0ICPC455

AY 2024-25

# Machine Learning Lab

## Quality Laboratory Manual

**Experiment No. 4**

# Study and Implementation of Multi-class Classification

Course Instructor -

**DR. TAHSEEN A. MULLA**

**ASSOCIATE PROFESSOR**

# Experiment No. 4

**Title of Experiment:** To study and implement Multi-class Classification

**Aim of Experiment:** To implement and understand the working of Logistic Regression, a statistical method used for binary classification problems in Machine Learning

**System Requirements –** Win 8 and above OS, 4GB RAM, 2.33 GHz Processor

**Software/s Needed for Experiment** – Jupyter Notebook/ Anaconda Navigator/ Google Colaboratory/ Spyder, Python 3.x [With libraries such as Numpy, Pandas, Matplotlib and Scikit-Learn]

**Experiment Outcomes –**

1. Understand the principles and fundamentals of multi-class classification
2. Gain insights into how multi-class classification fits into the broader landscape of Machine Learning models
3. Able to evaluate the performance of multi-class classification model using appropriate metrics
4. Extend the possibilities to handle multi-class classification problems with logistic regression, support vector machines (SVM), etc…

**Theory –**

Multiclass classification is a classification task with more than two classes and makes the assumption that an object can only receive one classification. A common example requiring multiclass classification would be labeling a set of fruit images that includes oranges, apples and pears

A classification task with more than two classes, e.g., classifying a set of fruit images that may be oranges, apples or pears. Multiclass classification makes the assumption that each sample is assigned to one and only one label. A fruit can be either an apple or a pear but not both at the same time

Multi-class classification is the classification technique that allows us to categorize the test data into multiple class labels present in trained data as a model prediction.

Using Logistic Regression with the one-vs-rest (OvR) approach, where the model fits a separate binary classifier for each class. Another common approach is softmax regression, which generalizes logistic regression to multi-class problems

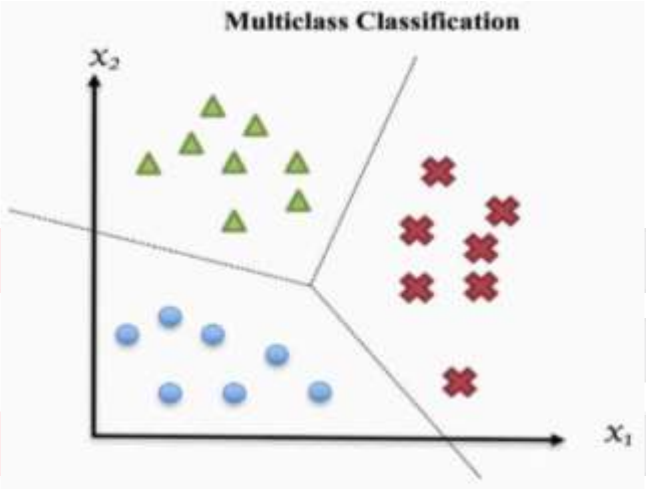
The logistic function for multi-class classification is –

$$P(y = 1) = \frac{1}{1 + \ e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n}}$$

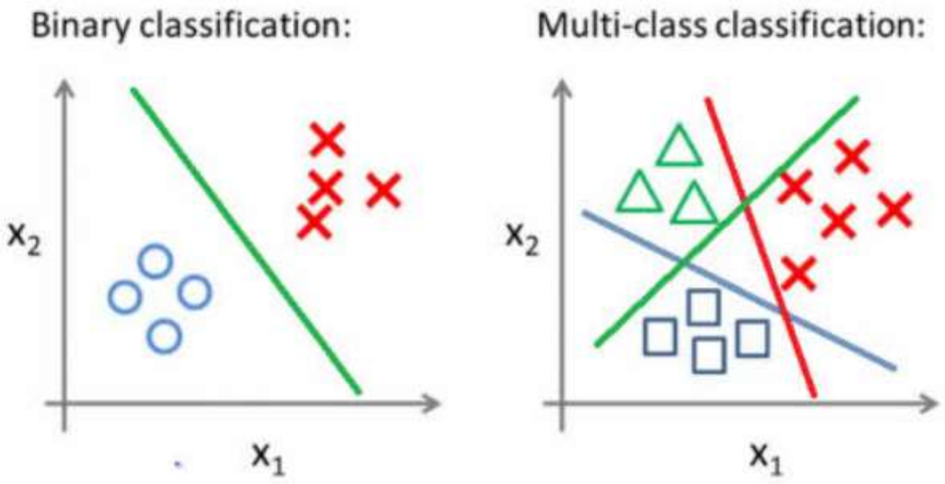
Where;
P(y=1) is the probability that the dependent variable y equals 1
$\beta_0$ is the intercept
$\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients for the independent variables $x_1, x_2, \ldots, x_n$

