GRIP : THE SPARKS FOUNDATION

Data Science and Business Analytics

Task 1 : prediction using supervised ML

In this task i will predict the percentage score of a student based on the number of hours studied. In this task, i have used two variables where the feature is the no. of hours studied and target value is the percentage score. This can be achieved with the help of Linear Regression.

importing required libraries

In [46]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [47]:
```
# reading the data
data_link = "stud.csv"
data = pd.read_csv(data_link)
data.head(27)
```

Out[47]:

	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 [48]:
```
data.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
```

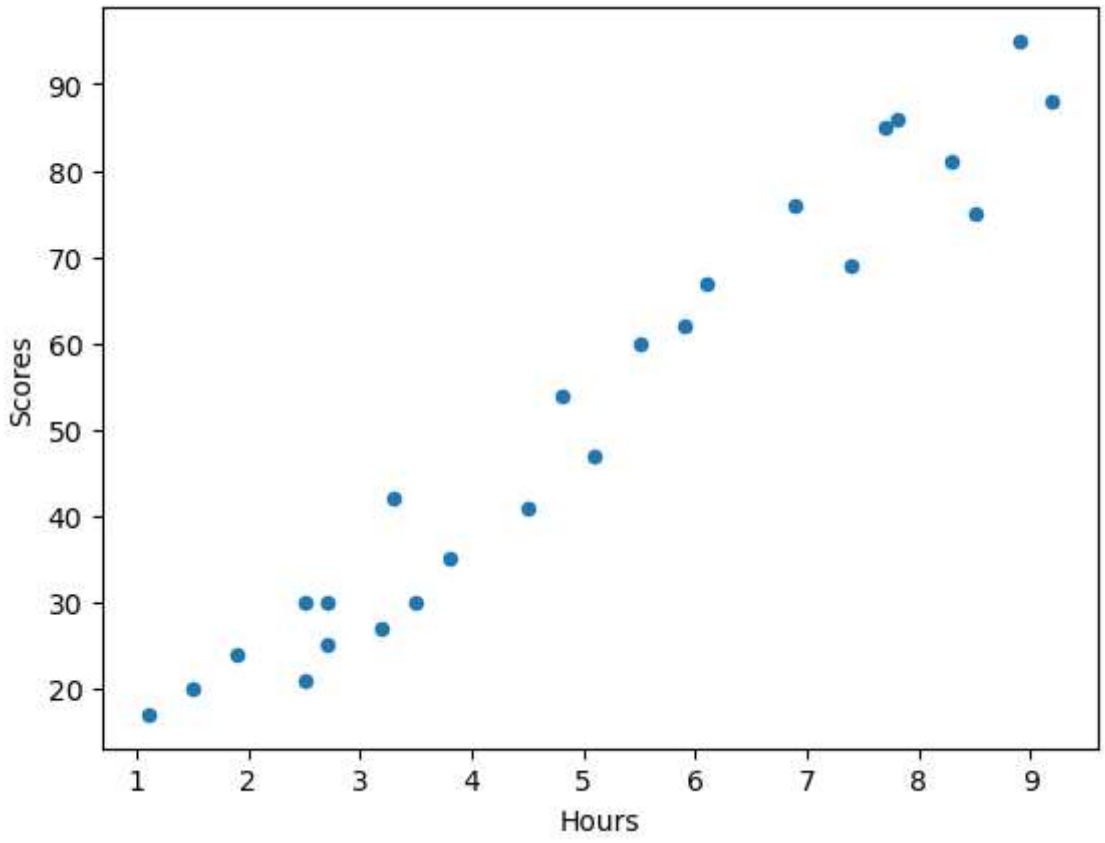
In [49]:
```
data.describe()
```

Out[49]:

	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 [12]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [50]:
```
data.plot(kind= 'scatter', x="Hours", y="Scores");
plt.show()
```



