Softmax Regression | Multinomial Logistic Regression

Softmax Regression

- **Purpose:** Extends binary logistic regression to **multi-class classification** problems.
- **Idea:** Instead of predicting a single probability (for class 1), Softmax outputs a probability distribution across all classes.
- **Softmax Function:**

$$P(y = k|\mathbf{x}) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

- z_k = linear combination of inputs for class kk
- Ensures probabilities are **between 0 and 1** and **sum to 1**.

Training Intuition

- Similar to logistic regression, we adjust model weights to minimize a loss function.
- For multi-class, we use **cross-entropy loss** instead of binary log loss.
- Gradient Descent updates all class weight vectors simultaneously.

Prediction

1. Compute the logits (z_k) for each class.
2. Apply Softmax to convert logits into probabilities.
3. Pick the class with the highest probability as the prediction.

Loss Function

- **Cross-Entropy Loss for multi-class:**

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log(p_{ik})$$

- y_{ik} = 1 if sample ii belongs to class k, else 0
- kp_{ik} = predicted probability for class kk
- Penalizes confident wrong predictions heavily.

Softmax Demo in Scikit-Learn

- Using LogisticRegression(multi_class='multinomial', solver='lbfgs') for multi-class problems.
- SKLearn handles the SoftMax computation and optimization internally.