There are mainly two types of multi-class classification techniques –

- One vs. All (one-vs-rest)
- One vs. One



Binary Classification v/s Multi-class Classification –



**Binary Classification –**

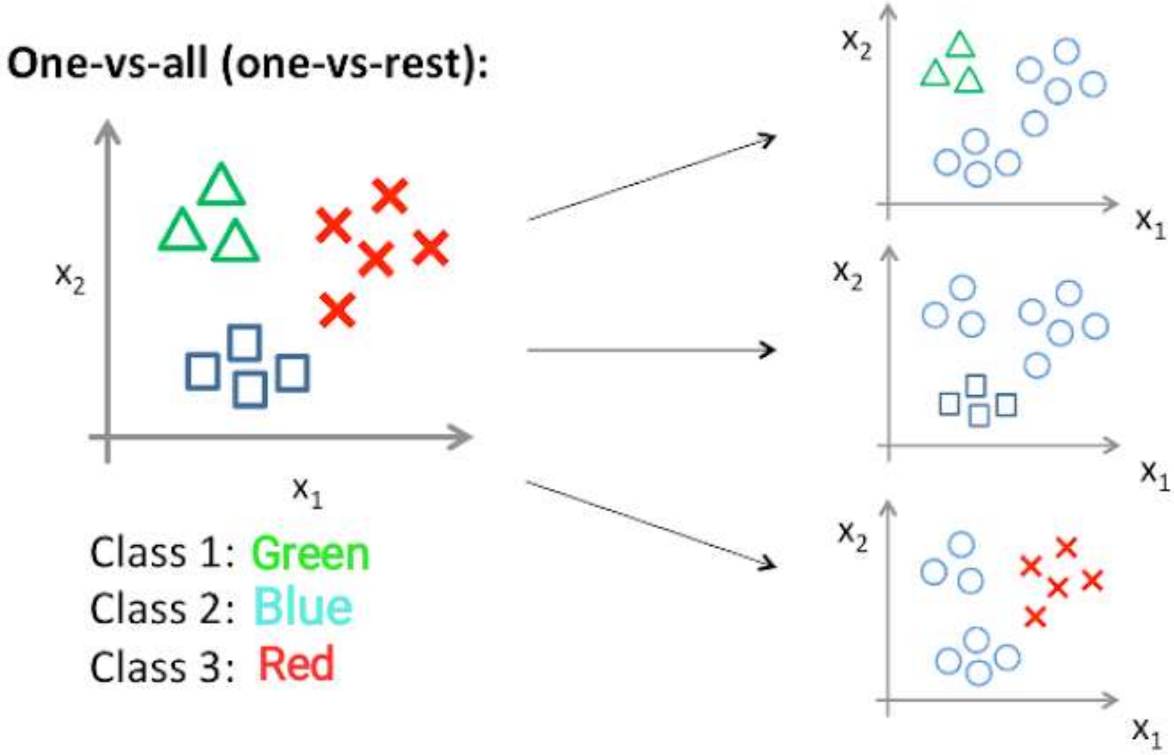
- Only two class instances are present in the dataset.
- It requires only one classifier model.
- Confusion Matrix is easy to derive and understand.
- Example - Check email is spam or not, predicting gender based on height and weight

**Multi-class Classification –**

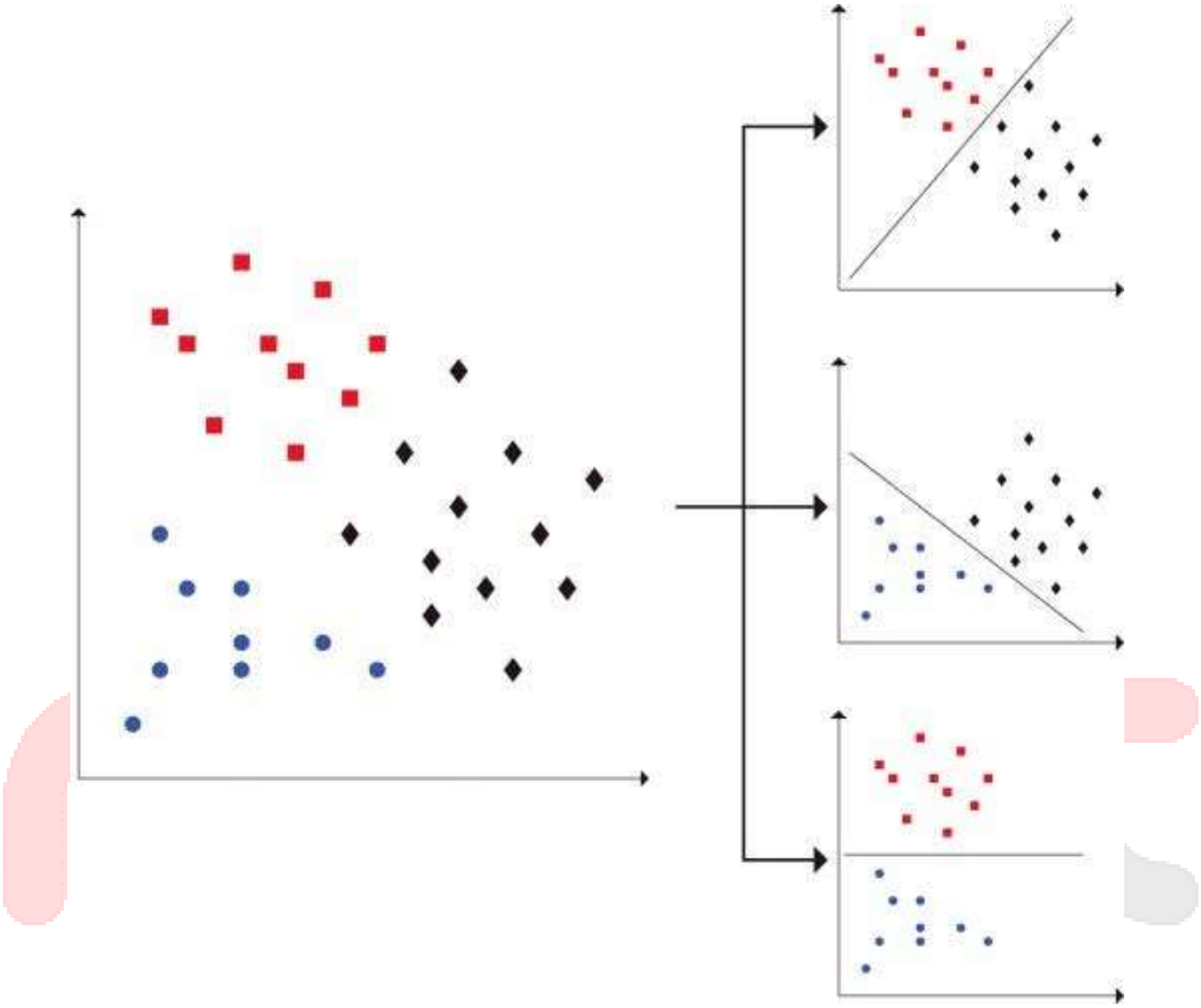
- Multiple class labels are present in the dataset.
- The number of classifier models depends on the classification technique we are applying to.
- One vs. All - N-class instances then N binary classifier models
- One vs. One - N-class instances then N* (N-1)/2 binary classifier models
- The Confusion matrix is easy to derive but complex to understand.
- Example - Check whether the fruit is apple, banana, or orange.

**One vs. All (One-v/s-Rest) –**

In one-vs-All classification, for the N-class instances dataset, we have to generate the N-binary classifier models. The number of class labels present in the dataset and the number of generated binary classifiers must be the same



