Comparative Analysis of Machine Learning Models for Wine Quality Prediction Based on Physicochemical Properties

Youssef Ashraf
Computer Science and Mathematics
University of Prince Edward Island
Charlottetown, Canada
yaali@upei.ca

***Abstract*—This study investigates the application of machine learning techniques to predict wine quality based on physicochemical properties. Using a dataset of 6,497 wine samples with 11 chemical features, we compared the performance of Random Forest, K-Nearest Neighbors, and XGBoost algorithms. To address challenges of high dimensionality and class imbalance, we implemented feature selection, dimensionality reduction via PCA, and class merging. After extensive hyperparameter tuning, XGBoost achieved the highest accuracy (76.41%), marginally outperforming Random Forest (76.13%) and KNN (74.81%). Analysis of feature importance revealed alcohol content, volatile acidity, and sulphates as the most influential predictors of wine quality. Learning curves showed that tree-based models exhibited overfitting tendencies, while KNN demonstrated potential for improvement with additional training data. This research demonstrates the effectiveness of ensemble methods for wine quality prediction and provides insights for the food and beverage industry in developing automated quality assessment systems.**

***Index Terms*—machine learning, wine quality, random forest, k-nearest neighbors, XGBoost, feature selection**

I. Introduction

Wine quality assessment traditionally relies on sensory evaluation by human experts, which is subjective, time-consuming, and costly. The development of accurate predictive models based on objective chemical measurements offers a promising alternative that could complement human expertise, reduce costs, and improve consistency in quality assessment. In recent years, machine learning has emerged as a powerful tool for analyzing complex relationships between chemical properties and quality ratings [1].

The wine industry faces significant challenges in maintaining consistent quality assessment due to the complex interplay of numerous factors affecting wine characteristics. While expert tasters remain essential, their evaluations can be influenced by various biases and contextual factors [2]. Machine learning models, by contrast, can identify patterns in chemical compositions that correlate with quality ratings, potentially revealing insights that might not be immediately apparent to human evaluators.

This research addresses several important questions pertaining to wine quality prediction:

1) Which machine learning algorithms perform best for predicting wine quality based on physicochemical properties?
2) How do preprocessing techniques such as feature selection and dimensionality reduction impact model performance?
3) Which chemical properties are most predictive of wine quality?

Our approach involves comprehensive preprocessing of the wine dataset, followed by implementation and comparison of three prominent machine learning algorithms—Random Forest, K-Nearest Neighbors (KNN), and XGBoost. We analyze model performance through various metrics, learning curves, and feature importance assessments. The remainder of this paper is organized as follows: Section II describes the dataset and methods used, Section III presents results and analysis, and Section IV provides conclusions and directions for future research.

II. Background

A. Dataset Description

The dataset used in this study consists of red and white variants of Portuguese "Vinho Verde" wines [1]. After removing duplicates, the final dataset contains 6,497 wine samples (1,599 red and 4,898 white). Each wine sample is characterized by 11 physicochemical properties: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol content. The target variable is a quality score ranging from 3 to 9 on a scale of 0 to 10, assigned by at least three wine experts.

Exploratory data analysis revealed that the dataset has no missing values, ensuring a clean dataset for modeling. The quality scores are unevenly distributed, with the majority of wines having ratings between 5 and 7, and very few samples

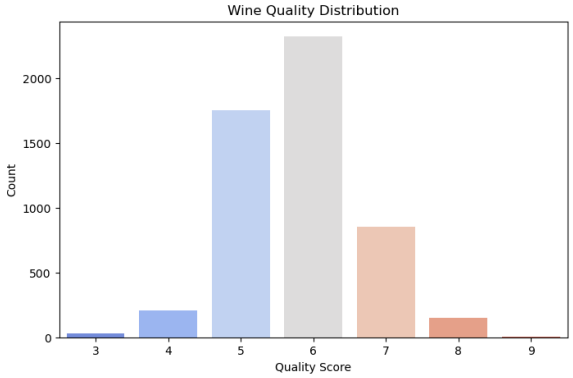


Fig. 1. Distribution of wine quality scores showing imbalance with most samples clustered around scores 5-7.

