Grip : The Spark Foundation

Data Science & Business Analytics Intern

Author : Chirag Kalgude

Task 1 : Prediction Using Supervised Learning

In [6]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Reading Data

In [7]:

```
url="http://bit.ly/w-data"
data = pd.read_csv(url)
```

Exploring Data

In [8]:

```
print(data.shape)
data.head()

```

```
(25, 2)
```

Out[8]:

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

In [9]:

```
1 data.describe()
```

Out[9]:

	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

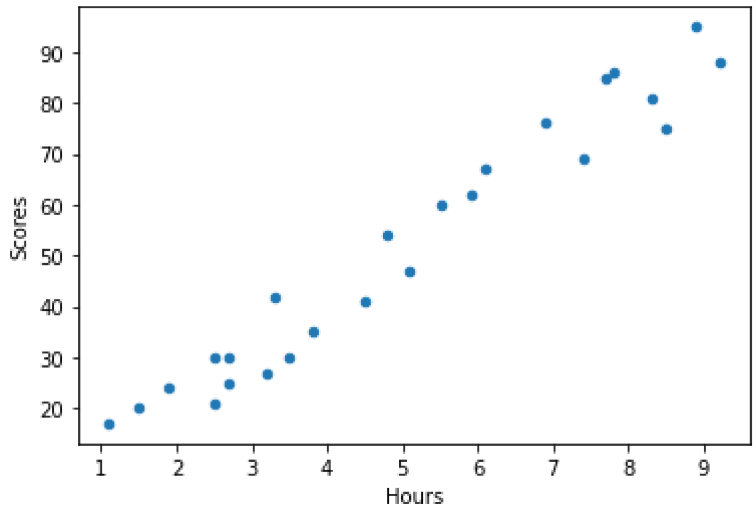
In [10]:

```
1 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 [11]:

```
1 data.plot(kind='scatter',x='Hours',y='Scores');
2 plt.show()
```



In [12]:

```
data.corr(method='pearson')
```

Out[12]:

	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

In [13]:

```
data.corr(method='spearman')
```

Out[13]:

	Hours	Scores
Hours	1.000000	0.971891
Scores	0.971891	1.000000

In [14]:

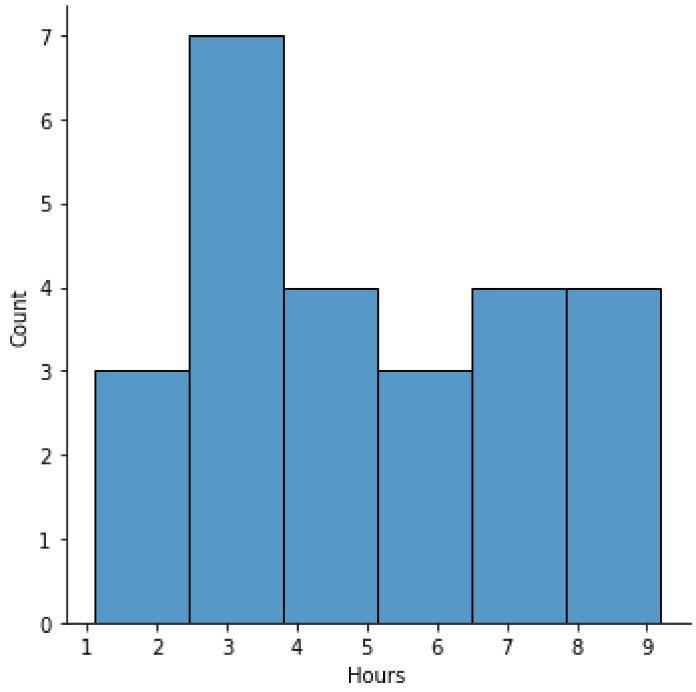
```
hours=data['Hours']
scores=data['Scores']
```

In [15]:

```
1 sns.displot(hours)
2
```

Out[15]:

```
<seaborn.axisgrid.FacetGrid at 0x1ac6b406d30>
```

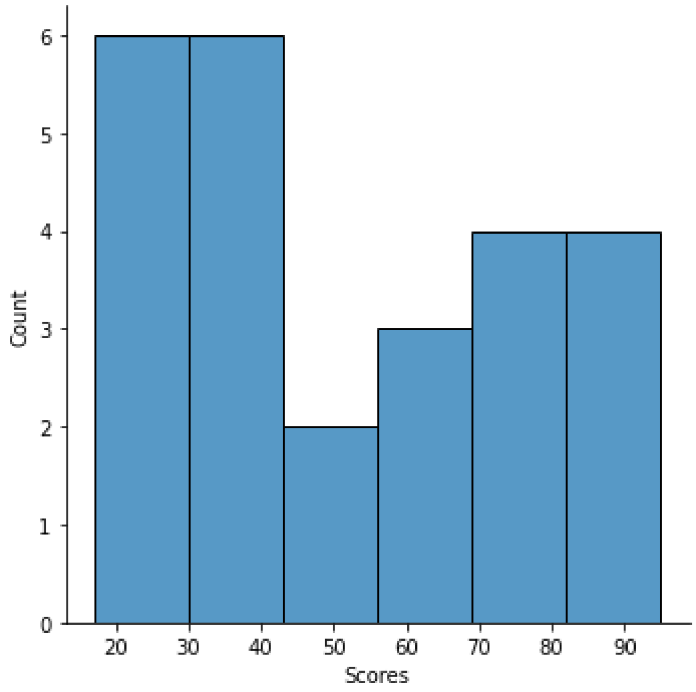


In [16]:

```
1 sns.displot(scores)
```

Out[16]:

```
<seaborn.axisgrid.FacetGrid at 0x1ac6b8f1070>
```



Linear Regression

In [17]:

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

In [18]:

```
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
```

In [23]:

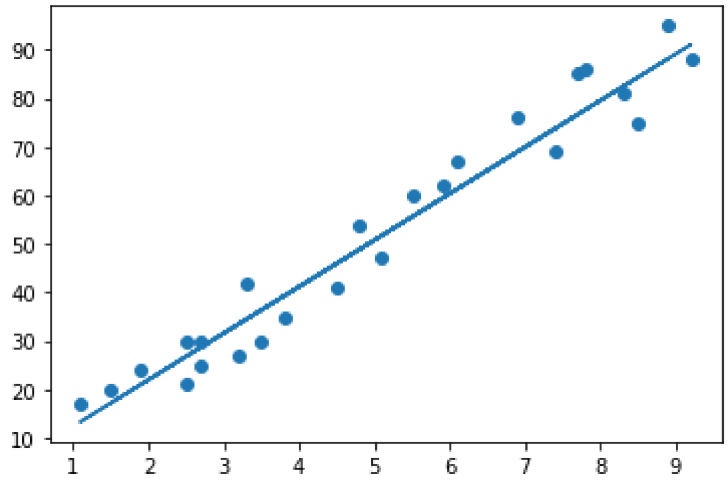
```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
reg.fit(X_train , y_train)
```

Out[23]:

```
LinearRegression()
```

In [24]:

```
m=reg.coef_
c = reg.intercept_
line=m*X+c
plt.scatter(X,y)
plt.plot(X,line);
plt.show()
```



In [27]:

```
y_pred =reg.predict(X_test)
```

In [28]:

```
actual_predicted = pd.DataFrame({'Target':y_test , 'predicted':y_pred})
actual_predicted
```

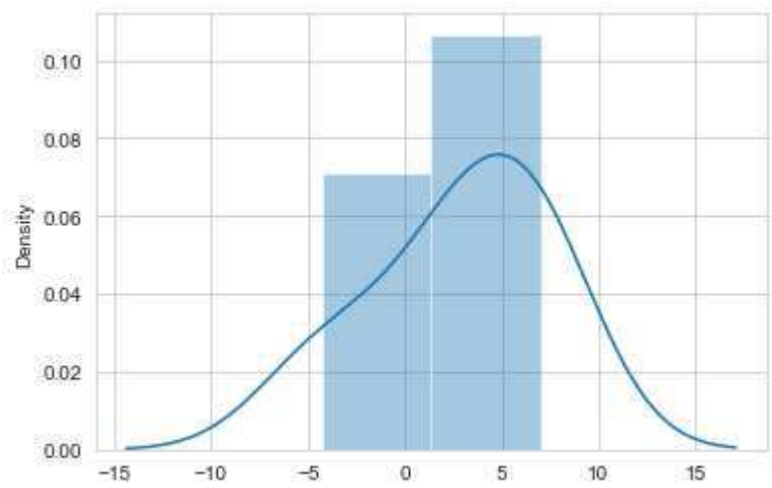
Out[28]:

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

In [29]:

```
sns.set_style('whitegrid')
sns.distplot(np.array(y_test-y_pred))
plt.show()
```

```
C:\Users\chira\OneDrive\Documents\Custom Office Templates\lib\site-packages
\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to us
e either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [32]:

```
h = 9.25
s = reg.predict([[h]])
print("If a student for {} hours per day he/she will score {} % in exam .".format (h,s)
```

```
If a student for 9.25 hours per day he/she will score [91.56986604] % in exa
m .
```

Model Evaluation

In [34]:

```
from sklearn import metrics
from sklearn.metrics import r2_score
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,y_pred))
print('R2 Score:',r2_score(y_test,y_pred))
```

```
Mean Absolute Error:  4.5916495300630285
R2 Score: 0.971014141329942
```

In [ ]:

```

```

In [35]:

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

In [ ]:

```

```