**One vs. One –**

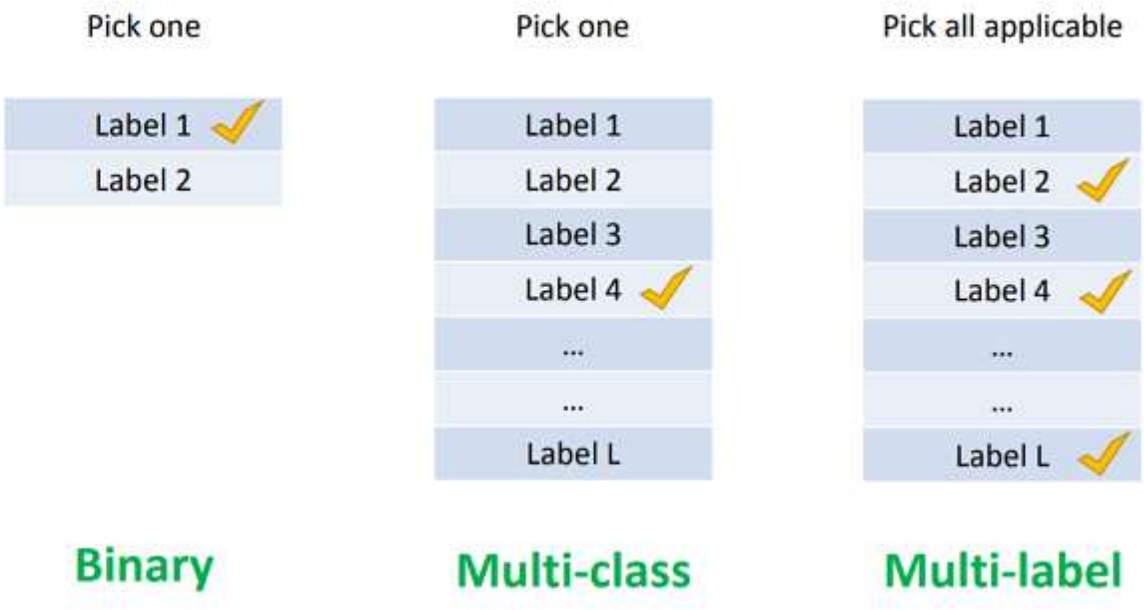


The N-class instances dataset, we have to generate the N* (N-1)/2 binary classifier models. Using this classification approach, we split the primary dataset into one dataset for each class opposite to every other class.

We divide this problem into N* (N-1)/2 = 3 binary classifier problems:

Classifier 1: Green vs. Blue
Classifier 2: Green vs. Red
Classifier 3: Blue vs. Red

Each binary classifier predicts one class label. When we input the test data to the classifier, then the model with the majority counts is concluded as a result.



**Procedure to implement Logistic Regression:**

1. **Data Collection:** Obtain a dataset suitable for multi-class classification. A dataset such as Iris, which classifies iris flowers into three species – Setosa, Versicolour and Virginica

2. **Data Preprocessing:**
   a. Load the dataset into a Pandas DataFrame.
   b. Handle missing values, outliers, and encode categorical variables if necessary.
   c. Normalize or standardize the data if needed.

3. **Exploratory Data Analysis (EDA):** Visualize relationships between the dependent and independent variables using scatter plots and correlation matrices.

4. **Splitting the Data:** Split the dataset into training and test sets to evaluate the model's performance.

5. **Implementing Multi-class Classification:**
   a. Use the Scikit-learn library to implement a multi-class classification model.
   b. Train the model using the training data.

6. **Model Prediction:**
   a. Use the trained model to make predictions on the test data.
   b. Generate a confusion matrix and classification report to assess accuracy

7. **Model Evaluation:** Evaluate the model using metrics such as accuracy, precision, recall, F1-score for each class and the overall weighted average

**Sample Code:**

1. **One v/s One -**

```
import numpy as np
from sklearn.preprocessing import LabelBinarizer

y = np.array(['apple','pear','apple','orange'])
y_dense = LabelBinarizer().fit_transform(y)
print(y_dense)

from scipy import sparse
y_sparse = sparse.csr_matrix(y_dense)
print(y_sparse)
```

2. **One v/s Rest –**

```
from sklearn import datasets
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import LinearSVC

X, y = datasets.load_iris(return_X_y=True)
OneVsRestClassifier(LinearSVC(random_state=0)).fit(X,y).predict(
X)
```

**Observations -**

- Record the predicted classes for the test data
- Observe the performance of the model using the confusion matrix and classification report

**Conclusion –**

Hence, the model summarizes the findings from the experiment, such as classifying iris species

**References –**

a. **Textbook –**
   i. Machine Learning with Python – An approach to Applied ML – Abhishek Vijayvargiya, BPB Publications, 1st Edition 2018
   ii. Machine Learning, Tom Mitchell, McGraw Hill Education, 1st Edition 1997

b. **Online references –**
   i. https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/
   ii. https://www.baeldung.com/cs/svm-multiclass-classification
   iii. https://scikit-learn.org/stable/modules/multiclass.html

**Expected Oral Questions –**

1. What is multi-class classification?
2. How does multi-class classification differ from binary classification?
3. What are some common algorithms used for multi-class classification?
4. Explain the difference between one-vs-rest (OvR) and one-vs-one (OvO) strategies in multi-class classification
5. How do interpret the output probabilities in a multi-class classification model?
6. What is confusion matrix in the context of multi-class classification, and how is it different from the binary case?
7. What are precision, recall, F1-score in the context of multi-class classification and how are they calculated for each class?
8. Explain the concept of class imbalance and how it affects multi-class classification
9. How do you evaluate the performance of a multi-class classification model?
10. State a basic example for multi-class classification?

**FAQ's in Interview –**

1. When would you use multi-class classification in a real-world application?
2. How does a classification algorithm learn from data?
3. How would you handle categorical features in a classification problem?
4. How do you evaluate the performance of a classification model?
5. What are the common classification algorithms?