In [1]:

```
#Expt-7
```

In [ ]:

```
# Name: Gauri Santosh Ingale
# Roll no: 20
# Sec: CSE-Ds
# Subject: ET 1
```

In [ ]:

```
import pandas as pd
```

In [2]:

```
import os
```

In [3]:

```
os.getcwd()
```

Out[3]:

```
'C:\\Users\\ASUS'
```

In [4]:

```
os.chdir('C:\\Users\\ASUS\\Desktop')
```

In [5]:

```
data=pd.read_csv("salary_data.csv")
```

In [6]:

```
data.head(30)
```

Out[6]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

In [7]:

```python
data.info
```

Out[7]:

```
<bound method DataFrame.info of     YearsExperience    Salary
0               1.1   39343.0
1               1.3   46205.0
2               1.5   37731.0
3               2.0   43525.0
4               2.2   39891.0
5               2.9   56642.0
6               3.0   60150.0
7               3.2   54445.0
8               3.2   64445.0
9               3.7   57189.0
10              3.9   63218.0
11              4.0   55794.0
12              4.0   56957.0
13              4.1   57081.0
14              4.5   61111.0
15              4.9   67938.0
16              5.1   66029.0
17              5.3   83088.0
18              5.9   81363.0
19              6.0   93940.0
20              6.8   91738.0
21              7.1   98273.0
22              7.9  101302.0
23              8.2  113812.0
24              8.7  109431.0
25              9.0  105582.0
26              9.5  116969.0
27              9.6  112635.0
28             10.3  122391.0
29             10.5  121872.0>
```

In [8]:

```python
data.shape
```

Out[8]:

```
(30, 2)
```

In [9]:

```python
data.size
```

Out[9]:

```
60
```

In [10]:

```python
data.ndim
```

Out[10]:

```
2
```

In [11]:

```python
data.columns
```

Out[11]:

```
Index(['YearsExperience', 'Salary'], dtype='object')
```

In [12]:

```python
data.describe()
```

Out[12]:

	YearsExperience	Salary
count	30.000000	30.000000
mean	5.313333	76003.000000
std	2.837888	27414.429785
min	1.100000	37731.000000
25%	3.200000	56720.750000
50%	4.700000	65237.000000
75%	7.700000	100544.750000
max	10.500000	122391.000000

In [13]:

```python
data.isna()
```

Out[13]:

	YearsExperience	Salary
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
25	False	False
26	False	False
27	False	False
28	False	False
29	False	False

In [14]:

```
x=data.drop('Salary',axis=1)
```

In [15]:

```
x.head()
```

Out[15]:

	YearsExperience
0	1.1
1	1.3
2	1.5
3	2.0
4	2.2

In [16]:

```
y=data.Salary
```

In [17]:

```
y.head
```

Out[17]:

```
<bound method NDFrame.head of 0      39343.0
1      46205.0
2      37731.0
3      43525.0
4      39891.0
5      56642.0
6      60150.0
7      54445.0
8      64445.0
```

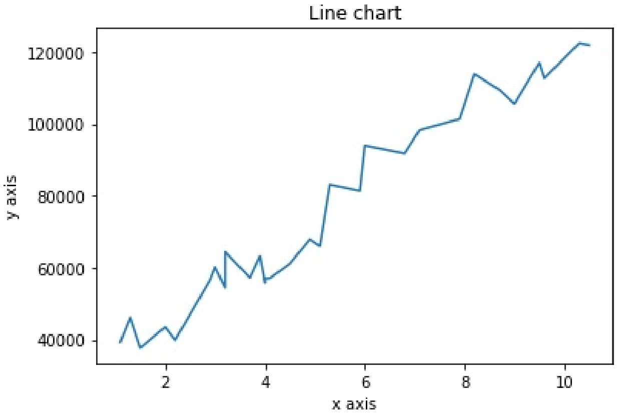
```
9      57189.0
10     63218.0
11     55794.0
12     56957.0
13     57081.0
14     61111.0
15     67938.0
16     66029.0
17     83088.0
18     81363.0
19     93940.0
20     91738.0
21     98273.0
22    101302.0
23    113812.0
24    109431.0
25    105582.0
26    116969.0
27    112635.0
28    122391.0
29    121872.0
Name: Salary, dtype: float64>
```

In [18]:

```
#importing the basic library
import numpy as np
from matplotlib import pyplot as plt
```

In [33]:

```
plt.plot(x,y)
plt.title("Line chart")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.show()
```



In [34]:

```
from sklearn.model_selection import train_test_split
```

In [35]:

```
x_train,x_test,y_test,y_train=train_test_split(x,y,test_size=.3,random_state=50)
```

In [36]:

```
x_train,x_test,y_test,y_train=train_test_split(x,y,random_state=0,test_size=0.30)
```

In [37]:

```
print(x_train.shape)
```

```
(21, 1)
```

In [38]:

```
x_test.shape
```

Out[38]:

```
(9, 1)
```

In [49]:

```
y_test.shape
```

Out[49]:

```
(6,)
```

In [50]:

```
y_train.shape
```

Out[50]:

```
(24,)
```

Model Fitting

In [40]:

```
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=42)
```

In [41]:

```
x_train=np.array(x_train).reshape(-1,1)
x_test = np.array(x_test).reshape(-1,1)
```

In [56]:

```
LR = LinearRegression()
```

In [57]:

```
LR.fit(x_train,y_train)
```

Out[57]:

```
LinearRegression()
```

In [58]:

```
m = LR.coef_
```

In [59]:

```
print("Coefficient  :" , m)
```

```
Coefficient  : [9423.81532303]
```

In [60]:

```
#Assigning Y-intercept to a
c = LR.intercept_
```

In [61]:

```
print("Intercept : ", c)
```

```
Intercept :  25321.583011776813
```

y=mx+c

prediction

In [63]:

```
y_pred=LR.predict(x_test)
```

In [64]:

```
y_pred
```

Out[64]:

```
array([115790.21011287,  71498.27809463, 102596.86866063,  75267.80422384,
        55477.79204548,  60189.69970699])
```

In [65]:

```
y_test
```

Out[65]:

```
27    112635.0
15     67938.0
23    113812.0
17     83088.0
8      64445.0
9      57189.0
Name: Salary, dtype: float64
```

Evaluation matrices

In [66]:

```
from sklearn import metrics
```

In [68]:

```
Accuracy = LR.score(x_test, y_test)
Accuracy
```

Out[68]:

```
0.9024461774180497
```

MAE

In [69]:

```
print(metrics.mean_absolute_error(y_test,y_pred))
```

```
6286.453830757749
```

MSE

In [70]:

```
print(np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
```

```
7059.04362190151
```

RMSE

In [71]:

```
rmse = np.sqrt(metrics.mean_squared_error(y_test,y_pred))
print("RMSE:", rmse)
```

```
RMSE: 7059.04362190151
```

In [ ]:

```

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js