Practical Machine Learning: ML Algorithm Selection

Milana M. Wolff

April 27, 2024

1 Introduction

In this assignment, we approach the problem of selecting the best-performing model machine learning model for a wine classification prediction problem. Wines are rated based on expert assessments; using a number of physicochemical quantitative measures associated with wines, such as density and acidity, we attempt to replicate expert opinions and predict the score assigned to a wine. In order to solve the problem of determining the best-performing machine learning algorithm to predict expert scoring of red Vinho Verde wines, we test a variety of well-known classification models, starting with default hyperparameter configurations, and evaluate across metrics commonly used for classification problems, such as balanced accuracy and confusion matrices.

2 Dataset Description

The dataset used for this assignment contains physicochemical quantitative input features and sensory quantitative output features (i.e., an expert wine score) for the red variant of the Portuguese "Vinho Verde" wine [1]. The dataset includes 1599 observations and eleven input features, including fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol content. According to the UC Irvine Machine Learning Repository website, "the classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones)", with a total of 1319 observations rated as 5 or 6 and a mere 28 observations rated with the highest and lowest scores (3 and 8) [2]. This robust dataset includes no missing values to be imputed. We use the eleven listed input features to predict the wine quality measurement.

	Accuracy	Balanced Accuracy
Ridge	0.5919	0.2331
AdaBoost	0.5244	0.2898
Bagging	0.6731	0.3828
Random Forest	0.7069	0.3690
Logistic Regression	0.6038	0.2428
Perceptron	0.4206	0.2514
SGD	0.4781	0.2718
MLP	0.5913	0.2806

3 Experimental Setup

We use Python 3.10.12 in a Jupyter/interactive Python notebook. After importing the `scikit-learn` library, we use out-of-the-box implementations of the following classification algorithms with default hyperparameter settings: Ridge classifier, AdaBoost classifier, bagging classifier, random forest classifier, logistic regression classifier, perceptron classifier, SGD classifier, and an MLP classifier.

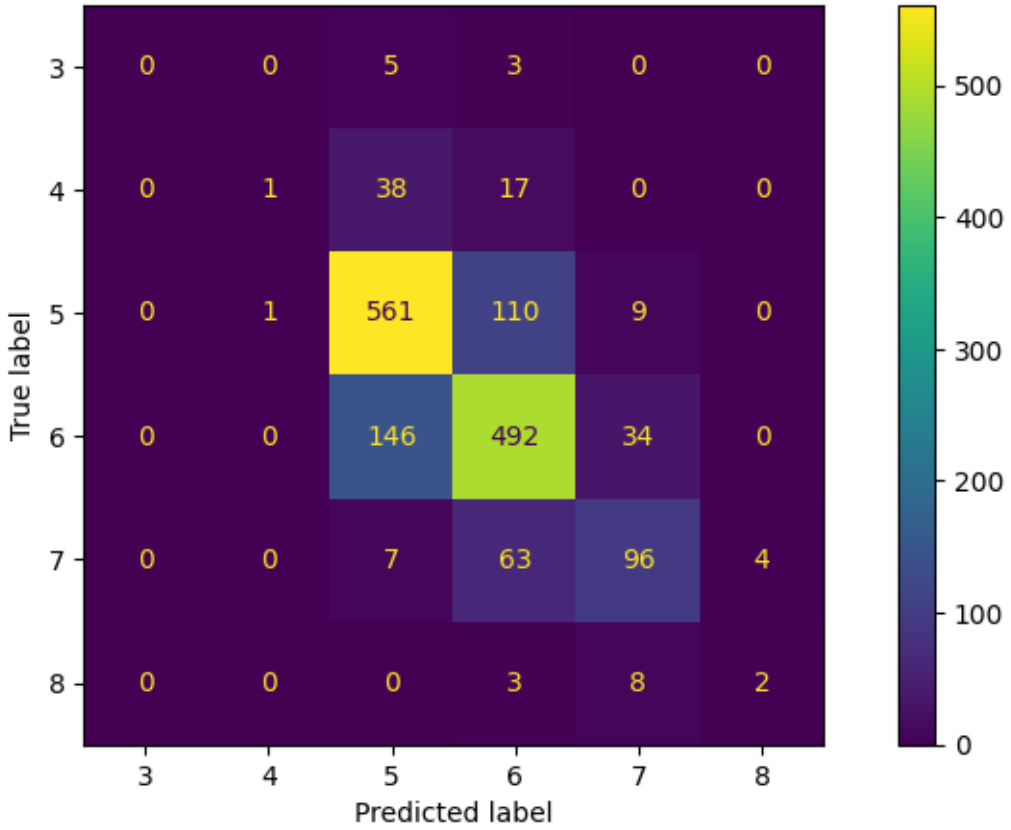
We load the wine data using the `pandas` library and use all eleven features for classification without additional pre-processing steps. For each machine learning algorithm, we use a nested resampling approach as described below.

We perform an 80/20 train/test split on the data using the `train_test_split` method from `sklearn`. Using the training data alone, we use ten-fold cross-validation and evaluate the best-scoring model, as measured by balanced accuracy, on the held-out data. We repeat this procedure five times per model type (i.e., five times for AdaBoost, five times per SGD, etc.) We save the performance results for each model. If the balanced accuracy score averaged over these five outer CV folds exceeds that of the previous model, we update the maximum balanced accuracy score and store the type of algorithm (i.e. the type of model that performs best of the data). We report the averaged balanced accuracy of the best-performing algorithm.

4 Results

We observed the highest performance using the balanced accuracy metric with the bagging classifier method, at 0.3828 (performance of best algorithms averaged across five trials) and an overall accuracy of 0.6731. Note that random forest outperforms the bagging classifier when we use accuracy as the metric for comparison. See below for a comparison performance estimators for this approach to all less performant methods. The reported metrics are derived from the average performance of the best parameter settings determined by 10-fold inner cross validation across 5 outer folds.

The confusion matrix below contains the performance of the bagging classifier across five outer folds.



Code is uploaded to GitHub at the following link: https://github.com/COSC5557/ml-algorithm-selection-mwolff2021-1

References

[1] In: (). URL: `http://www.vinhoverde.pt/en/`.

[2] Paulo Cortez, A. Cerdeira, F. Almeida, et al. "Wine Quality". In: (2009). DOI: https://doi.org/10.24432/C56S3T.