Predicting Red Wine Quality Using Machine Learning

Student: 23048257

1. Introduction

Traditional wine quality assessment involves expert tasters using their senses to judge the wine's aroma, flavour, and colour. This method is subjective and can take a lot of time since it often requires several tasters to agree on the quality (Jackson, 2009). Additionally, chemical analysis is performed to measure specific compounds like acidity, sugar levels, and sulfur dioxide content, which also contributes to the overall quality score. Wine quality prediction is a task that can benefit from the use of machine learning. By training a model on a dataset of wine characteristics and quality scores, a machine learning algorithm can be used to predict the quality of new wines by their characteristics. This might be useful for wine producers and sellers, as it can help them to identify the better wines so they can do product segmentation and pricing accordingly. Also, achieving a good working model will reduce the costs of traditional quality test methods.

The dataset was taken from Kaggle, Red Wine Quality (Cortez et al., 2009).

The code and visualisations for this project can be found within the Jupyter Notebook on GitHub:

https://github.com/alpertoo/Machine-Learning-Projects/tree/main

2. Data Exploration

The dataset contains 1599 entries and 12 features. The features include various chemical properties of the wine. The data type of all features is numeric. The input variables are float, and the output variable "quality" is integer.

Fixed acidity: The fixed nonvolatile acids.

Volatile acidity: The acetic acid amount in wine. Which can lead to an unpleasant vinegar taste if it reaches prominent levels.

Citric acid: Increases the freshness and flavour of wine, and acts as a preservative to increase acidity.

Residual sugar: The amount of sugar left after fermentation. The aim is to reach a perfect balance of sweetness and sourness. Wine > 45g/litre is sweet.

Chlorides: The salt amount in the wine. For better quality wines chloride levels should be lower

Free sulfur dioxide: The free form of SO2 exists as a dissolved gas. It is used for oxidation and microbial spoilage prevention.

Total sulfur dioxide: The free and bound form amounts of S02.

Density: The density of wine is like that of water, but it varies based on the alcohol and sugar content. Higher-quality wines tend to have lower densities, while sweeter wines typically have higher densities.

pH: The level of acidity on a scale of 0 (acidic) to 14 (basic). Most wines are between pH 3 and 4.

Sulphates: An antibacterial and antioxidant agent added to wine which can contribute to sulphur dioxide gas (S02) levels.

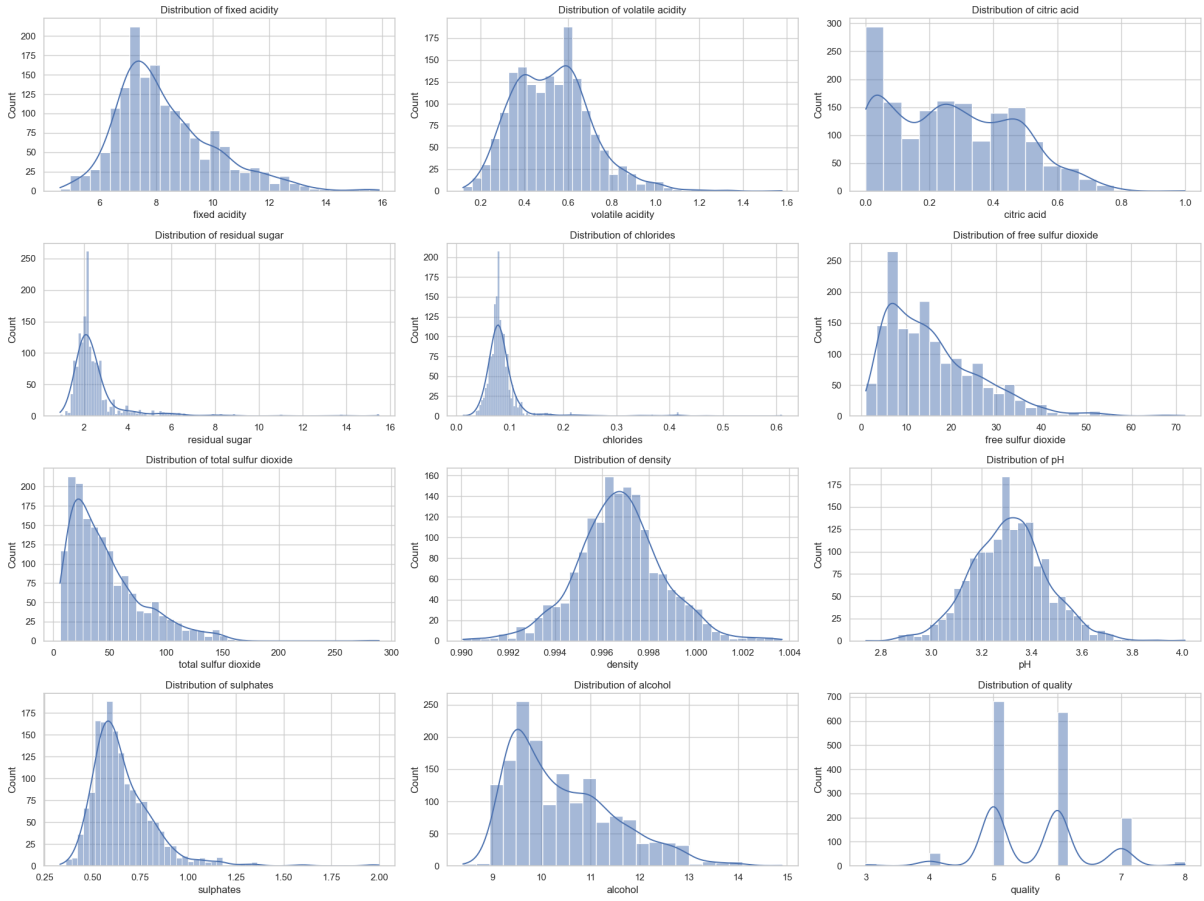
Alcohol: The alcohol amount in wine by percentage. A higher amount makes the quality better.

Quality: The output variable scores between 0 and 10. (Kniazieva, 2023).

Missing values

The dataset doesn't contain any missing values, which simplifies data preprocessing.

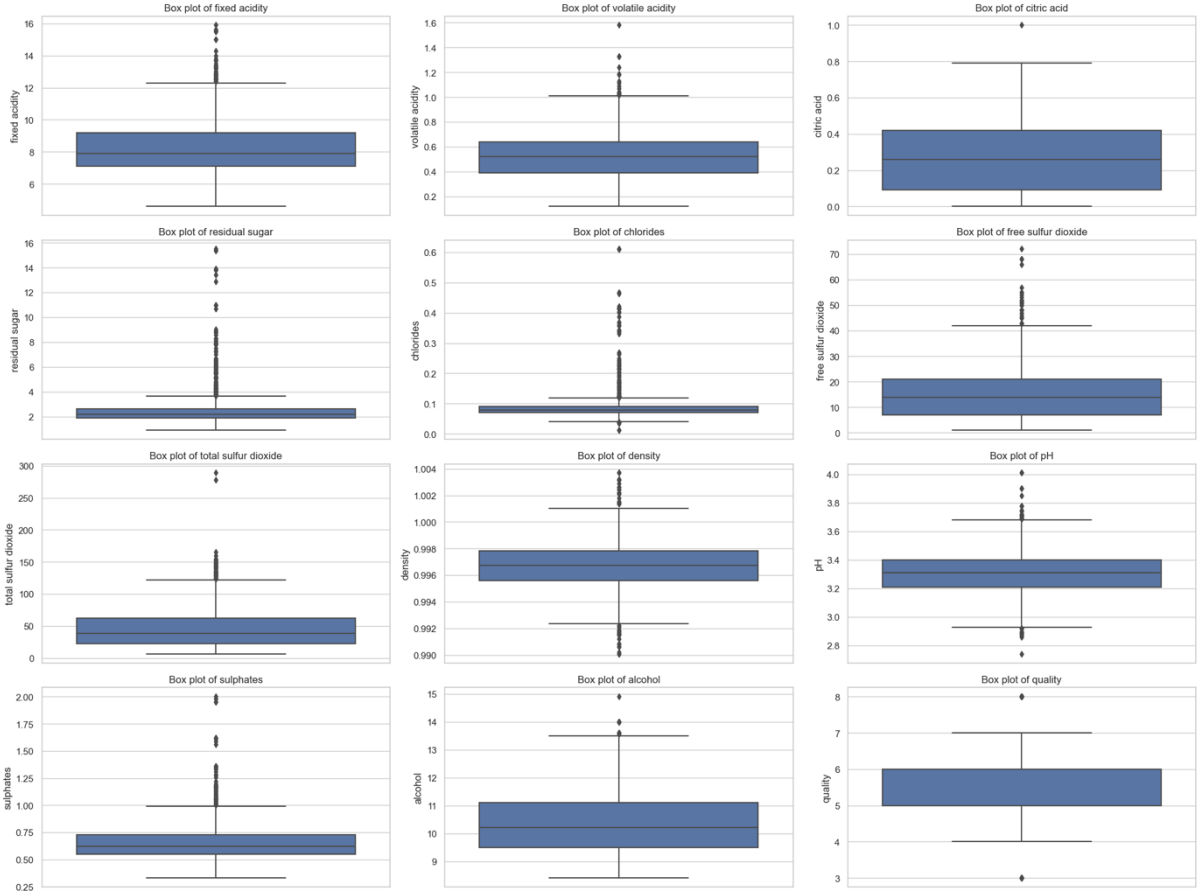
Distribution of the features



Most features exhibit a normal distribution with some skewness. Chlorides, Residual Sugar, and Sulfur Dioxide Levels (free and total) exhibit long tails, which might indicate the presence of some higher values or diverse wine types. Fixed and Volatile Acidity Levels, Citric Acid, and Sulphates show a skewed distribution. Transformations can be considered for better model performance.

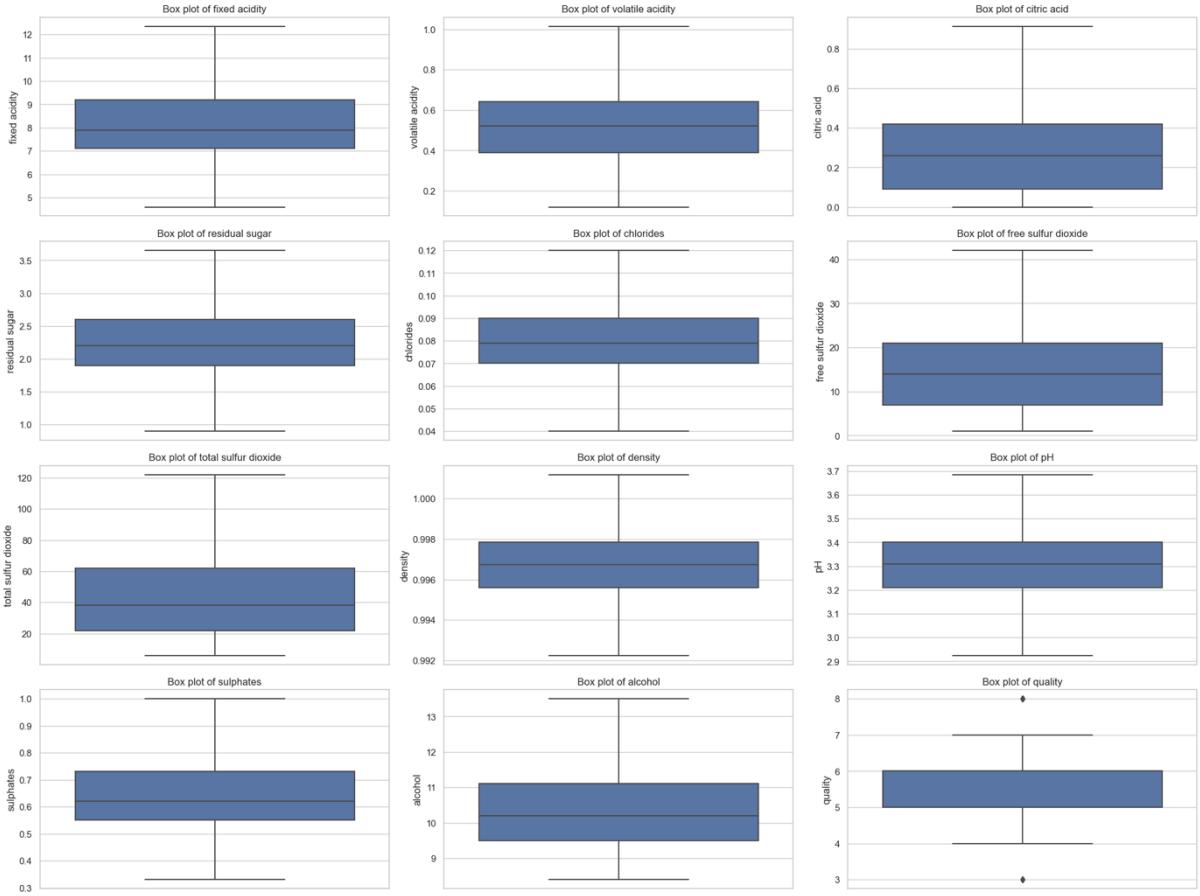
Density and pH distributions appear to be relatively symmetrical, indicating that these variables might not need transformation. Alcohol content varies widely, suggesting it could be an influential factor in predicting wine quality.

Identifying and handling outliers

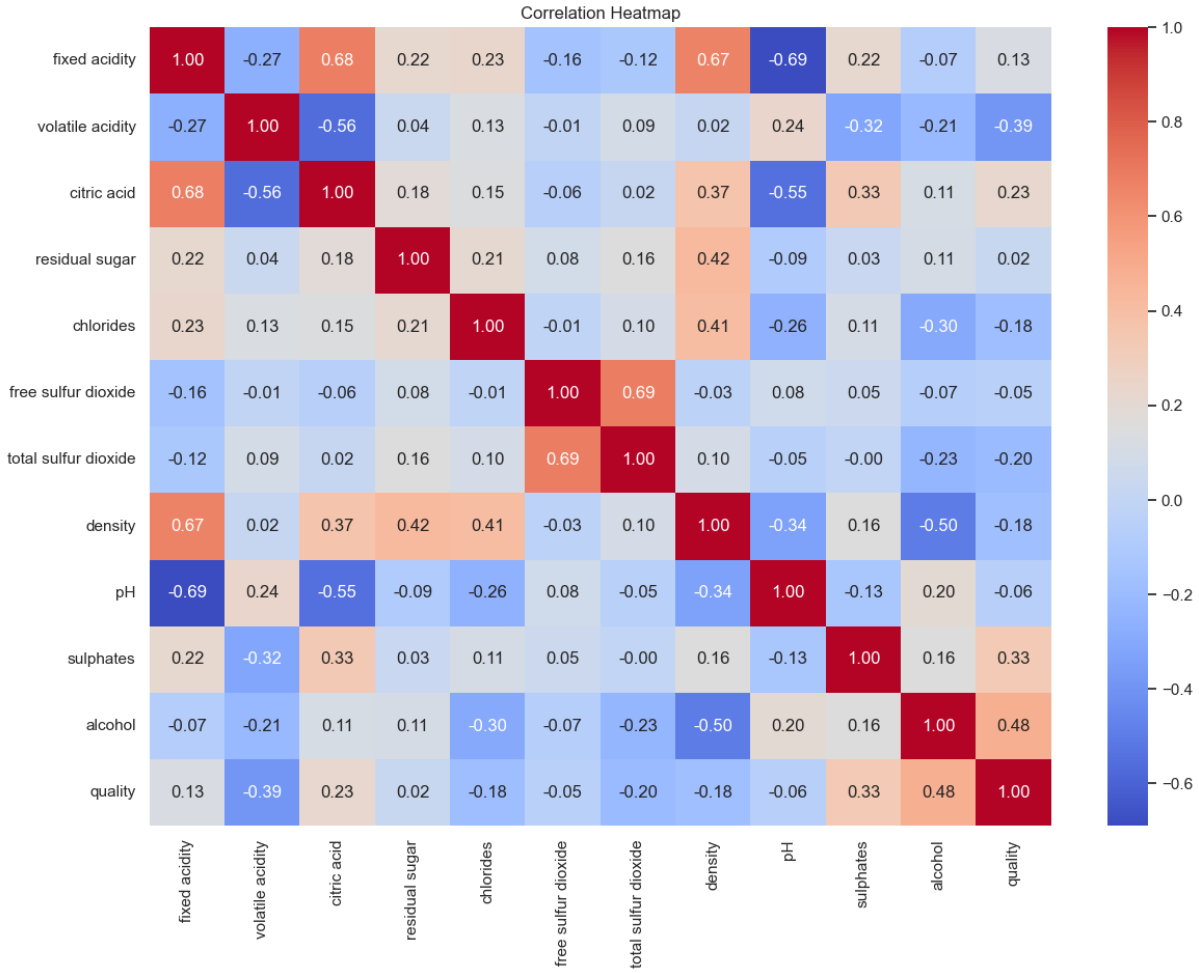


The box plots reveal that all features have outliers. Being a categorical variable, the quality does not show outliers in the same sense. Handling outliers is crucial in a machine learning project because it ensures the accuracy and reliability of the model, preventing skewed results and improving overall performance. The outliers are detected using the IQR method and then capped, reducing the extreme values while retaining most of the data. The Interquartile Range (IQR) method and capping method are common techniques for handling outliers in data analysis. The IQR method calculates the range between the first quartile (Q1) and the third quartile (Q3) and identifies outliers as values that are below (Q1 - 1.5IQR) or above (Q3 + 1.5IQR) (Fan et al., 2021). On the other hand, the capping method, also known as Winsorizing, involves replacing outliers with the nearest value within the acceptable range, effectively "capping" the extreme values (Buhl, 2023). This approach reduces the influence of extreme values without entirely removing them and helps to retain more data. However, it also limits the maximum and minimum values, which could affect the distribution of the data.

The box plots below show the outliers have been capped.

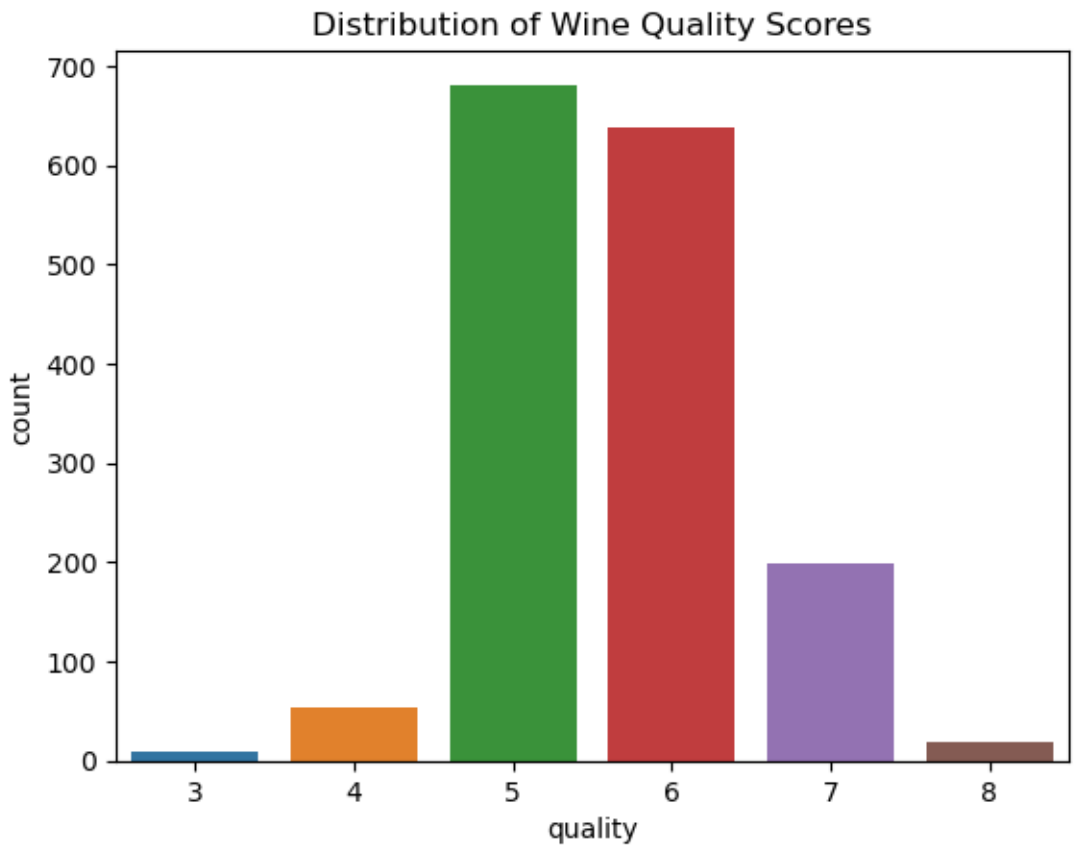

Box plot of fixed acidity
fixed acidity
Box plot of volatile acidity
volatile acidity
Box plot of citric acid
citric acid
Box plot of residual sugar
residual sugar
Box plot of chlorides
chlorides
Box plot of free sulfur dioxide
free sulfur dioxide
Box plot of total sulfur dioxide
total sulfur dioxide
Box plot of density
density
Box plot of pH
pH
Box plot of sulphates
sulphates
Box plot of alcohol
alcohol
Box plot of quality
quality

Correlation Heatmap



The correlation heatmap reveals that Alcohol has a positive correlation and Volatile acidity has a negative correlation with quality. While higher alcohol content may be associated with higher quality ratings, lower volatile acidity levels might be preferable for higher-quality wines. Moderate correlations exist between Acidity Levels and pH, which is expected due to their chemical relationship. Density is strongly correlated with Residual Sugar, possibly because sugars increase the density of the liquid.

Common thresholds for identifying multicollinearity are when the correlation coefficient is over 0.8 (or under -0.8). Features with high correlation might be redundant and could be candidates for removal or further analysis. None of the correlation coefficients of the features exceed the 0.8 threshold, suggesting that there is no severe multicollinearity. There is no need for feature selection or dimensionality reduction techniques like PCA (Principal Component Analysis).



Wine quality scores are mostly concentrated between 5 and 6, with fewer wines receiving lower or higher ratings. This indicates an imbalanced dataset that might benefit from resampling techniques.

3. Data Preprocessing

Feature Scaling

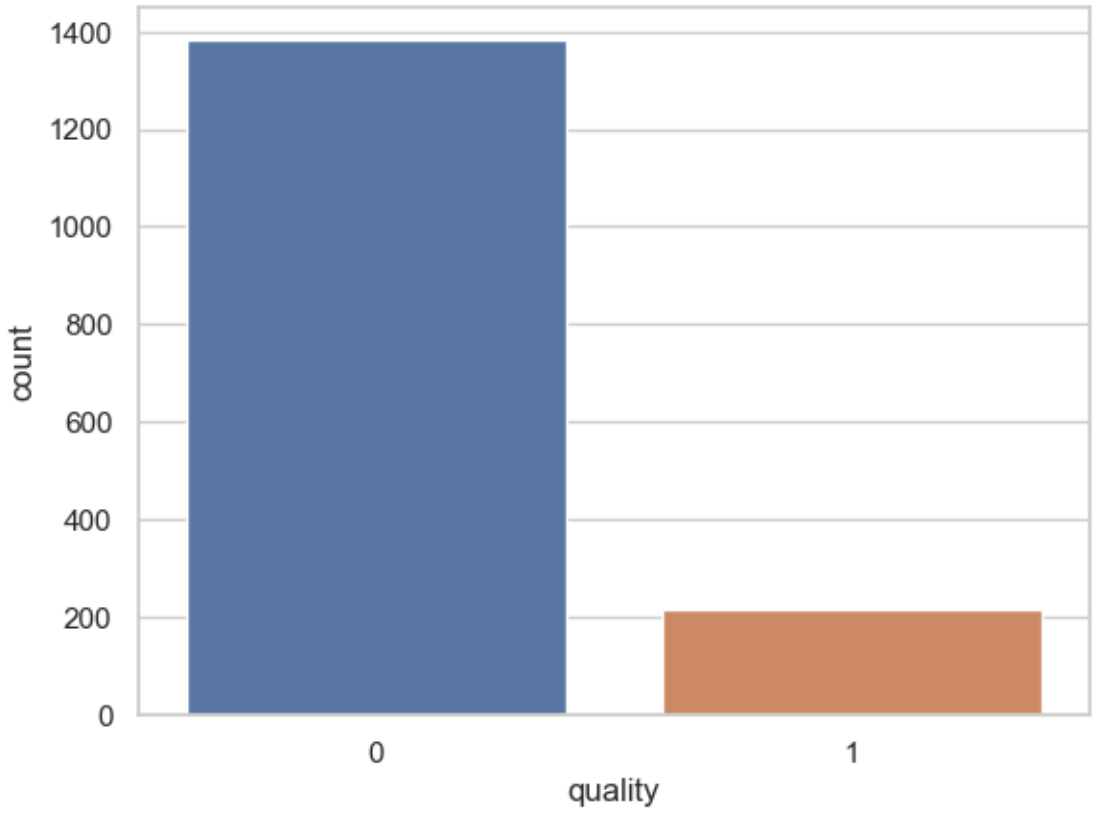
Feature scaling allows to put features into the same scale and is essential for many machine learning algorithms to perform optimally. If not applied, features with a higher value range start to dominate when performing distance calculations. Algorithms like Logistic Regression, SVM, and KNN rely on distance calculations, so having features on the same scale can impact their performance.

StandardScaler from sklearn is used for scaling the features, scaling features to have mean 0 and standard deviation 1.

Binarisation of target variable

Binarisation is applied to categorise the quality scores. The wines that have a quality score of 7 and above are considered "good quality wines" and values are labelled as 1, and the wines that have a quality score under 7 are considered "worse quality wines" and labelled as 0.

Binarisation results in a distribution of 217 "good quality wine (1)" instances and 1382 "worse quality wine (0)" instances. This indicates an imbalance in the dataset, with significantly more instances of worse quality wine compared to good quality wine.



Handling class imbalance

To handle imbalance, the minority class can be oversampled using replacement or some rows can be deleted randomly from the majority class to match with the minority class (undersampling). The loss of valuable data is the disadvantage of undersampling.

One of the most widely used methods for oversampling is the Synthetic Minority Oversampling Technique (SMOTE). It aims to balance class distribution by randomly increasing the number of minority class examples through replacement. SMOTE generates virtual training records for the minority class using linear interpolation. The benefit of this approach is that it creates synthetic data points that are slightly different from the original ones, rather than simply duplicating them (Chawla, N. V. et al., 2002).

4. Model Selection

The problem of predicting wine quality can be handled with two different aspects. Since the target variable "quality" is a score, this could be a regression problem or a classification task if the scores are treated as categorical. In this project, quality scores are binarised with labels 0 (below 7) and 1 (7 and above), making "quality" a categorical variable and several supervised learning classification models are trained and evaluated.

Logistic Regression is not selected for predicting wine quality because of its limitations in handling complex, non-linear relationships within the dataset. The assumption of a linear relationship between the features and the target might not fit this data well. Models like Random Forest and XGBoost can

handle complex, non-linear relationships better and usually perform better on diverse datasets (Breiman, 2001; Chen & Guestrin, 2016). While Logistic Regression is easy to understand, it is not as flexible as these advanced models, making it less suitable for this project (Hosmer et al., 2013).

K-Nearest Neighbors (KNN) can be computationally intense during prediction due to the need to calculate distances to all training points. Its memory intensive, vulnerability to the curse of dimensionality and sensitivity to scale and noise would have led to a decision to reject the method. However, considering the dataset's properties (2764 entries after oversampling, 11 features, scaled features, handled outliers and class imbalance and no missing values), KNN remains a simple model option, making no assumptions about data distribution. It is not rejected but, decided to focus on more efficient and robust models.

5. Model Evaluation

80% of the dataset is split as train data and 20% is split as test data. Decision Tree, Random Forest, SVM, KNN, Gradient Boosting, and XGBoost models are trained and evaluated. Half of the models show signs of overfitting and require attention.

	Model	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
0	Decision Tree	1.000000	0.902351	0.908088	0.894928	0.901460
1	Random Forest	1.000000	0.962025	0.944251	0.981884	0.962700
2	SVM	0.899141	0.911392	0.901060	0.923913	0.912343
3	KNN	0.920398	0.896926	0.834862	0.989130	0.905473
4	Gradient Boosting	0.943917	0.929476	0.907216	0.956522	0.931217
5	XGBoost	1.000000	0.956600	0.943662	0.971014	0.957143

The Decision Tree shows perfect train accuracy but a significant drop in test accuracy, indicating overfitting. In contrast, Random Forest and XGBoost exhibit both perfect train accuracy and high test accuracy, suggesting strong generalisation and robust performance. SVM and Gradient Boosting display balanced train and test accuracies, implying effective generalisation without overfitting. KNN, while having good metrics, shows slightly lower test accuracy and higher recall relative to precision, indicating more false positives.

For selecting the best parameters for the models, hyperparameter tuning is applied using GridSearchCV. The best parameters are then used to train the final model, which is evaluated on the test set.

The technique used to better estimate model performance and avoid overfitting is k-fold cross-validation. K-fold cross-validation is a technique used to evaluate the performance of a machine-learning model. It involves dividing the dataset into K equal parts, or "folds." The model is trained on K-1 folds and tested on the remaining folds. This process is repeated K times, with a different fold used as the test set each time (James et al., 2021). This method helps prevent overfitting and gives a more accurate estimate of the model's performance by averaging the results from all folds. K-fold cross-validation is popular because it uses the data efficiently and provides a reliable assessment of model accuracy.

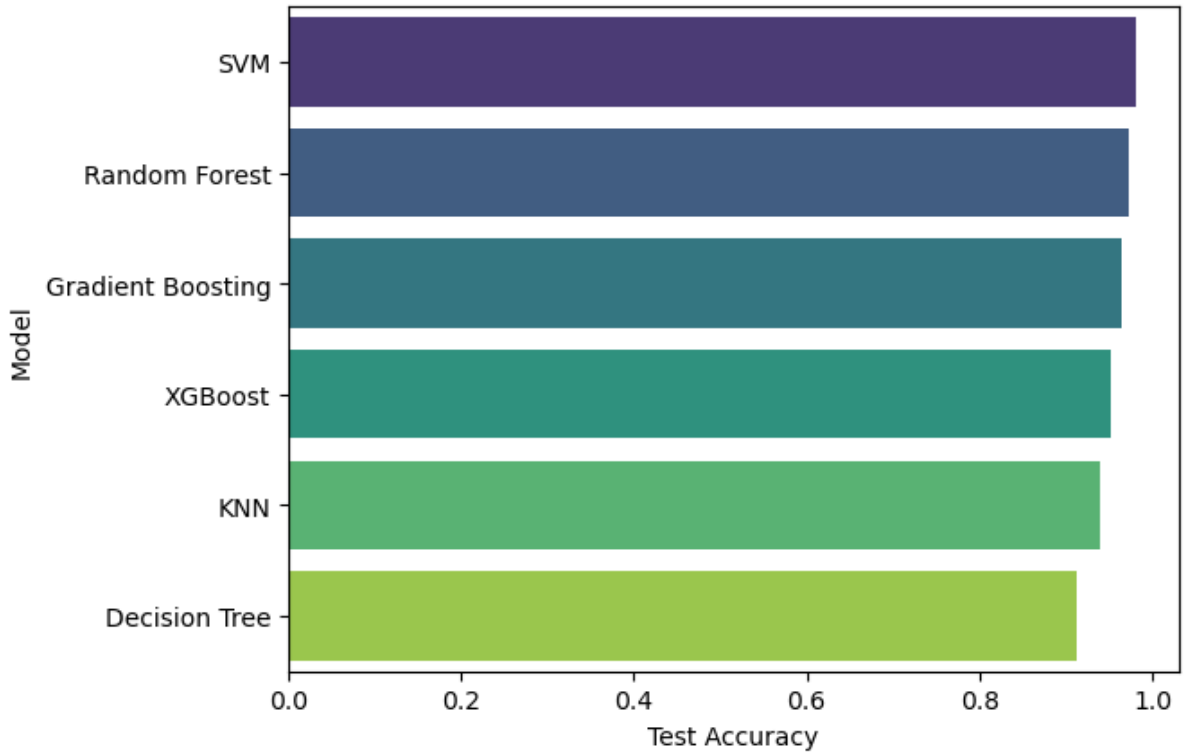
In this project, 5-fold cross-validation (k=5) is employed. The dataset is divided into five equal parts, and the model undergoes training and evaluation five times. Each iteration uses a different part as the

validation set while the remaining parts serve as the training set. The reported cross-validation accuracy for each model is the average accuracy across all five folds.

The accuracy scores after hyper tuning and cross-validation:

	Model	Cross-Validation Accuracy	Test Accuracy	Precision	Recall	F1 Score
0	Decision Tree	0.881953	0.913201	0.890411	0.942029	0.915493
1	Random Forest	0.949342	0.972875	0.961131	0.985507	0.973166
2	SVM	0.976478	0.981917	0.985401	0.978261	0.981818
3	KNN	0.914058	0.940325	0.893204	1.000000	0.943590
4	Gradient Boosting	0.953413	0.963834	0.947552	0.981884	0.964413
5	XGBoost	0.946175	0.951175	0.933798	0.971014	0.952043

SVM is the best performing model with the highest test accuracy and balanced scores of precisions, recall, and F1. Indicating that the model is well-generalized and performs very well on the test set. Random Forest also performs exceptionally well with high test accuracy and balanced precision, recall, and F1 score (Shung, 2018). However, it falls behind SVM with higher costs. It takes 1 minute 57 seconds to hyper-tune and train while SVM only needs 7 seconds. Gradient Boosting has robust performance metrics but has the highest time and computational costs. It took 5 minutes and 7 seconds to hyper-tune and train. KNN has perfect recall, indicating that it classifies all positive instances correctly, but its precision is lower compared to other models. XGBoost also performs well, though its metrics are slightly lower compared to Random Forest and SVM. Decision Tree has the lowest performance metrics among the models listed, indicating that it may not generalise as well as the others. Decision Tree and KNN show some potential issues with precision and recall balance, which might need further tuning.



Conclusion

This project successfully developed a machine learning model capable of accurately classifying the red wine quality. The SVM model demonstrated the highest performance, making it an ideal choice for deployment. This model can provide significant value to the wine industry by enabling stakeholders to predict wine quality reliably and efficiently, thereby enhancing decision-making processes and potentially increasing profitability.

By following a structured approach to data preprocessing, feature selection, model training, and evaluation, the development of a robust and reliable model is ensured. The methodologies and techniques applied in this project can be extended to similar classification problems in other domains, demonstrating the versatility and power of machine learning.

For further improvements, more samples can be collected for better training, research can be specialised in adding wine types and widened by adding white wine properties as well. Also, developing a user-friendly application can be useful to easily assess the quality of the wines at production facilities.

Word count: 2108

Bibliography

Jackson, R. S. (2009). *Wine Tasting: A Professional Handbook*. San Diego: Academic Press.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T. and Reis, J. (2009) Modeling Wine Preferences by Data Mining From Physicochemical Properties. *Decision Support Systems* [online]. 47 (4), pp. 547-553. [Accessed 12 May 2024].

Kniazieva, Y. (2023) *How to Build a Wine Quality Prediction Model Using Machine Learning?*. Label Your Data [blog]. 12 October. Available from: https://labelyourdata.com/articles/machine-learning-for-wine-quality-prediction [Accessed 10 May 2024].

Fan, C., Chen, M., Wang, X. and Huang, B. (2021) A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data. *Frontiers in Energy Research* [online]. 9 [Accessed 13 May 2024].

Buhl, N. (2023) *Mastering Data Cleaning & Data Preprocessing*. Encord [blog] Available from: https://encord.com/blog/data-cleaning-data-preprocessing/ [Accessed 11 May 2024].

Breiman, L. (2001) Random Forests. *Machine Learning* [online]. 45, pp. 5-32. [Accessed 13 May 2024].

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York, USA, August 2016. Association for Computing Machinery, pp. 785-794.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* [online]. 16, 321-357 [Accessed 13 May 2024].

Hosmer, D.W., Lemeshow, S. and Sturdivant, R.X. (2013) *Applied Logistic Regression* [online]. 3rd ed. Hoboken, New Jersey: John Wiley & Sons, Inc. [Accessed 15 May 2024].

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2021) *An Introduction to Statistical Learning with Applications in R* [online]. 2nd ed. New York, NY: Springer. [Accessed 15 May 2024].

Shung, K. P., 2018. *Accuracy, Precision, Recall or F1?*. [Online]. Available at: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9 [Accessed 10 May 2024].