Abdul Raffy
W25-PAK-INP-AI-16

Logistic regression

> It is a supervised classification model

Independent / dependent
helps to   To be predicted
predict
dependent.

•) In linear regression data is continuous
•) but in the logistic the date to
be predicted is categorical, binary (discrete)
(0, 1) (yes, no)

A student studies 5 hours and predict
he will pass or not on basis
of given data of study hours and
Exam result as 0,1 (find probability)

Sigmoid function is used

$$y = \frac{1}{1 + e^{-(a_0 + a_1 x)}}$$

↓
Predicting
variable
(0, 1)
only between
0 and 1 high or low-

Maximum likely hood
function Estimation
i.e. cos function
is optimized again and
Gradient ... again.

If x is 0 when
↗ y will intercept
a_0 = Intercept
a_1 = coefficient
If x is incremented by
1 how much y will
change.

> Training
D.ots

not continuous
like
linear
if is
yes or No
0 to 1

study hours →X
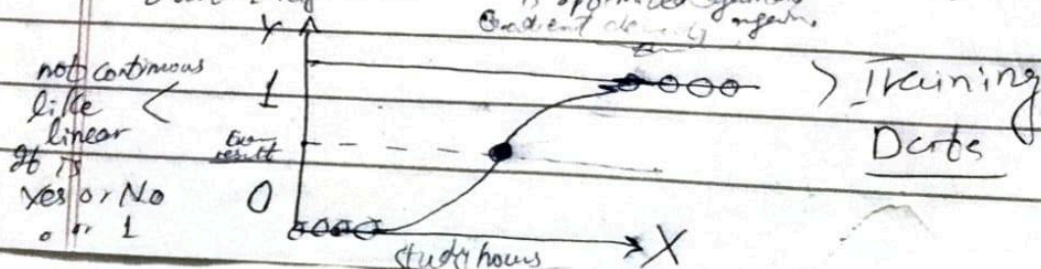
Maximum likely hood method is used
to calculate a_0 and a_1.
In which cos function is further
optimized by Gradient decent method

For now we assume $x = 5$ - hours of study
$a_0 = -1.5$ $a_1 = 0.6$

$y = \frac{1}{1+e^{(a_0+a_1 x)}}$ $a_0 + a_1 x = (-1.5 + 0.6 \times 5)$

$\rightarrow Y = \frac{1}{1+e^{-1.5}}$ ← $= 1.5$

$\approx .818 \rightarrow 81$ percent chance he will
Pass if he studies 5 hours
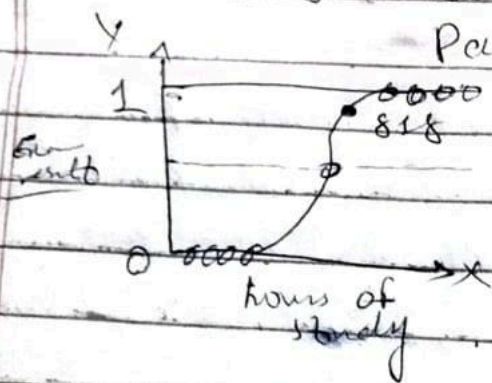
Linear regression	Logistic regression
→ Continuous	→ Categorized (Yes or No)
→ Prize hike based on house or land area	→ On base of study hours student will pass or not Yes or no (classified)

Screenshot 1 (top):

```python
 5   # Data: Hours studied and pass (1 = Yes, 0 = No)
 6   hours_studied = np.array([1, 2, 3, 4, 5, 6, 7]).r
 7   pass_exam = np.array([0, 0, 0, 1, 1, 1, 1])
 8
 9   # Split the data into training and test sets
10   X_train, X_test, y_train, y_test = train_test_spli
11
12   # logistic regression formula
13   def sigmoid(x, b_0, b_1):
14       """Sigmoid function (Logistic function)"""
15       return 1 / (1 + np.exp(-(b_0 + b_1 * x)))
16
17   # Experiment with different values of b_0 (interc
18   b_0 = -5  # Intercept
19   b_1 = 1.5  # Coefficient for hours studied
20
21   # Predicting the probability using the logistic r
22   predictions_probabilities = sigmoid(X_test, b_0,
23
24   # Predict the class labels based on the probabili
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
[0.69141896 0.30858104]]
PS D:\Doc\AIC\Assignments\Python\16th_Assignment_AIC\Task_
IC/Assignments/Python/16th_Assignment_AIC/Task_2/Logistic_
Test Set Hours Studied: [1 2]
Predicted Probabilities of Passing: [[0.02931223]
 [0.11920292]]
Predicted Pass/Fail (0 or 1): [[0]
 [0]]
```