In [15]:
```
data.corr(method= "pearson")
```

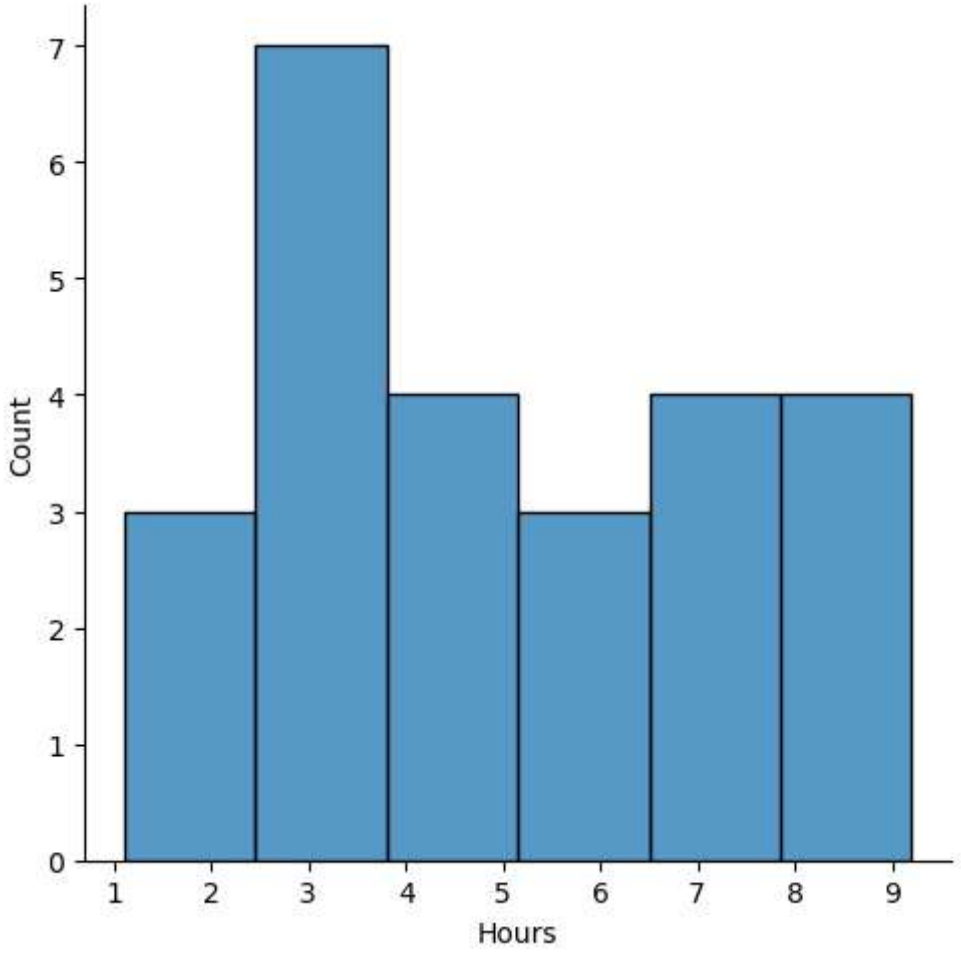
Out[15]:

	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

In [51]:
```
hours =data["Hours"]
scores = data["Scores"]
```

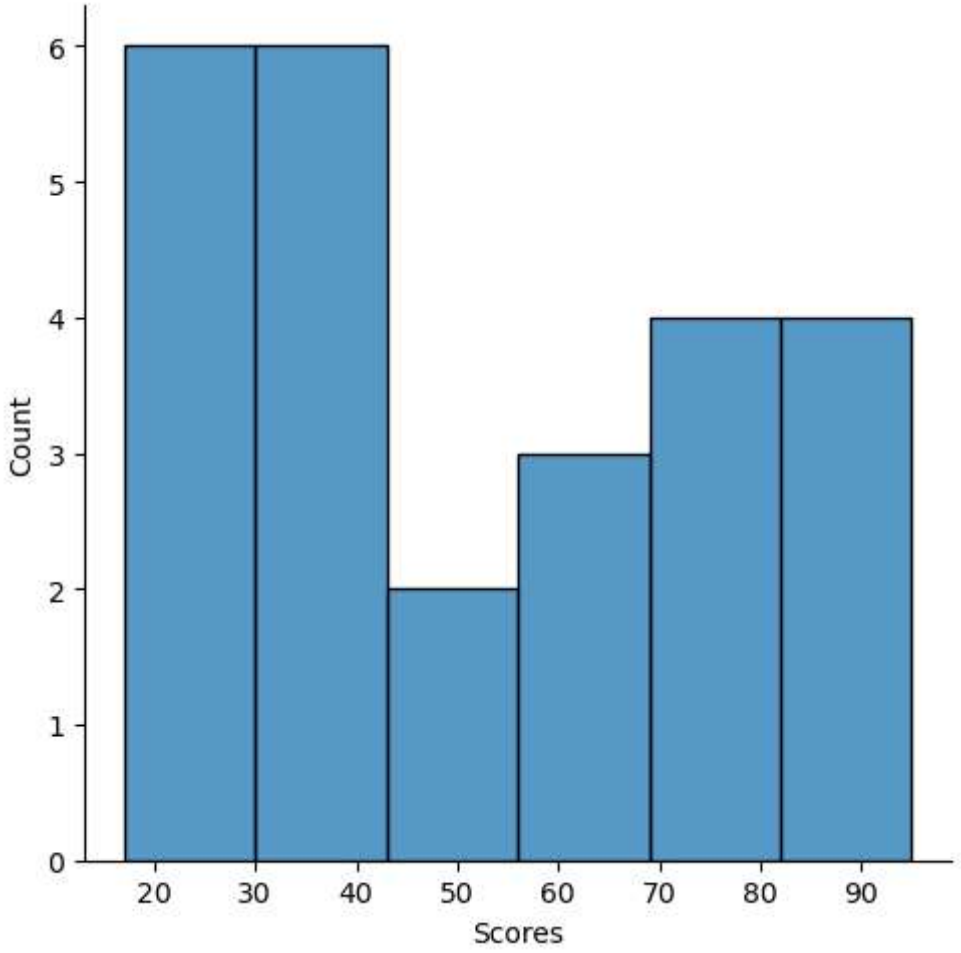
In [52]:
```
sns.displot(hours)
```

Out[52]:
```
<seaborn.axisgrid.FacetGrid at 0x1d90b97ad90>
```



In [53]:
```
sns.displot(scores)
```

Out[53]:
```
<seaborn.axisgrid.FacetGrid at 0x1d90cfe88e0>
```



Linear Regression

In [58]:
```
a= data.iloc[:,:-1].values
b= data.iloc[:,1].values
```

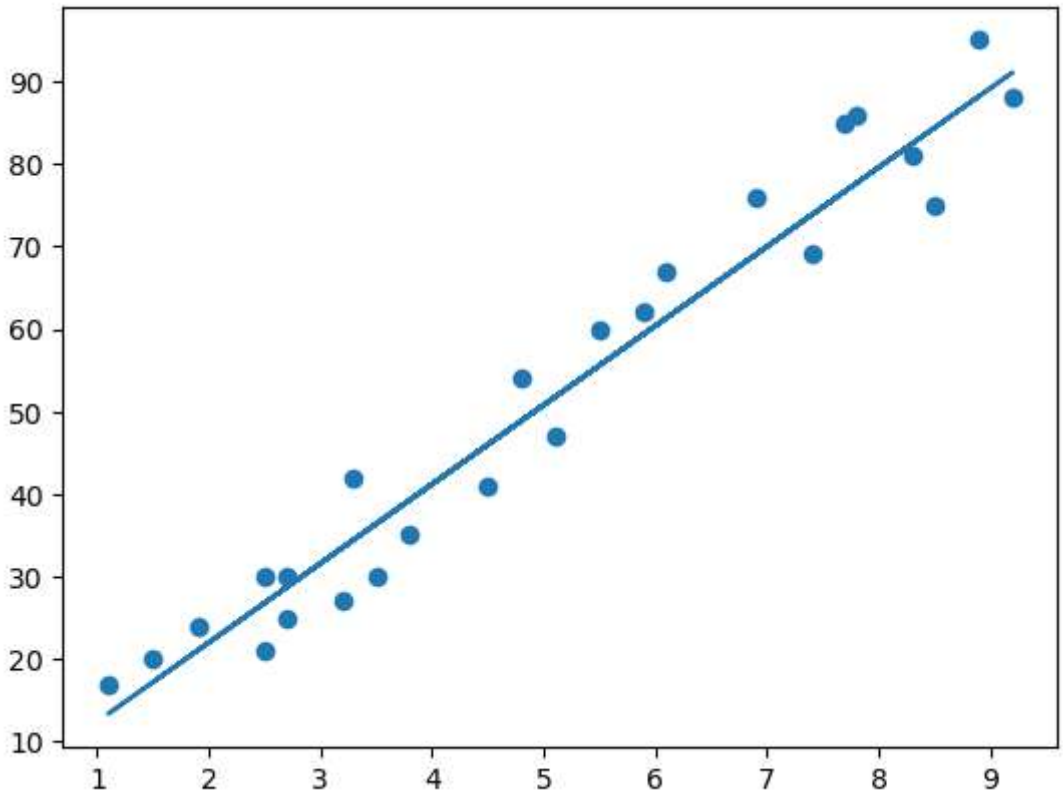
In [76]:
```
from sklearn.model_selection import train_test_split
a_train,a_test,b_train,b_test= train_test_split(a,b, test_size=0.2, random_state=50)
```

