Experiment 5: Implement Logistic Regression by using Insurance Dataset

In [5]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```
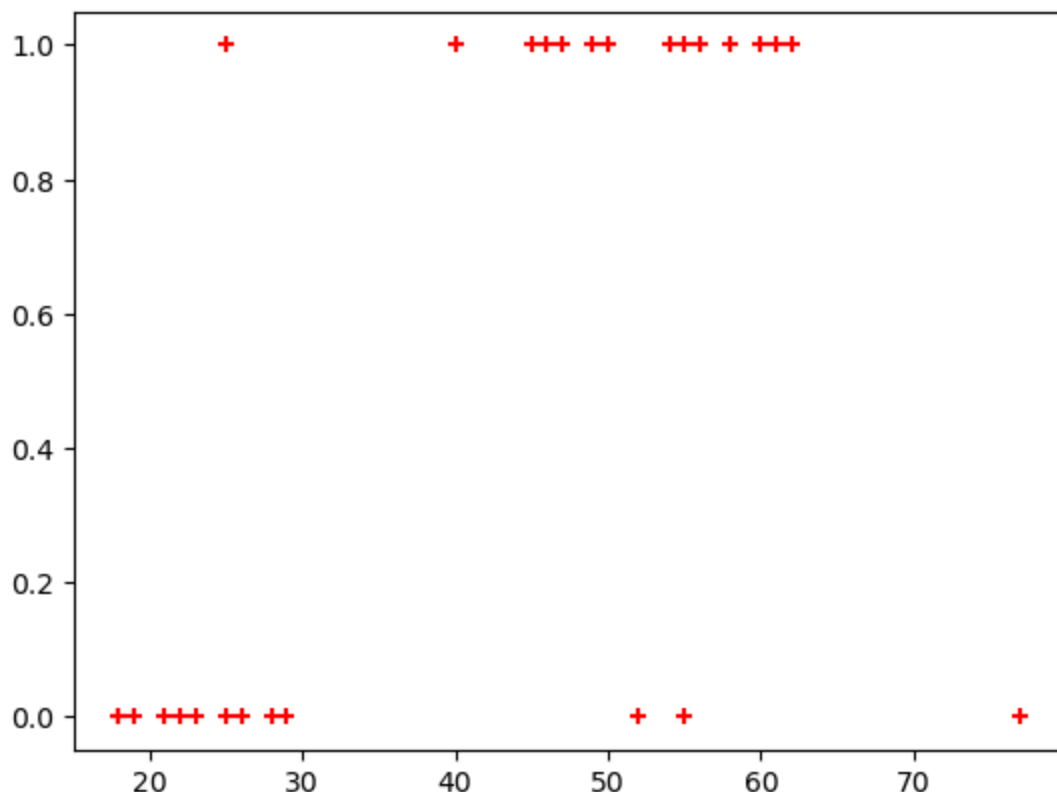
In [6]:
```python
df = pd.read_csv('insurance_data.csv')
df.head()
```

Out[6]:

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

In [7]:
```python
plt.scatter(df.age, df.bought_insurance, marker = '+', color = 'red')
```

Out[7]: <matplotlib.collections.PathCollection at 0x1b38d8d3530>

```python
In [8]:  from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(df[['age']], df.bought_i
```

```python
In [9]:  x_test
```

Out[9]:

	age
21	26
22	40
23	45
24	50
25	54
26	23

```python
In [10]: from sklearn import linear_model
         from sklearn.linear_model import LogisticRegression
```

```python
In [11]: reg = linear_model.LogisticRegression()
```

```python
In [12]: reg.fit(x_train,y_train)
```

Out[12]:
```
▼   LogisticRegression  ⓘ ?

LogisticRegression()
```

```python
In [13]: reg.coef_
```

Out[13]: array([[0.06987406]])

```python
In [14]: reg.intercept_
```

Out[14]: array([-3.07609549])

```python
In [15]: reg.predict([[47]])
```

```
C:\Users\Rishi\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning:
X does not have valid feature names, but LogisticRegression was fitted with
feature names
  warnings.warn(
```

Out[15]: array([1], dtype=int64)

```python
In [17]: y_pred = reg.predict(x_test)
```

```python
In [19]: from sklearn.metrics import accuracy_score, confusion_matrix, classificatior
         accuracy = accuracy_score(y_test, y_pred)
         conf_matrix = confusion_matrix(y_test, y_pred)
         class_report = classification_report(y_test, y_pred)
```

```
accuracy, conf_matrix, class_report
```

Out[19]: (0.8333333333333334,
 array([[2, 0],
        [1, 3]], dtype=int64),
 '   precision   recall f1-score   support\n\n   0
0.67   1.00   0.80   2\n   1   1.00   0.75
0.86   4\n\n   accuracy   0.83   6\n
   macro avg   0.83   0.88   0.83   6\nweighted avg   0.89
0.83   0.84   6\n')

In [ ]: