DV2599 - Assignment 1

1st Viktor Fransson
DVAMI22h
Blekinge Institute of Technology
Karlskrona, Sweden
vifr22@student.bth.se

2nd Tobias Gustafsson
DVAMI22h
Blekinge Institute of Technology
Karlskrona, Sweden
togu22@student.bth.se

I. Introduction

The white wine dataset was used together with a random forest classifier from the `sklearn.ensemble` library called `RandomForestClassifier()` and decision tree classifier from `sklearn.tree` called `DecisionTreeClassifier()`.

II. Inspection of the Data

The data contains 4898 entries with 11 features (including the quality feature). There are no null values for any of the features. The ratio of each quality class in the data was calculated along with what values the features of each class were most likely to have.

TABLE I
Quality Class Ratios

Quality Class	Ratio (%)
Class 3	0.41
Class 4	3.33
Class 5	29.75
Class 6	44.88
Class 7	17.97
Class 8	3.57
Class 9	0.10

The quality classes looks to be somewhat normally distributed, with class 6 in the middle, and 3 and 9 at the outer edges. Some interesting takeaways from what feature values were more likely to lead to higher classes were: high alcohol, low density and low chlorides. This was found out by discretizing the values of the features into low, medium and high.

III. Repeated k-Fold Cross-Validation

To evaluate the classifiers, accuracy score was used as metric as it measures the proportion of correctly classified instances and the data had been min-max scaled. The RF-classifier got a higher average score with lower standard deviation, proving it to be the better classifier for this problem.

TABLE II
Repeated k-Fold Cross-Validation Results

Random Forest: average: 0.64, standard deviation: 0.011
Decision Tree: average: 0.55, standard deviation: 0.015

IV. Model Performance

After fitting the RF-model on the scaled train-data it was used to predict the test-data. Once again accuracy-score was used to evaluate the model and after 3 runs it got an average score of 0.70.

V. Balanced and Scaled set

The scaled train-data was balanced using `SMOTE` from the imblearn library, that performs up-sampling of the minority-classes. It works by adding new samples of a minority class based on existing ones. This led to the data-entries being equally distributed between the classes 3-9, with $\approx 14.29\%$ of the entries each.

TABLE III
Balanced Quality Class Ratios

Quality Class	Ratio (%)
Class 3	14.29
Class 4	14.29
Class 5	14.29
Class 6	14.29
Class 7	14.29
Class 8	14.29
Class 9	14.29

VI. New Model's Performance

A new RF-model was trained using the balanced train-data and was then evaluated using the accuracy score on the test data. After 3 runs the average improvement compared to the old model was -0.023, which is a degradation in performance.

It is difficult to say exactly why there was no improvement. Using `SMOTE` to have evenly large class sizes is probably not beneficial with this data. Some classes were very small, and performing up-sampling on those will not contribute to a higher accuracy, but rather add noise to the model. Up-sampling those classes will also create many instances that are very similar, so even though there are many instances of the class, the quality and variety of the data will decrease.

Furthermore, the accuracy is calculated based on comparison to the test data. In the test data, the classes with a large ratio will be over-represented. For a high accuracy, it is most important to correctly classify instances of classes with a large ratio. Adding instances of a class with a low ratio might lead to a better result for that class, but the overall accuracy might decrease.