

**Tribhuvan University**

**A REPORT ON**

**“RED WINE QUALITY CLASSIFICATION”**

**Submitted to:**

**Department of Computer Science and Information Technology**

**National College of Computer Studies**

*In partial fulfillment of the requirements for the Bachelor degree in Science in Computer Science*

*and Information Technology (B.Sc.CSIT) of Tribhuvan University, Nepal.*

**Submitted By:**

**Name: Rabina Shakya**

**TU Reg. No.: 5-2-551-32-2020**

**Under the Supervision of**

Mausam Rajbanshi

September 2024

Abstract

**Wine Quality Classification** is a challenging task, as taste is one of the most subjective human senses. Accurate wine quality predictions are valuable for the certification process. My goal in this project is to develop an algorithm that classifies the quality of red wines as good or bad. Basically, traditional methods of assessing wine quality, such as taste testing, can be subjective and time-consuming. To address these limitations, machine learning techniques are being used to create automated models that can predict wine quality efficiently and accurately.

In this research, we applied machine learning algorithms to a publicly available dataset of red wine to predict its quality. We implemented two machine learning models: **Logistic Regression** and **Random Forest**. The Logistic Regression model achieved an accuracy of 89%, while the Random Forest classifier provided a higher accuracy of 91%. These models offer a scientific approach to wine quality prediction, improving the ability to classify wines based on their chemical properties and helping winemakers and consumers make more informed decisions.

Acknowledgement

First of all, we would like to express our gratitude to the National College of Computer Studies (NCCS) for providing a supportive learning environment and the opportunity to work on the “Red Wine Quality Classification” project using Machine Learning and Data Science in Python.

We are also deeply thankful to our supervisor **Mr. Mausam Rajbanshi**, for his invaluable support, encouragement, and expert guidance to make our project much better. His insights greatly improved our work and helped us produce a high-quality report.

Finally, we appreciate everyone who contributed to the completion of this report. The support and collaboration of many individuals made this project possible, and we are truly grateful for their contributions.

Table of Contents

[Abstract i](#_Toc177229041)

[Acknowledgement ii](#_Toc177229042)

[1. Introduction 1](#_Toc177229043)

[2. Objectives 1](#_Toc177229044)

[3. Problem With the System 1](#_Toc177229045)

[4. Dataset 2](#_Toc177229046)

[5. Implementation 3](#_Toc177229047)

[5.1 Importing the necessary modules 3](#_Toc177229048)

[5.2 Loading the data 3](#_Toc177229049)

[5.3 Exploratory Data Analysis 4](#_Toc177229050)

[5.4 Data Visualization 6](#_Toc177229051)

[5.5 Preprocessing Data 8](#_Toc177229052)

[5.6 Training the Model 10](#_Toc177229053)

[5.6.1 Logistic Regression 11](#_Toc177229054)

[5.6.2 Random Forest Classifier 13](#_Toc177229055)

[Conclusion 15](#_Toc177229056)

[References 16](#_Toc177229057)

# Introduction

Red wine is an alcoholic beverage made by fermenting the juice of dark-skinned grapes. Wine quality is determined by multiple factors, including physicochemical tests like acidity, sugar content, and alcohol levels. The first and most obvious characteristic of red wine is the color. Red wines range in hue from deep, opaque purple to pale ruby and everything in between. [1] As red wine ages, its bright, youthful colors turn garnet and even brown. Predicting wine quality accurately is a challenge in the wine industry. By implementing a machine learning model, we can improve the efficiency and accuracy of assessing wine quality. This project explores the relationship between different features of wine and their impact on quality, as well as builds models to predict wine quality.

# Objectives

The main objectives of this project are:

* To explore and analyze the dataset of red wine.
* To preprocess the data for machine learning algorithms.
* To build and evaluate machine learning models to classify wine quality.
* To compare the performance of different classification algorithms.

# Problem With the System

In this wine quality classification, the dataset was unbalanced, with more bad wine samples than good ones. This caused the model to be biased toward predicting the more common class, affecting its overall accuracy. There were some challenges with cleaning the dataset, such as removing duplicates, which needed to be addressed before building a reliable model.

# Dataset

A machine learning dataset is a collection of data that is used to train the model. A dataset acts as an example to teach the machine learning algorithm how to make predictions [2]. In this study we have utilized a comprehensive dataset, there are 1599 different wine as row data and 12 features as columns. With 11 variables and 1 output variable (quality) given, let us examine each of these features:

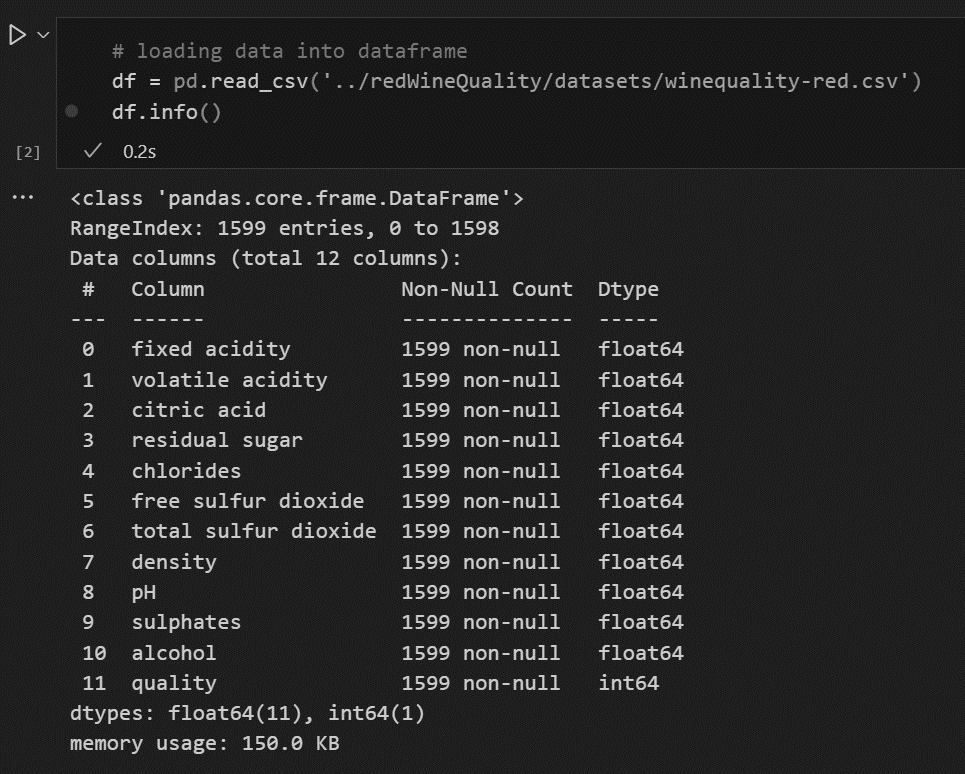
* volatile acidity
* citric acid
* residual sugar
* chlorides
* free sulfur dioxide
* total sulfur dioxide
* density
* pH
* sulphates
* alcohol
* quality (target variable)

# Implementation

## Importing the necessary modules



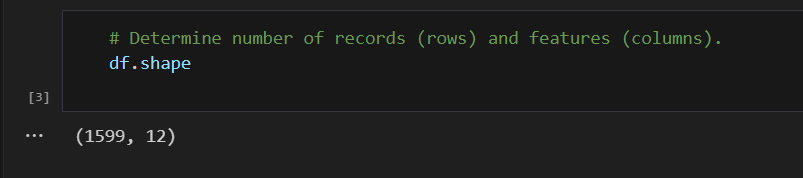
## Loading the data



**Result:** The **df.info()** provides a summary of the dataset. There are 1599 number of entries, 12 column names, non-null values, and 11 float values for input variables and 1 int value for target variable.

