**A STUDY ON MODELING WINE PREFERENCE**

1. Project objectives and members’ contributions

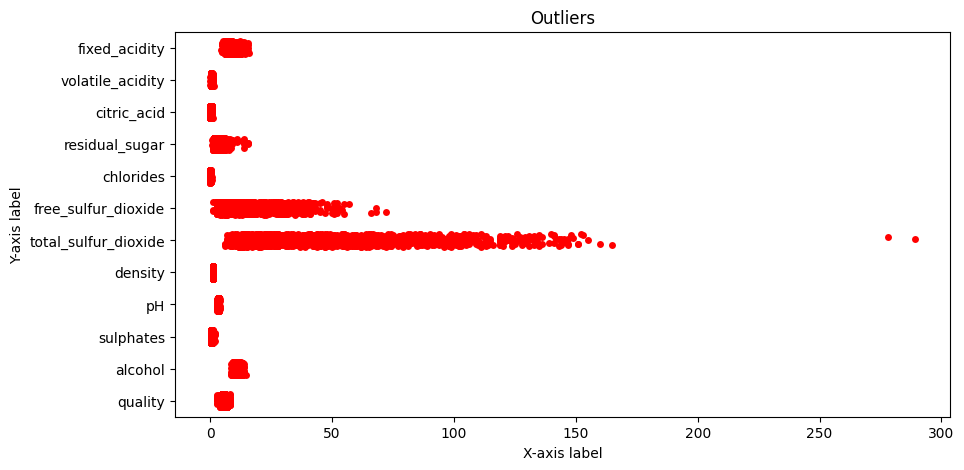
Wine, once considered a luxury item, is now enjoyed by a diverse range of consumers. To ensure the highest quality products reach the market, wine certification is conducted through physicochemical and sensory evaluations. Physicochemical analyses typically involve measuring density, alcohol content, and pH levels in a laboratory setting, while sensory tests rely on human experts. However, taste remains the least understood of the human senses, making wine classification a challenging task. Furthermore, the relationship between physicochemical properties and sensory perception is complex and not yet fully deciphered.

Team members’s contributions are specified as the table below:

| **Name** | **Contribution** |
| --- | --- |
| Thi-Thanh-Lam Vu | * Data cleaning * Data exploration and analysis * Implement a Neural network for wines’ quality classification * Implement and examine techniques to handle the imbalanced dataset |
| Sangjun Hong | * Classification of good wines * Graph representation of wine distribution * Data clustering |
| Xiangji Li | * Data Analysis with Classification * Find the optimal model fit the data * Trying to improve the accuracy of the data |

1. Data cleaning

Data cleaning is a crucial step in classification and regression tasks, as it enhances model accuracy and overall performance. In our project, we began by identifying and removing outliers from the dataset. To do this, we analyzed the distribution of key variables, as shown in Figure 1. Based on this visualization, we chose to exclude data points where total\_sulfur\_dioxide exceeded 200. Additionally, we removed duplicate entries to ensure data integrity.

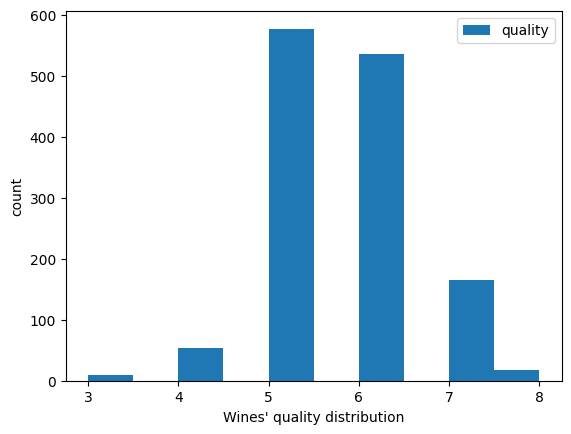


1. Data Exploration and Analysis

After removing outliers and duplicate entries, 1,357 data points remained. Each data point includes the variables described in Table 1 below:

| **ID** | **Variable name** | **Meaning** |
| --- | --- | --- |
| 1 | fixed acidity | Acidity level that remains after fermentation |
| 2 | volatile acidity | Acetic acid amount affecting taste |
| 3 | citric acid | Enhances flavor, adds freshness |
| 4 | residual sugar | Sugar left after fermentation |
| 5 | chlorides | Salt content |
| 6 | free sulfur dioxide | SO₂ not bound and acts as an antimicrobial |
| 7 | total sulfur dioxide | Total amount of SO₂ (bound + free) |
| 8 | density | Wine density, related to alcohol and sugar content |
| 9 | pH | Acidity level |
| 10 | sulphates | Contributes to microbial stability |
| 11 | alcohol | Alcohol percentage |
| 12 | quality | Sensory quality score (0 to 10). Wines with quality greater than 6.5 are considered as good ones. |

The distribution of wine quality is shown in Figure 2. The dataset is imbalanced, with most wines rated as medium quality, and relatively few classified as low or high quality.



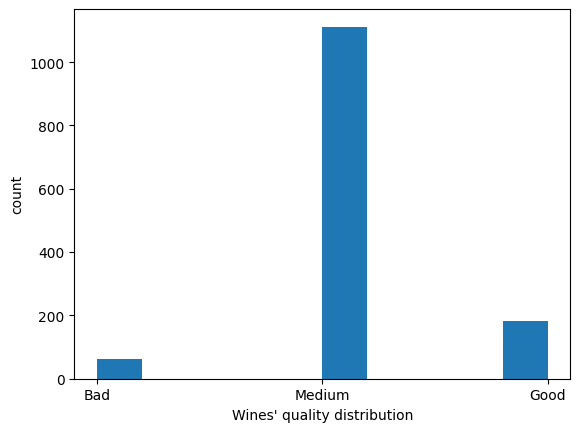
We used boxplots to examine the relationship between wine quality and other variables. As shown in Figure 4, certain variables exhibit clear trends with quality. Specifically, higher levels of alcohol, sulphates, and citric acid are associated with better wine quality, while lower volatile acidity also corresponds to higher quality.

| a) | b) |
| --- | --- |
| c) | d) |

1. Techniques and tools for wines’ quality classification

In this project, we investigated the application of machine learning models for the classification of wines into three distinct categories: Bad, Medium, and Good. Wines with a quality score below 5 were categorized as Bad, those with scores of 5 or 6 as Medium, and those with scores above 7 as Good.

As illustrated in Figure XX, the class distribution is notably imbalanced, with a disproportionately large number of samples falling into the Medium category. Therefore, our focus was on applying techniques specifically designed to address this class imbalance during model training.



1. Machine Learning Models

In this project, we employed a range of machine learning models to address the classification task, including K-Nearest Neighbors (KNN), Gradient Boosting Machines (GBM), Decision Trees, Support Vector Machines (SVM), Random Forests, and Neural Networks.

1. Techniques to handle imbalanced dataset

To address the class imbalance in the dataset, we experimented with several techniques, including synthetic sample generation using **SMOTE** and dimensionality reduction via **PCA**. Additionally, we explored methods specifically designed for handling imbalanced data in neural network training, such as **Weighted Cross-Entropy Loss**, **Focal Loss**, and **One-vs-All** training.

* SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) is a sampling method designed to address class imbalance in datasets—particularly when one class has significantly fewer samples than others. Rather than simply duplicating existing minority class examples, SMOTE generates new synthetic samples by first selecting a minority class instance, identifying its nearest neighbors from the same class, and then creating a new sample along the line connecting the selected instance and one of its neighbors.

* Principal Component Analysis

Principal component analysis(PCA) is an unsupervised dimensionality reduction method, which is usually used to reduce the number of features, improve model efficiency, or for visualization. Specifically, it converts the original variables into a set of new unrelated variables (principal components) through linear transformation. These principal components are sorted according to the size of the variance explained by the original data, thereby simplifying the data structure and removing redundant information.

* Weighted Cross-Entropy Loss

Weighted Cross Entropy Loss is a variant of standard cross-entropy loss used to handle class imbalance in classification tasks. In standard cross entropy loss, the loss function in a multi-class setting with classes is as follows:

where is the ground truth one-hot label for class , is the predicted probability for class . In the weighted cross entropy loss, a weight is added to each class to increase/decrease its contribution to the loss as follows:

Usually, minority classes get larger weights to emphasize their loss during training.

* Focal Loss

Focal Loss is another modified version of cross-entropy loss, designed specially to tackle class imbalance by focusing training on hard-to-classify examples.

* One-vs-All Training

One-vs-All (OvA), also known as One-vs-Rest (OvR), is a strategy for adapting binary classifiers to handle multi-class classification problems. Given a task with classes, OvA involves training separate binary classifiers—one for each class - where each classifier distinguishes class from all the other classes. Specifically, the target label for classifier is set to 1 for samples belonging to class , and 0 for all other samples.

During inference, each classifier outputs a score for a given input . The final prediction is made by selecting the class whose classifier yields the highest score.

1. Results

We allocated 80% of the data for training and the remaining 20% for testing. Due to the significant class imbalance, we adopted the macro F1-score as the primary evaluation metric. Initially, we investigated the impact of SMOTE on various classification models, including K-Nearest Neighbors (KNN), Gradient Boosting Machines (GBM), Decision Trees, Support Vector Machines (SVM), and Random Forests. The corresponding results are summarized in Table XX. These results demonstrate that SMOTE generally improves model performance, with particularly notable gains observed in the SVM and Random Forest classifiers.

|  | KNN | GBM | Decision Trees | SVM | Random Forests |
| --- | --- | --- | --- | --- | --- |
| Original | 0.47 | 0.53 | 0.51 | 0.50 | 0.49 |
| SMOTE | 0.48 | 0.54 | 0.49 | 0.58 | 0.61 |

**Figure XX** presents the confusion matrix of the **Random Forest classifier**. The results reveal that *Bad* wines are poorly identified, with only 5 out of 15 samples correctly classified, highlighting the model's difficulty in recognizing this minority class. In contrast, *Good* wines are classified with moderate accuracy; however, the frequent misclassification as *Medium* suggests potential overlap in feature representations between these two categories.

| Original | SMOTE |
| --- | --- |

Subsequently, we employed the SMOTE-augmented training set to train Neural Networks (NNs). As shown in Table XX, the neural network models outperformed the classical machine learning approaches. The baseline NN achieved performance comparable to that of the Random Forest classifier, while enhanced variants demonstrated notably improved micro F1-scores. In particular, the One-vs-All approach achieved a macro F1-score of 0.68, representing a significant improvement over the traditional classifiers, most of which attained scores below 0.60.

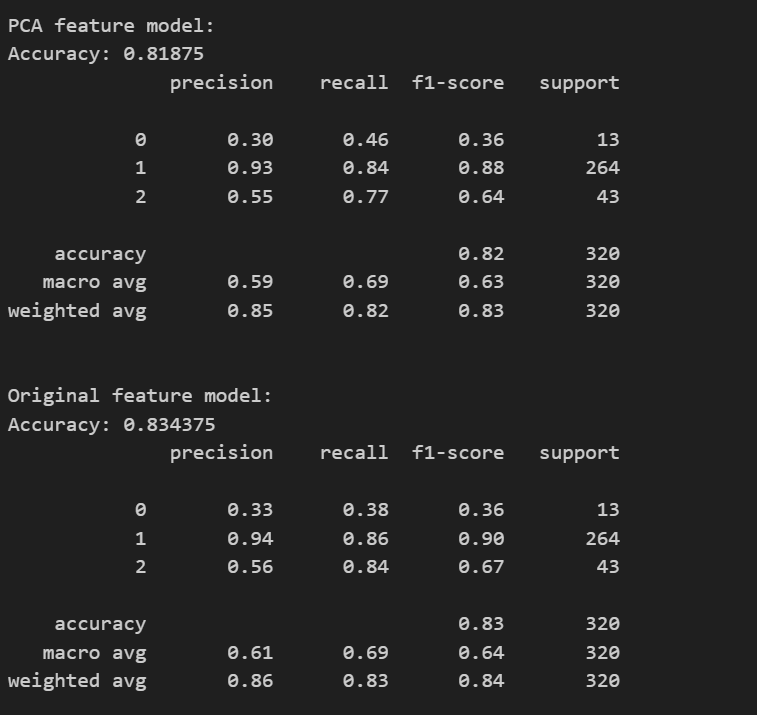
| Baseline | Weighted CE Loss | Focal Loss | One-vs-All |
| --- | --- | --- | --- |
| 0.61 | 0.64 | 0.65 | 0.68 |

