

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
```

```
In [2]: df=pd.read_csv("insurance_data.csv")
df.head()
```

```
Out[2]:
```

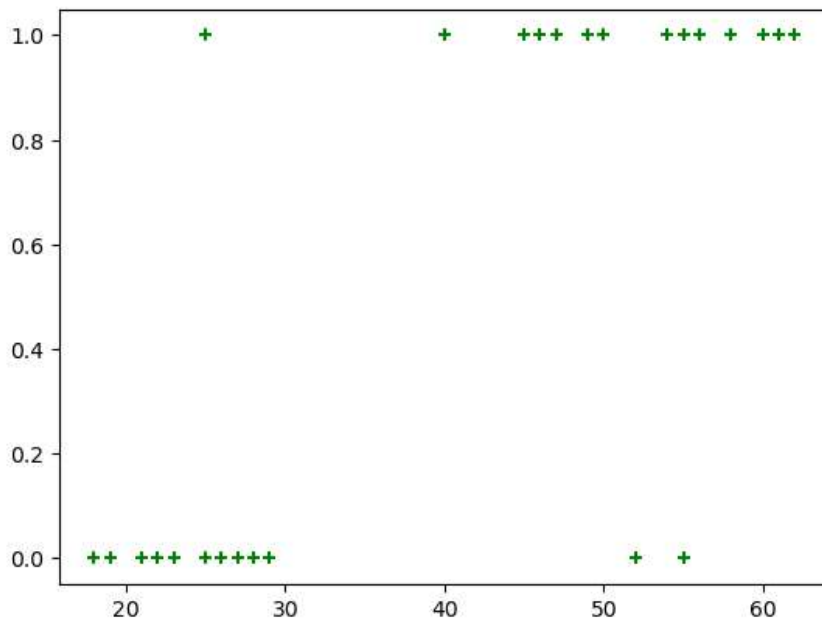
	age	bought_insurance
0	22	0
1	25	0
2	47	1
3	52	0
4	46	1

```
In [3]: df.shape
```

```
Out[3]: (27, 2)
```

```
In [4]: plt.scatter(df.age,df.bought_insurance,marker='+',color='green')
```

```
Out[4]: <matplotlib.collections.PathCollection at 0x146ce18b7f0>
```



```
In [5]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(df[['age']],df.bought_insurance,test_size=0.3,random_state=
```

```
In [6]: x_test.shape
```

```
Out[6]: (9, 1)
```

```
In [7]: x_train.shape
```

```
Out[7]: (18, 1)
```

```
In [8]: from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

```
In [10]: model.fit(x_train,y_train)
```

```
Out[10]: LogisticRegression
LogisticRegression()
```

```
In [11]: x_test
```

```
Out[11]:
```

	age
17	58
14	49
18	19
3	52
23	45
10	18
20	21
21	26
4	46

```
In [13]: y_pred=model.predict(x_test)
```

```
In [14]: model.score(x_test,y_test)
```

```
Out[14]: 0.8888888888888888
```

```
In [15]: model.score(x_train,y_train)
```

```
Out[15]: 0.8888888888888888
```

```
In [18]: from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

```
Out[18]: array([[4, 1],
               [0, 4]], dtype=int64)
```

```
In [19]: model.coef_
```

```
Out[19]: array([[0.1312933]])
```

```
In [20]: model.intercept_
```

```
Out[20]: array([-4.92847769])
```

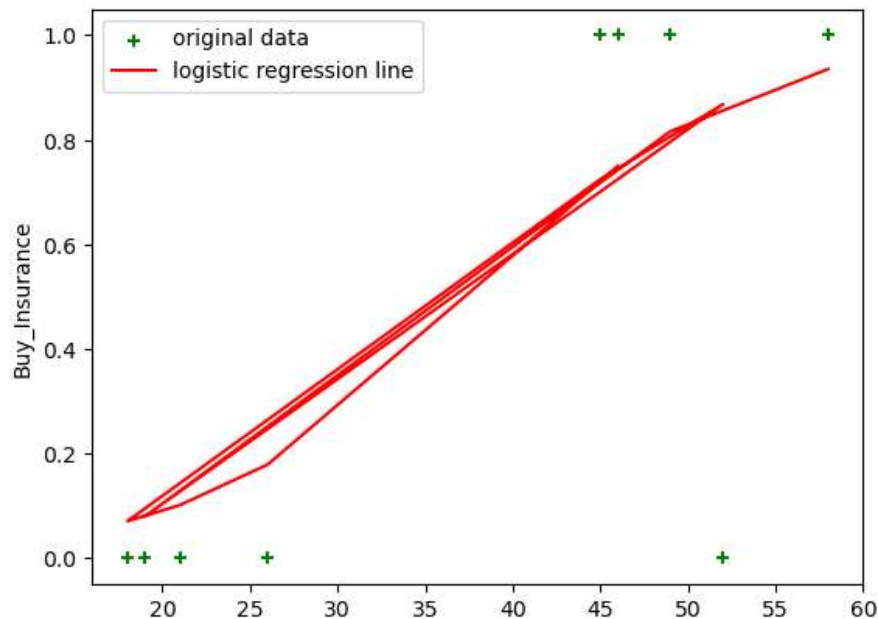
```
In [21]: import numpy as np
def sigmoid(x):
    return 1.0/(1+np.exp(-x))
```

```
In [22]: def pred_function(age):
z=0.131*age - 4.928
y=sigmoid(z)
return y
```

```
In [23]: age=35
pred_function(age)
```

```
Out[23]: 0.4150809257548097
```

```
In [29]: #to response predict vector
response_Vec= sigmoid(0.131*x_test-4.928)
#placing labels
plt.scatter(x_test,y_test,marker='+',color='green',label='original data')
plt.plot(x_test,response_Vec,"r",label='logistic regression line')
plt.xlabel('Age')
plt.ylabel('Buy_Insurance')
#to display plot
plt.legend()
plt.show()
```



In []: