Data Science and Economics
Department of Economics, Management and Quantitative Methods
Department of Computer Science "Giovanni degli Antoni"
Università degli Studi di Milano

Statistical Learning (Supervised):

Binary classification of Wine Quality

Soudabeh Masoudisoltani - 961506

September 2024

Table of Contents:

Chapter 1: Abstract

In this study we have analyzed the Wine Quality dataset from Kaggle. This data set consists of different wines and their quality score from low 0 to high 10 based on the effect of the chemicals present among its ingredients.

After preprocessing the data and having transformed the research question into a binary classification problem, we have used three supervised learning algorithms, Logistic regression, Decision tree and Knn, to train our models and make predictions.

For Knn, some hyper parameter optimization has been applied for the variable k, and the one outputting a better performance has been selected.

For each of the three models, F score, precision and recall have been reported as metrics for analyzing the modes' performance. Based on the findings of this project, the model that performs best on this data is the Decision tree.

Link to public repository with R notebook and Dataset:

Link to repository

Chapter 2: Problem definition

The objective of this project is to forecast the quality of wine based on the various chemicals present among its ingredients. We consider this project a binary classification task, and our aim is to assign one of the two labels: {high, low} to each wine based on the ingredients.

The results from this study are useful in the real world, particularly, for stakeholders in the wine industry, from production to sales by improving both efficiency and quality.

To provide some examples, wine makers can assess the quality of wine in different steps of production, using predictive models. They can identify the chemicals that highly affect wine quality to effectively enhance the quality of their products.

Retailers can use the outcome of these predictive models to understand which wine has a better market reception in order to stock products that are more probable to sell faster at a premium price.

Chapter 3: Data and Preprocessing

3.1. Data

To conduct this supervised binary classification task, we have used the available Wine_Quality dataset from Kaggle.

This dataset describes the amount of different chemical ingredients present in wine and their effects on its quality.

Our dataset consists of about 1600 rows per 12 columns, of which 11 columns are wine ingredients and 1 column is the target variable (quality). The target variable is a score between 0 (Low quality wine) and 10 (High quality wine)

3.1.1. Attributes' information

The dataset is structured with these 12 numerical features:

1 - fixed acidity: Acidity level of the wine.

2 - volatile acidity: Acidity that leads to vinegar taste.

3 - citric acid: Found in small quantities, adds freshness.

4 - residual sugar: Sugar left after fermentation.

5 - chlorides: Salt content in the wine.

6 - free sulfur dioxide: Free SO_2 preventing spoilage.

7 - total sulfur dioxide: Total SO_2, free and bound.

8 - density: Wine's density, related to sugar/alcohol content.

9 - pH: Acidity or basicity of the wine.

10 - sulfates: Adds flavor and acts as a preservative.

11 - alcohol: Alcohol percentage.

12 - quality (score between 0 and 10): Output variable (based on sensory data): Target variable, wine quality score (0–10).

3.2. Pre-processing of the data

- The existence of missing values has been checked and no null value is reported.
- The only textual column, ‘description’ has been dropped.
- The categorical target variable ‘wine quality’ was originally a number between 0 and 10, putting the wines into 10 different classes based on quality of wine, and the project was a multi classification task. has been changed into numerical and binary classification by using only two labels ‘low quality’ and ‘high quality’ wine. To make this distinction we have considered wines belonging to quality score 1-6 as Low and those belonging to classes 7-10 as High quality wine.

3.2.1. Correlation among variables

The correlation among the features is shown through the graph below, stating that some features such as {ph, quality} and {ph, fixed.acidity} have strong negative correlations and other ones such as {alcohol, quality} have strong positive correlation with each other. And no correlation is significant among some other pairs of variables.