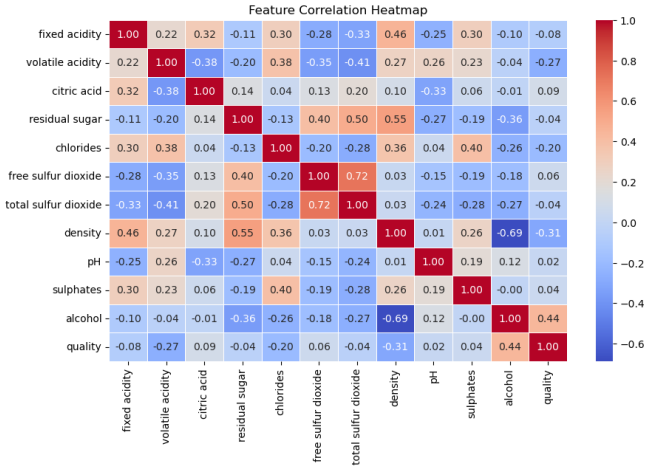


Fig. 2. Feature correlation heatmap showing relationships between physicochemical properties and wine quality.

with extremely low (3) or high (9) ratings. This imbalance in the distribution of quality scores necessitated the conversion to a binary classification problem, where wines rated below 6 were classified as ”bad” (0) and those rated 6 or above as ”good” (1). Additionally, due to the very small number of wines with extreme ratings, we merged quality scores 3 with 4 and 9 with 8 to reduce class imbalance.

Correlation analysis highlighted several important relationships between features and wine quality. Alcohol content showed a moderate positive correlation (0.44) with quality, suggesting that higher alcohol levels may contribute to better wine ratings. Volatile acidity had a negative correlation (-0.27) with quality, indicating that wines with higher acidity tend to have lower ratings. These findings align with the general understanding of wine characteristics, where balanced acidity and appropriate alcohol content are often associated with better quality wines.

B. Machine Learning Methods

1) Random Forest: Random Forest is an ensemble learning method that operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) of the individual trees [3]. It combines the concepts of bagging (bootstrap aggregating) and random feature selection to create a robust classifier that is less prone to overfitting than individual decision trees.

Random Forest was chosen for this study due to its ability to handle high-dimensional data without feature scaling, robustness to outliers, and capacity to provide feature importance metrics. Key hyperparameters tuned in our implementation included n_estimators (number of trees), max_depth (maximum depth of trees), min_samples_split (minimum samples required to split an internal node), min_samples_leaf (minimum samples required to be at a leaf node), and max_features (number of features to consider for the best split).

2) K-Nearest Neighbors: K-Nearest Neighbors (KNN) is a non-parametric, instance-based learning algorithm that classifies a data point based on the majority class of its k nearest neighbors in the feature space [4]. The algorithm makes no assumptions about the underlying data distribution, making it suitable for problems where the decision boundaries are irregular.

For high-dimensional data like our wine dataset, KNN can suffer from the ”curse of dimensionality,” where the concept of distance becomes less meaningful as the number of dimensions increases. To address this challenge, we implemented feature selection and Principal Component Analysis (PCA) before applying KNN. Additionally, feature scaling was essential since KNN is distance-based and sensitive to the scale of input features. The most critical hyperparameter tuned was k, the number of neighbors to consider for classification.

3) XGBoost: XGBoost (Extreme Gradient Boosting) is an optimized implementation of gradient boosting decision trees designed for speed and performance [5]. Unlike Random Forest, which builds trees in parallel, XGBoost constructs trees sequentially, with each new tree correcting errors made by the previous ensemble.

XGBoost was selected for this study due to its well-documented performance advantages in various machine learning competitions and its effectiveness in handling structured data. Key hyperparameters tuned included n_estimators (number of trees), max_depth (maximum depth of trees), learning_rate (step size shrinkage used to prevent overfitting), and subsample (fraction of samples used for fitting individual trees).

