**Aim:** Develop and evaluate logistic regression models for multi-class classification tasks using machine learning.

**Software Used:** Google Collaborative Python

**Theory:**

**Logistic Regression**

Logistic Regression is a supervised machine learning algorithm used for classification problems. Unlike linear regression which predicts continuous values it predicts the probability that an input belongs to a specific class. It is used for binary classification where the output can be one of two possible categories such as Yes/No, True/False or 0/1. It uses sigmoid function to convert inputs into a probability value between 0 and 1. In this article, we will see the basics of logistic regression and its core concepts.

**Logistic Regression for Multi-Class Classification** is an extension of binary logistic regression used when the target variable contains more than two classes. Instead of predicting a probability for just two classes, the model estimates probabilities for all possible classes and assigns the one with the highest probability.

**Concept:**

* Unlike binary classification, where the output is either 0 or 1, multi-class classification deals with multiple categories (e.g., predicting if a fruit is apple, banana, or orange).
* The algorithm uses a generalized form of logistic regression to handle more than two outcomes.

**Approaches:**

* One-vs-Rest (OvR): Trains one binary classifier per class, treating that class as "positive" and all others as "negative".
* Multinomial Logistic Regression: Directly models all classes simultaneously using the softmax function.

**Mathematical Model:**

* For K classes, the probability that an observation belongs to class jjj is calculated as:



* Here, the softmax function ensures all probabilities sum to 1.

**Model Training:**

* The model parameters are estimated using Maximum Likelihood Estimation (MLE).
* Optimization algorithms like Gradient Descent are used to minimize the loss function (often Cross-Entropy Loss).

**Advantages:**

* Works well for linearly separable classes.
* Produces probabilistic outputs, which can be useful in decision-making.

**Limitations:**

* Assumes linear decision boundaries between classes.
* May underperform if classes are highly imbalanced or non-linear in nature.

**Evaluation Metrics:**

* Accuracy – percentage of correctly classified samples.
* Precision, Recall, F1-score – calculated for each class.
* Confusion Matrix – for detailed class-wise performance.

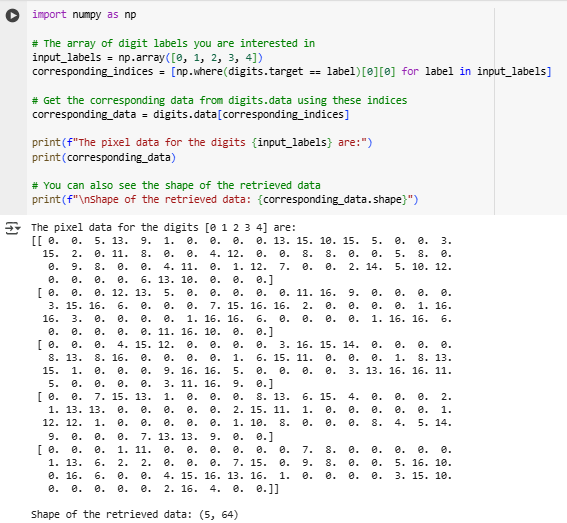
**Implementation in Python:**

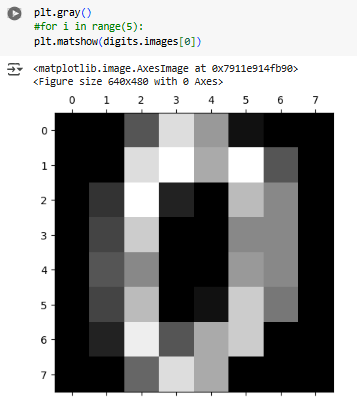
* Libraries like Scikit-learn provide built-in support for multi-class logistic regression.
* Common parameters include multi\_class='multinomial'.

