**Wine Classification with Multi-Class Logistic Regression**

**Introduction**

Wine classification is a significant application of machine learning, enabling the categorization of wines based on their chemical properties. This project utilizes the UCI Wine Dataset, which contains 178 samples with 13 chemical features (e.g., alcohol, proline, flavanoids) and three class labels representing different cultivars (Classes 1, 2, and 3). By employing Multi-Class Logistic Regression with the softmax approach, the model accurately predicts the cultivar of a wine sample. This project demonstrates the application of supervised learning techniques to achieve perfect classification performance on a well-separated dataset.

**Problem Statement**

The challenge is to classify wines into three cultivars (Classes 1, 2, and 3) based on a dataset of 178 samples with 13 chemical features. Incorrect classification can impact quality control and market segmentation in the wine industry. The goal is to develop a Multi-Class Logistic Regression model that achieves high accuracy and minimizes misclassification errors, leveraging the dataset’s distinct feature distributions.

**Algorithm**

Multi-Class Logistic Regression extends binary logistic regression to handle multiple classes using the softmax approach. In this project, the model computes probabilities for each class (1, 2, and 3) and assigns the wine sample to the class with the highest probability. The model is optimized using the L-BFGS solver, which applies gradient descent to minimize the cost function. Standardized features ensure robust performance across varying scales.

**Objective**

* To develop a Multi-Class Logistic Regression model to classify wines into three cultivars (Classes 1, 2, and 3) based on 13 chemical features.
* To achieve perfect classification accuracy (100%) on the test set with minimal misclassification errors.
* To evaluate the model using accuracy, precision, recall, F1-score, and a confusion matrix.
* To visualize the class distribution to understand the dataset’s balance.

**Motivation**

Accurate wine classification is crucial for quality assurance, product labeling, and market differentiation in the beverage industry. The UCI Wine Dataset provides a real-world benchmark for multi-class classification, offering insights into feature importance and model performance. This project enhances understanding of supervised learning, data preprocessing, and evaluation metrics, preparing students for practical machine learning applications.