C. Preprocessing Techniques

Several preprocessing techniques were applied to enhance model performance:

1) **Feature Selection:** To reduce dimensionality and focus on the most relevant features, we used SelectKBest with f_classif as the scoring function to identify the top 10 features most strongly associated with wine quality. This approach helped eliminate irrelevant or redundant features that could introduce noise into the models.
2) **Principal Component Analysis (PCA):** For the KNN model, we applied PCA to further reduce dimensionality

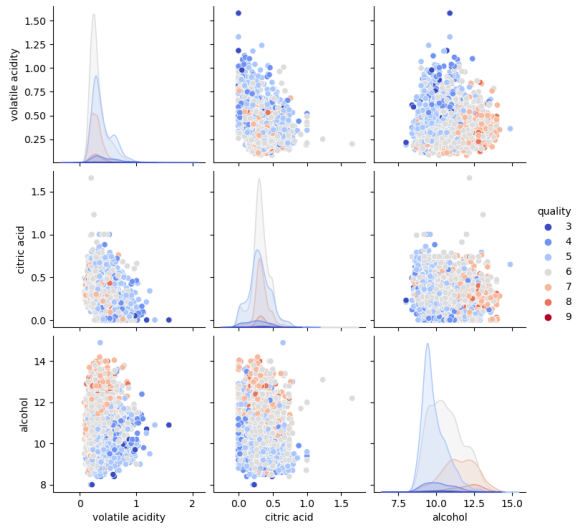


Fig. 3. Pairplot showing relationships between selected features (volatile acidity, citric acid, alcohol) and wine quality.

while preserving as much variance as possible. The number of components was selected to retain 98% of the original variance, which resulted in a more manageable feature space where distance calculations became more meaningful.

3) **Data Scaling:** Standard scaling was applied to the dataset, particularly for KNN, to ensure that all features contributed equally to the distance calculations. The features were transformed to have a mean of zero and a standard deviation of one using StandardScaler.
4) **Stratified Sampling:** To address class imbalance during model training and evaluation, we used stratified sampling in the train-test split to ensure that both training and test sets maintained the same class distribution as the original dataset.

III. Results and Analysis

A. Feature Analysis and Selection

The correlation heatmap (Fig. 2) revealed complex relationships between the physicochemical properties of wine. Most notably, alcohol showed the strongest correlation with quality (0.44), followed by volatile acidity (-0.27) and sulphates (0.04). Additionally, strong correlations were observed between related chemical properties, such as free sulfur dioxide and total sulfur dioxide (0.72), which was expected given their chemical relationship.

Feature selection using SelectKBest identified the top features most predictive of wine quality. The most important features, in descending order of importance, were alcohol, volatile acidity, sulphates, total sulfur dioxide, and pH. This finding is consistent with domain knowledge in winemaking, where alcohol content, acidity, and preservatives (sulphates and sulfur dioxide) are known to significantly impact wine quality.

TABLE I
Optimal Hyperparameters for Each Model

Model	Optimal Hyperparameters
Random Forest	n_estimators = 300 max_depth = 30 min_samples_split = 10 min_samples_leaf = 2 max_features = 'sqrt'
KNN	n_neighbors = 11 with PCA components preserving 98% variance
XGBoost	n_estimators = 500 max_depth = 7 learning_rate = 0.01 subsample = 0.8

TABLE II
Performance Metrics for Each Model

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	76.13%	0.70/0.79	0.63/0.84	0.66/0.82
KNN	74.81%	0.69/0.77	0.59/0.84	0.64/0.81
XGBoost	76.41%	0.70/0.80	0.64/0.84	0.67/0.82
Note: Precision, Recall, and F1-Score are presented as "bad wine/good wine"				

Fig. 3 displays the relationships between selected key features and wine quality. Higher-quality wines (represented by warmer colors) tend to cluster in regions with higher alcohol content and lower volatile acidity, while lower-quality wines show the opposite pattern. This visualization confirms the importance of these features in differentiating between wine quality levels.

B. Model Tuning and Performance

Extensive hyperparameter tuning was performed for each model using GridSearchCV with 5-fold cross-validation to ensure robust parameter selection. Table I summarizes the optimal hyperparameters for each model.

The three models were evaluated using accuracy, precision, recall, and F1-score metrics. Table II presents the performance metrics for each model on the test dataset.

XGBoost achieved the highest accuracy (76.41%), marginally outperforming Random Forest (76.13%), while KNN achieved a slightly lower accuracy (74.81%). All models demonstrated higher precision and recall for the majority class (good wines) compared to the minority class (bad wines), reflecting the underlying class imbalance in the dataset.

C. Confusion Matrices and Classification Analysis

Confusion matrices provide deeper insights into each model's classification performance. Fig. 5, 6, and 7 show the confusion matrices for Random Forest, KNN, and XGBoost, respectively.

All three models showed similar patterns in their confusion matrices, with a notable tendency to misclassify bad wines as good (false positives) more frequently than good wines as bad (false negatives). This pattern is consistent with the class

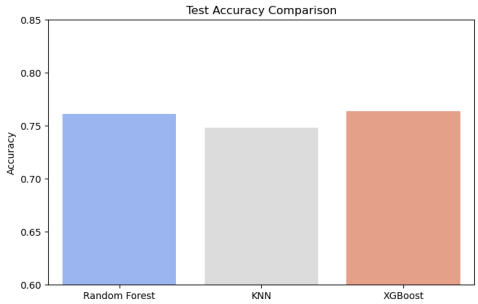


Fig. 4. Comparison of test accuracy across the three models.

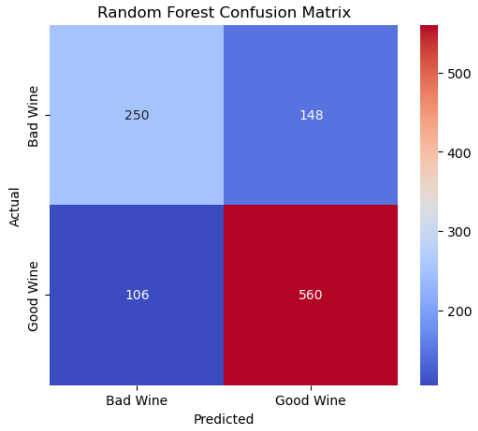


Fig. 5. Confusion matrix for Random Forest showing 250 true negatives, 148 false positives, 106 false negatives, and 560 true positives.

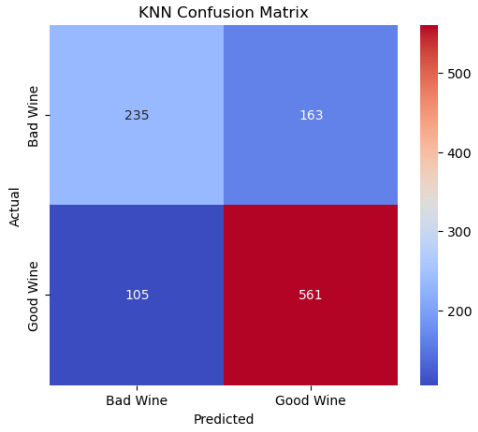


Fig. 6. Confusion matrix for KNN showing 235 true negatives, 163 false positives, 105 false negatives, and 561 true positives.

imbalance in the dataset, where good wines outnumber bad wines.

XGBoost performed the best in correctly identifying bad wines (true negatives: 256), followed by Random Forest (250) and KNN (235). This ability to correctly identify the minority class is particularly important in scenarios where misclassifying bad wine as good would be costly.