In [60]:
```
from sklearn.linear_model import LinearRegression
reg= LinearRegression()
reg.fit(a_train,b_train)
```

Out[60]:
```
LinearRegression()
```

In [67]:
```
m= reg.coef_           #coef_ and intercept_ are attributes of LinearRegression
c= reg.intercept_
line= m*a+c

plt.scatter (a,b)
plt.plot(a,line)
plt.show()
```



In [68]:
```
b_pred = reg.predict(a_test)
```

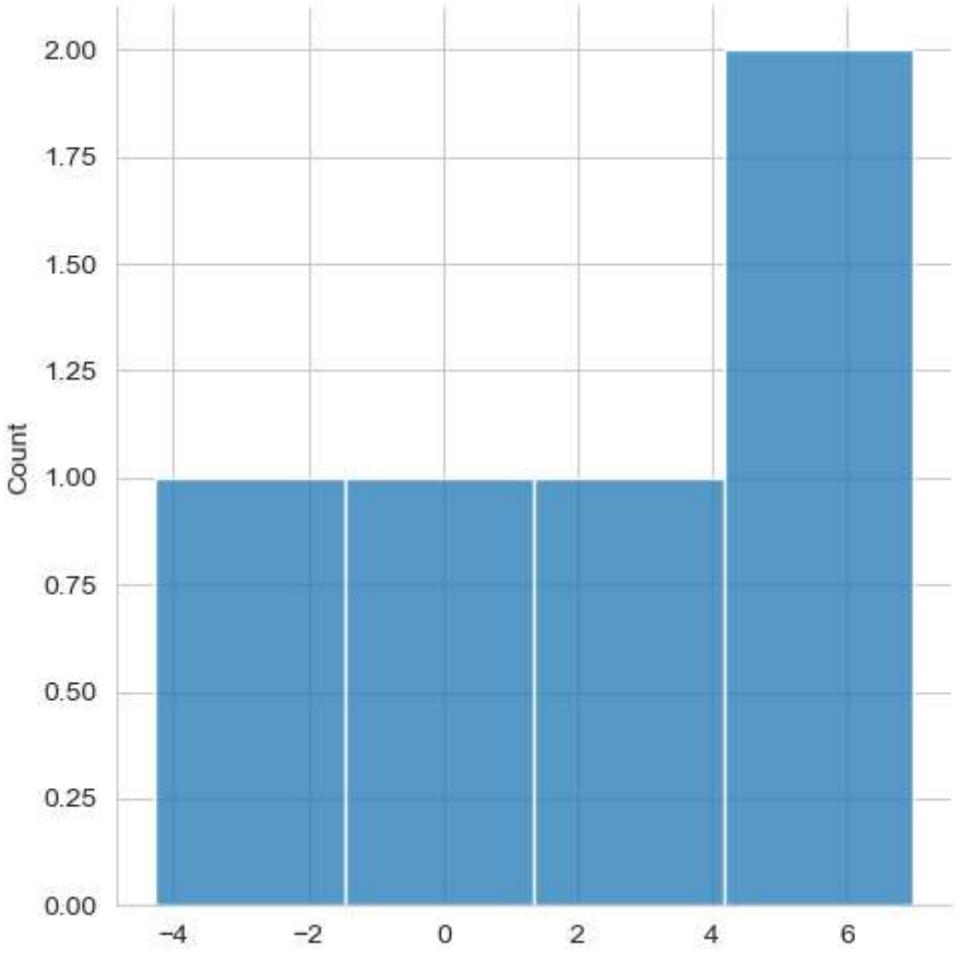
In [84]:
```
actual_pred = pd.DataFrame({"target": b_test, "predicted":b_pred})
actual_pred
```

Out[84]:

	target	predicted
0	95	88.211394
1	30	28.718453
2	76	69.020122
3	35	39.273652
4	17	13.365436

In [85]:
```
sns.set_style("whitegrid")
sns.displot(np.array(b_test - b_pred))
```

Out[85]:
```
<seaborn.axisgrid.FacetGrid at 0x1d90e0ce9d0>
```



what would be the predicted score of a student if he/she studies for 9.25 hours/day?

In [90]:
```
h = 9.25
s = reg.predict([[h]])
print("if a student studies {} hours a day then he/she will score {} % in exam".format
```

```
if a student studies 9.25 hours a day then he/she will score [91.56986604] % in exam
```