Figure 1 — Logistic Regression - Exam Pass Prediction

Screenshot 2 (bottom):

```python
 5   # Data: Hours studied and pass (1 = Yes,
 6   hours_studied = np.array([1, 2, 3, 4, 5,
 7   pass_exam = np.array([0, 0, 0, 1, 1, 1,
 8
 9   # Split the data into training and test
10   X_train, X_test, y_train, y_test = train
11
12   # logistic regression formula
13   def sigmoid(x, b_0, b_1):
14       """Sigmoid function (Logistic functi
15       return 1 / (1 + np.exp(-(b_0 + b_1 *
16
17   # Experiment with different values of b_
18   b_0 = -3  # Intercept
19   b_1 = 1.5  # Coefficient for hours studi
20
21   # Predicting the probability using the l
22   predictions_probabilities = sigmoid(X_te
23
24   # Predict the class labels based on the
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORT

```
 [0]]
PS D:\Doc\AIC\Assignment\Python\16th_Assignment_
IC/Assignments/Python/16th_Assignment_AIC/Task_2
Test Set Hours Studied: [1 2]
Predicted Probabilities of Passing: [[0.18242552
 [0.5        ]]
Predicted Pass/Fail (0 or 1): [[0]
 [1]]
```

Figure 1 — Logistic Regression - Exam Pass Prediction

Top screenshot:

```
6   hours_studied = np.array([1, 2, 3, 4, 5,
7   pass_exam = np.array([0, 0, 0, 1, 1, 1,
8
9   # Split the data into training and test
10  X_train, X_test, y_train, y_test = train
11
12  # logistic regression formula
13  def sigmoid(x, b_0, b_1):
14      """Sigmoid function (Logistic functi
15      return 1 / (1 + np.exp(-(b_0 + b_1 *
16
17  # Experiment with different values of b_
18  b_0 = -7  # Intercept
19  b_1 = 9   # Coefficient for hours studied
20
21  # Predicting the probability using the l
22  predictions_probabilities = sigmoid(X_te
23
24  # Predict the class labels based on the
```

```
[1]]
PS D:\Doc\AIC\Assignments\Python\16th_Assignment
IC/Assignments/Python/16th_Assignment_AIC/Task_2
Test Set Hours Studied: [1 2]
Predicted Probabilities of Passing: [[0.88079708
 [0.9999833 ]]
Predicted Pass/Fail (0 or 1): [[1]
[1]]
```

Logistic Regression - Exam Pass Prediction

(x, y) = (0.37, 0.271)

Bottom screenshot:

```
6   hours_studied = np.array([1, 2, 3, 4
7   pass_exam = np.array([0, 0, 0, 1, 1,
8
9   # Split the data into training and t
10  X_train, X_test, y_train, y_test = t
11
12  # logistic regression formula
13  def sigmoid(x, b_0, b_1):
14      """Sigmoid function (Logistic fu
15      return 1 / (1 + np.exp(-(b_0 + b
16
17  # Experiment with different values o
18  b_0 = -1  # Intercept
19  b_1 = 8   # Coefficient for hours stu
20
21  # Predicting the probability using t
22  predictions_probabilities = sigmoid(
23
24  # Predict the class labels based on
```

```
[1]]
PS D:\Doc\AIC\Assignments\Python\16th_Assign
IC/Assignments/Python/16th_Assignment_AIC/Tas
Test Set Hours Studied: [1 2]
Predicted Probabilities of Passing: [[0.99908
 [0.99999969]]
Predicted Pass/Fail (0 or 1): [[1]
[1]]
```

Logistic Regression - Exam Pass Prediction