**PART A**

**EXPERIMENT NO. 6**

**A.1 AIM: -** To Implement Multi-Category Classification Using Binary Linear Classifiers

**A.2 Prerequisite**

* Different programming language (Python or Java), Understanding of Machine Learning Algorithms, Machine Learning Algorithms

**A.3 Outcome**

After successful completion of this experiment students will be able to implement Multi-Category Classification Using Binary Linear Classifiers

**A.4 Theory**

**Definition of Classification**

In machine learning, Classification, as the name suggests, classifies data into different parts/classes/groups. It is used to predict from which dataset the input data belongs to.

For example, if we are taking a dataset of scores of a cricketer in the past few matches, along with average, strike rate, not outs etc, we can classify him as “in form” or “out of form”.

Classification is the process of assigning new input variables (X) to the class they most likely belong to, based on a classification model, as constructed from previously labeled training data.

Data with labels is used to train a classifier such that it can perform well on data without labels (not yet labeled). This process of continuous classification, of previously known classes, trains a machine. If the classes are discrete, it can be difficult to perform classification tasks

**Types of Classification**

There are two types of classifications;

1. Binary classification
2. Multi-class classification

* **Binary Classification**

It is a process or task of classification, in which a given data is being classified into two classes. It’s basically a kind of prediction about which of two groups the thing belongs to.

Let us suppose, two emails are sent to you, one is sent by an insurance company that keeps sending their ads, and the other is from your bank regarding your credit card bill. The email service provider will classify the two emails, the first one will be sent to the spam folder and the second one will be kept in the primary one.

This process is known as binary classification, as there are two discrete classes, one is spam and the other is primary. So, this is a problem of binary classification.

Binary classification uses some algorithms to do the task, some of the most common algorithms used by binary classification are .

1. Logistic Regression
2. k-Nearest Neighbors
3. Decision Trees
4. Support Vector Machine
5. Naive Bayes

* **Multiclass Classification**

Multi-class classification is the task of classifying elements into different classes. Unlike binary, it doesn’t restrict itself to any number of classes.

Examples of multi-class classification are

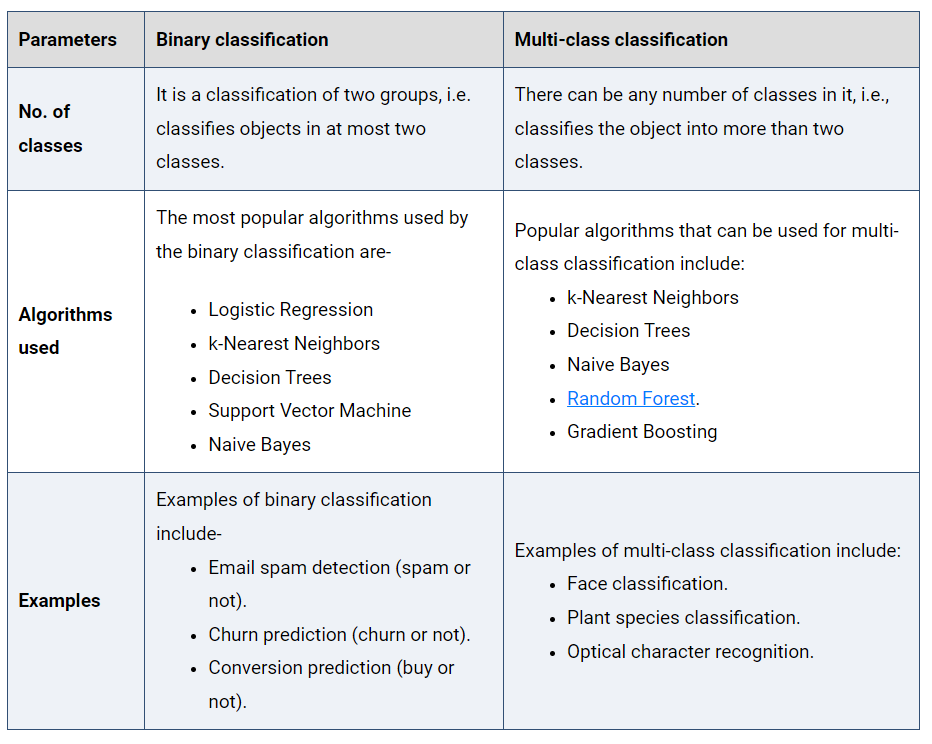
1. classification of news in different categories,
2. classifying books according to the subject,
3. classifying students according to their streams etc.

In these, there are different classes for the response variable to be classified in and thus according to the name, it is a Multi-class classification

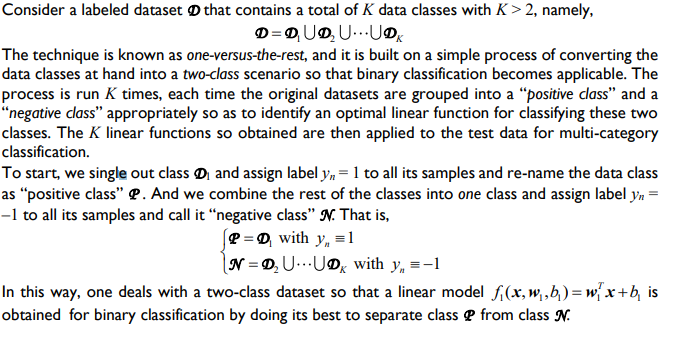
Can a classification possess both binary or multi-class?

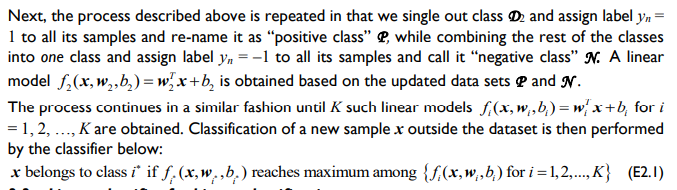
Let us suppose we have to do sentiment analysis of a person, if the classes are just “positive” and “negative”, then it will be a problem of binary class. But if the classes are “sadness”, happiness”, “disgusting”, “depressed”, then it will be called a problem of Multi-class classification.

* **Binary vs Multiclass Classification**

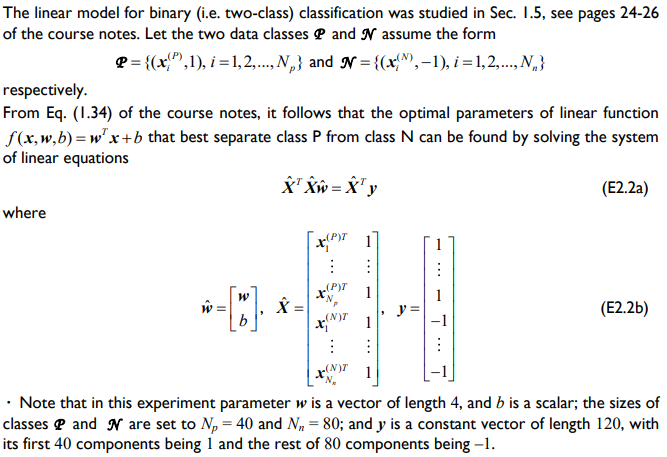


* **The idea of multi-category classification using linear binary classifiers**



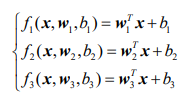


* **Linear classifier for binary classification**

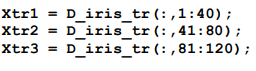


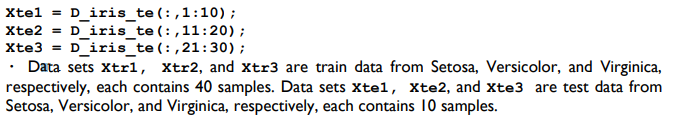
**A5. Task**

1. **12 c**

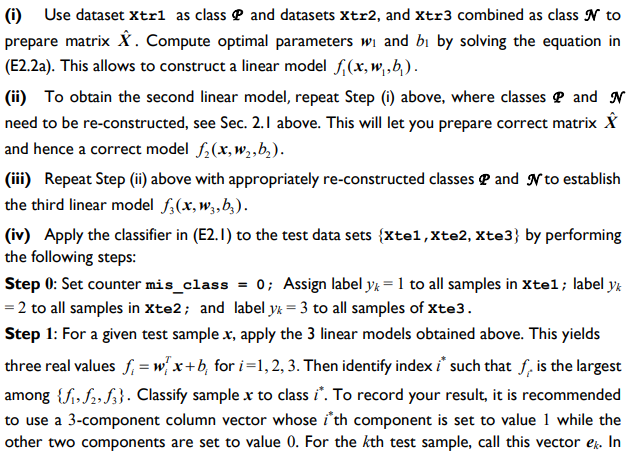


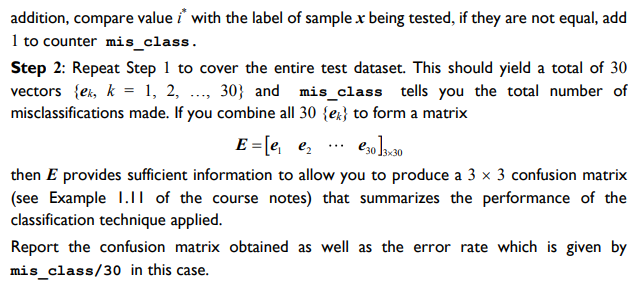
1. Prepare datasets as:





1. Perform the following steps





PART B

|  |  |
| --- | --- |
| Roll No. C009 | Name: Samarth Borade |
| Class : BTI SEM 10 | Batch : EB1 |
| Date of Experiment: 1/3/24 | Date of Submission |
| Grade : |  |

**B.1 Documentation written by student:**

# Importing necessary libraries

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, classification\_report

from sklearn.datasets import load\_iris

import pandas as pd

iris\_data = load\_iris()

iris\_df = pd.DataFrame(iris\_data.data, *columns*=iris\_data.feature\_names)

iris\_df['target'] = iris\_data.target

iris\_df\_setosa = iris\_df.copy()

iris\_df\_setosa['target'] = (iris\_df['target'] == 0).astype(int) # Class Setosa vs rest

iris\_df\_versicolor = iris\_df.copy()

iris\_df\_versicolor['target'] = (iris\_df['target'] == 1).astype(int) # Class Versicolor vs rest

iris\_df\_virginica = iris\_df.copy()

iris\_df\_virginica['target'] = (iris\_df['target'] == 2).astype(int) # Class Virginica vs rest

trained\_classifiers = []

# Training and evaluating logistic regression classifiers for each binary classification task

for index, data in enumerate([iris\_df\_setosa, iris\_df\_versicolor, iris\_df\_virginica]):

class\_names = ['Setosa', 'Versicolor', 'Virginica']

print(f"Training and evaluating classifier for class: {class\_names[index]}")

X\_train, X\_test, y\_train, y\_test = train\_test\_split(data.iloc[:, :4], data.iloc[:, -1], *test\_size*=0.2, *random\_state*=42)

classifier = LogisticRegression()

classifier.fit(X\_train, y\_train)

predictions = classifier.predict(X\_test)

print(f"Accuracy: {accuracy\_score(y\_test, predictions)}")

print(classification\_report(y\_test, predictions))

trained\_classifiers.append(classifier)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(iris\_df.iloc[:, :4], iris\_df.iloc[:, -1], *test\_size*=0.2, *random\_state*=42)

individual\_predictions = [classifier.predict(X\_test) for classifier in trained\_classifiers]

# Combining predictions using majority voting

final\_predictions = [max(range(len(trained\_classifiers)), *key*=lambda *i*: preds[*i*]) for preds in zip(\*individual\_predictions)]

accuracy = accuracy\_score(y\_test, final\_predictions)

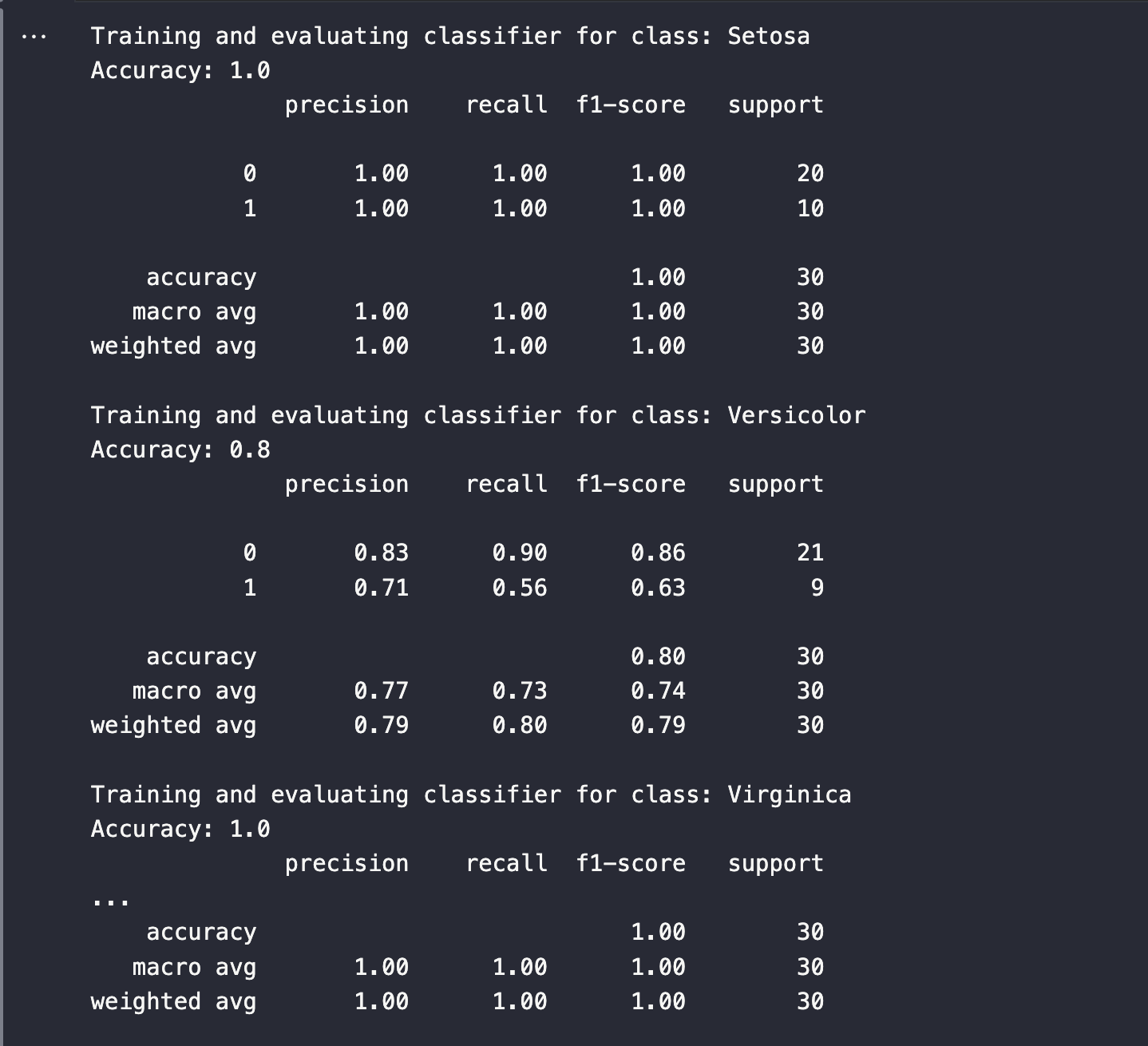
print(classification\_report(y\_test, final\_predictions))

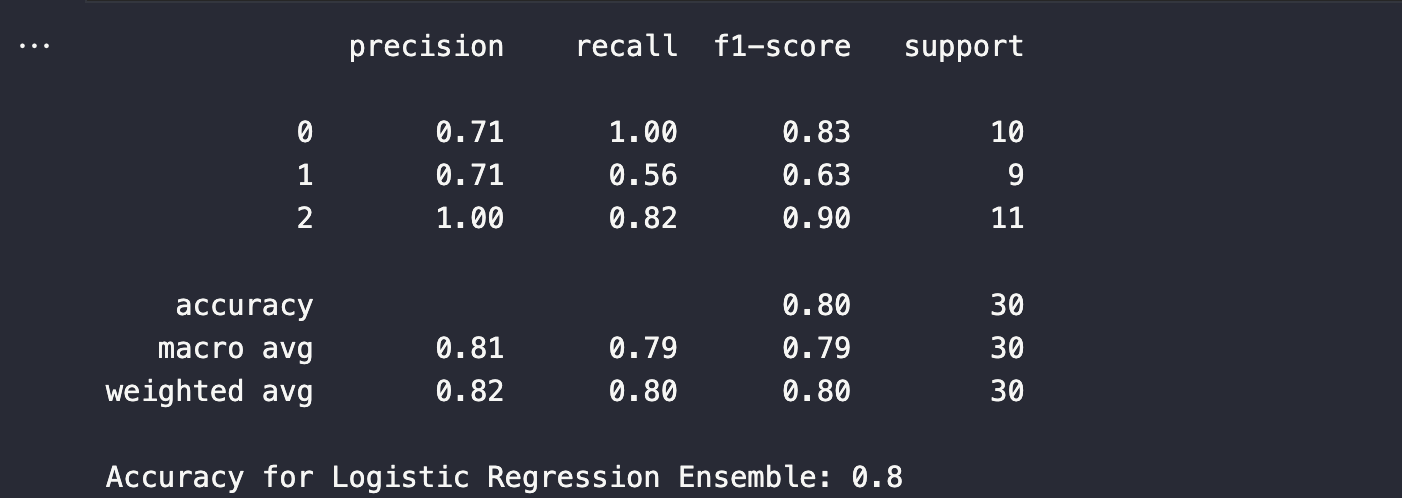
print(f"Accuracy for Logistic Regression Ensemble: {accuracy}")

**Output:**

**A screenshot of a computer program

Description automatically generated**

****

****

**B.2 Observations and learning:**

1. **Class Setosa Classification:**
   * Achieved perfect accuracy (1.0), indicating that the classifier effectively differentiated Setosa flowers from the rest.
   * Both precision and recall values were 1.00, signifying the classifier's ability to correctly identify all Setosa instances without misclassifications.
2. **Class Versicolor Classification:**
   * Achieved an accuracy of 80%, indicating a relatively good performance but with some misclassifications.
   * Precision and recall for Versicolor were 0.71 and 0.56 respectively, suggesting that the classifier struggled to correctly classify all instances of Versicolor.
3. **Class Virginica Classification:**
   * Similar to Setosa, achieved perfect accuracy (1.0), indicating excellent differentiation from the rest.
   * Precision and recall values were 1.00, showing accurate identification of all Virginica instances.
4. **Logistic Regression Ensemble:**
   * Achieved an accuracy of 0.8 when combining predictions from individual classifiers.
   * Precision and recall values varied across classes, with Versicolor showing slightly lower performance compared to Setosa and Virginica.

**B.3 Conclusion:**

The logistic regression classifier performed exceptionally well in distinguishing Setosa and Virginica classes with perfect accuracy but showed slightly lower performance in classifying Versicolor with 80% accuracy. Overall, the ensemble maintained a commendable 80% accuracy in predicting the Iris flower classes, reflecting the effectiveness of logistic regression in this context.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*