Classification of Red Wine Quality based on physicochemical properties

Keith Barnes
Department of Computer Science and Engineering
University of Bologna
Bologna, Italy
keith.barnes@studio.unibo.it

I. Introduction

The goal of this project was to develop a machine learning model that classifies red wines as either "good" or "bad" based on their physicochemical characteristics. The dataset used for this study is the Wine Quality dataset [1], which includes 4,898 samples of red wine described by eleven features: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol content. Each wine was rated by professional tasters in a blind test on a scale from 1 to 10, but the dataset contains scores ranging from 3 to 8.

An exploratory analysis was conducted to understand the data better. I looked at the distribution of features and their correlation with the quality label using Pearson's correlation coefficient. Notably, alcohol showed a strong positive correlation with wine quality (around 0.45), while volatile acidity was negatively correlated (approximately 0.35). To evaluate our models properly, the dataset was split into training and testing subsets with an 80/20 ratio, preserving the class distribution. The training data was then balanced using the Synthetic Minority Oversampling Technique (SMOTE) to address the class imbalance, and all features were standardized to have zero mean and unit variance. Principal Component Analysis (PCA) was applied to reduce dimensionality by retaining components that explain 90% of the variance, which allowed us to test how some classifiers perform with fewer, combined features.

The classification models tested include decision trees, random forests, AdaBoost, gradient boosting, k-nearest neighbors, and support vector machines. The final evaluation used accuracy, precision, recall, and F1-score, which is especially important for this imbalanced classification problem on the test set.

II. Related Work

Since its release, the Wine Quality dataset has been widely studied. *Cortez et al.* [1] originally treated the problem as both regression and classification, finding moderate results using support vector regression and neural networks. Subsequent research framed wine grading as binary classification, most commonly labeling wines with score ≥ 7 as positive, though some authors have experimented with a threshold of ≥ 6 [2]. Ensemble methods such as random forests and gradient boosting emerged as strong baselines because they handle skewed distributions naturally and require minimal preprocessing. When combined with SMOTE, these approaches typically achieve F1-scores in the range of 0.52–0.55.

Other techniques like support vector machines and k-nearest neighbors have also been applied, frequently incorporating PCA to reduce noise and dimensionality [3]. While PCA can boost performance for distance-based classifiers, it reduces feature interpretability since principal components mix original measurements. To further address class imbalance, advanced sampling strategies such as SMOTE-ENN [4] and cost-sensitive learning have been explored. Although these methods often increase recall, they tend to lower precision, and F1 rarely surpasses 0.60. This pattern suggests that physicochemical attributes alone may not capture the full nuance of expert wine evaluations.

III. Proposed Method

The analysis began with loading the dataset and performing an initial assessment of data quality. There were no missing values in the dataset, which simplified preprocessing. However, 240 duplicate records were identified and subsequently removed to prevent potential data leakage or bias during model training and evaluation. Following this, an initial assessment of the data's distribution was conducted through histograms for all features.

Next, the problem was framed as a binary classification task. Although the wine quality ratings in the dataset range from 3 to 8, a binary classification was chosen to simplify the problem and make it more suitable for many supervised learning algorithms that perform better with clearly defined target classes. Furthermore, classifying wines as either "good" or "bad" is a practical approach aligned with how such models might be used in real-world applications, such as quality control or product recommendation systems.

To create the binary target variable, wines rated 7 or above were labeled as "good," while those rated below 7 were labeled as "bad." This threshold was not arbitrarily selected but follows a widely adopted convention in academic and industry practice. Although the original dataset authors [1] did not define a specific cutoff, many subsequent studies and tutorials have used 7 as a boundary because it generally captures wines rated one standard deviation above the mean. This makes the

label ”good” representative of high-quality wines while still preserving a meaningful minority class. In this dataset, wines labeled as ”good” accounted for approximately 13% of the total data, highlighting a class imbalance issue.

To address this imbalance, the dataset was first split into training (80%) and testing (20%) subsets, using a `random_state` of 42 for reproducibility. Prior to applying SMOTE, a log transformation (specifically `np.log1p`) was applied to several skewed features in both the training and test sets, including 'fixed acidity', 'volatile acidity', 'residual sugar', 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'sulphates', and 'alcohol'. This transformation aimed to reduce skewness and normalize their distributions. All features were then standardized using `StandardScaler` to have a mean of zero and a standard deviation of one. Standardization is essential for algorithms that rely on distance or gradient-based calculations, ensuring consistent feature scaling. The Synthetic Minority Over-sampling Technique (SMOTE) was then applied only to the scaled training set to generate synthetic examples of the minority class, ensuring the test data remained untouched for unbiased evaluation.

In addition to the primary preprocessing pipeline, a secondary analysis was conducted using Principal Component Analysis (PCA) on the standardized training set. PCA was used to reduce dimensionality and potentially remove noise from the data. Enough components were retained to explain 90% of the variance, and the PCA-transformed features were used to train models that are sensitive to high-dimensional data, such as k-nearest neighbors (k-NN) and support vector machines (SVM).

The model training process was divided into two tracks. The first track included tree-based ensemble methods: Decision Tree, Random Forest, AdaBoost, and Gradient Boosting Classifiers. These models were trained using the balanced and standardized training data. For tree-based models, `random_state=42` was set for reproducibility. The second track involved training k-NN and SVM models using both the PCA-transformed dataset.

