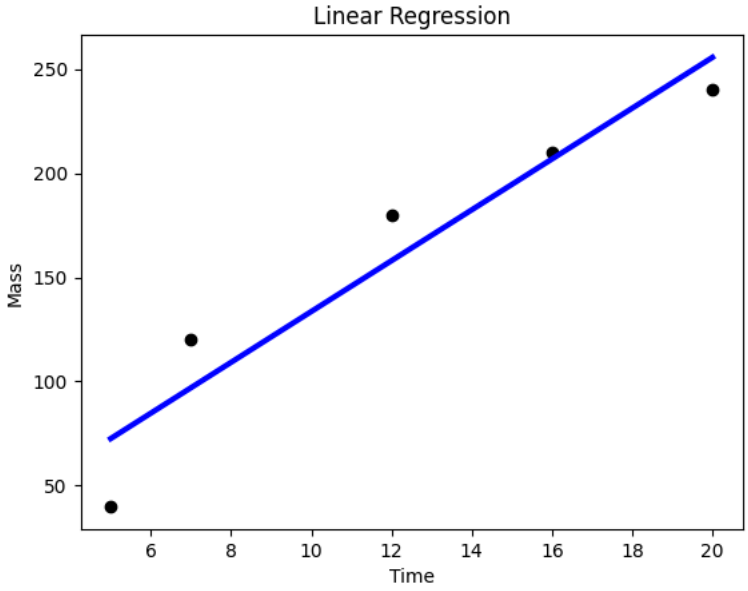
```
from sklearn.linear_model import LinearRegression
LR=LinearRegression()
t=[[5],[7],[12],[16],[20]]
m=[40,120,180,210,240]
LR.fit(t,m)
print(LR.predict([[5.5]]))
print(LR.predict([[25]]))
```

```
     [78.64935065]
     [316.7012987]
```

```
import matplotlib.pyplot as plt
plt.scatter(t,m,color='black')
y_pred=LR.predict(t)
plt.plot(t,y_pred,color="blue",linewidth=3)
plt.xlabel("Time")
plt.ylabel("Mass")
plt.title("Linear Regression")
plt.show()
```




```
# Distance and corresponding probability data
import numpy as np
from sklearn.linear_model import LogisticRegression
distances = np.array([1,2,5,10,15,20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30,35,40,41,47,50]).reshape(-1, 1)
probabilities = np.array([1,1,1,1,1,1,0.9, 0.85, 0.73, 0.67, 0.5, 0.47, 0.39, 0.31, 0.25, 0.15,0,0,0,0,0])
threshold=0.5
binary_labels=(probabilities>threshold)
logr=LogisticRegression()
logr.fit(distances,binary_labels)
p=logr.predict([[25]])
print(p)
```

```
     [False]
```

```
dist=np.linspace(1,50,100).reshape(-1,1)
print(dist)
prob=logr.predict_proba(dist)
print(prob)
plt.plot(dist,prob,color="blue",label="Logistic Regression")
plt.title("Distance vs probability of scoring a Goal")
plt.xlabel("Distance")
plt.ylabel("Probability")
plt.legend()
plt.grid()
plt.show()
```

```
       [[ 1.        ]
        [ 1.49494949]
        [ 1.98989899]
        [ 2.48484848]
        [ 2.97979798]
        [ 3.47474747]
        [ 3.96969697]
        [ 4.46464646]
        [ 4.95959596]
        [ 5.45454545]
        [ 5.94949495]
        [ 6.44444444]
        [ 6.93939394]
        [ 7.43434343]
        [ 7.92929293]
        [ 8.42424242]
        [ 8.91919192]
        [ 9.41414141]
        [ 9.90909091]
        [10.4040404 ]
        [10.8989899 ]
        [11.39393939]
        [11.88888889]
        [12.38383838]
        [12.87878788]
        [13.37373737]
        [13.86868687]
        [14.36363636]
        [14.85858586]
        [15.35353535]
        [15.84848485]
        [16.34343434]
        [16.83838384]
        [17.33333333]
        [17.82828283]
        [18.32323232]
        [18.81818182]
        [19.31313131]
        [19.80808081]
        [20.3030303 ]
        [20.7979798 ]
        [21.29292929]
        [21.78787879]
        [22.28282828]
        [22.77777778]
        [23.27272727]
        [23.76767677]
        [24.26262626]
        [24.75757576]
        [25.25252525]
        [25.74747475]
        [26.24242424]
        [26.73737374]
        [27.23232323]
        [27.72727273]
        [28.22222222]
        [28.71717172]
```

```
import pandas as pd
df=pd.read_csv("/content/demodt.txt",sep=",")
df
```

	State	Literacy	Cleanliness	Crime_Rate	Good
0	A	92	90	54	0
1	B	56	67	50	1
2	C	78	85	62	0
3	D	63	72	48	1
4	E	85	79	55	0
5	F	71	68	58	0
6	G	80	83	51	0
7	H	67	74	47	1
8	I	89	88	53	0
9	J	58	65	49	1
10	K	82	81	60	0
11	L	75	78	57	0
12	M	69	70	46	1
13	N	87	86	52	0
14	O	61	63	45	1
15	P	93	91	56	0
16	Q	55	66	61	0
17	R	76	77	59	0
18	S	84	82	44	1
19	T	70	69	50	1
20	U	94	92	57	0
21	V	59	64	52	0
22	W	83	80	43	1
23	X	74	76	63	0
24	Y	68	73	41	1
25	Z	88	84	47	1

```
[1.21339773e-02 9.87866023e-01]
```

```
cr=np.array(df['Crime_Rate']).reshape(-1,1)#it is independent variable so convert it to 2D array
print(cr)
y=np.array(df["Good"])
```

```
[[54]
 [50]
 [62]
 [48]
 [55]
 [58]
 [51]
 [47]
 [53]
 [49]
 [60]
 [57]
 [46]
 [52]
 [45]
 [56]
 [61]
 [59]
 [44]
 [50]
 [57]
 [52]
 [43]
 [63]
 [41]
 [47]]
[[54]
 [50]
 [62]
 [48]
 [55]
 [58]
 [51]
 [47]
 [53]
```

```
[49]
[60]
[57]
[46]
[52]
[45]
[56]
[61]
[59]
[44]
[50]
[57]
[52]
[43]
[63]
[41]
[47]]
[1.00000000e+00 1.22627861e-13]
```

```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(cr,y)
n=int(input("Enter crime rate:"))
p=model.predict([[n]])
if(p==1):
  print("Good")
else:
  print("Bad")
```

```
Enter crime rate:20
Good
```


```
import pandas as pd
df=pd.read_csv("/content/demodt.txt",sep=",")
df
```

	State	Literacy	Cleanliness	Crime_Rate	Good
0	A	92	90	54	0
1	B	56	67	50	1
2	C	78	85	62	0
3	D	63	72	48	1
4	E	85	79	55	0
5	F	71	68	58	0
6	G	80	83	51	0
7	H	67	74	47	1
8	I	89	88	53	0
9	J	58	65	49	1
10	K	82	81	60	0
11	L	75	78	57	0
12	M	69	70	46	1
13	N	87	86	52	0
14	O	61	63	45	1
15	P	93	91	56	0
16	Q	55	66	61	0
17	R	76	77	59	0
18	S	84	82	44	1
19	T	70	69	50	1