CP8318 Machine Learning - Assignment 2

Student: Nebojsa Djosic, `nebojsa.djosic@torontomu.ca`

Lecturer: Elodie Lugez, Ph.D., `cps8318@cs.torontomu.ca`

Introduction

Multi-class classification is one of the fundamental problems in machine learning. The goal is to map some input into one of many discrete categories. The most commonly used approach to address this problem is by extending binary classifiers. Logistic regression was originally designed for binary classification. However, it can be extended to handle multi-class problems through techniques like one-vs-rest (OvR), one-vs-one (OvO) and softmax, which allow the model to predict the probability of each class and select the most likely one.

Logistic regression calculates a weighted sum of the input features plus a bias term as in $z = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$ where $x_1, x_2, \ldots, x_n$ are input features and $w_1, w_2, \ldots, w_n$ are weights.

A logistic function, such as *softmax*, is typically used to transform the linear combination into a probabilistic value. *Softmax* is an extension of the *sigmoid* function used for binary classification. Each class is assigned a probability and the class with the highest probability is used to label the prediction instance.

Logistic regression is well-suited for problems where the decision boundaries between classes are approximately linear such as handwritten digit recognition, disease diagnosis, text categorization, etc. In this report, we present the performance of multi-class logistic regression in the classification of dried bean grains.

Methodology

We selected the Dry Bean dataset from UCI Machine Learning Repository [1] introduced in [2]. The dataset consists of images of 13,611 grains, representing 7 different types of dry beans. Images were taken with a high-resolution camera. The dataset is created by extracting 16 numeric features, of which 12 are size dimensions and 4 are numerical representations of the bean's shape.

Features in the dataset are scaled using the Standard Scaller from the sklearn.preprocessing library. Scaling helps to standardize the range of the features ensuring that all features contribute equally to the model. This prevents features with large ranges from dominating.

The dataset is split into two: 20% for testing and 80% for training, with a class balancing stratification option to ensure each class is represented proportionally and a random state to ensure repeatability.

The experiments consist of training, testing, and evaluating performance. To run experiments We constructed an array of three Logistic Regression models using the sklearn.linear_model library. The first model is pure Logistic Regression, the second is wrapped into One vs Rest, and the third is wrapped into One vs One, both wrappers are from sklearn.multiclass library. Experiments are executed in a loop and results are persisted to a directory on the local disk.

Results

Table 1 shows performance scores for the pure Logical Regression model, while Figure 1 shows the confusion matrix for the same model.

Table 2 shows performance scores for the One vs. One model, while Figure 2 shows the confusion matrix for this model.

Table 3 shows performance scores for the One vs. Rest Logical Regression model, while Figure 3 shows the confusion matrix for this model.

Conclusion

The results show no significant difference between the models. Performance is almost identical between the pure Logistic Regression and One vs. One models, with One vs Rest showing only minor differences. This is probably because the modern Python libraries are well-optimized out-of-the-box. Future work should put this assumption to the test by experimenting with different experimental setups, different libraries and different datasets.

Figures and Tables

Table 1: Logical Regression Performance Scores

bean	precision	recall	f1-score	support
BARBUNYA	0.95	0.89	0.92	265
BOMBAY	1.00	1.00	1.00	104
CALI	0.93	0.94	0.94	326
DERMASON	0.92	0.91	0.92	709
HOROZ	0.96	0.95	0.96	386
SEKER	0.93	0.95	0.94	406
SIRA	0.85	0.88	0.86	527
accuracy	0.92	0.92	0.92	1
macro avg	0.94	0.93	0.93	2723
weighted avg	0.92	0.92	0.92	2723