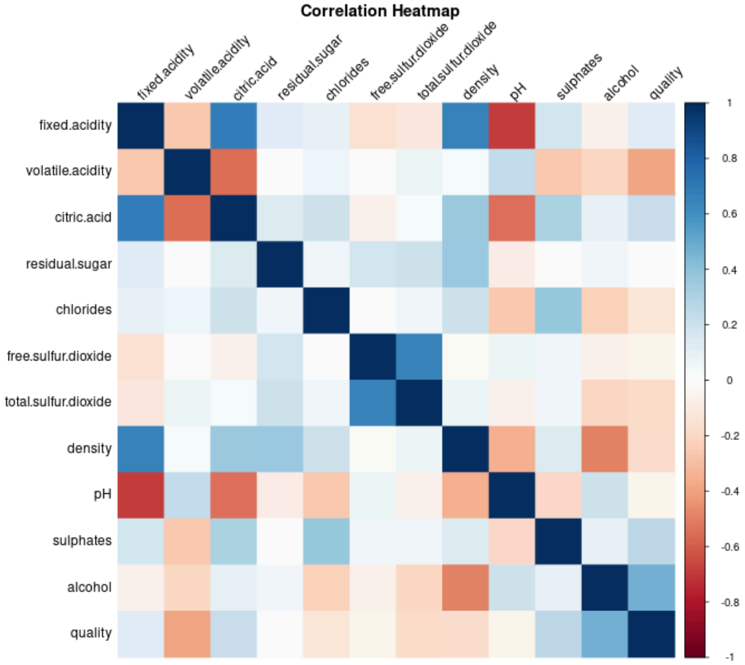


Figure 3.1. Correlation heatmap

As has been mentioned before, wine quality >= 6 is considered high quality.

- Finally we have normalized the data (excluding the target variable|) such that:

  Mean = 0

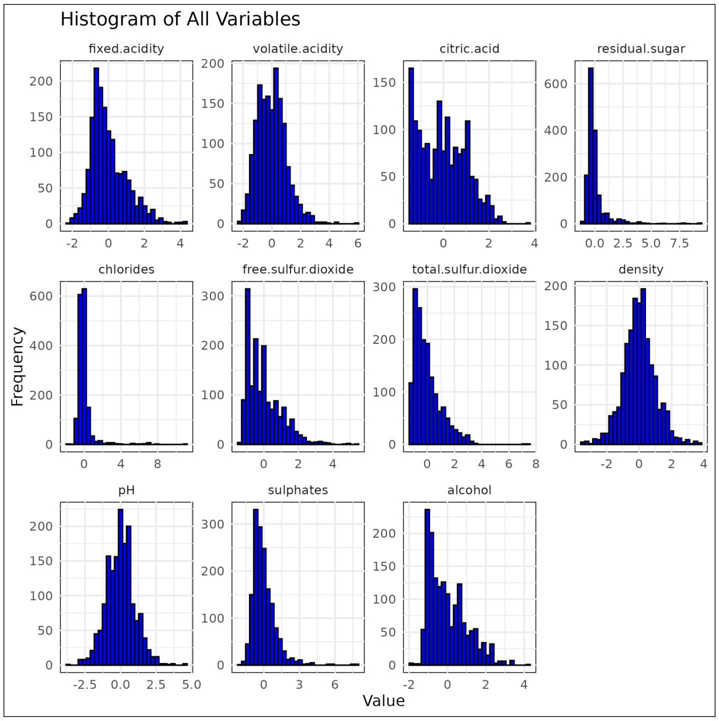
  Standard deviation = 1

Scaling ensures that all features contribute equally to the model, especially important for algorithms sensitive to feature magnitudes.

3.2.2. Distribution of features and target variable

To show the distribution of each variable, we have used histograms on normalized data, this shows that for example, more than 750 of data points are high quality wine.

Finally the distribution of the target variable, wine quality shows that the data is balanced and we have almost the same amount of each type of wine quality and no predominance of one type.



