17. Data Science – Machine Learning – Logistic Regression

Contents

17. Data Science – Machine Learning – Logistic Regression

1. Logistic Regression

- ✓ Logistic regression comes under supervised Learning.
- ✓ It is a technique that is used to solve for classification problems.
- ✓ It is used for predicting the categorical dependent variable using a given set of independent variables.
- ✓ Examples
    - Email spam or not
    - Customer will buy product or not

2. Types of logistic regression

- ✓ Binary classification
    - This is having two classes

- ✓ Multiclass classification
    - This is having more than two classes

3. Binary classification

- ✓ In binary classification, there can be only two possible types of the dependent variables, such as,
    - 0 or 1
    - Pass or Fail
    - Yes or No etc.

4. Multiclass classification

- ✓ In multiclass classification, there can be 3 or more possible unordered types of the dependent variable, such as,
    - Ok, good, best
    - Cat, dot, sheep etc

5. Data set

- ✓ Its insurance dataset
    - ○ ZERO means didn't buy the insurance
    - ○ ONE means will buy the insurance
- ✓ We can understand one pattern here like, young people not buying the insurance
- ✓ Whereas person age increasing then that person more likely to buy the insurance

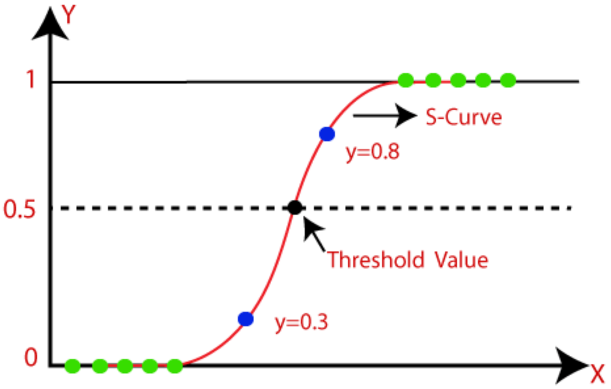
6. Problem statement

- ✓ Based on the age, we wanted to predict for persons will chose insurance or not.

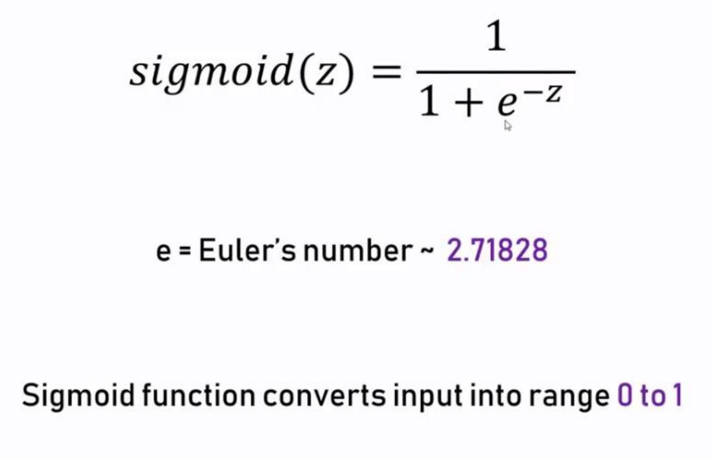
age	**Insurance status**
22	0
25	0
47	1
52	0
46	1
56	1
55	0
60	1
62	1
61	1

7. Logistic function or Sigmoid

- ✓ The logistic function, also called the sigmoid function.
- ✓ It maps any real value into another value within a range of 0 and 1.
- ✓ The value of the logistic regression must be between 0 and 1.
- ✓ In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1.



Formula



$$sigmoid(z) = \frac{1}{1 + e^{-z}}$$

e = Euler's number ~ 2.71828

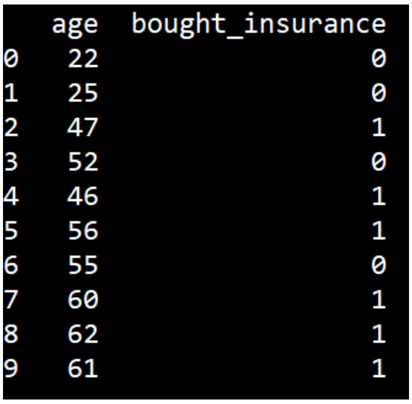
Sigmoid function converts input into range 0 to 1

Program Name

Loading insurance dataset
demo1.py

```
import pandas as pd

# Loading the dataset
df = pd.read_csv("insurance_data.csv")

print(df.head(10))
```

Output



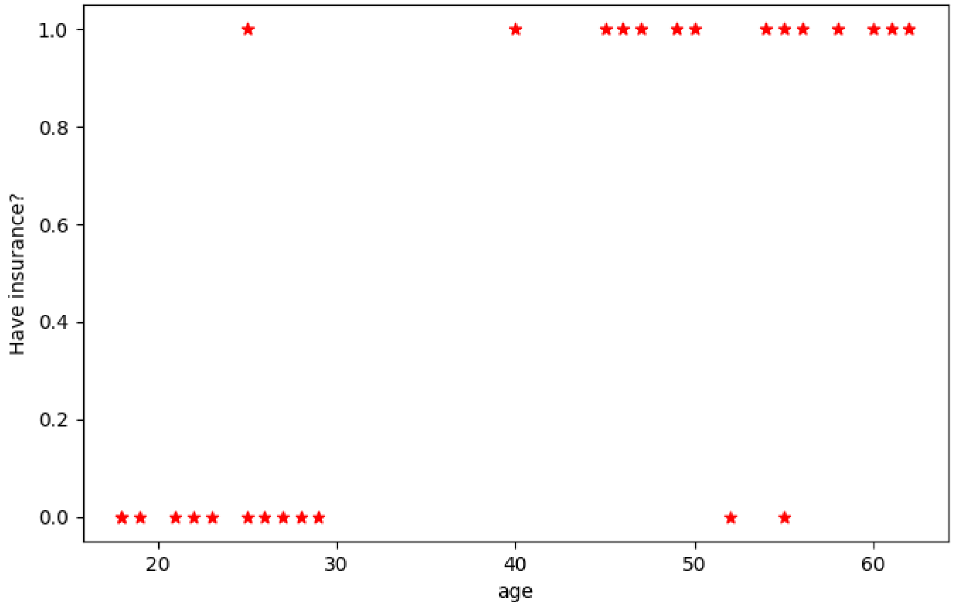
```
   age  bought_insurance
0   22                 0
1   25                 0
2   47                 1
3   52                 0
4   46                 1
5   56                 1
6   55                 0
7   60                 1
8   62                 1
9   61                 1
```

Program Name

Plotting the dataset
demo2.py

```
import pandas as pd
from matplotlib import pyplot as plt

# Loading the dataset
df = pd.read_csv("insurance_data.csv")

# plotting the data
plt.scatter(df.age, df.bought_insurance, marker = '*', color = 'red')

plt.xlabel('age')
plt.ylabel('Have insurance?')

plt.show()
```

Output

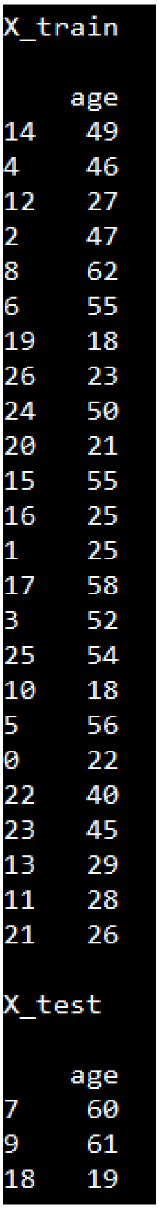


Program Name

Splitting the dataset
demo3.py

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Loading the dataset
df = pd.read_csv("insurance_data.csv")
X = df[['age']]
y = df.bought_insurance

# Splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.1, random_state=52)

print("X_train", '\n')
print(X_train,'\n')

print("X_test", '\n')
print(X_test)
```

Output



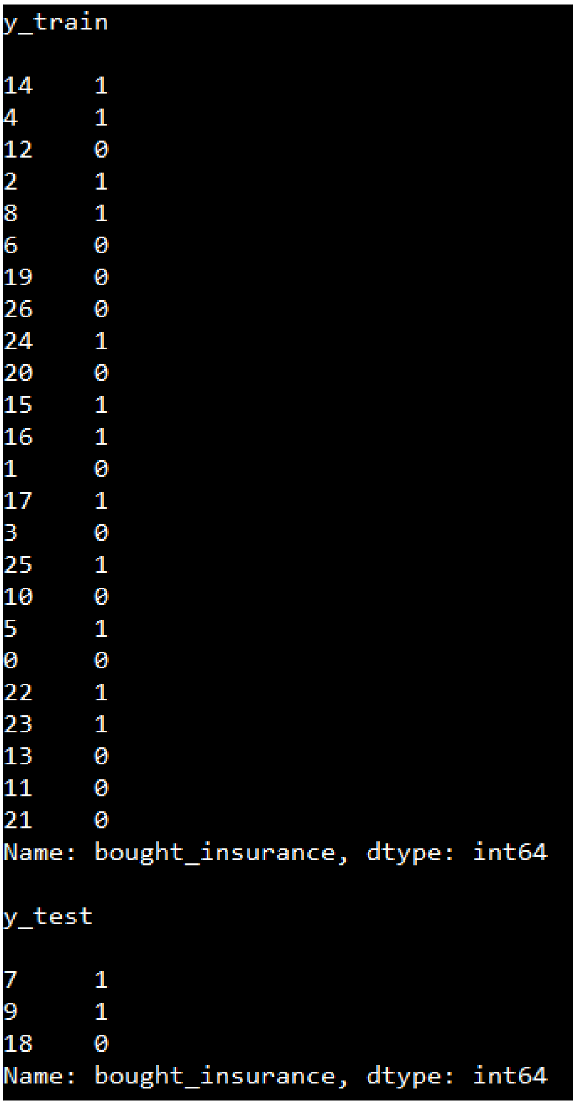
```
X_train

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

X_test

    age
7    60
9    61
18   19
```

Program Name

Splitting the dataset
demo4.py

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Loading the dataset
df = pd.read_csv("insurance_data.csv")

X = df[['age']]
y = df.bought_insurance

# Splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.1, random_state = 52)


print("y_train", '\n')
print(y_train,'\n')

print("y_test", '\n')
print(y_test)
```

Output



```
y_train

14     1
4      1
12     0
2      1
8      1
6      0
19     0
26     0
24     1
20     0
15     1
16     1
1      0
17     1
3      0
25     1
10     0
5      1
0      0
22     1
23     1
13     0
11     0
21     0
Name: bought_insurance, dtype: int64

y_test

7      1
9      1
18     0
Name: bought_insurance, dtype: int64
```

Program Name

Training the model
demo5.py

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Loading the dataset
df = pd.read_csv("insurance_data.csv")

X = df[['age']]
y = df.bought_insurance

# Splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1, random_state = 52)

# Creating and training the model
model = LogisticRegression()
model.fit(X_train, y_train)

print("Model got trained")
```

Output

```
Model got trained
```

Program Name

Prediction with single value
demo6.py

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Loading the dataset
df = pd.read_csv("insurance_data.csv")

X = df[['age']]
y = df.bought_insurance

# Splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.1, random_state = 52)

# Creating and training the model
model = LogisticRegression()
model.fit(X_train, y_train)

# Prediction
print(model.predict([[50]]))
print(model.predict([[25]]))
```

Output

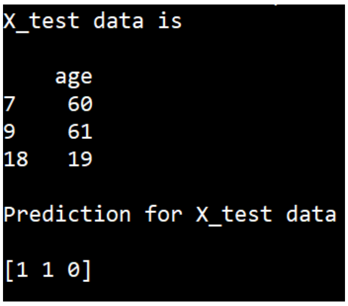
```
[1]
[0]
```

Program Name

Prediction the result
demo7.py

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Loading the dataset
df = pd.read_csv("insurance_data.csv")

X = df[['age']]
y = df.bought_insurance

# Splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.1, random_state = 52)

# Creating and training the model
model = LogisticRegression()
model.fit(X_train, y_train)

# Prediction
y_predicted = model.predict(X_test)

print("X_test data is \n")
print(X_test, '\n')

print("Prediction for X_test data \n")
print(y_predicted)
```

Output



```
X_test data is

    age
7    60
9    61
18   19

Prediction for X_test data

[1 1 0]
```

Prediction

- ✓ The one who has 19 years age he will not buy the insurance
- ✓ Both who has 60 and 61 years age persons will buy the insurance

Program Name

Prediction score
demo8.py

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Loading the dataset
df=pd.read_csv("insurance_data.csv")

X = df[['age']]
y = df.bought_insurance

# Splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1, random_state = 52)

# Creating and training the model
model = LogisticRegression()
model.fit(X_train, y_train)

# Prediction score
print(model.score(X_test, y_test))
```

Output

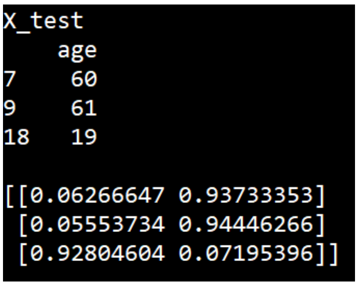
```
1.0
```

Program Name

Prediction probability
demo9.py

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Loading the dataset
df = pd.read_csv("insurance_data.csv")

X = df[['age']]
y = df.bought_insurance

# Splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.1, random_state = 52)

# Creating and training the model
model = LogisticRegression()
model.fit(X_train, y_train)

print("X_test")
print(X_test)

print()

# Prediction probability
print(model.predict_proba(X_test))
```

Output



```
X_test
    age
7    60
9    61
18   19

[[0.06266647 0.93733353]
 [0.05553734 0.94446266]
 [0.92804604 0.07195396]]
```