Finally, all trained models were evaluated on the untouched test set using standard performance metrics: accuracy, precision, recall, and F1-score. These metrics were selected to provide a comprehensive view of model performance, especially given the class imbalance, with precision and recall being of particular interest due to their insights into minority class identification. The combination of multiple models and preprocessing approaches allowed for a robust comparison of techniques under the constraints of an imbalanced dataset.

IV. Results

Table I summarizes the classification performance of all models on the untouched test set.

As shown in Table I, the classifiers in our pipeline generally achieved strong accuracy, ranging from 73.90% to 86.76%. However, this high accuracy largely reflects correct predictions on the majority class—wines rated below 7, which represented approximately 87% of the data. Precision values across all models remained relatively low, between 29.67% and 47.92%, indicating a notable number of false positives among predicted ”good” wines. This suggests that a significant proportion of wines labeled as good were, in fact, not, thereby undermining the models' reliability in identifying true high-quality wines.

TABLE I
Performance on Test Set

Model	Accuracy	Precision	Recall	F1-Score
AdaBoost	0.8346	0.4203	0.8529	0.5631
Rand. Forest	0.8676	0.4792	0.6765	0.5610
Grad. Boost.	0.8346	0.4098	0.7353	0.5263
SVC (PCA)	0.7904	0.3580	0.8529	0.5043
Dec. Tree	0.8419	0.4082	0.5882	0.4819
KNN (PCA)	0.7390	0.2967	0.7941	0.4320

Among the evaluated models, AdaBoost and the Support Vector Classifier (SVC) trained on PCA-reduced features attained the highest recall at 85.29% each. This indicates their strong ability to identify most of the true positive cases (actual good wines). However, this high recall often came at the expense of precision, a classic trade-off in imbalanced classification: broader coverage of positive instances versus a higher rate of false alarms.

In terms of the precision–recall trade-off, the Random Forest and AdaBoost models offered the best balance, achieving the highest F1-scores of 0.5610 and 0.5631, respectively. This made them the top performers in comprehensively measuring a model's ability to classify both positive and negative instances correctly. The Gradient Boosting Classifier also performed reasonably well with an F1-score of 0.5263.

Simpler methods, including the Decision Tree and k-nearest neighbors (k-NN), even after PCA, generally lagged behind in overall metrics. The Decision Tree yielded an F1-score of 0.4819, struggling with subtle class boundaries due to its inherent rigidity in creating decision regions. Similarly, the k-NN model, despite PCA transformation, achieved the lowest F1-score of 0.4320. Its performance suffered because classifying a point based on its closest neighbors becomes unreliable when different classes are heavily mixed, even if PCA reduces dimensionality and noise.

These results highlight two main insights. First, the application of SMOTE effectively improved sensitivity to the minority class (high recall values across models) but could not perfectly replicate the complex distribution of real ”good” wines, which consequently reduced precision. By generating artificial minority examples, SMOTE aimed to force the models to learn their characteristics, but these synthetic points might not have accurately represented the nuanced decision boundaries, leading to an increased number of false positives. Second, Principal Component Analysis (PCA) successfully reduced feature redundancy and potentially helped certain models avoid overfitting by simplifying the data structure. However, PCA did not create new discriminative information or inherently separate classes when the underlying distributions were already overlapping. Essentially, PCA could reorient the feature space and reduce noise, but it could not fundamentally

disentangle classes that were intrinsically mixed based on the given physicochemical measures.

In summary, our models demonstrate a strong ability to identify most high-quality wines (as indicated by high recall) but exhibit limitations in ensuring the correctness of those predictions (as evidenced by low precision). This challenge primarily stems from the dataset's imbalanced and overlapping nature, underscoring the inherent difficulty of distinguishing subjective wine quality solely from physicochemical measures.

V. Conclusions

In this project, multiple machine learning workflows were implemented to classify red wines as "good" or "bad" using physicochemical characteristics. All models demonstrated strong overall accuracy and high recall, particularly for identifying quality wines, yet precision consistently lagged, many wines predicted as good were in fact bad. This imbalance in precision reflects two core issues: first, the intrinsic overlap in feature distributions between classes; and second, the limited ability of SMOTE-generated samples to introduce truly new information.

Ensemble models such as random forests, AdaBoost and gradient boosting achieved the most balanced performance, reaching F1-scores around 0.5631, 0.5610 and 0.5263, while PCA-enhanced SVM and k-nearest neighbors highlighted the trade-offs between dimensionality reduction and class separability. Simpler approaches like decision trees underscored sensitivity to noisy splits and overfitting in the absence of feature combination strategies.

The primary reasons for these outcomes are twofold. First, despite using SMOTE, the inherent imbalance in the dataset means the models still favor the majority class, resulting in lower precision. Second, the dataset lacks the richness needed to fully distinguish subjective quality attributes that trained tasters assess. I believe that these factors explain why precision remains low even when recall is high, and they highlight important considerations when applying machine learning to real-world quality assessments.

References

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Ramos, and J. Santos, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems*, vol. 47, no. 4, pp. 547–553, 2009.

[2] S. García, J. Luengo, and F. Herrera, "An application of evolutionary undersampling for classification with imbalanced data: An empirical study," *Evol. Comput.*, vol. 17, no. 3, pp. 275–306, 2012.

[3] A. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, 2014.

[4] G. Batista, R. Prati, and M. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004.