DV2599 – Assignment 1

Simon Lindqvist
Blekinge Institute of Technology
Karlskrona, Blekinge
siln22@student.bth.se

Abdalrahman Mohammed
Blekinge Institute of Technology
Karlskrona, Blekinge
abmm22@student.bth.se

I. Introduction

The goal of this assignment was to evaluate the performance of two supervised classification models on a wine quality prediction task. Using K-fold cross validation the best model was to be chosen, trained, and scored both balanced and unbalanced. The red wine dataset was chosen at random over the white wine dataset, meaning the red wine dataset is the dataset being used in this assignment.

II. Dataset inspection

A. Initial inspection

The dataset contains 1,599 rows and 12 columns. The features consist of 11 continuous variables (e.g., fixed acidity, alcohol content) and 1 target variable, "quality," which represents the wine's quality on a scale from 3 to 8 resulting in the dataset having 6 classes. All 1599 instances were non-null, meaning the data is complete.

B. Calculating class ratios

By using the value counts and sort index methods of the pandas dataframe the class ratios were calculated as percentages and rounded to two decimals. A severe skew towards the 5 and 6 class was observed with them together having over 80% of all instances in the dataset. Even class 7 had a comparatively higher ratio than the other classes 3, 4, and 8. See Table. 1.

TABLE I. The calculated class ratios

Class	*Ratio (%)*
3	0.63
4	3.31
5	42.59
6	39.90
7	12.45
8	1.13

III. Data preprocessing

Before using K-fold cross validation or training any models the data had to undergo two preprocessing steps. First it was divided into test and train data using the sample method of the dataframe. An 80-20 split was used, meaning 80% of the data was used for training, and 20% was used for testing. After the train-test split was performed, min-max scaling was applied on the data. Min-max scaling was used since it preserves outliers, whilst still narrowing the range of variables and at the same time keeping the data interpretable. One of the used classifiers, SVM, is also known to be sensitive to feature magnitude, making min-max scaling especially suitable.

IV. Classifiers

A. Choosing classifiers

Support Vector Classifier and Random Forest Classifier, both from Scikit-learn, was chosen as the classifiers to be assessed. These classifiers were chosen since both can handle multiple attributes and non-binary classes, which both are characteristics of the chosen dataset.

B. Evaluating classifiers

To evaluate classifier performance, we performed Repeated k-Fold Cross-Validation, using the parameters n_splits = 3 and k_folds = 10, on the training set using our two classifiers. The results were examined and the Random Forest Classifier was chosen as the best performing model due to it's notably higher mean accuracy and comparable standard deviation. See Table. 2.

TABLE II. Unbalanced K-fold cross-validation results

Model	*Mean Accuracy*	*Standard Deviation*
SVC	0.5898	0.0195
RFC	0.6519	0.02

The Random Forests Classifier model was then trained on the entire training dataset and evaluated on the test dataset yielding an accuracy score of 0.7031.

V. Balancing the data

Balancing of the data was performed using the SMOTE oversampling function from the imbalanced learn library. Balancing the data means all classes should be represented equally in hope of improving performance on models trained on skewed data. Using the balanced data Repeated K-Fold Cross-Validation was performed as before using both classifiers. It was again observed that the Random Forest Classifier performed the best with a higher mean accuracy and lower standard deviation compared to the SVC classifier. See Table. 3.

TABLE III. Balanced K-fold cross-validation results

Model	*Mean Accuracy*	*Standard Deviation*
SVC (balanced)	0.701	0.0107
RFC (balanced)	0.8626	0.0097

The balanced Random Forests Classifier model was then trained on the entire balanced training dataset and evaluated on the test dataset yielding an accuracy score 0.6719.

VI. Results

After balancing the data and retraining the model on the balanced data, a lower accuracy score was achieved. This is however likely due to the skewed nature of the dataset, meaning that we remedy the skewness via the balancing, making the model better at non-skewed data. Our testing data is however from the same dataset, making it equally skewed resulting in a lower accuracy score for the balanced model. This shows how balancing a model doesn't necessarily yield better accuracy if the application is skewed too.