Figure xx shows the confusion matrices for different techniques addressing class imbalance. The baseline neural network heavily favors the majority class (Medium), with poor performance on Bad and Good wines. Weighted Cross-Entropy and Focal Loss improve balance, with Focal Loss better identifying minority classes. One-vs-All training achieves the most balanced results, significantly improving accuracy across all classes, particularly for the minority ones.

| 1. Baseline | 1. Weighted CE Loss |
| --- | --- |
| 1. Focal Loss | 1. One-vs-All training |

Taking our current project as an example, the data set has 11 features. PCA can compress these 11 features into 7-8 principal components, but can still explain, for example, 90% of the information.

* Apply PCA on Random Forest



The results are obvious, the accuracy is not better.There are two reasons for the guess: 1. Each variable or feature is important for the quality of the wine. 2. Random Forest does not need PCA to deal with feature collinearity or scaling issues because it is not sensitive to these issues. Using other models such as KMeans will have a more significant effect.

1. Clustering of good wines

Each winery produces wine with a unique profile, and even similarly rated wines can differ based on the producer. Assuming that high-quality wines (scores 6–8) exhibit more distinct and consistent traits than lower-quality ones, we limited our analysis to this premium range and applied clustering techniques to identify representative patterns.

Our initial use of K-means clustering on the raw data yielded unclear groupings. This was due to K-means' assumption of convex clusters and its need for a predefined cluster count—limitations that poorly fit complex, high-dimensional datasets like wine profiles. Even after attempting the elbow method, no clear inflection point emerged. We manually set the number of clusters and visualized the results via PCA, but the boundaries remained ambiguous.

| Error graph for elbow method | Previous clustering using KMeans |
| --- | --- |

To improve results, we included wines with quality scores from 5 to 8 and restructured the dataset as a graph. Features were normalized using MinMax scaling, and each wine was connected to its closest 25% of peers based on L² distance. This graph structure allowed us to apply the Louvain algorithm, which detects communities without predefining cluster counts by maximizing modularity.

| Graph generated by L² norm |
| --- |

After removing small, irregular clusters and rerunning the algorithm on the filtered graph, we obtained five distinct clusters. We then calculated the centroid of each cluster as its representative profile. As shown in the table below, these clusters demonstrate clear differences in feature composition.

| Clustered graph before filtering  Modularity: 0.2196 | Clustered graph after filtering  Modularity: 0.2228 |
| --- | --- |

| Table of centroid wines of each cluster | | | | |
| --- | --- | --- | --- | --- |
| cluster | 0 | 1 | 2 | 3 |
| fixed acidity | 9.44 | 10.93 | 8.25 | 6.87 |
| volatile acidity | 0.40 | 0.38 | 0.32 | 0.58 |
| citric acid | 0.42 | 0.54 | 0.39 | 0.11 |
| residual sugar | 5.57 | 2.60 | 2.24 | 2.32 |
| chlorides | 0.09 | 0.08 | 0.07 | 0.07 |
| free sulfur dioxide | 8.56 | 9.38 | 13.12 | 16.32 |
| total sulfur dioxide | 23.19 | 25.79 | 27.58 | 42.58 |
| density | 1.00 | 1.00 | 0.99 | 0.99 |
| pH | 3.27 | 3.15 | 3.30 | 3.46 |
| sulphates | 0.69 | 0.80 | 0.73 | 0.70 |
| alcohol | 11.74 | 10.96 | 11.94 | 11.61 |

**URL**

[xiangjil/AMS-561---Project](https://github.com/xiangjil/AMS-561---Project)