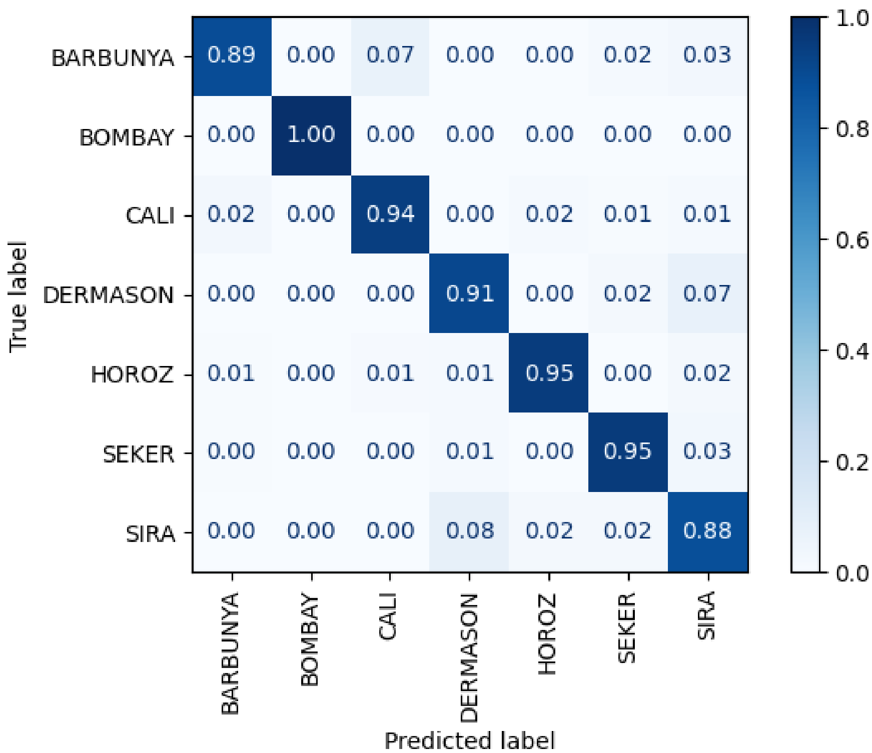

Figure 1: Logistic Regression Confusion Matrix

Table 2: One vs. One Performance Scores

	precision	recall	f1-score	support
BARBUNYA	0.95	0.89	0.92	265
BOMBAY	1.00	1.00	1.00	104
CALI	0.93	0.94	0.94	326
DERMASON	0.92	0.91	0.92	709
HOROZ	0.96	0.95	0.95	386
SEKER	0.93	0.95	0.94	406
SIRA	0.85	0.87	0.86	527
accuracy	0.92	0.92	0.92	1
macro avg	0.93	0.93	0.93	2723
weighted avg	0.92	0.92	0.92	2723

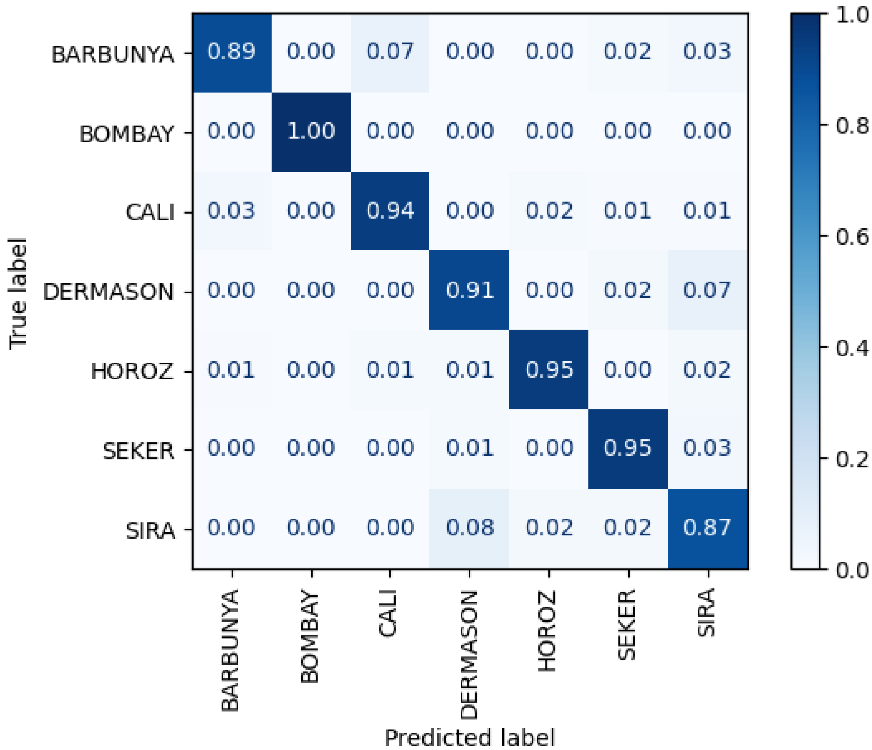

Figure 2: One vs. One Confusion Matrix

Table 3: One vs. Rest Performance Scores

	precision	**recall**	**f1-score**	**support**
BARBUNYA	0.96	0.86	0.90	265
BOMBAY	1.00	1.00	1.00	104
CALI	0.92	0.95	0.93	326
DERMASON	0.92	0.89	0.91	709
HOROZ	0.97	0.94	0.95	386
SEKER	0.95	0.95	0.95	406
SIRA	0.81	0.89	0.85	527
accuracy	0.92	0.92	0.92	1
macro avg	0.93	0.93	0.93	2723
weighted avg	0.92	0.92	0.92	2723

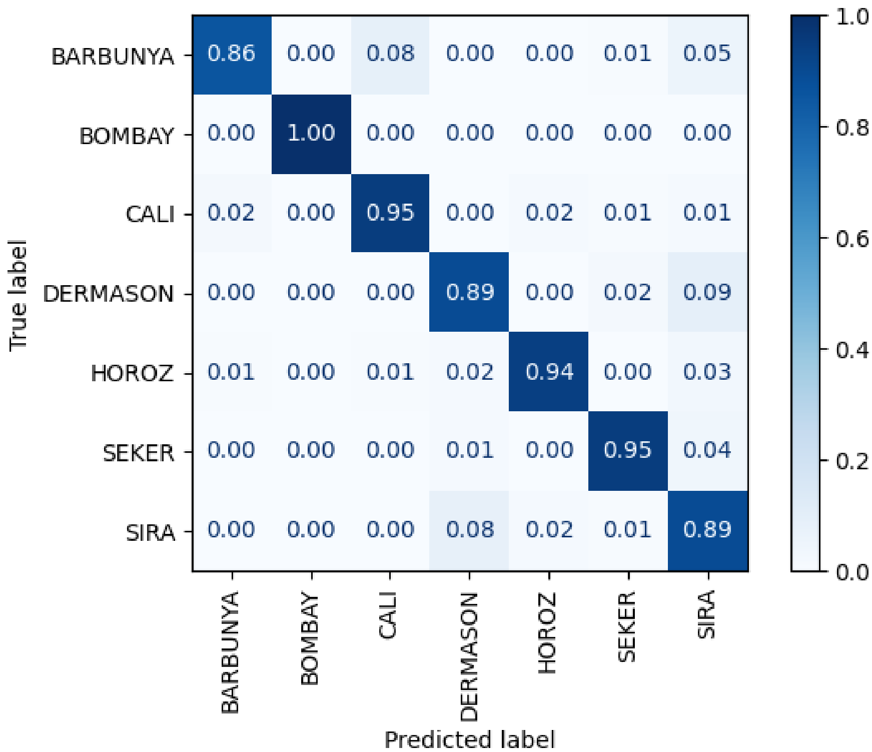

Figure 3: One vs. Rest Confusion Matrix

References

[1] Dry Bean. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C50S4B.

[2] M. Koklu and I. A. Özkan. Multiclass classification of dry beans using computer vision and machine learning techniques. *Comput. Electron. Agric.*, 174:105507, 2020.