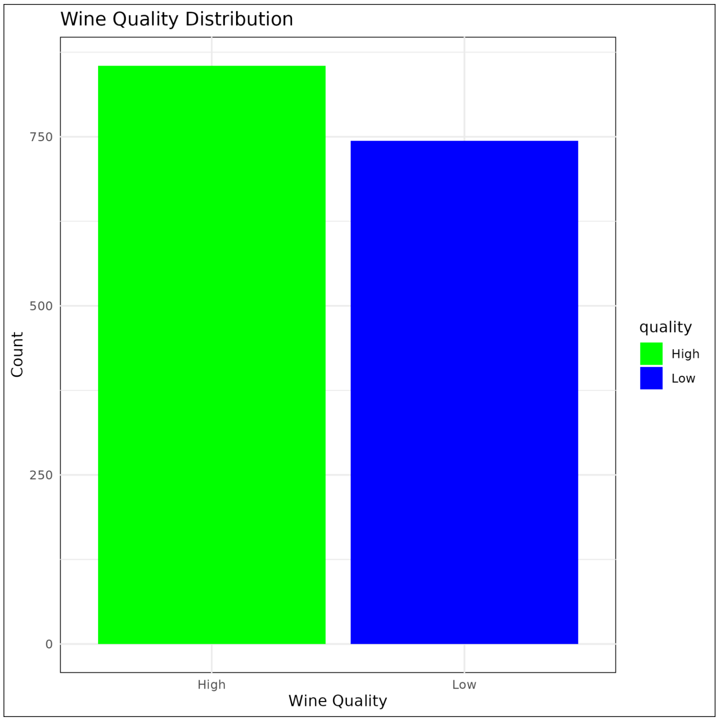


Figure 3.2. Histograms of all variables

Figure 3.3. Distribution of Wine Quality

It is interesting to explore a little bit the outliers among the data and to do so we have utilized box plots for every variable as below, showing that some features such as Chlorides, residual sugar and citric acid have many outliers, suggesting that they contain more extreme values, whereas others, alcohol for example, seem to be more homogenous.

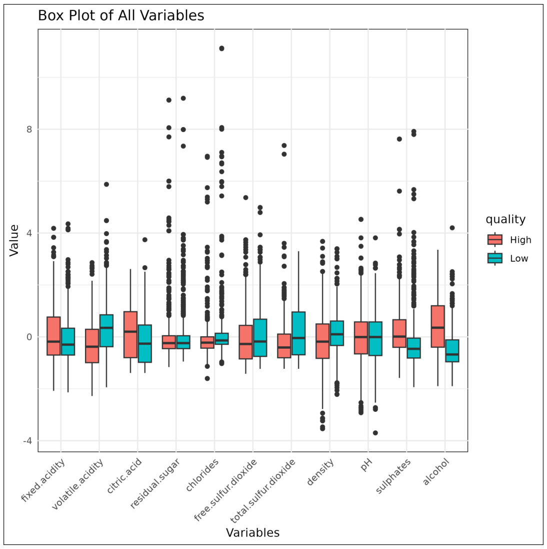


Figure 3.4. Box Plot of variable

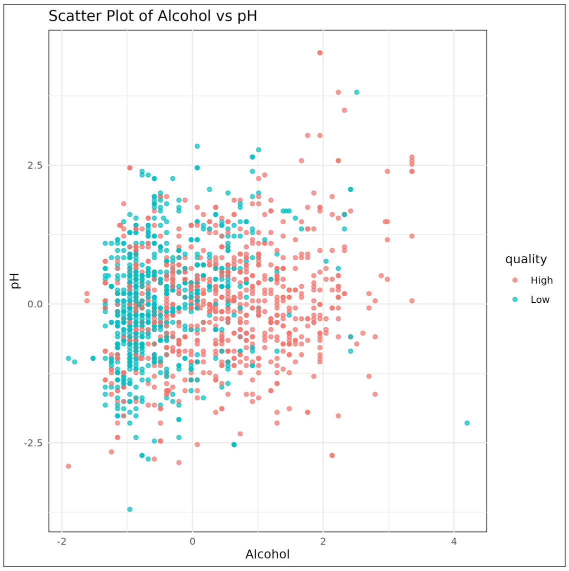


Figure 3.5. Scatterplot

The box plot in figure 3.4 compares the distribution of each feature on the x axis between "Low" and "High" quality wines. Differences in the box plots can indicate how feature values vary between quality categories.

The scatter plot in figure 3.5 tries to reveal clustering or separation between the two quality categories and find existing trends or patterns between alcohol content and ph levels.

Chapter 4: Supervised Algorithms

In this supervised task, we have utilized three common supervised learning algorithms. For the sake of simplicity in future classification of our data points, we consider 2 labels: {high, low} quality wine, thus the task is a binary classification for which we will train Logistic regression, Decision tree and Knn algorithms in order to make binary prediction on the quality of wine.

Data has been split into 70% train and 30% test data.

In this chapter we will explore three supervised learning algorithms theoretically and then we will apply them to our cleaned data. Finally the performance of each algorithm will be reported and compared.

4.1. Logistic Regression

As a statistical method for binary classification tasks, unlike linear regression that outputs a continuous outcome, Logistic Regressions's goal is to predict one of the two possible outcomes, in our case, whether the wine containing certain ingredients has low or high quality. Logistic regression predicts the probability with which an instance belongs to a specific class.
The key idea behind this kind of regression is that the model should show the relationship between the input features and the probability of belonging to a particular class using a Sigmoid function. This function takes values between 0 and 1, and using a decision threshold predicts the probability for an instance to belong to a class. The threshold is set to 0.5 here.

$$\frac{e^{(\beta_0+\beta_1 x)}}{1+e^{(\beta_0+\beta_1 x)}}$$

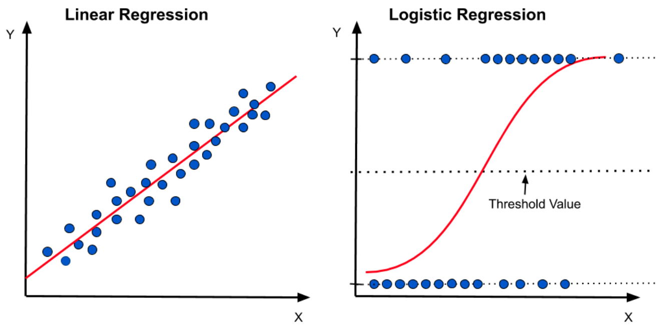


Figure 4.1. Linear vs Logistic regression

To predict wine quality as either ”High” or “Low”, Logistic regression is applied to the data.
I have trained the model using the ‘glm()’ function in R, for the binomial family as logistic regression is specific for this kind of task.
The threshold value used is 0.5 in our task.
This approach allows binary decision making on the output probabilities and to predict with what probability a specific wine is classified as one or the other target values.

4.2. Decision tree

Another widely used algorithm in supervised learning tasks is Decision tree and it can be applied to both classification and regression tasks.
The tree-like structure of this algorithm, models the decision using internal nodes representing a test on a feature and each branch represents the outcome of that test. Finally the leaf node represents the final prediction which can be a class, label or a continuous value.
The goal of the decision tree is to lead to the most significant separation among classes.
The tree is created by setting the feature and threshold that divides the data in the best way according to a criterion such as Gini impurity or information gain. This splitting continues until we achieve a perfect classification or when a stopping criterion is met.
To classify wine quality as low or high, a decision tree is implemented in this section using the ‘rpart’ package. The decision tree algorithm creates a model that splits the data according to the most informative features to gain the best separation of the data.
After training our algorithm, the algorithm uses entropy and makes the first split is based on the percentage of alcohol (0.16)
Among the instances that have alcohol < 0.16, percentage of Sulphates is selected as the next node and has split the remaining data using the threshold -0.078, we can see from figure 4.2 that A wine with ‘alcohol’ > 0.16 and ‘Sulfates’ > -0.078 and ‘total sulfur dioxide’< 1.8 is a “high quality” wine.

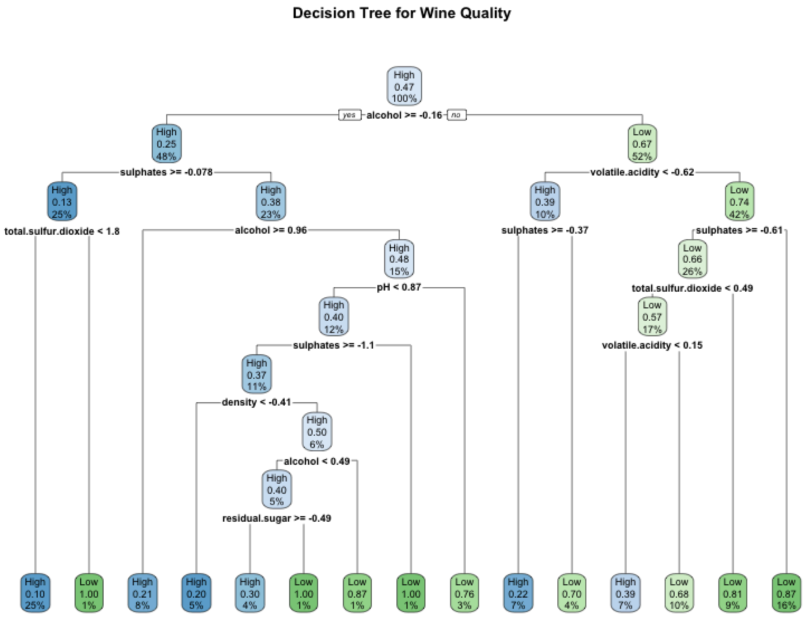


Figure 4.2. Decision tree for Wine Quality

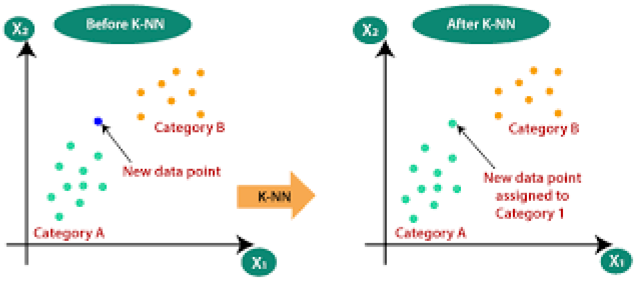
4.3. K-Nearest Neighbors

This simple, non-parametric and instance based algorithm can be used for both regression and classification tasks.

The idea behind this algorithm is the principle that similar data points are likely to be close to each other in the feature space.

By measuring the closeness of data points on the feature space, this algorithm chooses the label for the unknown data point by looking at the majority of the neighboring instances. The number of instances that vote is the k which is usually chosen following hyperparameter optimization.

Figure 4.3. Knn visualization

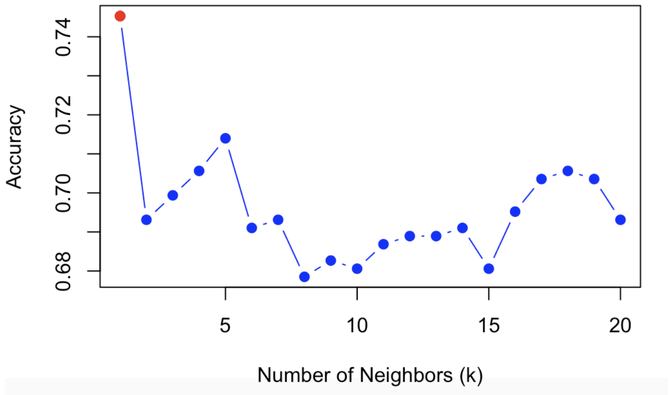


4.3.1. Hyper parameter optimization for KNN

To find the best value for K, we have checked values between 1 and 20.

By finding the accuracy value of the algorithm in each of these 20 models, the k value that allows for the highest accuracy will be chosen as the k that optimizes our Knn algorithm, finally we will build the KNN model to make the wine quality prediction according to the result of accuracy.

Figure 4.4. Knn accuracy vs K value



Based on the result of Accuracy, excluding k = 1 which is meaningless, it is noticeable that K = 5 gives the highest accuracy among the options so this is the value of k with which we will continue to work with.

Knn with k = 5 is created and the algorithm is trained on the data to be able to predict an unknown wine's quality using the votes of 5 closest instances.

Chapter 5: Model evaluation and the results

5.1. Theoretical definitions

To compare the performances of the three models shown discussed in the previous chapter, we have used three evaluation criterions:

- Precision: this criterion measures the ratio between true positives (labels predicted correctly to be positive) and total number of positive answers (correctly or incorrectly predicted as positive by the model)

$$Precision = \frac{TP}{TP + FP}$$

High precision indicates that most of the positive predictions made by our model are correct. Low precision means that the model often incorrectly predicts a positive class when it should not.

- Recall: also known as sensitivity or true positive rate, is another machine learning metric that measures the ability of the model to identify all relevant instances.
  Recall shows the ratio between true positive predictions and total number of actual positive cases.

$$Recall = \frac{TP}{TP + FN}$$

High recall indicates that most of the actual positive cases are correctly identified(few FN) and Low recall indicates that the model misses many positive cases.(maneyFN)

- F1 score: This machine learning metric is used as a balance between recall and precision particularly in cases of imbalanced datasets..

$$\text{F1 Score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

High F1 score indicates that both precision and recall are high, this means that the model's performance is good in identifying positive instances with minimal false positive and negatives.

Low F1 score indicates that either recall or precision or both are low, this model does not perform well with either correctly identifying positive cases or avoiding false positives.

5.2. Model evaluation and results

5.2.1. Logistic Regression Evaluation

metric	Precision	Recall	F1 score
performance	0.3125	0.2734	0.2917

This model's performance is shown in the below table:
Precision results in this model indicate that out of all the instances predicted as positive by this model, only 31.25% are actually correct, meaning that this model has a relatively high number of false positives.
Recall here indicates that out of all the actual positive instances, the model correctly identifies 27.34% meaning the model misses many true positive instances.
F1 score results combine precision and recall. So a score of 0.2917 indicates that the model performs poorly in terms of balancing precision and recall.

5.2.2. Decision Tree Evaluation

This model's performance is shown in the below table:

metric	Precision	Recall	F1 score
performance	0.734	0.742	0.738

Precision score of the decision tree implies that 73.4% of the instances are correctly predicted as positive. This is a high precision value that indicates the model has relatively few false positives.
Recall of 74.2% means that the model correctly finds a good percentage of actual positive instances. High recall indicates that the model is good at getting most true positives.

The F1 score is also strong, showing good overall performance.
These metrics show that the decision tree model balances both precision and recall well, that results in good overall performance.
None of the metrics outperform the other, by F1 score being near to both recall and precision. This is usually a good sign of a balanced model.

5.2.3. KNN Evaluation

This model's performance is shown in the below table:

metric	Precision	Recall	F1 score
performance	0.696	0.734	0.715

This model shows fairly good performance based on the provided metrics.
The precision metric here indicates that about 70% of positive predicted instances are correct. Implying that the model has a moderate level of false positives.
Recall of about 74% shows that the model is reasonably effective at capturing true positives,
Finally, an F1 score of 0.715 shows a good trade off between precision and recall.

5.3. Model selection and conclusion

To compare Knn, Decision Tree and Logistic regression models utilized in this study based on their performance metrics (Precision, Recall, F1 score) we can look at the findings of this project summarized in the table below:

	Logistic Regression	*Decision Tree*	*KNN*
Precision	0.3125	0.7336	0.6963
Recall	0.2734	0.7422	0.7344
F1 score	0.2917	0.7379	0.7148

It is easy to notice that among the models built and trained and used for prediction in this study, the Decision Tree stands out with its best performance. It has the highest precision, recall and f1 score.

This model provides us with the best trade-off between correctly finding positive instances (high recall) and minimizing false positives (high precision).

Therefore, we recommend Decision Tree as the choice of model for this classification task.

We tried to highlight the weaknesses and strengths of each of the three models in different chapters and justify the selection of the Decision Tree as the best option based on empirical evidence.

5.2. Conclusion and Key findings

1. Decision tree performs best as a predictive model for this dataset for example, easily visible which features (like alcohol, pH) are most influential in predicting wine quality and how they split the data into different classes.
2. After tuning, if k = 5 results in the highest accuracy compared to other values of k, it means that having 5 neighbors provides the best balance between model complexity and performance. Choosing the optimal k helps in achieving the best predictive performance and avoids both underfitting and overfitting.
3. Some factors contribute to logistic regression being less effective in scenarios where more sophisticated models capture the underlying data patterns better such as in this case where simplicity might lead to underfitting in cases where the data has complex patterns..