**Output:**

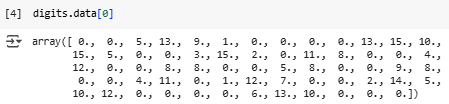
**1)Digit Dataset**

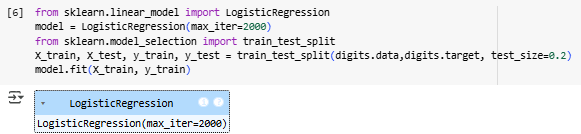


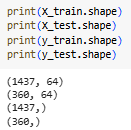






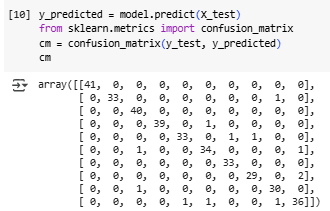


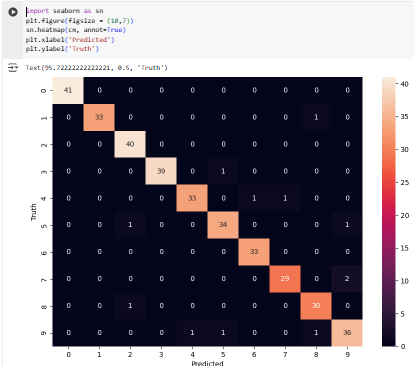


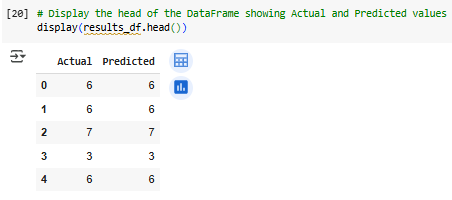




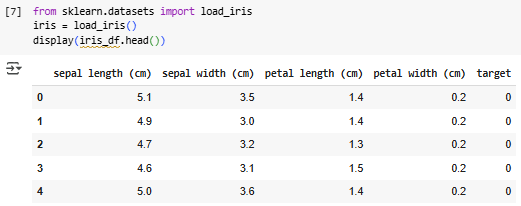


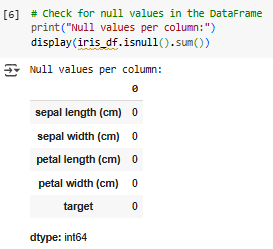


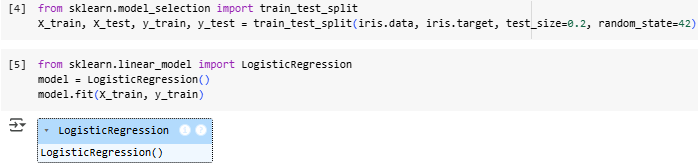


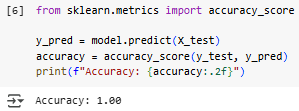


**2)Iris**

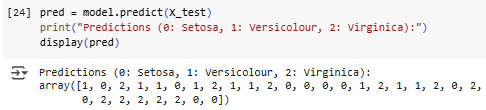


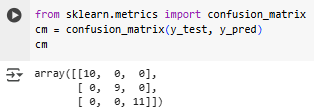


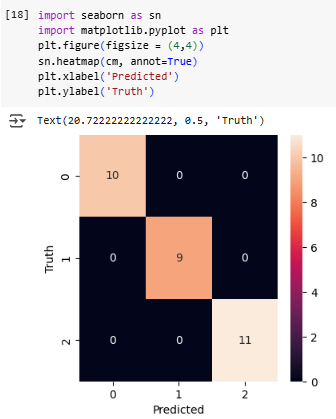


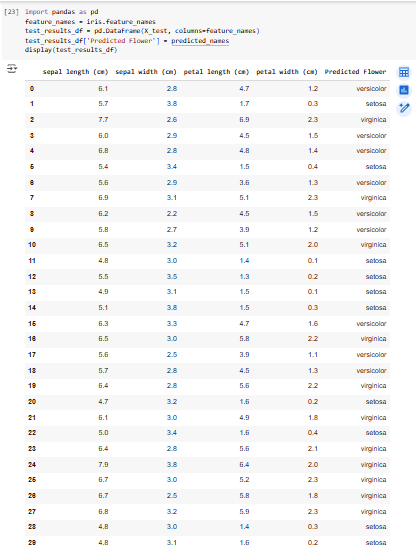


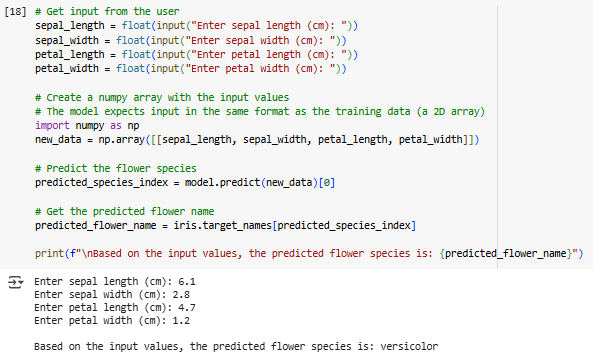












**Conclusion:** Hence, through this experiment, we learned how to develop and evaluate Logistic Regression models for multi-class classification tasks, train them using approaches like One-vs-Rest and Multinomial Logistic Regression, and assess performance. This process is essential for building reliable multi-class models that can accurately classify data into multiple categories.