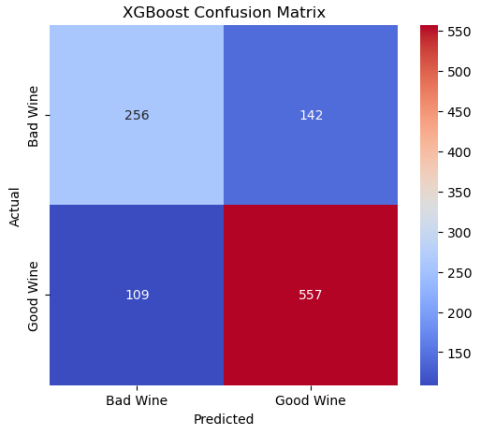


Fig. 7. Confusion matrix for XGBoost showing 256 true negatives, 142 false positives, 109 false negatives, and 557 true positives.

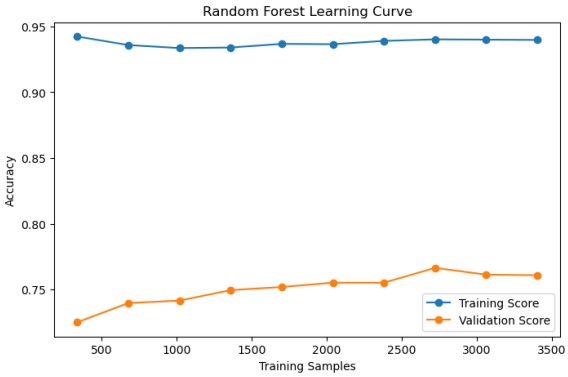
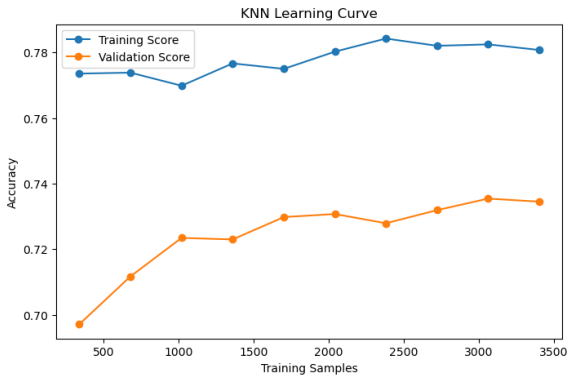


Fig. 8. Learning curve for Random Forest showing training and validation accuracy as a function of training samples.



W

Fig. 9. Learning curve for KNN showing training and validation accuracy as a function of training samples.

D. Learning Curves and Overfitting Analysis

Learning curves were analyzed to assess model behavior with increasing training data and to identify potential overfitting or underfitting. Fig. 8, 9, and 10 show the learning curves for Random Forest, KNN, and XGBoost, respectively.

The Random Forest learning curve (Fig. 8) shows signs

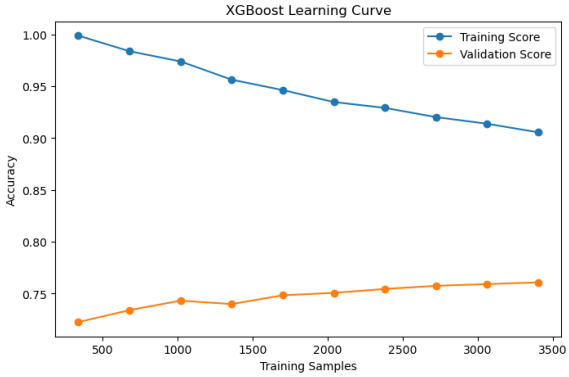


Fig. 10. Learning curve for XGBoost showing training and validation accuracy as a function of training samples.

of overfitting, with training accuracy (94-95%) significantly higher than validation accuracy (76%). Despite this gap, the validation curve plateaus, suggesting that additional training data would not substantially improve performance.

The KNN learning curve (Fig. 9) shows a smaller gap between training and validation accuracy, indicating less overfitting compared to the tree-based models. The gradually increasing validation accuracy suggests that KNN might benefit from additional training data.

