Module-04, Python for Machine Learning

Classification Algorithms (Logistic Regression)

Dostdar Ali
Instructor

Data science and Artificial Intelligence
3-Months Course
at
Karakaroum international Univrsity

January 31, 2024

Table of Contents

Logistic Regression

Definition

Logistic Regression is a statistical method used for binary classification. It models the probability of a binary outcome.

- We want to learn about Logistic Regression as a method for Classification.
- Some examples of classification problems,
  Spam versus "Ham" emails
  Loan Default (yes/no)
  Disease Diagnosis
- Above were all examples of Binary Classification

Logistic Regression

Definition

Logistic Regression is a statistical method used for binary classification. It models the probability of a binary outcome.

- We want to learn about Logistic Regression as a method for Classification.
- Some examples of classification problems,
  Spam versus "Ham" emails
  Loan Default (yes/no)
  Disease Diagnosis
- Above were all examples of Binary Classification

Logistic Regression

Definition

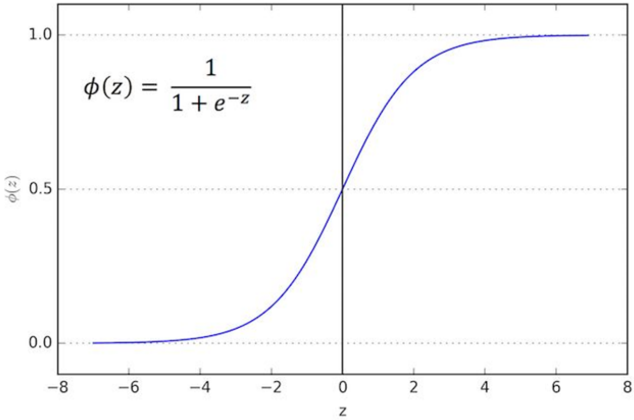
Logistic Regression is a statistical method used for binary classification. It models the probability of a binary outcome.

- We want to learn about Logistic Regression as a method for Classification.
- Some examples of classification problems,
  Spam versus "Ham" emails
  Loan Default (yes/no)
  Disease Diagnosis
- Above were all examples of Binary Classification

Sigmoid Function

Definition

The Sigmoid Function takes in any value and outputs it to be between 0 and 1.



- **Mathematical Formulation:**

$$P(Y=1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}} \quad (1)$$

- **History:** Originated in statistics, widely used in economics. Applied to machine learning for binary classification tasks.
- **Working Principle:** Fits a logistic curve to the data, mapping input features to probabilities. A threshold is applied for classification.

Mathematical Formulation

- **Mathematical Formulation:**

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}} \quad (1)$$

- **History:** Originated in statistics, widely used in economics. Applied to machine learning for binary classification tasks.
- **Working Principle:** Fits a logistic curve to the data, mapping input features to probabilities. A threshold is applied for classification.

- **Mathematical Formulation:**

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}} \tag{1}$$

- **History:** Originated in statistics, widely used in economics. Applied to machine learning for binary classification tasks.
- **Working Principle:** Fits a logistic curve to the data, mapping input features to probabilities. A threshold is applied for classification.

Logistic Regression Example

- **Problem:** Predict whether an email is spam or not.
- **Mathematical Formulation:**

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \ldots + \beta_n \cdot X_n)}}$$

- **Parameters:**

$$\begin{aligned} P(Y = 1) &: \text{Probability of email being spam} \\ X_1, X_2, \ldots, X_n &: \text{Features of the email} \\ \beta_0, \beta_1, \ldots, \beta_n &: \text{Model coefficients} \end{aligned}$$

- **Training:** Adjust coefficients using training data to maximize likelihood.

Logistic Regression Example

- **Problem:** Predict whether an email is spam or not.
- **Mathematical Formulation:**

$$P(Y=1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \ldots + \beta_n \cdot X_n)}}$$

- **Parameters:**

$$P(Y=1) : \text{Probability of email being spam}$$
$$X_1, X_2, \ldots, X_n : \text{Features of the email}$$
$$\beta_0, \beta_1, \ldots, \beta_n : \text{Model coefficients}$$

- **Training:** Adjust coefficients using training data to maximize likelihood.

Logistic Regression Example

- **Problem:** Predict whether an email is spam or not.
- **Mathematical Formulation:**

$$P(Y=1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \ldots + \beta_n \cdot X_n)}}$$

- **Parameters:**

$$\begin{aligned} P(Y=1) &: \text{Probability of email being spam} \\ X_1, X_2, \ldots, X_n &: \text{Features of the email} \\ \beta_0, \beta_1, \ldots, \beta_n &: \text{Model coefficients} \end{aligned}$$

- **Training:** Adjust coefficients using training data to maximize likelihood.

Logistic Regression Example

- **Problem:** Predict whether an email is spam or not.
- **Mathematical Formulation:**

$$P(Y=1) = \frac{1}{1 + e^{-(\beta_0+\beta_1 \cdot X_1+\beta_2 \cdot X_2+\ldots+\beta_n \cdot X_n)}}$$

- **Parameters:**

$$\begin{aligned} P(Y=1) &: \text{Probability of email being spam} \\ X_1, X_2, \ldots, X_n &: \text{Features of the email} \\ \beta_0, \beta_1, \ldots, \beta_n &: \text{Model coefficients} \end{aligned}$$

- **Training:** Adjust coefficients using training data to maximize likelihood.

Logistic Regression Example

- **Example:**
  - $\beta_0 = -5$
  - $\beta_1 = 0.1$ (positive coefficient for the presence of a certain word)
  - $\beta_2 = 0.2$ (negative coefficient for the absence of another word)
- **Prediction:**

$$P(Y = 1) = \frac{1}{1 + e^{-(\ldots)}}$$

If $P(Y = 1) > 0.5$, classify as spam.

Logistic Regression Example

- **Example:**
  - $\beta_0 = -5$
  - $\beta_1 = 0.1$ (positive coefficient for the presence of a certain word)
  - $\beta_2 = 0.2$ (negative coefficient for the absence of another word)
- **Prediction:**

$$P(Y = 1) = \frac{1}{1 + e^{-(\ldots)}}$$

If $P(Y = 1) > 0.5$, classify as spam.

Logistic Regression: Hand Calculations

- **Problem:** Predict whether a student passes ($y = 1$) or fails ($y = 0$) based on the number of hours studied.
- **Data:**

Hours Studied	Pass (1) / Fail (0)
2	0
3	0
4	0
5	1
6	1

- **Model:**

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot \text{Hours Studied})}}$$

- **Parameters:** $\beta_0 = 0.5, \beta_1 = 0.4$
- **Prediction:**

$$P(Y = 1) = \frac{1}{1 + e^{-(-0.5 + 0.4 \cdot \text{Hours Studied})}}$$

- **Problem:** Predict whether a student passes ($y = 1$) or fails ($y = 0$) based on the number of hours studied.
- **Data:**

Hours Studied	Pass (1) / Fail (0)
2	0
3	0
4	0
5	1
6	1

- **Model:**

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot \text{Hours Studied})}}$$

- **Parameters:** $\beta_0 = 0.5, \beta_1 = 0.4$
- **Prediction:**

$$P(Y = 1) = \frac{1}{1 + e^{-(-0.5 + 0.4 \cdot \text{Hours Studied})}}$$

Logistic Regression: Hand Calculations

- **Problem:** Predict whether a student passes ($y = 1$) or fails ($y = 0$) based on the number of hours studied.
- **Data:**

Hours Studied	Pass (1) / Fail (0)
2	0
3	0
4	0
5	1
6	1

- **Model:**

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot \text{Hours Studied})}}$$

- **Parameters:** $\beta_0 = 0.5, \beta_1 = 0.4$
- **Prediction:**

$$P(Y = 1) = \frac{1}{1 + e^{-(-0.5 + 0.4 \cdot \text{Hours Studied})}}$$

Logistic Regression: Hand Calculations

- **Problem:** Predict whether a student passes ($y = 1$) or fails ($y = 0$) based on the number of hours studied.
- **Data:**

Hours Studied	Pass (1) / Fail (0)
2	0
3	0
4	0
5	1
6	1

- **Model:**

$$P(Y=1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot \text{Hours Studied})}}$$

- **Parameters:** $\beta_0 = 0.5, \beta_1 = 0.4$
- **Prediction:**

$$P(Y=1) = \frac{1}{1 + e^{-(-0.5 + 0.4 \cdot \text{Hours Studied})}}$$

Logistic Regression: Hand Calculations

- **Problem:** Predict whether a student passes ($y = 1$) or fails ($y = 0$) based on the number of hours studied.
- **Data:**

Hours Studied	Pass (1) / Fail (0)
2	0
3	0
4	0
5	1
6	1

- **Model:**

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot \text{Hours Studied})}}$$

- **Parameters:** $\beta_0 = 0.5, \beta_1 = 0.4$
- **Prediction:**

$$P(Y = 1) = \frac{1}{1 + e^{-(-0.5 + 0.4 \cdot \text{Hours Studied})}}$$

Model Evaluating accuracy

- After we train a logistic regression model on some training data, we will evaluate our model's performance on some test data.
- We can use a confusion matrix to evaluate classification models.
- We can use a confusion matrix to evaluate our model.
- For example, imagine testing for disease
- Example:
  Test for presence of disease
  NO = negative test = False = 0
  YES = positive test = True = 1.

Model Evaluating accuracy

- After we train a logistic regression model on some training data, we will evaluate our model's performance on some test data.
- We can use a confusion matrix to evaluate classification models.
- We can use a confusion matrix to evaluate our model.
- For example, imagine testing for disease
- Example:
  Test for presence of disease
  NO = negative test = False = 0
  YES = positive test = True = 1.

Model Evaluating accuracy

- After we train a logistic regression model on some training data, we will evaluate our model's performance on some test data.
- We can use a confusion matrix to evaluate classification models.
- We can use a confusion matrix to evaluate our model.
- For example, imagine testing for disease
- Example:
  Test for presence of disease
  NO = negative test = False = 0
  YES = positive test = True = 1.

Model Evaluating accuracy

- After we train a logistic regression model on some training data, we will evaluate our model's performance on some test data.
- We can use a confusion matrix to evaluate classification models.
- We can use a confusion matrix to evaluate our model.
- For example, imagine testing for disease
- Example:
  Test for presence of disease
  NO = negative test = False = 0
  YES = positive test = True = 1.

Model Evaluating accuracy

- After we train a logistic regression model on some training data, we will evaluate our model's performance on some test data.
- We can use a confusion matrix to evaluate classification models.
- We can use a confusion matrix to evaluate our model.
- For example, imagine testing for disease
- Example:
  Test for presence of disease
  NO = negative test = False = 0
  YES = positive test = True = 1.

Confusion Matrix

- Confusion Matrix

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Basic Terminology:

- True Positives (TP)
- True Negatives (TN)
- False Positives (FP)
- False Negatives (FN)

Python Code: Logistic Regression

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# Assuming 'X' is feature matrix and 'y' is target varia
X_train, X_test, y_train, y_test = train_test_split(X, y

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

# Making predictions
predictions = model.predict(X_test)

# Evaluating accuracy
accuracy = accuracy_score(y_test, predictions)
```

Great Job
Thank yo