Predicting if a person would buy life insurnace based on his age using logistic regression

Above is a binary logistic regression problem as there are only two possible outcomes (i.e. if person buys insurance or he/she doesn't).

In [1]:
```
import pandas as pd # The Pandas library is used for importing and managing the d
import matplotlib.pyplot as plt # The second library is matplotlib, which is a Py
```

In [2]:
```
df = pd.read_csv("insurance_data.csv") # We are reading csv file named insurance_
df.head() # with df.head() function we can see the first five data points
```

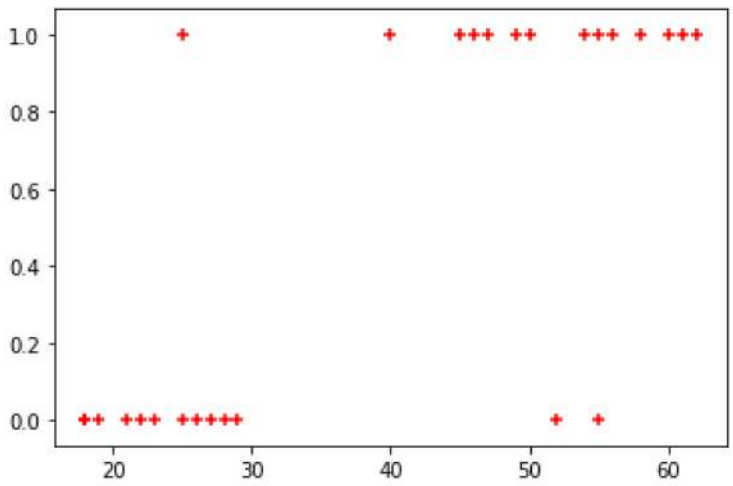
Out[2]:

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

The logistic function is a Sigmoid function, which takes any real value between zero and one. And if we plot it, the graph will be S curve,

In [11]:
```
plt.scatter(df.age,df.bought_insurance,marker='+',color='red') # We are using sc
```

Out[11]:
```
<matplotlib.collections.PathCollection at 0x1cbc2a4dda0>
```



In [4]:
```
x=df[['age']] # storing age in x  , here the value of x will be two dimensional
y=df['bought_insurance'] # string bought_insurance value in y , Y value will be
```

x_train: features for the training data

x_test: features for testing data

y_train: Dependent variables for training data

y_test: Independent variable for testing data

In train_test_split() function, we have passed four parameters in which first two are for arrays of data, and test_size is for specifying the size of the test set. The test_size maybe .5, .3, or .2, which tells the dividing ratio of training and testing sets. o The last parameter random_state is used to set a seed for a random generator so that you always get the same result.

In [5]:
```
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_sta
```

In [6]:
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(x_train, y_train)
```

```
C:\Users\HP\Anaconda3\envs\opencv\lib\site-packages\sklearn\linear_model\logist
ic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Sp
ecify a solver to silence this warning.
  FutureWarning)
```

Out[6]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

In [7]:
```
y_predicted = model.predict(x_test)
```

In [8]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_predicted)
```

Out[8]:
```
array([[2, 1],
       [0, 3]], dtype=int64)
```

In [9]:
```
model.score(x_test,y_test)
```

Out[9]:
```
0.8333333333333334
```

In [ ]:

In [ ]: