

## ex4-logistic-regression

August 12, 2024

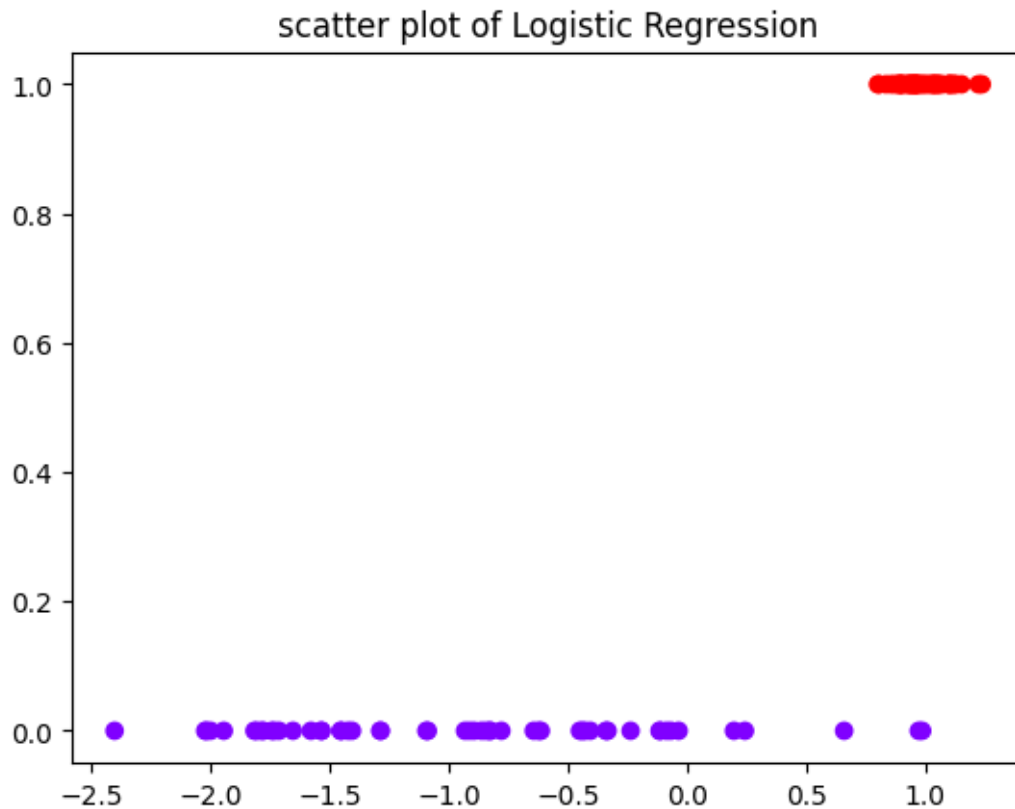
```
[1]: import pandas as pd
      from sklearn.model_selection import train_test_split
      from sklearn.linear_model import LogisticRegression
      from sklearn.metrics import confusion_matrix

      from sklearn.datasets import make_classification
      from matplotlib import pyplot as plt
```

```
[2]: x, y = make_classification(
      n_samples=100,
      n_features=1,
      n_classes=2,
      n_clusters_per_class=1,
      flip_y=0.03,
      n_informative=1,
      n_redundant=0,
      n_repeated=0,
      )
      print(y)
```

```
[1 0 1 1 0 0 1 1 0 0 1 1 1 0 1 0 1 0 0 1 0 1 1 0 1 0 1 0 0 0 0 1 1 1 1 1 1 1
 0 0 1 0 0 0 0 0 0 1 0 0 1 1 1 1 0 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 0 0 0 0 1
 1 0 0 0 1 1 1 1 1 0 0 0 0 0 0 1 0 1 0 1 1 1 0 1 1 0]
```

```
[3]: plt.scatter(x, y, c=y, cmap="rainbow")
      plt.title("scatter plot of Logistic Regression")
      plt.show()
```



```
[4]: x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=1)
```

```
[5]: x_train.shape
```

```
[5]: (75, 1)
```

```
[6]: # step 5 perform logidtic regression
log_reg = LogisticRegression()
log_reg.fit(x_train, y_train)
```

```
[6]: LogisticRegression()
```

```
[7]: # step 6 Make prediction using thr model
# performs prediction using the test dataset
y_pred = log_reg.predict(x_test)
```

```
[8]: # step 7 Display the Confusion matrix
confusion_matrix(y_test, y_pred)
```

```
[8]: array([[ 7,  1],
          [ 0, 17]])
```

```
[9]: df = pd.read_csv("datasets/insurance_data.csv")
```

```
[10]: df
```

```
[10]:
```

	age	bought_insurance
0	22	0
1	25	0
2	47	1
3	52	0
4	46	1
5	56	1
6	55	0
7	60	1
8	62	1
9	61	1
10	18	0
11	28	0
12	27	0
13	29	0
14	49	1
15	55	1
16	25	1
17	58	1
18	19	0
19	18	0
20	21	0
21	26	0
22	40	1
23	45	1
24	50	1
25	54	1
26	23	0

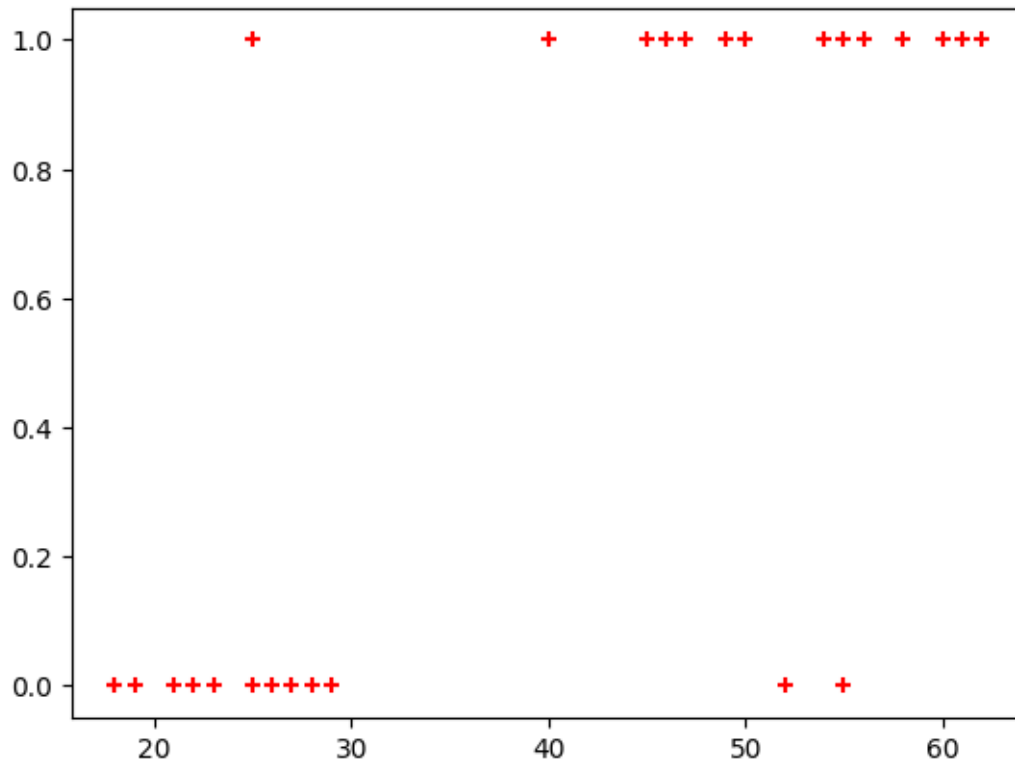
```
[11]: df.head()
```

```
[11]:
```

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

```
[12]: plt.scatter(df.age, df.bought_insurance, marker="+", color="red")
```

```
[12]: <matplotlib.collections.PathCollection at 0x7f199451fd50>
```



```
[13]: df.shape
```

```
[13]: (27, 2)
```

```
[14]: x_train, x_test, y_train, y_test = train_test_split(
      df[["age"]], df.bought_insurance, test_size=0.1
    )
```

```
[15]: x_test
```

```
[15]:      age
20    21
26    23
18    19
```

```
[16]: model = LogisticRegression()
```

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

```
[17]: LogisticRegression()
```

```
[18]: model.predict(x_test)
```

```
[18]: array([0, 0, 0])
```

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

```
[19]: 1.0
```

```
[20]: model.predict_proba(x_test)
```

```
[20]: array([[0.89425677, 0.10574323],  
        [0.86809806, 0.13190194],  
        [0.91573128, 0.08426872]])
```

```
[21]: df.describe()
```

```
[21]:
```

	age	bought_insurance
count	27.000000	27.000000
mean	39.666667	0.518519
std	15.745573	0.509175
min	18.000000	0.000000
25%	25.000000	0.000000
50%	45.000000	1.000000
75%	54.500000	1.000000
max	62.000000	1.000000