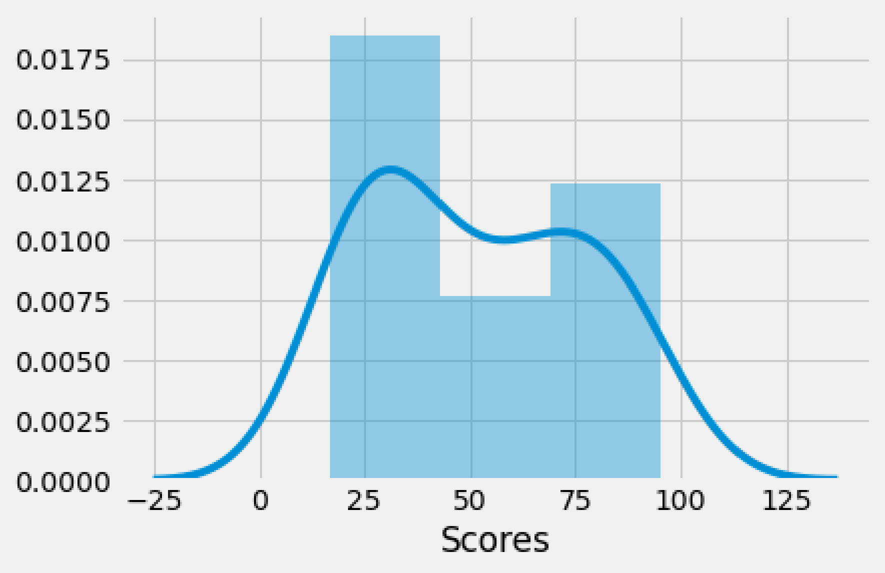
In [67]:

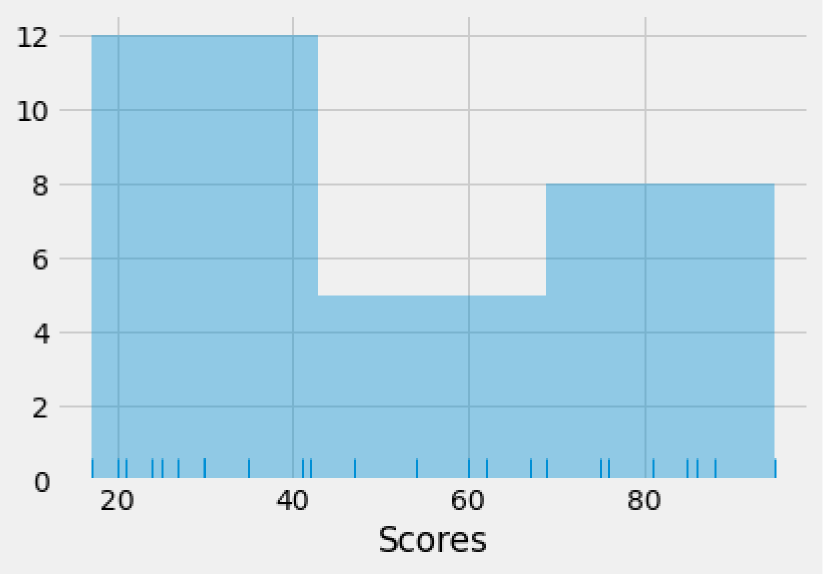
```
# converting to numpy
x = np.array(df['Hours']).reshape(-1,1)
y = np.array(df['Scores'])

# train test split
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25)
```

Distribution

In [68]:

```
sns.distplot(df["Scores"])
plt.show()
sns.distplot(df["Scores"], kde=False, rug=True)
plt.show()
```

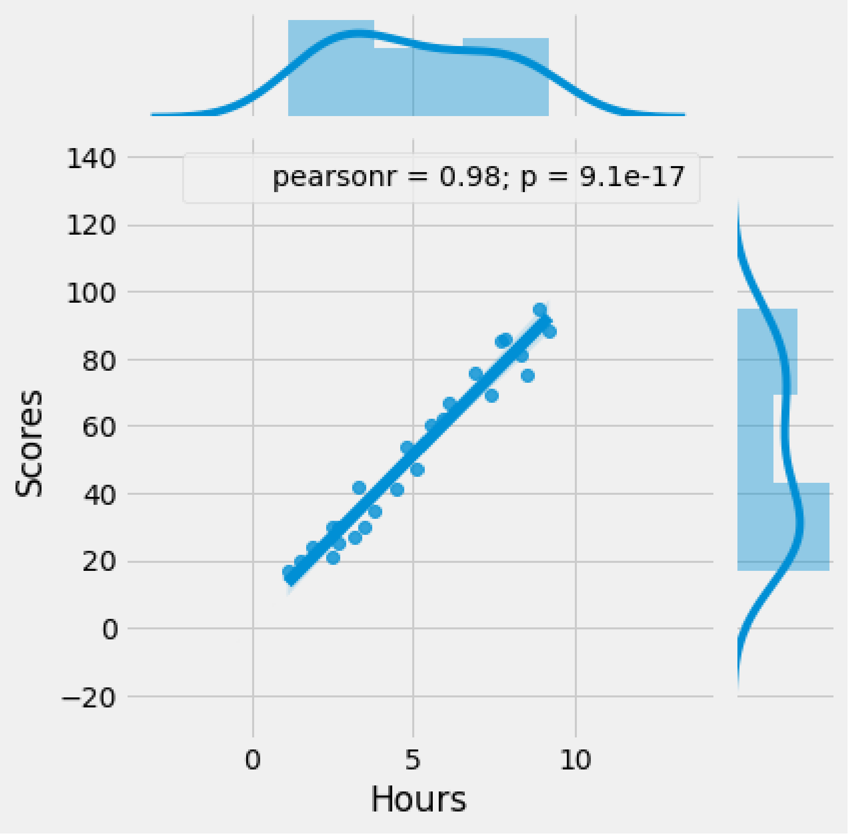




In [69]:

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

```
C:\Users\APC\anaconda3\lib\site-packages\seaborn\axisgrid.py:1840: UserWar
ning: JointGrid annotation is deprecated and will be removed in a future r
elease.
  warnings.warn(UserWarning(msg))
```

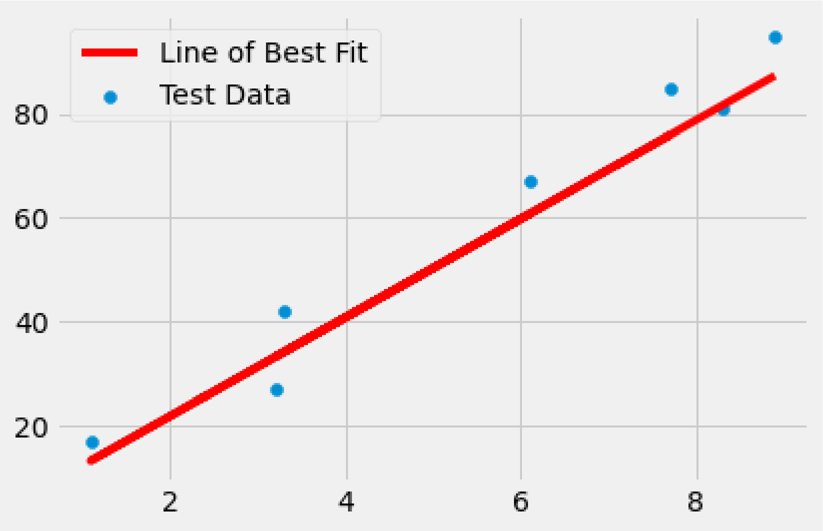


Linear Regression

Linear regression is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables.

In [70]:

```
# linear regression
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

lr = LinearRegression()
lr.fit(x_train,y_train)

# predictions
y_pred = lr.predict(x_test)

# making plot
plt.scatter(x_test,y_test,label="Test Data")
plt.plot(x_test,y_pred,color='red',label='Line of Best Fit')
plt.legend()
plt.show()

# mean squarred error
print(f"mean squared error : {mean_squared_error(y_test,y_pred)}")
```



```
mean squared error : 42.4500210876304
```

we can see that the line fits the testing data decently well. So, we can use this model to predict the scores of the new studets given the amount of time they studied

Plotting regression line

In [71]:

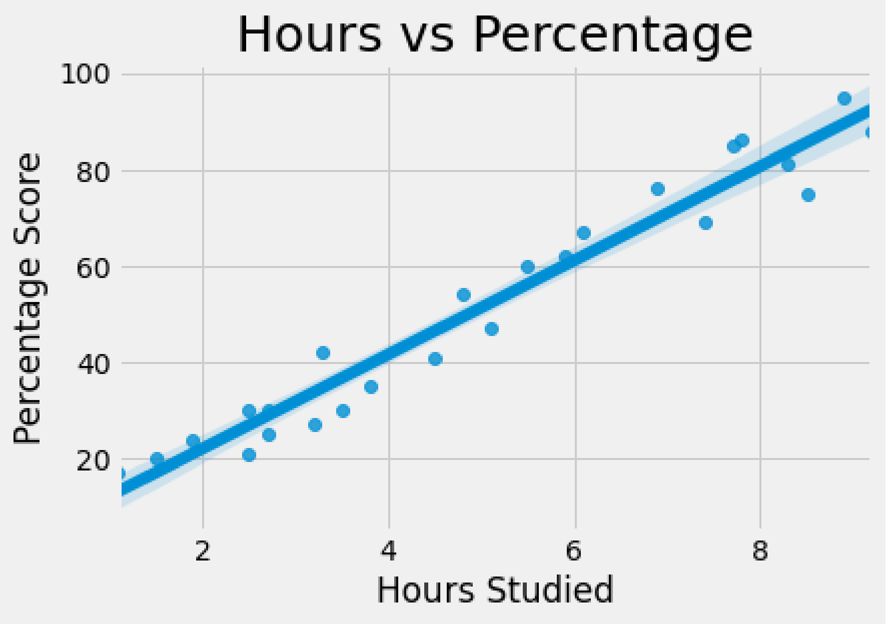
```
df.head
```

Out[71]:

```
<bound method NDFrame.head of     Hours  Scores
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>
```

In [72]:

```
ax = sns.regplot(x="Hours", y="Scores", data =df)
plt.title('Hours vs Percentage', fontsize=25)
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



Training the data

In [73]:

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

In [74]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0
)
```

In [75]:

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

print("Training complete.")
```

```
Training complete.
```

In [76]:

```
print(X_test)
y_pred = regressor.predict(X_test)
```

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

Making prediction

In [77]:

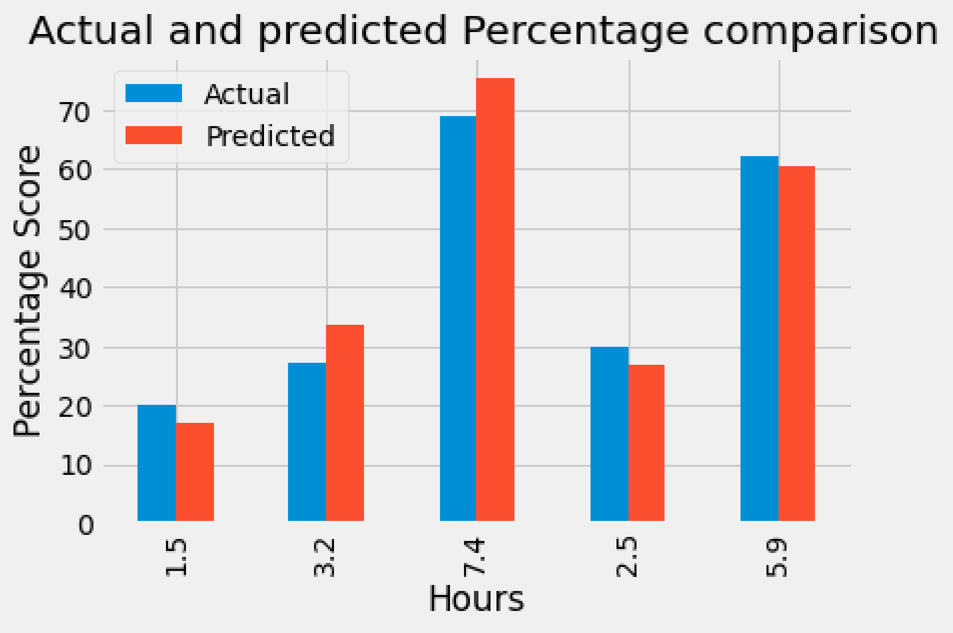
```
df1 = pd.DataFrame({'Hours':[1.5,3.2,7.4,2.5,5.9], 'Actual': y_test, 'Predicted': y_pre
d})
df1
```

Out[77]:

	Hours	Actual	Predicted
0	1.5	20	16.884145
1	3.2	27	33.732261
2	7.4	69	75.357018
3	2.5	30	26.794801
4	5.9	62	60.491033

In [78]:

```
df1.plot(x= "Hours", y=["Actual", "Predicted"], kind="bar")
plt.grid(linewidth='1')
plt.title(" Actual and predicted Percentage comparison")
plt.ylabel('Percentage Score')
plt.show()
```



What will be predicted score if a student studies for 9.25 hrs/ day?

In [79]:

```
hour = 9.25
own_pred = regressor.predict([[hour]])
print("No of Hours = {}".format(hour))
print("Predicted Score = {}".format(own_pred[0]))
```

```
No of Hours = 9.25
Predicted Score = 93.69173248737539
```

Predictions on User Input

In [80]:

```
# real time prediction
hours = float(input("Enter the number of hours : "))
print(f"the student is likely to score {(lr.predict([[hours]])[0]):.2f} marks")
```

```
Enter the number of hours : 11
the student is likely to score 107.25 marks
```

In [ ]: