# Wine Quality Classification with Random Forest: A Comprehensive Tutorial

## 1. Introduction

### 1.1 Motivation and Overview

Wine is a complex beverage whose quality is a function of a Number of chemical and sensory characteristics, namely acidity, sugar content and alcohol percentage. The properties of volatiles have for long been the subject of study by researchers and vintners in an effort to understand and predict overall wine quality. The Wine Quality from the UCI Machine Learning Repository has become the favorite dataset for playing with classification and regression methods in modern machine learning practice.

This tutorial is focused on predicting if a red wine is of 'good' or 'bad' quality, and we use a Random Forest Classifier for this. Particularly, the raw quality scores included in this dataset are binarized, ranging from 3 to 8 in those scores.

- **Good Quality (1)**: quality $\geq 6$
- **Bad Quality (0)**: quality $< 6$

By this way, the problem is simplified to a binary classification, which makes it easy to evaluate the model with classification metrics such as accuracy, precision, recall, etc. In addition, we show how to tune hyperparameters with GridSearchCV, emphasize the significance of data visualization as well as exploratory data analysis (EDA) and interpreting results in pleasing plot.

### 1.2 Why Random Forest?

Randam Forest is a bagging (bootstrap aggregation) ensemble of decision trees that makes use of the two main concepts, bagging, and feature randomness. In addition, it is very popular for several reasons. (Breiman, 2001)

1. **Robustness**: By averaging over multiple trees, Random Forest reduces the risk of overfitting compared to a single decision tree. (Breiman, 2001)
2. **Feature Importance**: It naturally provides feature importance scores, helping us interpret which chemical attributes most strongly influence wine quality.
3. **Hyperparameter Flexibility**: We can fine-tune parameters such as the number of trees, max depth, and leaf size to optimize performance. (Breiman, 2001)
4. **Works Well with Tabular Data**: For a dataset like Wine Quality, which is structured, Random Forest often yields excellent performance with minimal preprocessing.

In this tutorial, we will see in action how these advantages play out, ending up with a white (oops) high performing and interpretable classification model for the quality of red wine.

# 2. Dataset and Problem Setup

## 2.1 Data Source

We use the **Red Wine Quality** dataset from the UCI Machine Learning Repository. It has **1599 samples**, each representing a type of Portuguese "Vinho Verde" red wine, with 11 chemical/physical features and 1 quality score.

**Key columns** in the dataset:

- **fixed acidity**: Tartaric acid concentration ($g/dm^3$)
- **volatile acidity**: Acetic acid concentration ($g/dm^3$)
- **citric acid**: Citric acid content ($g/dm^3$)
- **residual sugar**: Sugar left after fermentation ($g/dm^3$)
- **chlorides**: Salt content ($g/dm^3$)
- **free sulfur dioxide**: Free $SO_2$ content ($mg/dm^3$)
- **total sulfur dioxide**: Total $SO_2$ content ($mg/dm^3$)
- **density**: Density of the wine ($g/cm^3$)
- **pH**: Acidity level
- **sulphates**: Potassium sulphate content ($g/dm^3$)
- **alcohol**: Alcohol percentage (% vol.)
- **quality**: An integer score from 3 to 8 (the higher the better)

## 2.2 Creating a Binary Target

We transform the **quality** score into a binary target called **good_quality**:

- **1** (Good) if quality $\geq 6$
- **0** (Bad) if quality $< 6$

This step is common in many classification tasks that require a simple yes/no outcome. In our dataset:

- **good_quality = 1** for 855 samples
- **good_quality = 0** for 744 samples

Thus, we have a relatively balanced distribution, although not perfectly even.

## 2.3 Splitting the Data

Once we can read the dataset in a pandas DataFrame and clean it (if needed), we split the data into the training and testing sets.

- **Training set size**: 1279
- **Test set size**: 320

We use train_test_split(..., stratify=y, random_state=42) to ensure reproducibility and maintain the class distribution.

# 3. Exploratory Data Analysis (EDA)

The dataset has to be carefully looked into through the process of exploratory data analysis, as this is crucial in forming hypotheses regarding what factors affect wine quality.

## 3.1 Pairplot of Selected Features

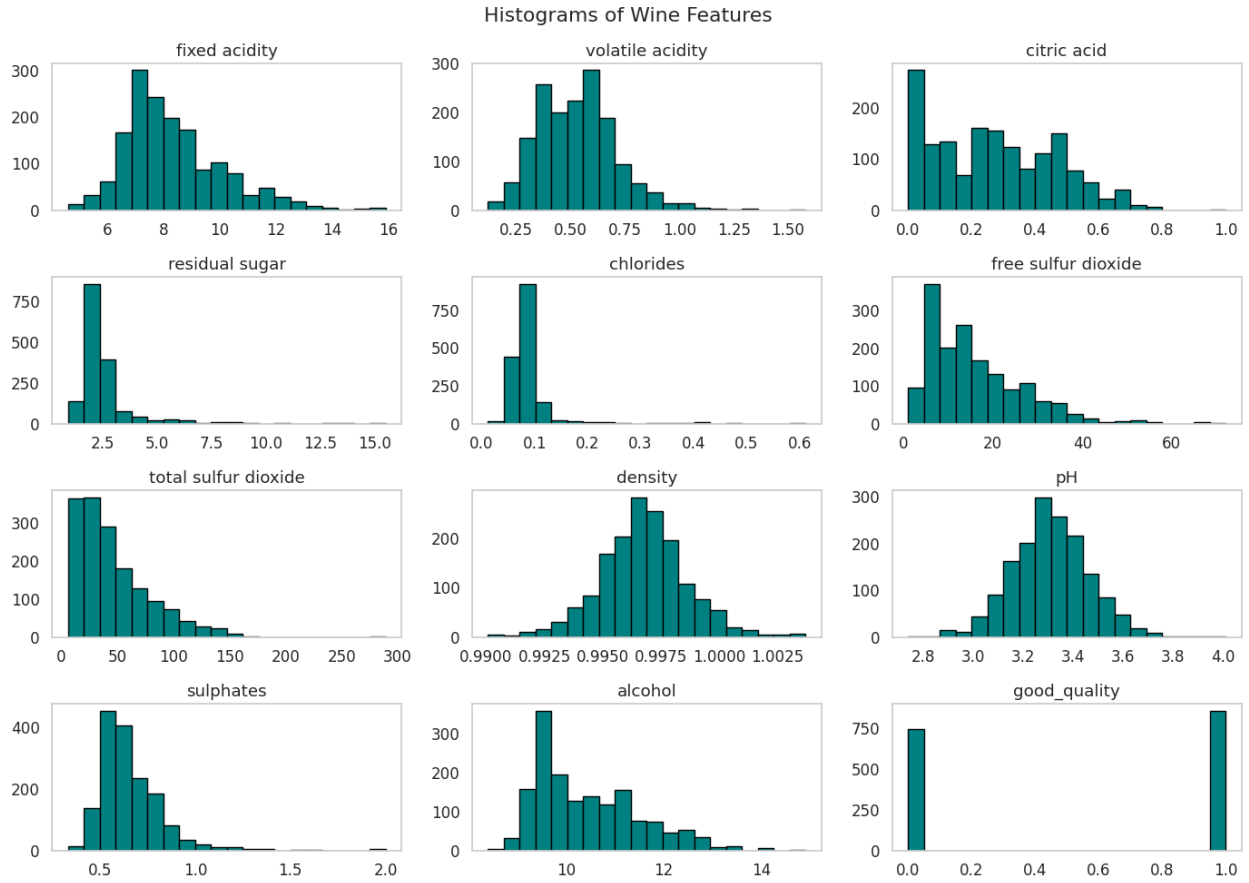
Using Seaborn's pairplot, we see how a pair of features such as fixed acidity, volatile acidity, citric acid, residual sugar, and alcohol relate to each other colored by the good_quality label. Observations:

- Wines with higher **alcohol** often appear more likely to be good (purple points for good_quality=1 cluster at higher alcohol levels).
- **Volatile acidity** tends to be lower in good wines, aligning with the domain knowledge that excessive volatile acidity can degrade flavor.

## 3.2 Histograms of Wine Features

Histograms of each feature (e.g., pH, sulphates, residual sugar) reveal:

- **pH** mostly clusters around 3.0 to 3.5, typical for acidic wines.
- **sulphates** distribution shows a right skew, indicating some wines have significantly higher sulphate content.
- **alcohol** peaks around 9-10% with a tail extending to about 14%, a key factor in perceived wine quality.

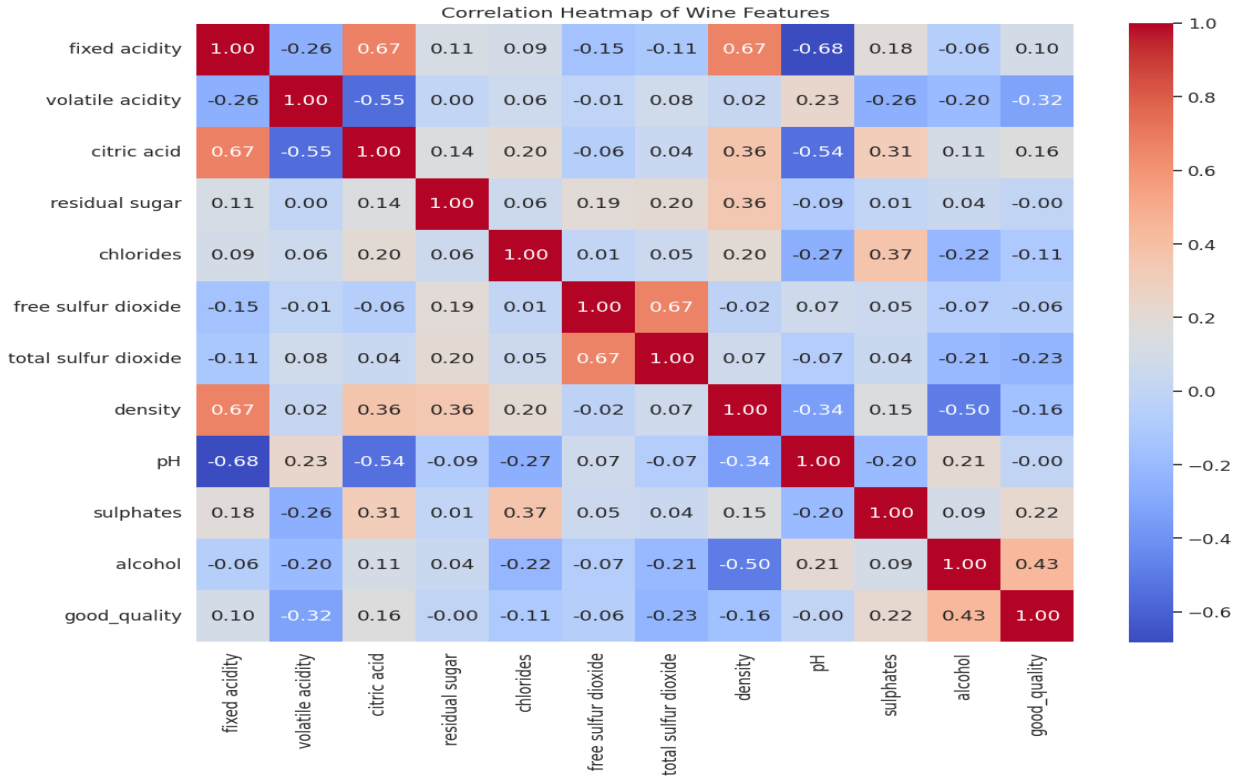



## 3.3 Correlation Heatmap

A correlation matrix with annotated values highlights relationships:

- **alcohol** is positively correlated (0.44) with good_quality, the highest among features.
- **volatile acidity** is negatively correlated (-0.39) with good_quality, confirming that high volatile acidity generally leads to lower quality.

- **total sulfur dioxide** and **free sulfur dioxide** have moderate positive correlations with each other, but less so with quality.




These insights guide our modeling approach and help interpret the final results.

# 4. Data Preprocessing and Feature Engineering

## 4.1 Data Cleaning

For this dataset, the number of missing values is usually minimal or zero. In case of abnormalities in your case, i.e. wrongly formatted rows, remove or correct them. UI takes in dataset from UCI which usually is clean. so we primarily focus on:

- **Dropping quality** once we create good_quality.
- **Checking for outliers** in chemical features. If extreme outliers exist, we can consider transformations like log-scaling. However, Random Forest is robust to outliers, so transformations might be optional.

## 4.2 Final Feature Matrix and Target Vector

We store:

- **X** = all features except good_quality
- **y** = good_quality (0 or 1)

Thus, each row in **X** is a wine's chemical profile, while **y** indicates if the wine is "good."

# 5. Model Building: Random Forest Classifier

## 5.1 Why Random Forest?

Each decision tree is trained on random samples (called bootstrap) of the training data in Random Forest. And at each split, it chooses randomly a subset of features leading to better generalization and smaller variance. (Breiman, 2001)

**Key Hyperparameters**:

1. **n_estimators**: Number of trees in the forest (e.g., 100, 200).
2. **max_depth**: Maximum depth of each tree (e.g., None means unlimited).
3. **min_samples_split**: Minimum samples required to split an internal node.
4. **min_samples_leaf**: Minimum samples required to be a leaf node.
5. **bootstrap**: Whether to use bootstrap samples (True) or the entire dataset (False).

## 5.2 Hyperparameter Tuning (GridSearchCV)

To find the best combination of hyperparameters, we define a **param_grid**:

```
# 5. Model Training: Random Forest Classifier with Hyperparameter Tuning
param_grid = {
    'n_estimators': [100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5],
    'min_samples_leaf': [1, 2],
    'bootstrap': [True, False]
}
```

```
grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=5, n_jobs=-1, scoring='accuracy')
grid_search.fit(X_train, y_train)

print("\nBest Parameters:", grid_search.best_params_)
print("Best CV Accuracy:", grid_search.best_score_)
```

We then run:

**Best CV Accuracy**: ~0.8108

With these settings aside we have a well tuned model of roughly equal parts complexity and generalization. We finalize the model as best_rf = grid_search.best_estimator_.

# 6. Model Evaluation and Results

## 6.1 Classification Report

On the test set of 320 wines, the final classification report is:

Classification Report:

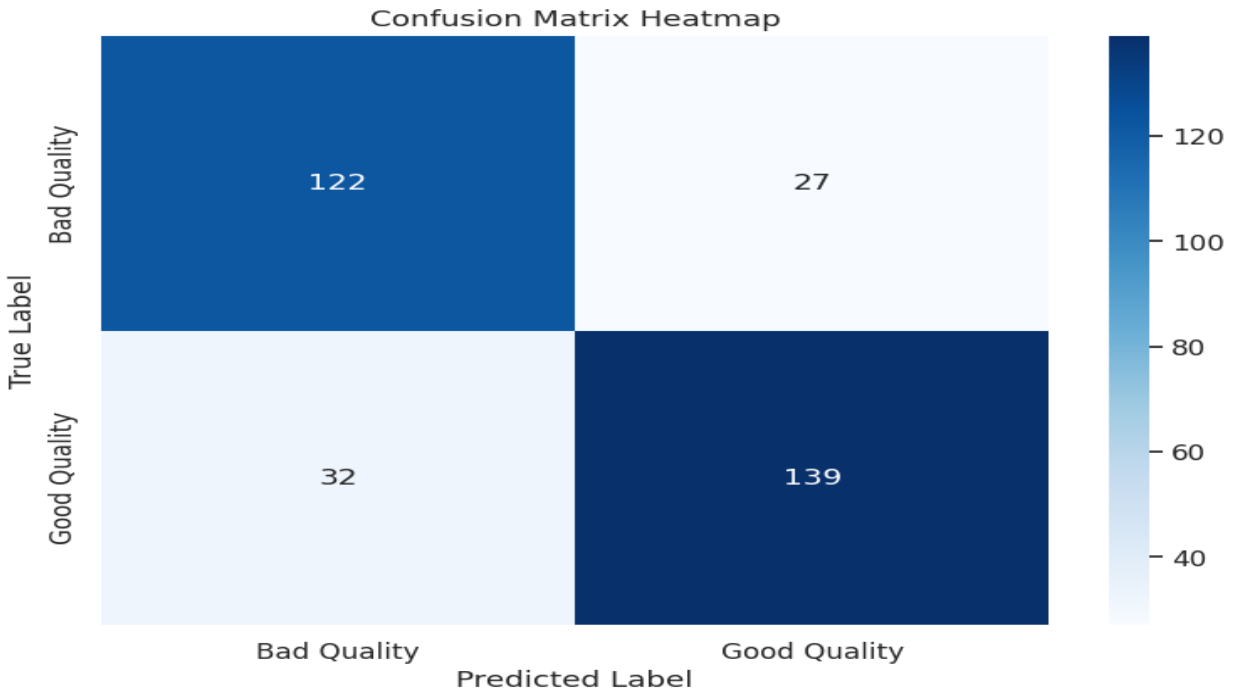
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.79 | 0.82 | 0.81 | 149 |
| 1 | 0.84 | 0.81 | 0.82 | 171 |
| accuracy | | | 0.82 | 320 |
| macro avg | 0.81 | 0.82 | 0.82 | 320 |
| weighted avg | 0.82 | 0.82 | 0.82 | 320 |

**Interpretation**:

- **Accuracy**: 82%, indicating that the model correctly classifies roughly 4 out of 5 wines.
- **Precision for Good Quality**: 0.84, meaning of all the wines predicted as good, 84% are actually good.
- **Recall for Bad Quality**: 0.82, meaning we correctly identify 82% of the bad wines.

## 6.2 Confusion Matrix

- **True Negatives (122)**: Wines that were truly bad and predicted bad.
- **False Positives (27)**: Wines that were bad but predicted good.
- **False Negatives (32)**: Wines that were good but predicted bad.



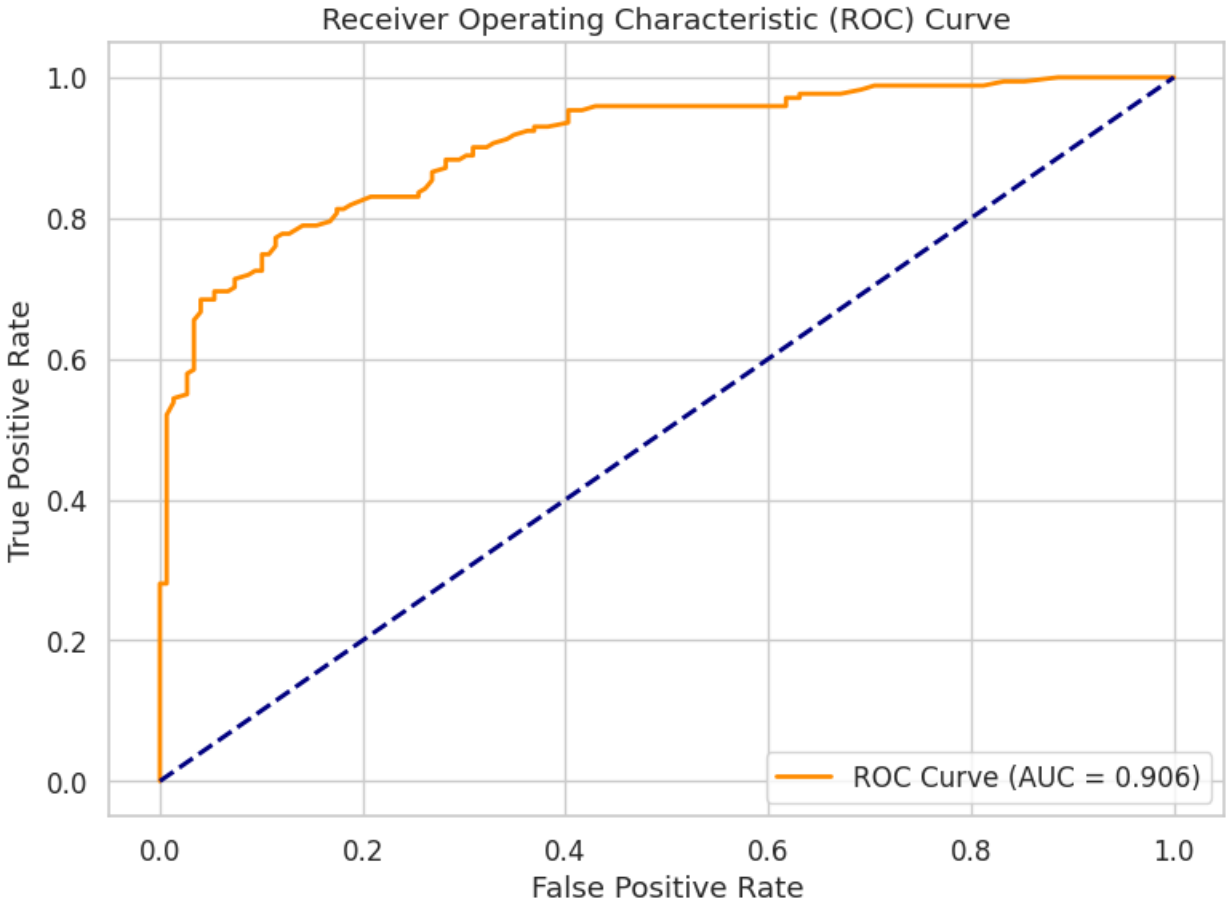

- **True Positives (139)**: Wines that were truly good and predicted good.

**Teaching Point**:

- Depending on how much higher the cost is of mislabeling a good wine to be bad (false negative), we might want to adjust our probability threshold to reduce false negatives. This would, however, result in more false positives.

## 6.3 ROC Curve and AUC

Once the class label is predicted, it is projected to the range of [0,1] by logistic function textitlogistic, which is described as below: Generally speaking, a value over 0.90 is excellent which means that Random Forest is indeed able to rank the wines in order of their chance to be



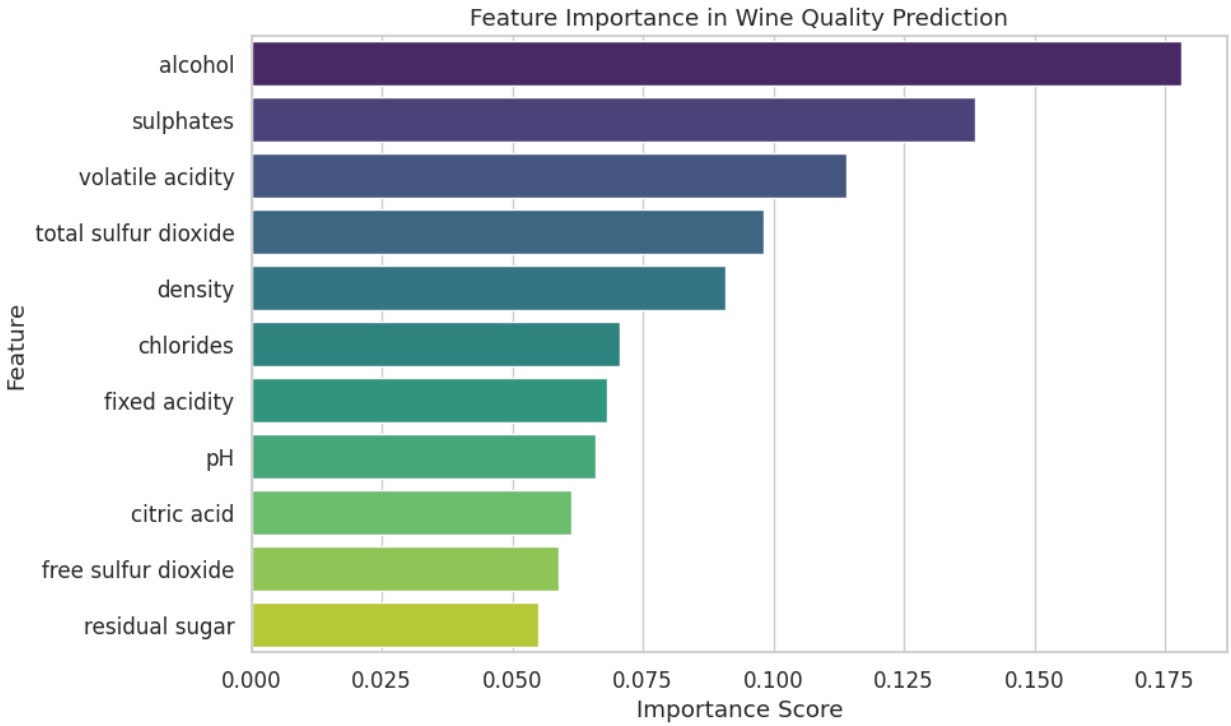

good to bad. (Pedregosa, 2011)

## 6.4 Feature Importance

A bar plot reveals the most important features:

1. **alcohol**
2. **sulphates**
3. **volatile acidity**
4. **total sulfur dioxide**
5. **density**

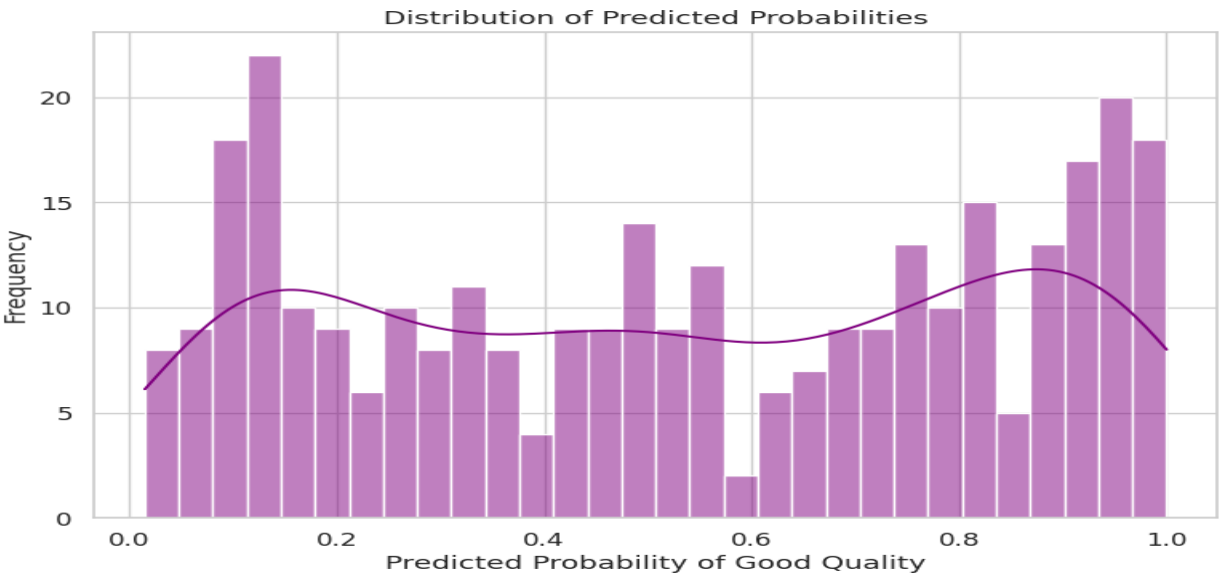
This also fits with domain knowledge about how alcohol content and sulphates affect the perceived quality of wine. One of the factors that correlate with more body and flavor intensity is




a high alcohol content while balanced acidity and controlled sulphate can improve taste.

## 6.5 Distribution of Predicted Probabilities

- The predicted probabilities for "good quality" are a histogram showing how the model's level of confidence in each wine. Observations: (Pedregosa, 2011)
- A lot of the predictions tend to cluster near 0.0 or 1.0, indicating high confidence in some wines.




- The model is less certain in the borderline cases of the 0.4–0.6 wines.

**Practical Relevance**: Winemakers or sellers might possibly adjust production processes or marketing strategies based on wine samples of wines the model is borderline about.

# 7. In-depth Analysis and Teaching Emphasis

## 7.1 Why This Approach Works

1. **Random Forest**: Bagging + feature randomness provides robust performance with tabular data.
2. **Hyperparameter Tuning**: GridSearchCV systematically explores combinations, leading to an optimal model configuration.
3. **Balanced Classes**: With 855 good wines vs. 744 bad wines, the dataset is fairly balanced, simplifying model training. (Pedregosa, 2011)

## 7.2 Potential Improvements

1. **Further Feature Engineering**: For instance, a ratio is created out of free and total sulfur dioxide, or a 'total acidity' is engineered out of fixed and volatile acidity.
2. **Threshold Tuning**: If we want to minimize false negatives, we might lower the classification threshold from 0.5 to ~0.4 or 0.45.
3. **Cross-validation**: We used 5-fold CV for the grid search, but if more thorough evaluation is needed or if more detailed approaches such as stratified repeated CV are preferred, then using aggregation methods can be more reliable for performance estimates.

4. **Explainable AI**: Instance level explanations provide how each feature contributes to a particular wine's classification; this could be done via tools like SHAP or LIME. (Pedregosa, 2011)

### 7.3 Accessibility

- **Colorblind-Friendly Plots**: We used viridis or coolwarm color maps to ensure the visual distinction is accessible.
- **Descriptive Captions and Titles**: Each plot has clear labeling to assist screen readers and color-impaired users.
- **Alt-Text**: In a web-based tutorial, alt-text for each figure would further enhance accessibility.

# 8. Conclusion

### 8.1 Key Findings

- The last Random Forest model also had an 82% accuracy rate on the testing set with an AUC of 0.906, which also speaks to great predictive power for separating good from bad wines.
- However, alcohol was the most important variable and coincided with the established importance of alcohol as a flavor component and body feature. In addition to the total hydrolysis of acids, sulphates, volatile acidity and total sulfur dioxide were all important in determining quality.
- As shown by analyzing the confusion matrix, classification metrics, and having equal recall for both good and bad classes, there is a balanced performance between them.

### 8.2 Lessons Learned

1. Debugging through data becomes vital in identifying the key patterns and they ensure data quality.
2. Random Forest makes for a great baseline or advanced method on tabular data, and performs well, and is interpretable.
3. The best CV accuracy of ~0.8108 shows the necessity of Hyperparameter Tuning for improving the results.
4. Visualization allows one to visualize the model's decisions and confidence levels at a more concrete level that the purely numerical metrics are not able to do.

### 8.3 Future Directions

- **Ensemble Stacking**: Combine multiple classifiers (e.g., Random Forest + Gradient Boosting + XGBoost) to see if performance improves.
- **Regression vs. Classification**: Compare this classification approach to the alternative of direct regression on the original quality score to judge which gives better real-world interpretability. (Pedregosa, 2011)

- **Explainable AI**: Using advanced interpretability libraries to examine individual predictions in more detail.
- **Domain Integration**: Engage domains experts (enologists or wine experts) to include domain specific knowledge on feature engineering or the thresholds adjustment.

# 9. References


1. **Breiman, L. (2001).** Random Forests. *Machine Learning*, 45(1), 5–32.
2. **UCI Wine Quality Dataset**: Red Wine Quality Data
3. **Pedregosa, F., et al. (2011).** Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.