The XGBoost learning curve (Fig. 10) initially shows severe overfitting, with training accuracy near 100%. However, as more training samples are included, the training accuracy gradually decreases while validation accuracy improves, indicating that the model's hyperparameters (particularly learning_rate=0.01 and subsample=0.8) effectively control overfitting.

IV. Conclusions and Discussion

A. Summary of Findings

This study compared the performance of three machine learning algorithms—Random Forest, KNN, and XGBoost—for predicting wine quality based on physicochemical properties. After extensive preprocessing and hyperparameter tuning, all three models achieved accuracies between 74.81% and 76.41%, with XGBoost slightly outperforming the others.

The analysis revealed several key insights:

1) **Feature Importance:** Alcohol content, volatile acidity, and sulphates emerged as the most influential predictors of wine quality, confirming existing domain knowledge in winemaking.
2) **Model Performance:** Tree-based ensemble methods (Random Forest and XGBoost) outperformed KNN, likely due to their ability to capture complex, non-linear relationships in the data and their robustness to irrelevant features.
3) **Class Imbalance:** All models struggled more with correctly classifying the minority class (bad wines), highlighting the challenges posed by class imbalance in wine quality prediction.
4) **Preprocessing Impact:** Feature selection and dimensionality reduction significantly improved model performance, particularly for KNN, which saw its accuracy increase from 50-55% to 74.81% after these preprocessing steps.

B. Limitations

Despite the promising results, this study has several limitations:

1) **Binary Classification:** Converting the multi-class problem to binary classification simplified the task but reduced the granularity of predictions, potentially limiting practical applications.
2) **Dataset Bias:** The dataset predominantly contained wines of average quality (5-7), with very few examples of extremely good or bad wines, which may limit the models' ability to generalize to wines at the extremes of the quality spectrum.
3) **Subjective Ratings:** The quality ratings, although provided by experts, inherently contain subjective elements that may introduce noise into the prediction task.
4) **Limited Feature Set:** The dataset included only 11 physicochemical properties, whereas wine quality is influenced by numerous other factors, including grape variety, aging process, and sensory characteristics not captured by chemical analysis.

C. Future Work

Several avenues for future research could address these limitations and further improve wine quality prediction:

1) **Multi-class Classification:** Returning to the original multi-class problem with techniques specifically designed for ordinal classification could provide more nuanced predictions.
2) **Advanced Sampling Techniques:** Exploring techniques such as SMOTE (Synthetic Minority Over-sampling Technique) or ADASYN (Adaptive Synthetic Sampling) could help address class imbalance more effectively.
3) **Feature Engineering:** Creating new features based on domain knowledge, such as ratios between certain chemical properties or polynomial features, could potentially capture more complex relationships.
4) **Deep Learning:** Investigating the application of neural networks, particularly those designed for tabular data, might reveal patterns that traditional machine learning algorithms miss.
5) **Ensemble Methods:** Combining predictions from multiple models through stacking or blending could potentially improve overall performance beyond what any single model can achieve.
6) **Additional Data:** Incorporating data on grape varieties, vineyard conditions, or aging processes could provide a more comprehensive basis for prediction.

In conclusion, this study demonstrates that machine learning models can effectively predict wine quality based on physicochemical properties, with ensemble methods like XGBoost

showing particularly promising results. While the achieved accuracy of 76.41% leaves room for improvement, it represents a solid foundation for automated quality assessment systems that could complement traditional sensory evaluation in the wine industry.

REFERENCES

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

[2] F. Brochet and D. Dubourdieu, "Wine descriptive language supports cognitive specificity of chemical senses," Brain and Language, vol. 77, no. 2, pp. 187-196, 2001.

[3] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[4] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21-27, 1967.

[5] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794.

[6] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[7] Lars Buitinck et al., "API design for machine learning software: experiences from the scikit-learn project," in ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108-122.

[8] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is 'nearest neighbor' meaningful?," in International Conference on Database Theory, 1999, pp. 217-235.