Practicing Logistic Regression Model

Predicting weather a person will buy insurance.

In [1]:
```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

In [2]:
```
df = pd.read_csv(r'D:\NEEL_FOLDER\Data Science\Logistics_Regression_codebasic
\insurance_data.csv')
df.head()
```

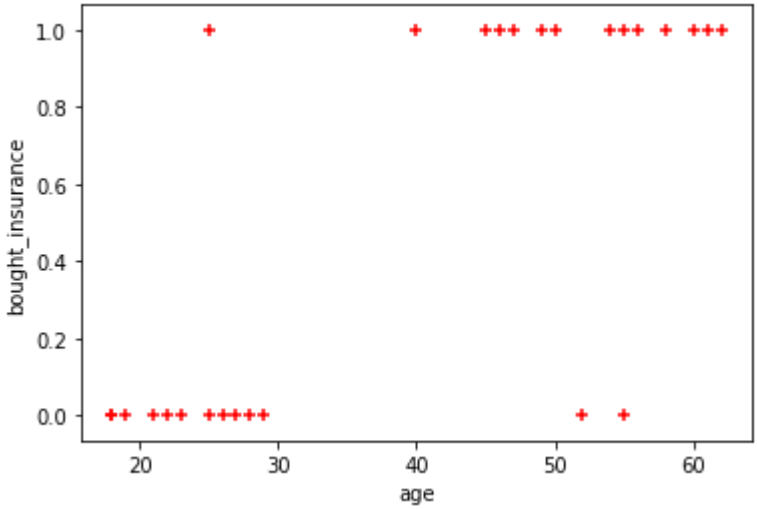
Out[2]:

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

Plot scatter plot to see data distribution.

In [3]:
```
plt.xlabel('age')
plt.ylabel('bought_insurance')
plt.scatter(df.age,df.bought_insurance,marker='+',color='red')
```

Out[3]: <matplotlib.collections.PathCollection at 0x8bd1f30>



Use train test split metthod to split our dataset

In [4]:
```
from sklearn.model_selection import train_test_split
```

In [5]:
```
X_train, X_test, y_train, y_test = train_test_split(df[['age']],df.bought_insurance,train_size=0.8)
```

In [6]:
```
X_test
```

Out[6]:

	age
8	62
24	50
22	40
13	29
6	55
2	47

In [7]:
```
from sklearn.linear_model import LogisticRegression
```

In [8]:
```
model = LogisticRegression()
```

In [9]:
```
model.fit(X_train,y_train)
```

```
C:\Users\admin\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```

Out[9]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, 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 [10]:
```
X_test
```

Out[10]:

	age
8	62
24	50
22	40
13	29
6	55
2	47

In [11]:
```
y_predicted = model.predict(X_test)
```

In [12]:
```
model.predict_proba(X_test)
```

Out[12]:
```
array([[0.15905366, 0.84094634],
       [0.26936303, 0.73063697],
       [0.39134422, 0.60865578],
       [0.54243394, 0.45756606],
       [0.21824002, 0.78175998],
       [0.30343382, 0.69656618]])
```

In [13]:
```
model.score(X_test,y_test)
```

Out[13]:
```
0.8333333333333334
```

In [14]:
```
y_predicted
#the prediction says person with age of 62, 50, 40, 55 & 47 have bought insura
nce. Age 29 have not bought insurance.
```

Out[14]:
```
array([1, 1, 1, 0, 1, 1], dtype=int64)
```

In [15]:
```
model.coef_
```

Out[15]:
```
array([[0.05561913]])
```

In [16]:
```
model.intercept_
```

Out[16]:
```
array([-1.78309979])
```

Lets defined sigmoid function now and do the math with hand

In [17]:
```
import math
def sigmoid(x):
  return 1 / (1 + math.exp(-x))
```

In [18]:
```
def prediction_function(age):
    z = 0.042 * age - 1.53 # 0.04150133 ~ 0.042 and -1.52726963 ~ -1.53
    y = sigmoid(z)
    return y
```

In [19]:
```
age = 35
prediction_function(age)
#0.485 is less than 0.5 which means person with 35 age will not buy insurance
```

Out[19]:
```
0.4850044983805899
```

In [20]:
```
age = 43
prediction_function(age)
#0.485 is more than 0.5 which means person with 43 will buy the insurance
```

Out[20]:
```
0.568565299077705
```

Source:Codebasic youtube channel