# Logistic Regression – Module 2 Comprehensive Guide

## 📌 Introduction to Logistic Regression

Logistic Regression is a fundamental classification technique used in supervised learning to predict categorical outcomes. Unlike Linear Regression, which predicts continuous values, Logistic Regression transforms predictions into a probability distribution using the sigmoid function.

Key Applications:

✔ Spam detection (spam vs. not spam).

✔ Disease diagnosis (positive vs. negative test results).

✔ Loan approval (approve vs. reject).

✔ Customer churn prediction (stay vs. leave).

### 🔹 Logistic Regression Model

Logistic Regression applies the sigmoid (logistic) function to transform linear outputs into probabilities between 0 and 1.

**Mathematical Representation:**

Where:

* If p(ŷ) ≥ 0.5, classify as Class 1.
* Otherwise, classify as Class 0.
* The decision boundary is determined by adjusting the threshold.

## 🔹 Training a Logistic Regression Model

Training a Logistic Regression model involves optimizing parameters (θ) to minimize prediction error.

**Log-Loss Function (Cross-Entropy Loss):**

* Penalizes incorrect confident predictions.
* The goal is to minimize log-loss to improve classification accuracy.

## 🔹 Optimization Techniques

There are two common optimization methods for training Logistic Regression:

# 1️. Gradient Descent: Iteratively updates parameters to minimize log-loss.

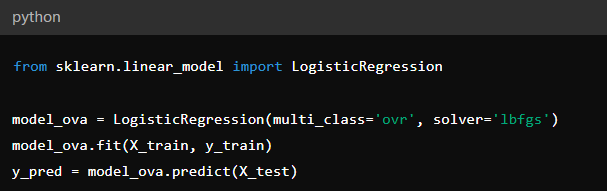
# 2. Stochastic Gradient Descent (SGD): A faster variation using random subsets of data for optimization.

## 🔹Multi-Class Classification with Logistic Regression

Logistic Regression can be extended for multi-class classification using:

### One-vs-All (OvA):

* If there are K classes in total, OvA strategy will train K binary classifiers, each one distinguishing one class (positive) from all other classes (negative).
* When making predictions, each classifier outputs a probability score.
* The class corresponding to the classifier with the highest probability is selected as the prediction.
* **Advantages of OvA:**
  + Simpler and computationally efficient (only K classifiers).
  + Works well when there are a moderate number of classes.
* **Disadvantages of OvA:**
  + Can produce imbalanced classifiers (one class vs. all others).
  + Might not perform as well when classes overlap heavily.



* Example: Suppose you have 3 classes:
  + Class A, Class B, Class C.

OvA would create three separate classifiers:

* + Classifier 1: **Class** **A** vs. **Classes** **B & C**
  + Classifier 2: **Class** **B** vs. **Classes** **A & C**
  + Classifier 3: **Class** **C** vs. **Classes** **A** **&** **B**

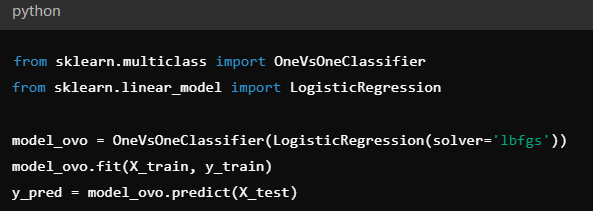
To predict, each classifier calculates probabilities for a data point, for example:

|  |  |
| --- | --- |
| Classifier | Probability |
| A vs. All | 0.35 ✅ |
| B vs. All | 0.30 |
| C vs. All | 0.25 |

Since Classifier 1 has the highest probability (0.35), the model predicts Class A.

### One-vs-One (OvO):

* For K classes, the OvO strategy trains K(K-1)/2 binary classifiers, each classifier distinguishing between a pair of classes.
* When making predictions, each classifier votes for one class.
* The class with the majority of votes across all classifiers wins.
* **Advantages of OvO:**
  + Each classifier handles a simpler task (only distinguishing between two classes).
  + Can handle class imbalance better.
* **Disadvantages of OvO:**
  + Computationally more expensive, especially with many classes (requires training many classifiers).
  + Can result in tie votes (solved by confidence scores or probability estimates).



* Example: For the same scenario (Classes A, B, C), OvO creates 3 classifiers:
  + Classifier 1: Class A vs. Class B
  + Classifier 2: Class B vs. Class C
  + Classifier 3: Class A vs. Class C
* To predict, each classifier gives a vote, for example:

|  |  |
| --- | --- |
| Classifier | Winner |
| A vs. B | A ✅ |
| B vs. C | C ✅ |
| A vs. C | A ✅ |

* + Votes: A = 2, C = 1, B = 0.
  + The predicted class is Class A, with the majority vote.

**📌 When to Use OvA vs. OvO:**

|  |  |  |
| --- | --- | --- |
| Consideration | OvA Strategy | OvO Strategy |
| Number of classes | Moderate | Large |
| Computational resources available | Low (efficient) | High (expensive) |
| Class imbalance | Less robust | More robust |
| Model interpretability | Higher (fewer models) | Lower (more models) |

## 🔹 Evaluating Logistic Regression Models

Evaluation metrics for classification tasks:

### 1. Accuracy:

Measures the proportion of correct predictions over total predictions.

**What it measures:** Overall effectiveness of the model, but can be misleading for imbalanced datasets.

### 2. Precision:

Measures the proportion of positive identifications that were actually correct.

**What it measures:** How reliable your model is when it claims a positive result. High precision means fewer false alarms.

### 3. Recall (Sensitivity):

Measures the proportion of actual positives correctly identified by the model.

**What it measures:** The model’s ability to detect positive instances. High recall ensures fewer false negatives (missed detections).

### 4. F1-Score:

Balances precision and recall into a single metric (harmonic mean).

**What it measures:** Overall balance between precision and recall. Useful when the dataset has uneven class distributions.

### 4. ROC Curve & AUC Score:

* **ROC Curve:** Plots the True Positive Rate (Sensitivity) against False Positive Rate, showing the trade-off between sensitivity and specificity.
* **AUC (Area Under Curve):** A single number summarizing the ROC curve. It quantifies the model’s ability to distinguish between classes (0.5 = random guessing, 1.0 = perfect discrimination).

**What it measures:** Model’s ability to discriminate between classes at different classification thresholds.

### 5. Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| PredictedActual | Positive | Negative |
| Positive | TP (True Positive) | FN (False Negative) |
| Negative | FP (False Positive) | TN (True Negative) |

* **TP (True Positive):** Correctly predicted positive outcomes.
* **TN (True Negative):** Correctly predicted negative outcomes.
* **FP (False Positive):** Incorrectly predicted positive outcomes (false alarms).
* **FN (False Negative):** Missed predictions for positive outcomes.

## 🔹 Insights from Labs:

Experiments demonstrated:

* The impact of feature selection and scaling on performance.
* How regularization (L1/L2) can prevent overfitting.
* The effect of adjusting the decision threshold on precision-recall trade-offs.

## 🔹 Key Takeaways

✅ Logistic Regression is essential for binary and multi-class classification tasks.

✅ The sigmoid function transforms linear predictions into probabilities.

✅ Log-loss is minimized to improve classification accuracy.

✅ Gradient Descent and SGD efficiently optimize logistic regression parameters.

✅ Evaluation metrics like precision, recall, and F1-score critically assess model performance.