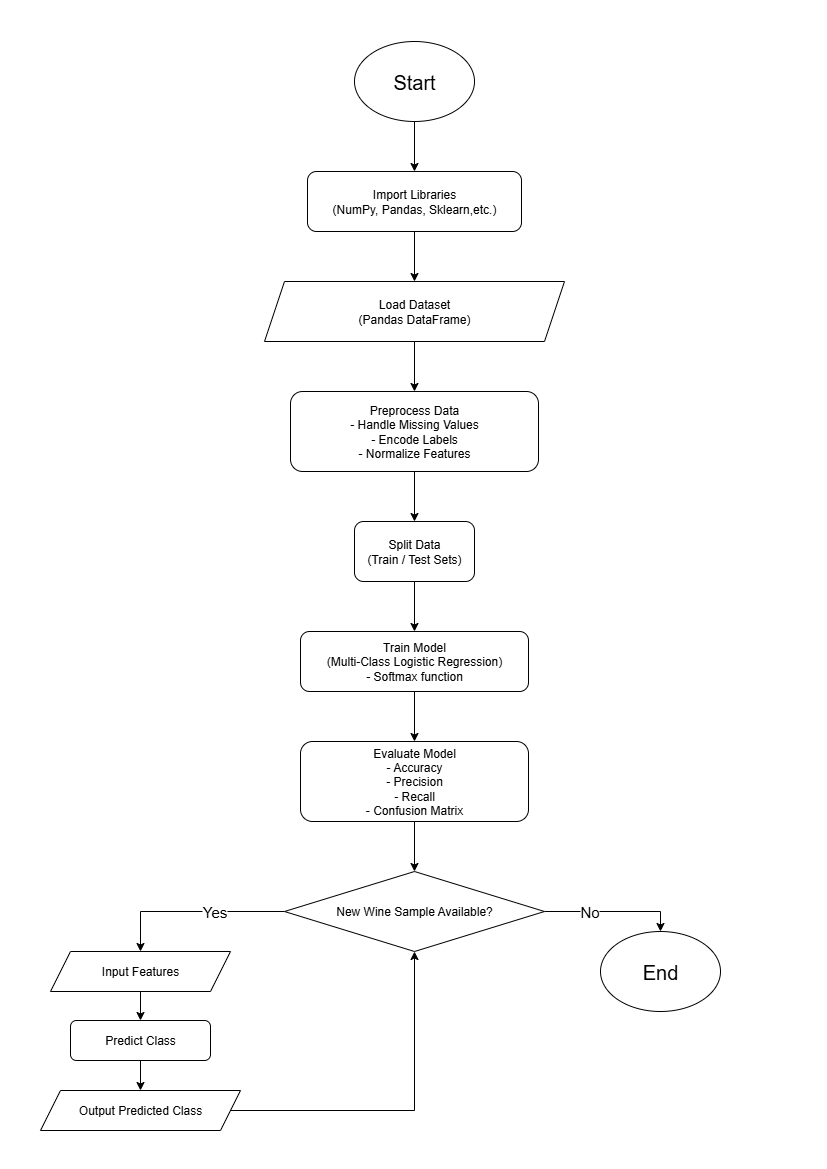
**Methodology**

* **Data Collection:** The UCI Wine Dataset is loaded using np.genfromtxt into a Pandas DataFrame, containing 178 samples with 13 features (e.g., alcohol, malic acid, proline) and class labels (1, 2, or 3).
* **Data Preprocessing:**
  + Check for missing values and duplicates (none found).
  + Detect outliers using the Interquartile Range (IQR) method for each feature.
  + Standardize features using StandardScaler to normalize scales.
  + Split the dataset into 80% training (142 samples) and 20% testing (36 samples) sets.
* **Model Training:** Implement Multi-Class Logistic Regression with the softmax approach using Scikit-learn’s LogisticRegression(solver='lbfgs'). Train the model on the standardized training data.
* **Model Evaluation:** Assess performance on the test set using accuracy, precision, recall, F1-score (weighted averages), and a confusion matrix.
* **Visualisation:** Generate a class distribution plot (countplot) to visualize the number of samples per class (59, 71, 48 for Classes 1, 2, 3).
* **Output:** Display test set predictions as a DataFrame showing true and predicted classes.

**Flow of the Program**

* **Import Libraries:** Use numpy, pandas, matplotlib, seaborn, and scikit-learn for data handling, model training, and visualization.
* **Load Dataset:** Read the UCI Wine Dataset using np.genfromtxt into a Pandas DataFrame.
* **Preprocess Data:** Check for missing values, duplicates, and outliers (IQR method). Standardize features and split data into training (80%) and testing (20%) sets.
* **Train Model:** Fit the Multi-Class Logistic Regression model using LogisticRegression(solver='lbfgs').
* **Predict and Evaluate:** Generate predictions on the test set and compute accuracy, precision, recall, F1-score, and the confusion matrix.
* **Visualise Results:** Plot the class distribution using seaborn.countplot.
* **Output Results:** Print evaluation metrics and a DataFrame of test set predictions (true vs. predicted classes).

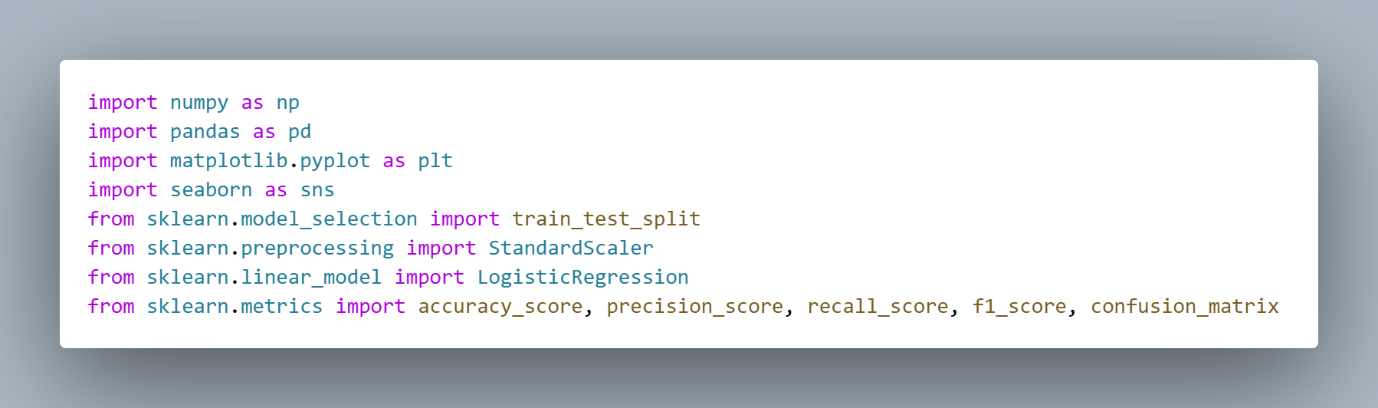
**Flow Chart**

****

**Glimpse**

Below are the key sections of the Python code used for the project.

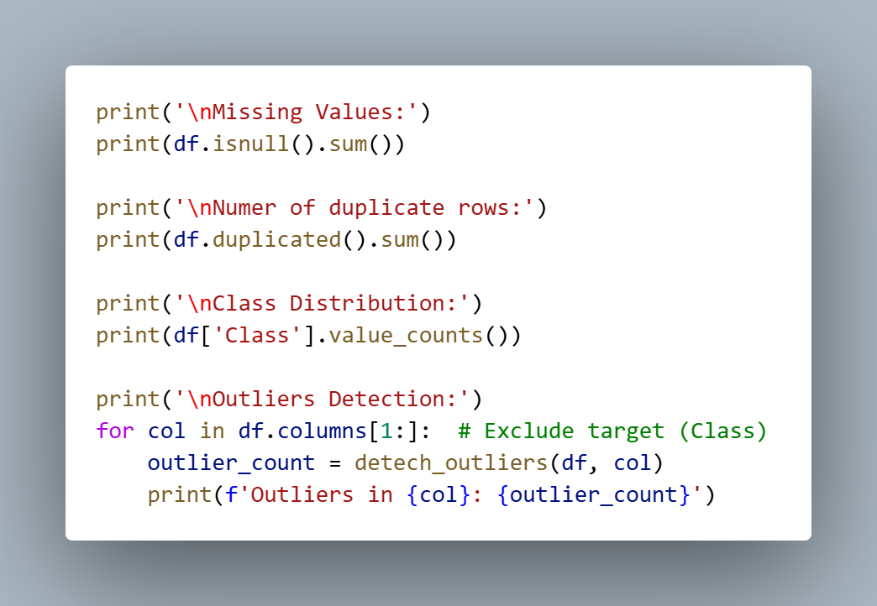
* **Libraries used**

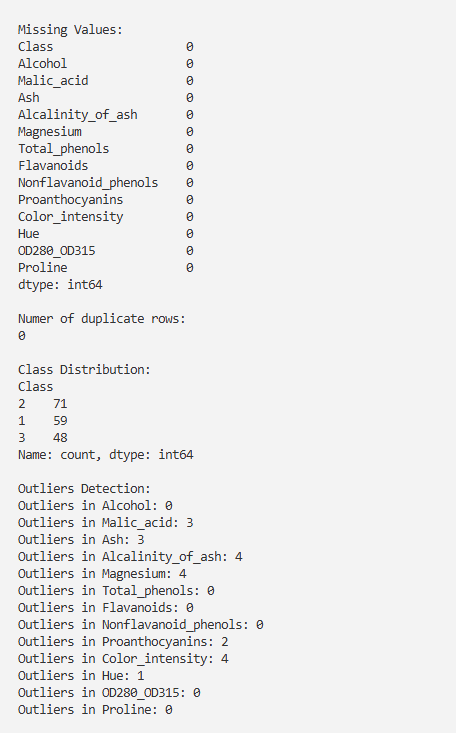
****

* **Loading dataset**

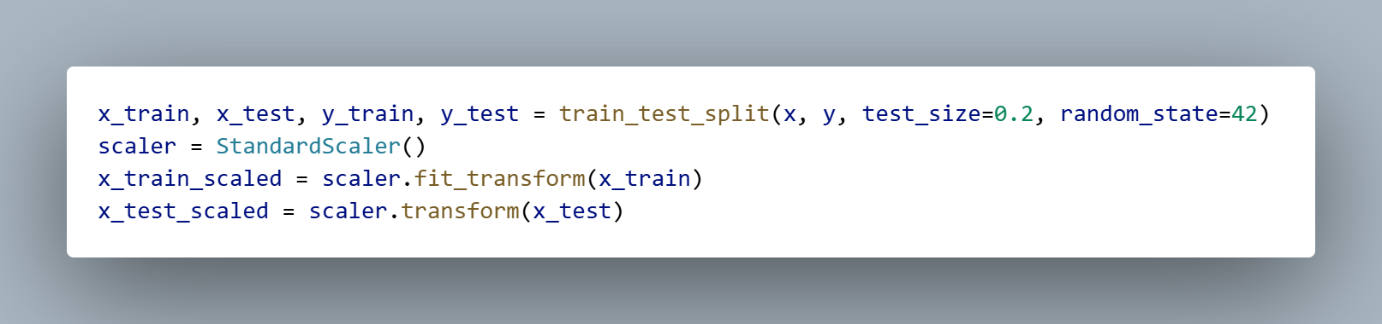
****

* **Preprocessing data**

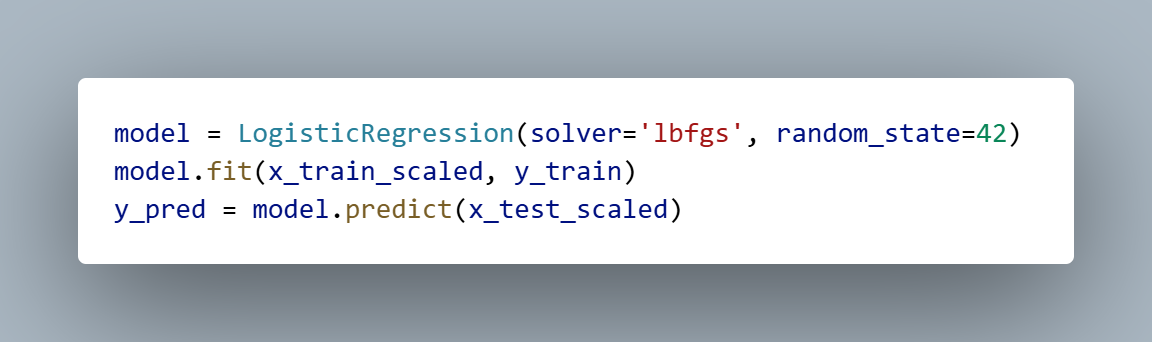
****

****

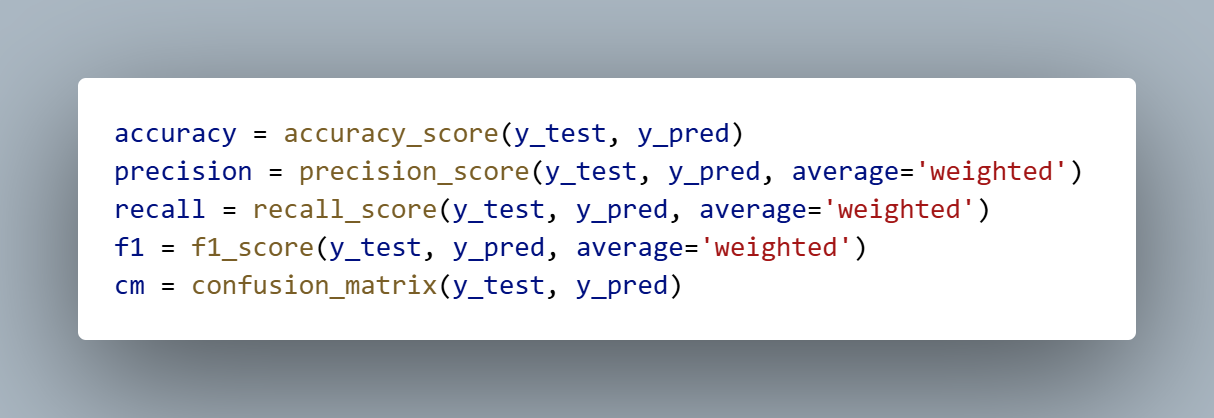
**• Split and scale**

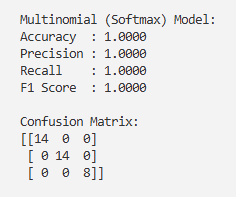
****

**• Training model**

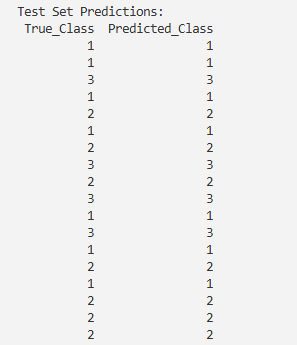
****

**• Evaluating model**

****

****

**• Output**

****

**Results**

The Multi-Class Logistic Regression model achieved 100% accuracy, precision, recall, and F1-score (weighted) on the test set (36 samples). The confusion matrix shows perfect classification, with 14, 14, and 8 samples correctly classified for Classes 1, 2, and 3, respectively. Key findings include:

**Data Quality:** No missing values or duplicates were found. Outlier detection (IQR method) identified minimal outliers across features (e.g., 4 in Magnesium, 3 in Proline), indicating a clean dataset.

**Class Distribution:** The dataset is balanced, with 59, 71, and 48 samples for Classes 1, 2, and 3, respectively, as shown in the countplot.

**Model Performance:** The model’s perfect performance is attributed to the dataset’s well-separated classes, as confirmed by the confusion matrix.

**Test Set Predictions:** A DataFrame displays true and predicted classes for the test set, with all predictions correct.

**References**

* **Scikit-learn Documentation:** Logistic Regression. Available at: <https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression>
* **UCI Machine Learning Repository:** Wine Dataset. Available at: <https://archive.ics.uci.edu/ml/datasets/wine>
* **Multi-Class Logistic Regression:** A Friendly Guide to Classifying the Many: <https://medium.com/@jshaik2452/multi-class-logistic-regression-a-friendly-guide-to-classifying-the-many-4a590c2e6c26>