Modeling Section Summary – Logistic Regression (One-vs-Rest) from Scratch

Model Selection and Implementation:

- **Chosen Model:** Logistic Regression implemented from scratch, extended to multi-class classification using the **One-vs-Rest (OvR)** strategy.
- **Why OvR:** Logistic regression is inherently binary. OvR trains separate classifiers for each class to enable multi-class predictions by selecting the class with the highest predicted probability.

Model Architecture:

- **Sigmoid Function:** Used for mapping outputs to probabilities.
- **Loss Function:** Binary Cross-Entropy computed per class.
- **Gradient Descent:** Manual implementation to iteratively minimize loss.
- **OvR Strategy:** For k classes, train k binary classifiers (1 vs. all).

Training and Hyperparameter Tuning:

- **Hyperparameters:**
    - Learning Rate (alpha): default set to 0.01
    - Number of Iterations: default 1000
- **Trial Results (Example Tuning):**
    - **Alpha = 0.01, Iterations = 1000** → Slower convergence but stable training.
    - **Alpha = 0.1, Iterations = 1000** → Faster convergence but risk of overshooting cost minima.
    - Observed trade-off: Higher learning rate improves speed but can degrade stability; lower values improve stability but may slow convergence.

Model Evaluation:

- **Predictions:** Made by computing sigmoid outputs for each classifier, then selecting the class with the highest probability.

- **Evaluation Metrics Used:**
  - **Accuracy Score**
  - **Classification Report:** Includes **precision**, **recall**, and **F1-score** per class.
  - **Confusion Matrix:** Visualized using Seaborn heatmap to observe misclassification patterns.

Conclusion (Model-Focused):

The scratch implementation of logistic regression using One-vs-Rest is functional and educationally valuable. It enables multi-class classification and gives insight into optimization techniques. However, for production or large datasets, advanced techniques like vectorization, regularization, and automated hyperparameter tuning would be essential enhancements.