Red wine classification

An Empirical Verification of the Bias-Variance Tradeoff and Permutation-Based Feature Importance in Non-Linear Classifiers

Clara García Pérez and Thomas Richard

December 2025

Contents

1 Introduction

Living in France, you may have asked yourself what makes French wine so good. Well, our objective here is to determine what physicochemical properties influence human sensory perception of quality. For that, we are going to explore tree and forest techniques. We've observed in other Kaggle notebooks that simple Linear Regression or default Random Forest models achieve near-perfect accuracy.That's why our goal here is not to demonstrate spectacular performance, but rather to validate the theoretical results presented in lecture.

Single decision trees suffer from a fundamental limitation: they achieve perfect accuracy on training data by memorizing irrelevant patterns rather than learning generalizable relationships. We will see that ensemble methods combined with feature sub-sampling can reduce variance without materially increasing bias, allowing ensemble error to converge to a minimum that a single tree would never achieve.

This work provides empirical validation of classical ensemble learning theory in a controlled setting, demonstrating that intelligent regularization (both through pruning and ensemble averaging) transforms weak, over-fitted learners into robust, generalizing models.

2 Dataset presentation

2.1 Data Extraction

We are working with a dataset available in Kaggle of Portuguese "Vinho Verde", wine containing 1599 observations from which we have extracted 11 physicochemical properties and determined their quality on a scale from 3 to 8. We face a vector space of dimension 11 where the objective variable is discrete and ordinal (3-10). Therefore, we can solve either a regression problem trying to predict the `quality` or a classification problem where we consider a wine 'good' if quality ≥ 7 (1) and 'not good' if not (0). The issue with treating Y as categorical (classification) is that we lose the distance metric, that is because an error when predicting 3 instead of 8 is much worse than an error predicting 7 instead of 8. In regression, this ordinal structure is naturally preserved.

2.2 Variables characteristics

2.2.1 Acidity Profile

The following variables define the acidic structure of the wine, governing its stability, taste, and preservation.

- **Fixed Acidity**
  - *Statistics:* $\mu \approx 8.32$ g/dm^3, Range: $[4.6, 15.9]$.
  - This measure refers to non-volatile acids, predominantly **tartaric acid** ($C_4H_6O_6$). Unlike volatile acids, these do not evaporate readily and provide the essential structural backbone and crispness associated with high-quality wines.
- **Volatile Acidity**
  - *Statistics:* $\mu \approx 0.53$ g/dm^3, Range: $[0.12, 1.58]$.
  - This quantifies the presence of **acetic acid** (CH_3COOH) in the wine. While trace amounts are intrinsic to fermentation, elevated levels (> 1.2 g/dm^3) are indicative of bacterial spoilage, often resulting in an undesirable vinegar-like character.
- **Citric Acid**
  - *Statistics:* $\mu \approx 0.27$ g/dm^3, Range: $[0.0, 1.0]$.
  - Found in lower concentrations than tartaric acid, **citric acid** ($C_6H_8O_7$) is often used to boost freshness and flavor complexity. It plays a minor role in the total acidity but a significant role in flavor profile.
- **pH**
  - *Statistics:* $\mu \approx 3.31$, Range: $[2.74, 4.01]$.

- A logarithmic measure of the active hydrogen ion (H^+) concentration. Lower pH values correlate with higher acidity, which improves microbial stability and enhances the color intensity of red wines.

2.2.2 Sulfur & Preservatives

Sulfur dioxide is the primary antiseptic and antioxidant used in winemaking.

- **Free Sulfur Dioxide**
  - *Statistics:* $\mu \approx 15.9$ mg/dm^3, Range: $[1, 72]$.
  - This variable measures the SO_2 fraction available to react immediately with oxidative or microbial threats. It is the active form of the preservative, essential for preventing oxidation and spoilage during aging.
- **Total Sulfur Dioxide**
  - *Statistics:* $\mu \approx 46.5$ mg/dm^3, Range: $[6, 289]$.
  - The aggregate of free and bound SO_2. While necessary for preservation, excessive concentrations can mask fruit flavors and produce pungent off-odors.
- **Sulphates**
  - *Statistics:* $\mu \approx 0.66$ g/dm^3, Range: $[0.33, 2.0]$.
  - Typically related to the addition of potassium sulphate (K_2SO_4). These additives contribute to the concentration of sulfur dioxide gas (SO_2) and assist in maintaining the wine's hygienic state.

2.2.3 Composition & Structural Attributes

These characteristics influence the body, mouthfeel, and specific taste elements of the wine.

- **Residual Sugar**
  - *Statistics:* $\mu \approx 2.54$ g/dm^3, Range: $[0.9, 15.5]$.
  - Refers to natural grape sugars remaining after the completion of alcoholic fermentation. The observed range suggests that the majority of samples fall within the "dry" classification, contributing body without overt sweetness.
- **Alcohol**
  - *Statistics:* $\mu \approx 10.4\%$, Range: $[8.4, 14.9]$.
  - The percent alcohol by volume (ABV) produced during fermentation. Alcohol content is a primary driver of the wine's "body" and provides a warming sensation on the palate.
- **Chlorides**
  - *Statistics:* $\mu \approx 0.087$ g/dm^3, Range: $[0.012, 0.611]$.
  - A measure of the salt content (sodium chloride, $NaCl$). Elevated chloride levels can result from soil salinity and may impart a briny or salty taste if the concentration is significant.
- **Density**
  - *Statistics:* $\mu \approx 0.9967$ g/cm^3, Range: $[0.990, 1.004]$.
  - Density is dependent on the concentration of alcohol, sugar, and dissolved solids. Since alcohol is less dense than water and sugar is denser, this metric provides insight into the balance between residual sugar and alcohol content.

2.2.4 Target Variable

- **Quality**
  - *Statistics:* Median: 6, Range: $[3, 8]$.
  - The target variable representing the sensory evaluation score awarded by wine experts. The scale ranges from 0 (very bad) to 10 (very excellent), serving as a benchmark for the physicochemical properties listed above.

3 EDA

3.1 Distribution and dependency structure[1]

The dataset is complete and devoid of missing values; consequently, no significant data cleaning or imputation steps were required. The only preprocessing step involved the removal of 240 duplicate rows. Although in a chemical/wine context, two samples can be similar, in our case when we do a random split we could have the "same" observation in Train and in Test and the algo wouldn't be robust.

We will analyze the geometry of the variables, their marginal distributions, and the dependency structure. To do this, we examine the skewness and kurtosis. For a uni-modal distribution, negative skew typically indicates that the tail is on the left side of the distribution, while positive skew indicates that the tail is on the right. In cases where one tail is long but the other tail is fat, skewness does not follow a simple rule.

From the statistics output (Figure 1) output, we observe that `chlorides`, `residual sugar`, and `sulphates` exhibit positive skewness and high kurtosis, suggesting the presence of a significant right tail. This is visually confirmed in the histograms shown in red (Figure 2). What does this imply for a decision tree? Fortunately, decision trees are invariant to monotonic transformations and handle skewed distributions well, so this is not catastrophic. However, individual decision trees do suffer from **high variance**, meaning small changes in the training data can produce very different structures. Random Forests mitigate this through **bagging and ensemble averaging**, techniques that we will discuss later. Since Random Forests do not require feature scaling to handle these distributions, we will not scale our data at this stage.

Y has an approximately gaussian distribution centered around 5-6. The extreme classes (3-8) are rare events and we have no samples with quality 0, 1, or 2. The 3rd quartile of `quality` is 6 so a naive classifier predicting 'not a good wine' (quality less than 7) would achieve 86% accuracy. A weighted loss function or resampling would be a good idea if we want to beat that baseline.

	count	mean	std	min	25%	50%	75%	max	skewness	kurtosis
fixed acidity	1599.0	8.319637	1.741096	4.60000	7.1000	7.90000	9.200000	15.90000	0.982751	1.132143
volatile acidity	1599.0	0.527821	0.179060	0.12000	0.3900	0.52000	0.640000	1.58000	0.671593	1.225542
citric acid	1599.0	0.270976	0.194801	0.00000	0.0900	0.26000	0.420000	1.00000	0.318337	-0.788998
residual sugar	1599.0	2.538806	1.409928	0.90000	1.9000	2.20000	2.600000	15.50000	4.540655	28.617595
chlorides	1599.0	0.087467	0.047065	0.01200	0.0700	0.07900	0.090000	0.61100	5.680347	41.715787
free sulfur dioxide	1599.0	15.874922	10.460157	1.00000	7.0000	14.00000	21.000000	72.00000	1.250567	2.023562
total sulfur dioxide	1599.0	46.467792	32.895324	6.00000	22.0000	38.00000	62.000000	289.00000	1.515531	3.809824
density	1599.0	0.996747	0.001887	0.99007	0.9956	0.99675	0.997835	1.00369	0.071288	0.934079
pH	1599.0	3.311113	0.154386	2.74000	3.2100	3.31000	3.400000	4.01000	0.193683	0.806943
sulphates	1599.0	0.658149	0.169507	0.33000	0.5500	0.62000	0.730000	2.00000	2.428672	11.720251
alcohol	1599.0	10.422983	1.065668	8.40000	9.5000	10.20000	11.100000	14.90000	0.860829	0.200029
quality	1599.0	5.636023	0.807569	3.00000	5.0000	6.00000	6.000000	8.00000	0.217802	0.296708

Figure 1: Statistics from wine variables

[1]Some of the plots are inspired by https://www.kaggle.com/code/aminesudesolak/eda-red-wine-quality

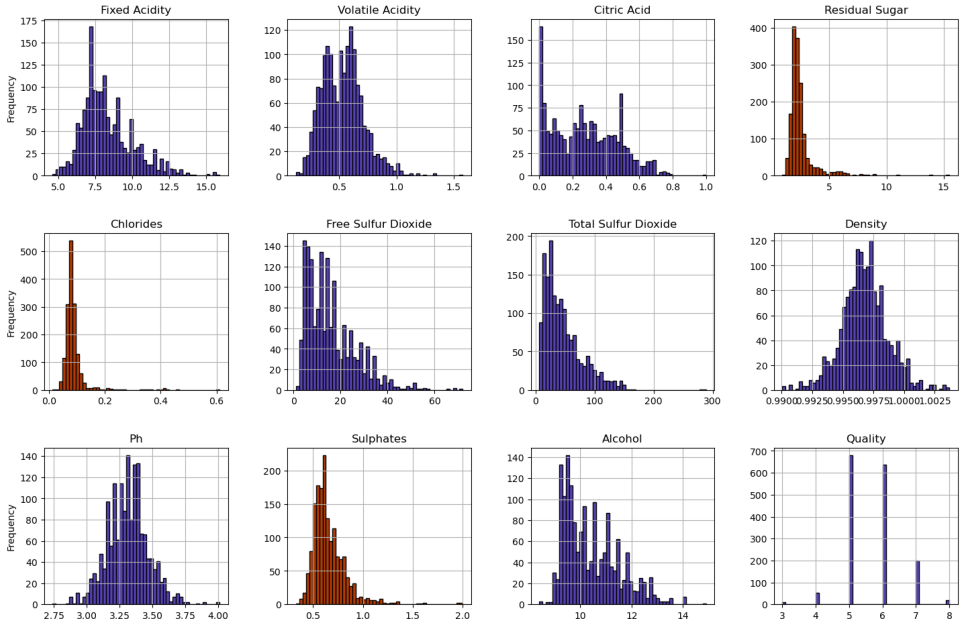


Figure 2:

3.2 Correlation analysis

The heatmap (Figure 3) displays the correlation structure of our wine features, and the scatterplots (Figure 4) provide visual evidence of the monotonic relationships observed in the heatmap. Our data is **ordinal**: an acidity of 7 is meaningfully less than an acidity of 11, and the order matters. This is why we use the Spearman correlation coefficient rather than Pearson, to apply a rank to the data. However, note that there is minimal difference between Pearson and Spearman correlations in our dataset, indicating that the dependencies are predominantly **linear** rather than just monotonic.

Difference:

- **Pearson Correlation** measures *linear* relationships: how well a straight line $y = \alpha x + \beta$ describes the relationship between two variables, where the rate of change is constant.
- **Spearman Correlation** measures *monotonic* relationships: whether two variables consistently move in the same direction (either always increasing or always decreasing), regardless of whether the rate of change is constant.

From `corr_matrix`, we observe strong Spearman correlations that reflect underlying chemical and physical laws[2]:

- **Fixed Acidity vs pH** ($\rho \approx -0.71$): Negative, since pH $\approx -\log[H^+]$.
- **Density vs Alcohol** ($\rho \approx -0.47$): Ethanol is less dense than water.
- **Density vs Residual Sugar** ($\rho \approx 0.41$): Dissolved sugar increases specific gravity.

These high linear correlations won't break our predictions, but they will break our interpretation. One of the great advantages of Random Forests (RF) is the explicability via feature importance, which is based on how much a feature decreases impurity when a split is made. The problem is that if Feature A and Feature B are highly correlated, they contain almost the same information. The tree will arbitrarily choose one or the other to make a split. Therefore, the "importance" score gets split between the two variables. Instead of Feature A having a score of 0.4, Feature A gets 0.2 and Feature B gets 0.2. This is why permutation importance and feature selection become critical (e.g., potentially removing one feature).

[2]Ref: ChatGPT

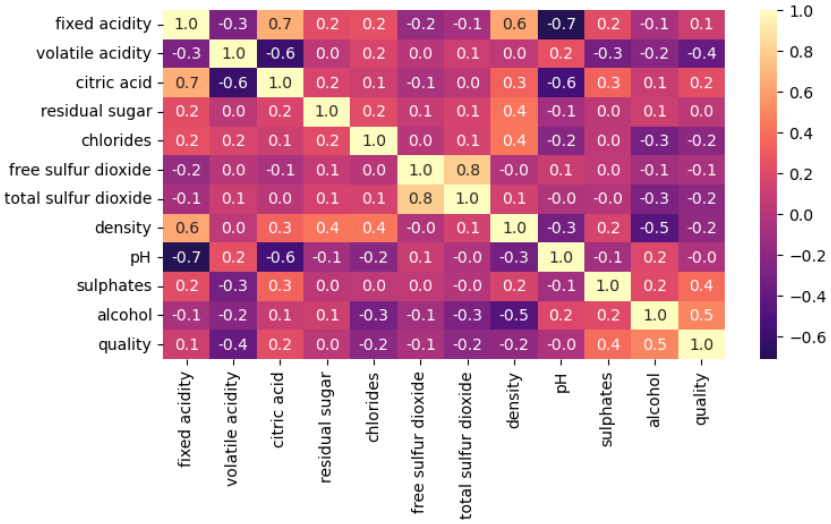


Figure 3: Spearman correlation matrix (monotonic relations)

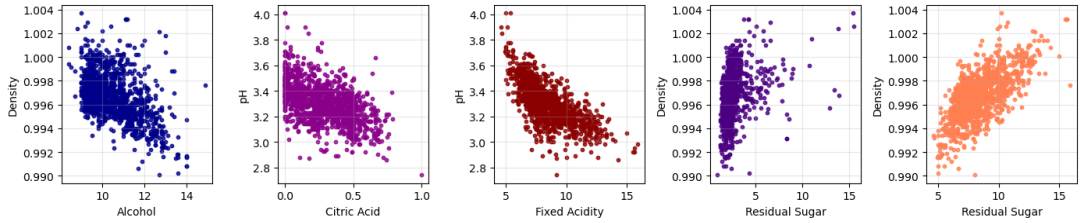


Figure 4: 1vs1 correlated variables

3.3 Class separation

The boxplots reveal that clear multiclass separation is not visually apparent. We observe that higher `alcohol` median quality increases. However, with `sulphates` and `density` features, we see significant overlap between the quality boxes, indicating that no simple hyperplane (single split) can cleanly separate good wines from bad wines. This overlap suggests that the problem requires a more complex, nonlinear decision that Random Forests can capture.

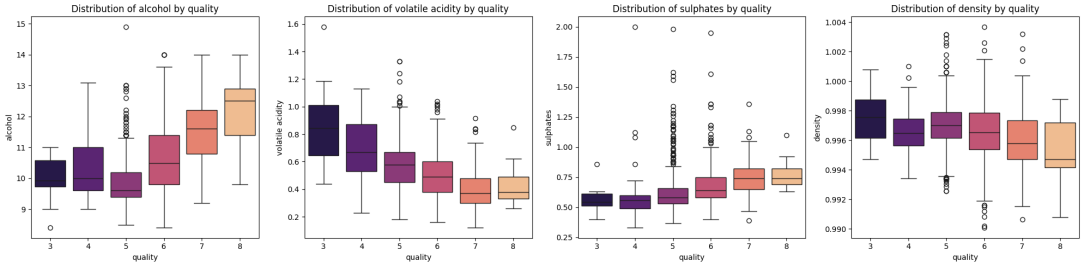


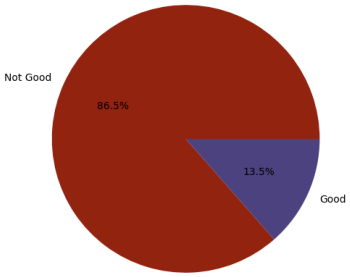
Figure 5: Distribution of some variables by quality

4 Establish baseline overfitting behavior with fully-grown decision trees

4.1 Binary classification task

As we discussed, we proceed with the binary classification task, distinguishing between good' and not good' wines. The original dataset is highly imbalanced, with 86% of wines having a quality rating below

7 (Figure 4.1).



To overcome the naive classifier bias (which would simply predict "Not Good" for all samples), we balance the dataset by ensuring equal representation of both classes: "Good Wine" (quality > 6) and "Not Good Wine" (quality ≤ 6). We proceeded to train fully grown decision trees on both dataset versions to establish a baseline performance and demonstrate overfitting. The initial results are summarized in Table 1.

Dataset	Depth	Train Acc	Test Acc	X_train	X_test	Overfitting Gap
Unbalanced	17	1.0	0.8824	(1087, 11)	(272, 11)	0.1176
Balanced	11	1.0	0.7297	(294, 11)	(74, 11)	0.2703

Table 1: Performance comparison: Unbalanced vs. Balanced Datasets

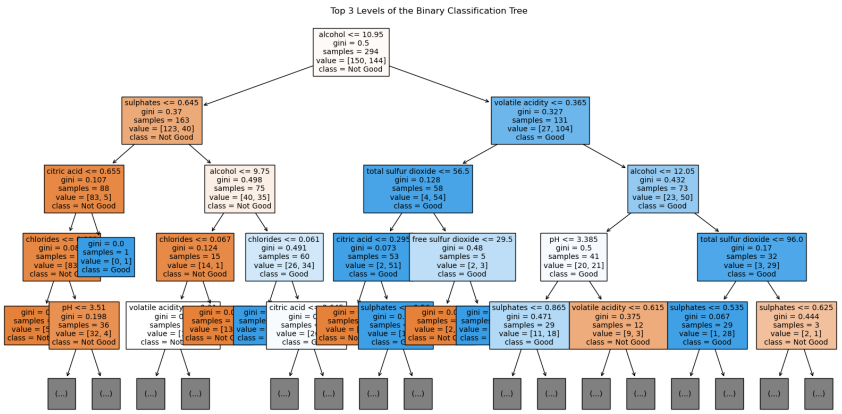


Figure 6: Top 3 Levels of the Binary Classification fully grown Tree

As expected, the model has learned **training-specific** noise rather than generalizable patterns. The tree achieves perfect **memorization** on 294 training samples but fails to generalize to the unseen 74 test samples.

The 88% accuracy in the unbalanced case might appear high, but it is actually close to the naive baseline of 86%, confirming that the model provides little value over a majority-class guess. The moment we balance the dataset, the lack of generalization becomes evident even for the overfitted tree.

At first glance, `alcohol` and `sulphates` appear to be key quality indicators; however, the depth-11 structure suggests the model exploits specific feature interactions that may not exist in the general population.

Regarding the splitting criterion, we adhere to the default **Gini impurity**, as it is the standard for classification and typically yields results comparable to entropy or misclassification error. The decreasing Gini values down the tree ($0.5 \rightarrow 0.37 \rightarrow 0.107$) indicate progressive class purity. However, this optimization occurs exclusively on the training data, creating a classic bias-variance tradeoff:

- **Low Bias:** Training Accuracy $= 100\%$
- **High Variance:** Test Accuracy drops to 73%

4.2 Addressing Variance: Cost-Complexity Pruning

To mitigate the high variance observed in the fully grown tree, we employ **Cost-Complexity Pruning**. This technique introduces a regularization parameter, $\alpha \geq 0$, which penalizes the tree's complexity. The algorithm minimizes the cost-complexity measure:

$$R_\alpha(T) = R(T) + \alpha|T|$$

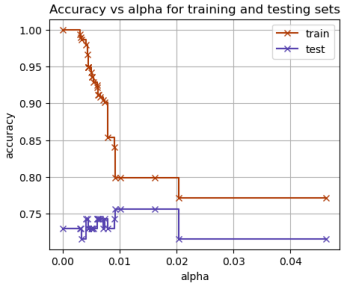
where $R(T)$ is the total misclassification rate (impurity) of the terminal nodes and $|T|$ is the number of leaves in T. As α increases, the penalty for having more leaves increases, forcing the tree to become smaller and simpler. For a fully-grown tree T_0, $\alpha = 0$ and $\mathrm{R}(T_0) = 0$

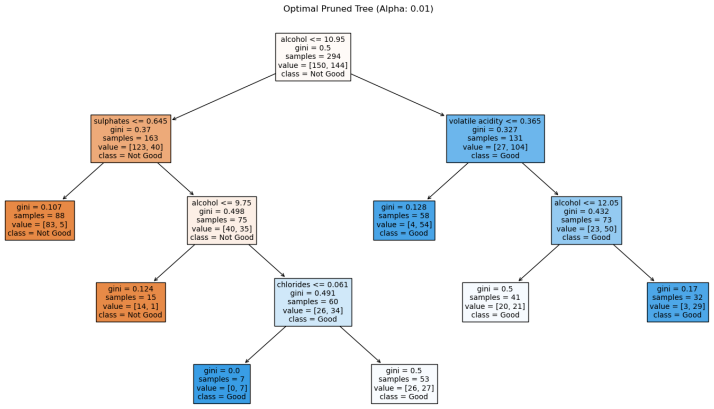
We computed the effective α values associated with the pruning path of our trained tree (using `cost_complexity_pruning_path`), which generated a sequence of candidate alphas ranging from the fully grown tree ($\alpha = 0$) to the root-only tree. From this sequence, we selected the optimal alpha that maximized test accuracy, along with two intermediate values for comparison. Table 2 summarizes the impact of varying α on the tree's structure and performance.

Alpha (α)	**Tree Depth**	**Testing Accuracy**
0.0000	11	72.97%
0.0033	11	71.62%
0.0093	**4**	**75.68%**
0.0463	1	71.62%

Table 2: Impact of Pruning Parameter α on Generalization

By varying the complexity parameter, we observe how constraining tree growth sacrifices training accuracy to achieve superior generalization. As shown in the table, the optimal α (≈ 0.009) drastically reduces the depth from 11 to 4, yet maximizes **Testing Accuracy** to 75.68%. This proves that deliberately "underfitting" the training data (sacrificing memorization) is necessary to uncover robust, generalizable patterns.





5 Random Forest

5.1 Key Mechanisms for Variance Reduction

Here is the core principle enabling Random Forests to generalize well, mechanisms reducing ρ (prediction variance) without significantly increasing bias, achieving the optimal bias-variance tradeoff. Let's discuss two of them.

5.1.1 Bootstrap Aggregation (Bagging)

Default Behavior: Random Forest uses bootstrap sampling by default (`bootstrap=True`). Each of the 50 trees in the ensemble is trained on a random sample of the training data drawn **with replacement**.

This means each tree sees approximately 63.2% of unique training samples, while some samples may be repeated and others omitted.

Advantages:

- **Variance Reduction:** By averaging predictions across multiple models trained on different data subsets, bagging reduces the variance of the ensemble compared to a single tree.
- **Decorrelation:** Trees trained on different bootstrap samples make different errors, so averaging these errors provides a smoother prediction.
- **Out-of-Bag (OOB) Evaluation:** we will talk about it later.
- **Computational Efficiency:** Each tree only processes a fraction of the data, reducing computational cost.

5.1.2 Feature Randomization at Each Split

Default Behavior: At each split, Random Forest considers only a random subset of features (`max_features='sqrt'` for classification). With 11 features in the wine dataset, approximately 3–4 features are randomly selected for each split decision across all trees.

Advantages:

- **Increased Diversity:** Forcing trees to use different features prevents them from all selecting the same "dominant" features, creating more varied decision boundaries.
- **Reduced Feature Correlation:** When all trees use the same best features, their predictions become highly correlated. Random feature selection breaks this correlation.
- **Robustness to Irrelevant Features:** Random feature selection allows weak features to occasionally influence splits, preventing the model from over-relying on potentially spurious correlations.

5.2 Empirical Analysis: Impact of Bagging and Feature Randomization

When disabling bootstrap aggregation and using all features for splits (`bootstrap=False`, `max_features=11`), the Random Forest achieves perfect training accuracy (100%) but catastrophic test generalization (56.87%), whereas the **default** configuration with bootstrap sampling and feature randomization (`bootstrap=True`, `max_features='sqrt'`) maintains the same training accuracy while dramatically **improving test** performance to 75.68%, demonstrating the critical importance of both mechanisms for variance reduction.

The interaction between bootstrap sampling and feature randomization creates an ensemble where each tree is a slightly different "expert", predictions are averaged across these diverse experts.

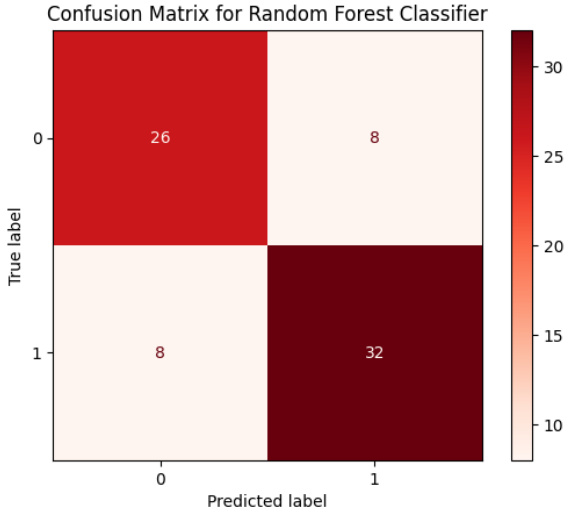
5.3 Hyperparameter Optimization Generalization Capability: CV vs. OOB error

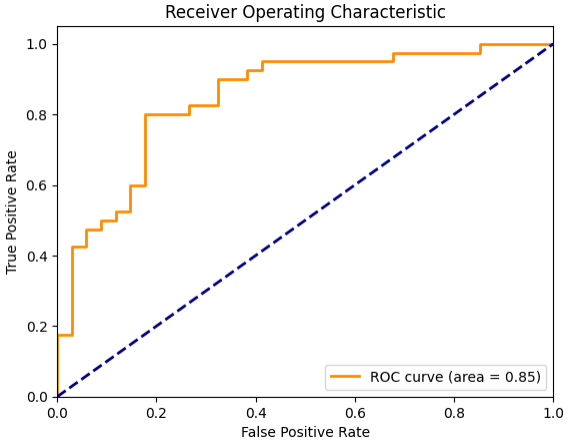
We employed **5-fold cross-validation with RandomizedSearchCV** to explore 100 random hyperparameter combinations across `n_estimators`, `max_depth`, `min_samples_leaf`, and `criterion`. The search yielded an optimal configuration defined by `criterion='entropy'`, `n_estimators=193`, `max_depth=35`, and `min_samples_leaf=4`. To assess the model's generalization stability, we compared the internal validation metrics:

- **Mean CV Accuracy:** 0.7959
- **OOB (Out-of-Bag) Accuracy:** 0.7585

The best cross-validation accuracy of **0.796** represents the average performance across the 5 folds, while the OOB score of **0.759** provides a conservative estimate using data points excluded during the bootstrapping of individual trees. The gap between these metrics is relatively small, indicating that the model is stable and not heavily overfitting to specific folds.

Ultimately, the model achieved a final **test accuracy of 0.7838**. This result falls between the conservative OOB estimate and the optimistic CV score, confirming the validity of the optimization process and significantly outperforming the baseline single decision tree (73.0% test accuracy).



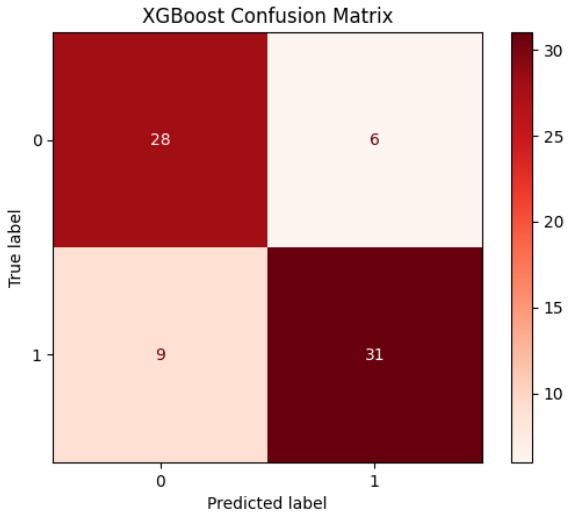


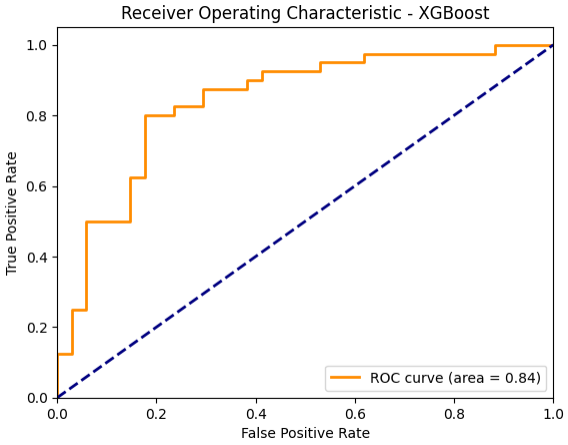
To further evaluate the model's discriminative capability, we analyze the Confusion Matrix (Figure 5.3) and the Area Under the Curve (AUC). The Random Forest achieved an **AUC of 0.841**, indicating a strong probability that the model ranks a randomly chosen "Good" wine higher than a "Not Good" one. The confusion matrix reveals a balanced classification performance, though the trade-off between sensitivity and precision remains a critical consideration depending on the specific business application (e.g., minimizing false positives to protect brand reputation).

6 Gradient Boosting (XGBoost)

To verify if the performance could be further improved by reducing bias, we implemented **XGBoost**, a gradient boosting framework that builds trees sequentially to correct the errors of previous estimators, rather than independently like Random Forest.

The XGBoost model achieved a final **test accuracy of 0.793** and an **AUC of 0.851**, slightly outperforming the optimized Random Forest (Accuracy: 0.784, AUC: 0.841). As shown in the Confusion Matrix (Figure 6), the boosting mechanism managed to correctly classify a marginally higher number of difficult instances. While the improvement is incremental ($\approx 1\%$), it demonstrates the effectiveness of boosting in minimizing residual errors that bagging could not eliminate, albeit at the cost of higher training complexity and sensitivity to noise.





7 Other models

In order to have a model to compare-with, we train two machine learning models for the binary classification of red wine quality: a Support Vector Machine (SVM) and a Multi-Layer Perceptron (MLP) neural network implemented using the *PyTorch* framework.

7.1 Support Vector Machine (SVM)

The SVM was configured with a Radial Basis Function (RBF) kernel to account for non-linear relationships between chemical features. The model achieved an Accuracy of 77.03% and an AUC score of 0.837. The confusion matrix (Table 3) demonstrates high reliability in identifying high-quality wines.

With SVM too, volatile acidity came before sulphates.

SVM	Predicted: Not Good	Predicted: Good
Actual: Not Good	25	9
Actual: Good	8	32

Table 3: Confusion Matrix for the SVM model

7.2 Multi-Layer Perceptron (MLP)

The artificial neural network consists of an input layer (11 features), two hidden layers (32 and 16 neurons with ReLU activation), and a final output layer with a Sigmoid activation function.

Following 100 training epochs, the MLP achieved an Accuracy of 66.22% and an AUC of 0.755. While slightly trailing the SVM on this specific test set, the network successfully captured the general patterns of the dataset.

MLP (PyTorch)	Predicted: Not Good	Predicted: Good
Actual: Not Good	21	13
Actual: Good	12	28

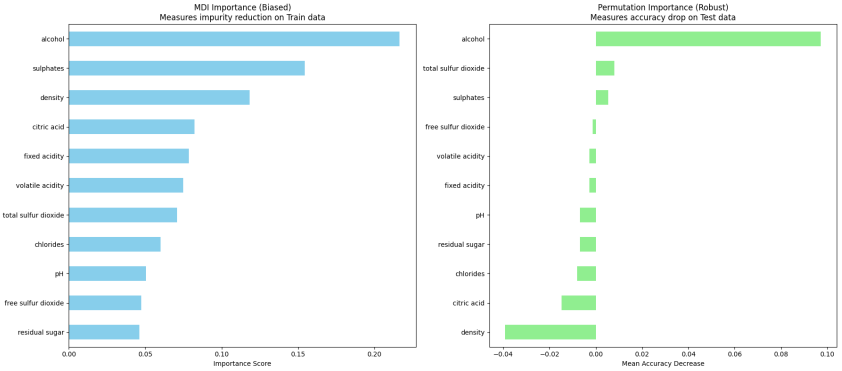
Table 4: Confusion Matrix for the MLP model

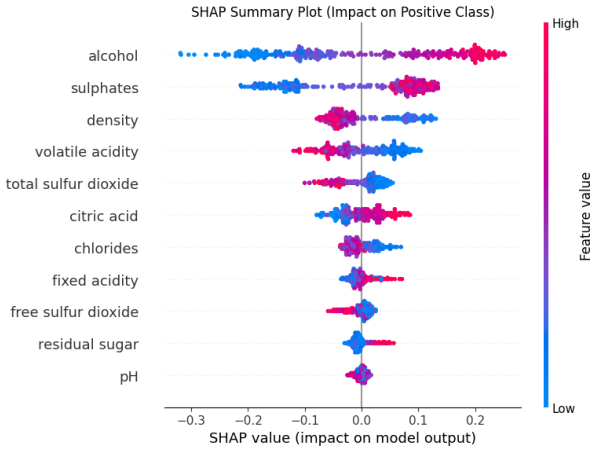
8 Feature importance

To interpret the model's decision-making process, we compared three distinct importance metrics: Mean Decrease in Impurity (MDI), Mean Decrease in Accuracy (MDA), and SHAP values of the best Random Forest. Across all methods, a consistent hierarchy of features emerges.

The following results were based on XGBoost and best RF MDI, MDA and SHAP. MDA, MDI and SHAP offer complementary perspectives on feature relevance. MDI delivers a fast global ranking and highlights alcohol and sulphates as the strongest predictors, but it is known to be biased toward features that are frequently used in splits. Permutation importance (MDA on `X test`) assesses impact on generalization and largely confirms alcohol and sulphates as top drivers while down-ranking features that MDI inflated (like fixed acidiy or volatile acidity). SHAP provides local, directional and interaction-aware explanations: it shows that higher alcohol and higher sulphates increase the model's probability of predicting "Good".

In summary, the Random Forest and the XGBoost successfully captured the fundamental enological "rules": high-quality wine requires a balance of sufficient alcohol and preservation, coupled with the absence of bacterial spoilage (volatile acidity).





The initial correlation analysis revealed significant dependencies between several features, notably (`volatile_acidity`, `citric_acid`), (`pH`, `fixed_acidity`), and (`citric_acid`, `pH`). To observe how these redundancies mask individual feature contributions, we retrained the Random Forest excluding `pH`. This intervention led to a substantial reconfiguration of the importance hierarchy: in the MDI ranking, `volatile_acidity` surged from 6th to 3rd place, while in the MDA (Permutation) ranking, it rose to surpass `sulphates`!!

This shift demonstrates the *importance dilution* effect inherent in ensemble models. When features are highly correlated, the Random Forest arbitrarily distributes importance scores among them during the splitting process. By removing `pH`, we reduced the noise in the acidity-related feature group, allowing the model to more consistently select `volatile_acidity` as a primary splitting criterion. The fact that its MDA score now exceeds `sulphates` suggests that `volatile_acidity` contains unique, high-variance information about wine spoilage that was previously partially attributed to or masked by its chemical relationship with the pH level. This confirms that feature selection is not merely a performance tool, but a necessary step for accurate model interpretability.

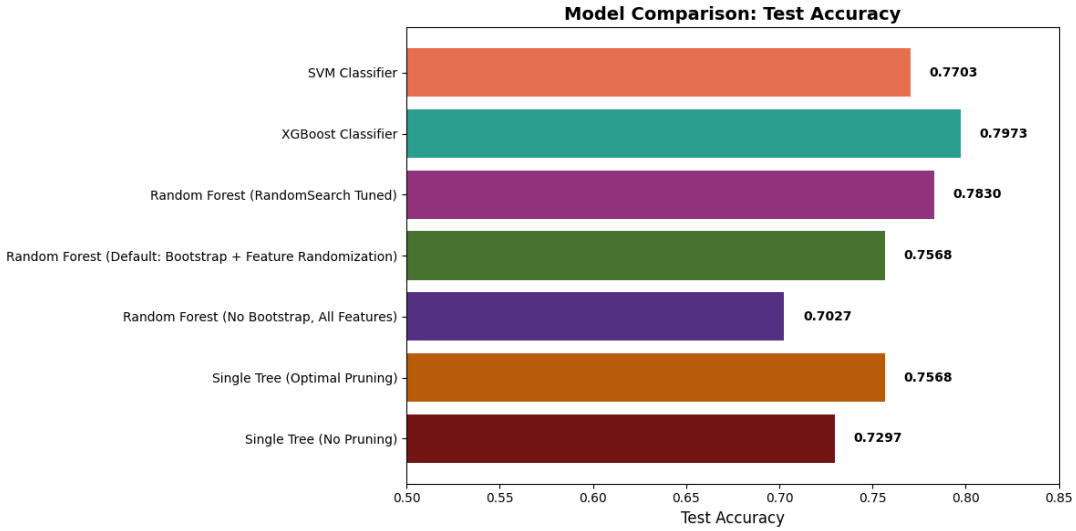
9 Conclusion

The driving force of this project was not merely to build a predictive tool, but to empirically verify fundamental Random Forest theories using a real dataset.

While XGBoost yielded the highest raw metrics, the Random Forest remains the most balanced candidate for this domain. It offers competitive accuracy ($\approx 1\%$ difference) while maintaining superior stability and more straightforward interpretability—qualities.

Table 5: Model Comparison: Test Accuracy Across All Approaches

Model	**Test Accuracy**
Single Tree (No Pruning, $\alpha = 0.0$)	0.7297
Single Tree (Optimal Pruning, $\alpha =$ best)	0.756
Random Forest (No Bootstrap, All Features)	0.705
Random Forest (Default: Bootstrap + Feature Randomization)	0.7568
Random Forest (RandomSearch Tuned)	0.783
XGBoost	0.797
SVM	0.77



Our analysis establishes a distinct physicochemical profile for superior wines. Foremost among these drivers is alcohol content, which serves as the primary positive indicator; this suggests that wines possessing greater body and associated with optimal grape ripeness are consistently favored by expert raters. Therefore, we should drink wine with moderation. Complementing this, sufficient sulphate levels emerge as a non-negotiable prerequisite, likely due to their essential role in preventing oxidation and maintaining stability. However, this structural strength must be paired with chemical "hygiene." The analysis highlights low density and high citric acid as critical negative filters.