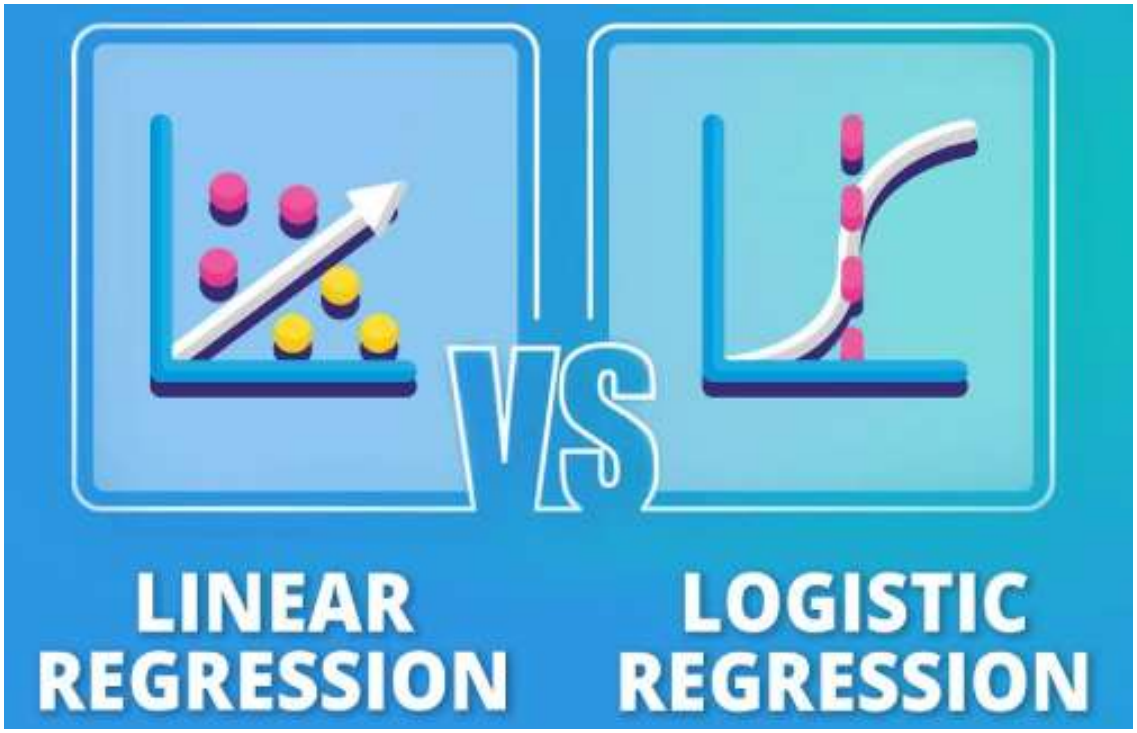
Code Academy
أكاديمية البرمجة

Loss Functions in Linear and Logistic Regression



Author: Ibreez Said Al-Aufi

Date: 16/5/2025

1. Introduction

In machine learning, regression algorithms are used to predict continuous or categorical outcomes based on input features. Two of the most common regression techniques are Linear Regression and Logistic Regression.

- Linear Regression is used for predicting continuous numerical values.
- Logistic Regression is used for binary classification problems, predicting probabilities of classes.

Both algorithms learn by minimizing a loss function, which measures how well the model's predictions match the true values.

2. What is a Loss Function?

A loss function quantifies the difference between predicted values and actual target values. The goal of training a machine learning model is to minimize this loss, thus improving prediction accuracy.

Different types of problems require different loss functions:

- For regression problems predicting continuous values, Mean Squared Error (MSE) is commonly used.
- For classification problems like logistic regression, Log Loss (Cross-Entropy Loss) is used.

3. Loss Function in Linear Regression

Linear Regression models the relationship between the input feature x and the continuous output y as:

$$\hat{y} = w\,x + b$$

where w is the weight (slope), and b is the bias (intercept).

The most common loss function for linear regression is Mean Squared Error (MSE):

$$\text{MSE} = \left(\frac{1}{n}\right)\sum(y_i - \hat{y}_i)^2$$

This function calculates the average of the squared differences between actual values y_i and predicted values $\hat{y}_i$. Squaring ensures all errors are positive and penalizes larger errors more.

4. Loss Function in Logistic Regression

Logistic Regression is used for binary classification problems, where the output y can be 0 or 1. The model predicts the probability of the positive class P(y=1|x) using the sigmoid function:

$$\hat{y} = \sigma(z) = \frac{1}{(1 + e^{-z})}, \text{where } z = w\,x + b$$

The loss function for logistic regression is Log Loss (also called Cross-Entropy Loss):

$$\text{Log Loss} = -\left(\frac{1}{n}\right)\sum [\, y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Log Loss measures the performance of a classification model whose output is a probability between 0 and 1. It penalizes wrong confident predictions more heavily.

5. Example: Applying Linear and Logistic Regression

To understand how loss functions, work in practice, we will demonstrate both **Linear Regression** and **Logistic Regression** using example datasets:

- For **Linear Regression**, we generate data based on a linear relationship with added noise.
- For **Logistic Regression**, we create a simple binary classification dataset.

We will then fit the appropriate models, calculate their respective loss functions (Mean Squared Error for linear, Log Loss for logistic), and plot the results to visualize how the models behave on real data.

Linear Regression

We simulate continuous data by generating random data points along a line with some noise. After fitting a linear regression model to this data, we calculate the **Mean Squared Error (MSE)**, which quantifies how far our predicted values are from the actual ones. The data points and the regression line are then visualized to show the model's fit.

Logistic Regression

We generate a binary classification dataset with two classes. A logistic regression model is trained to estimate the probability that a data point belongs to the positive class. The model's performance is evaluated using **Log Loss (Cross-Entropy)**. Finally, we visualize the classification data along with the sigmoid curve representing predicted probabilities.

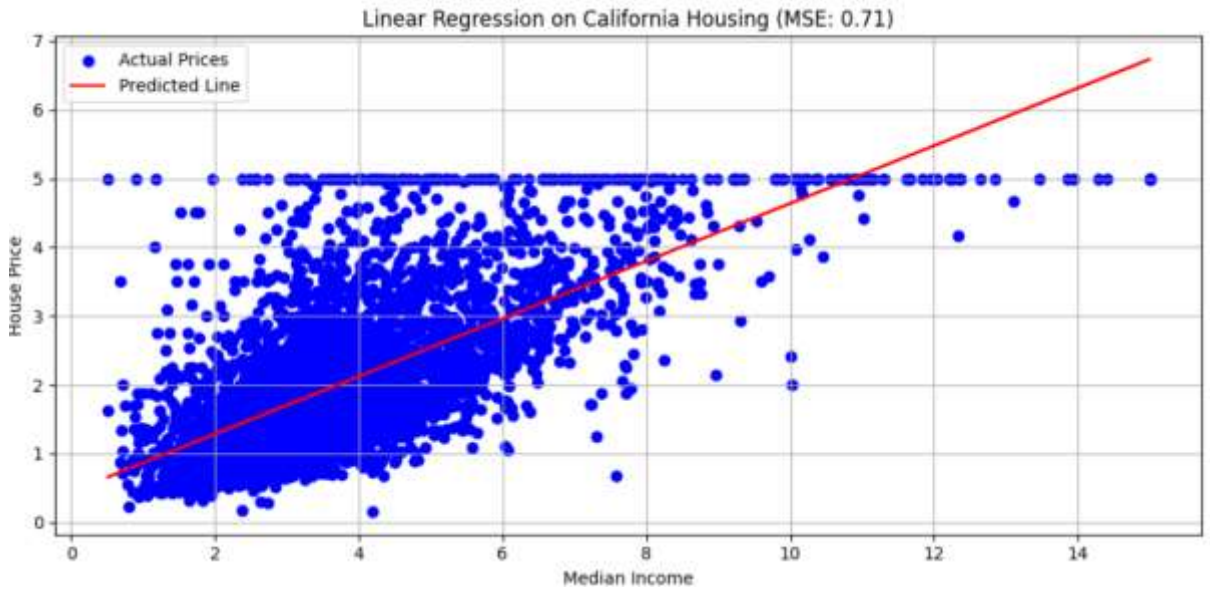
6. Python Implementation and Visualization

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_california_housing
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split

# Load dataset
data = fetch_california_housing()
X = data.data[:, [0]]  # Use only 'MedInc' (median income) as a feature for visualization
y = data.target

# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Fit model
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
```

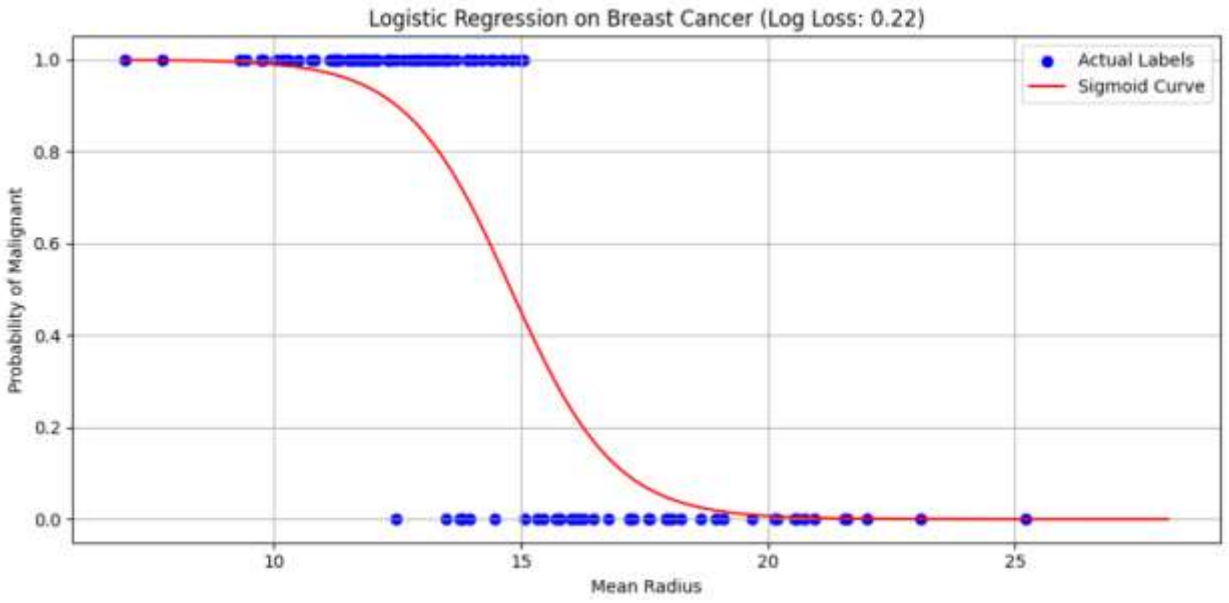


```python
from sklearn.datasets import load_breast_cancer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import log_loss, accuracy_score

# Load dataset
data = load_breast_cancer()
X = data.data[:, [0]]  # Use only 'mean radius' for visualization
y = data.target

# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Fit logistic regression
log_model = LogisticRegression()
log_model.fit(X_train, y_train)
y_prob = log_model.predict_proba(X_test)[:, 1]
logloss = log_loss(y_test, y_prob)
```



7. Interpretation of Results

- The linear regression plot shows data points scattered around the red regression line. The MSE value indicates how closely the line fits the data — lower MSE means better fit.

- The logistic regression plot shows the data points labeled 0 or 1, with a sigmoid curve representing the predicted probability of class 1. The log loss reflects the accuracy of these probability predictions — lower log loss means better classification performance.

8. Conclusion

Loss functions play a crucial role in training machine learning models by quantifying prediction errors. Linear regression uses MSE to measure error for continuous predictions, while logistic regression uses log loss to evaluate probability predictions in classification tasks. Visualizing these loss functions alongside data helps understand how models learn from data and improve predictions.