## Exploratory Data Analysis

* + 1. Determining rows and columns



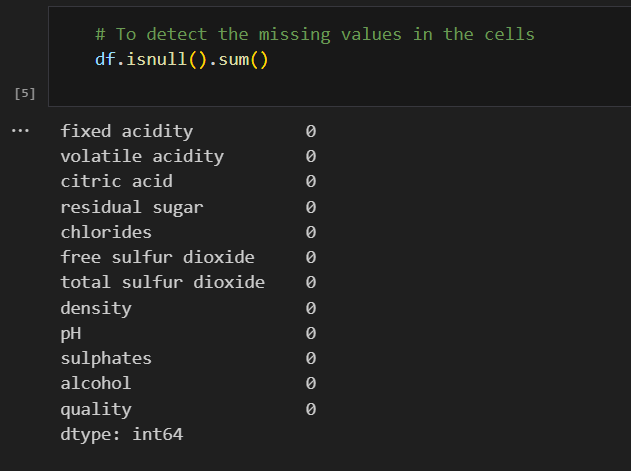
**Result:** The **df.shape** returns the shape of the dataset as **(1599, 12)**, indicating 1599 rows and 12 columns.

* + 1. Data describe



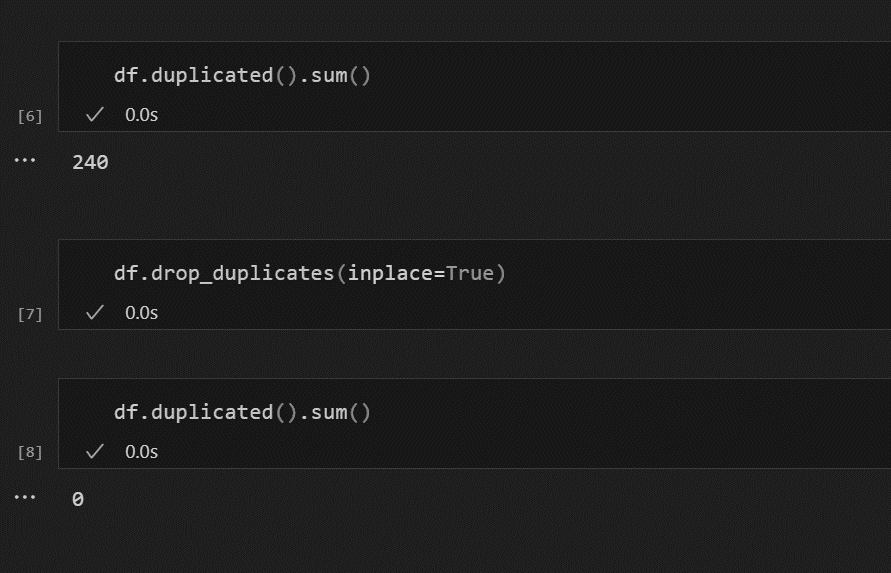
**Result:** The **df.describe()** gives a statistical summary of numerical columns, including metrics like mean, standard deviation, and quartiles.

* + 1. Checking for Null or Missing values



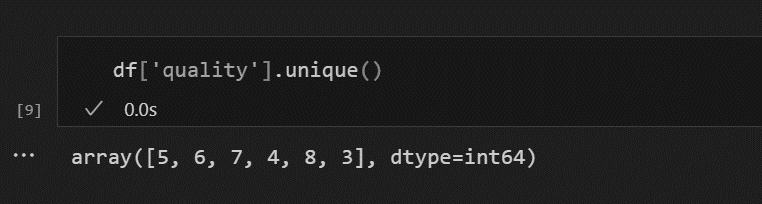
**Result:** The method **df.isnull().sum()** tells us whether there are any missing values present or not. It looks like there are no missing value. It means dataset can be processed.

* + 1. Duplicate data



**Result:** The **df.duplicated().sum** displays the number of duplicated rows in the dataset. There are 240 duplicates datasets, which will be removed using **df.drop\_duplicates (inplace=True)**

* + 1. Unique data



**Result**: The output shows that we can classify the column quality based on its unique rating of 3,4,5,6,7,8 where: 3-6: Bad Wine – 0 7-8: Good Wine – 1

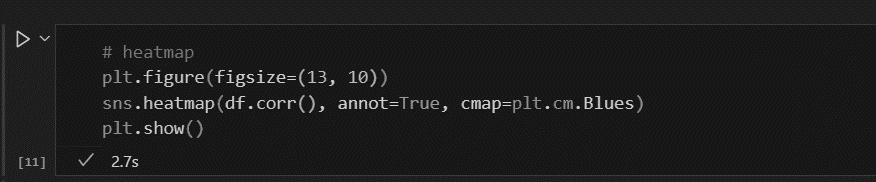
## Data Visualization

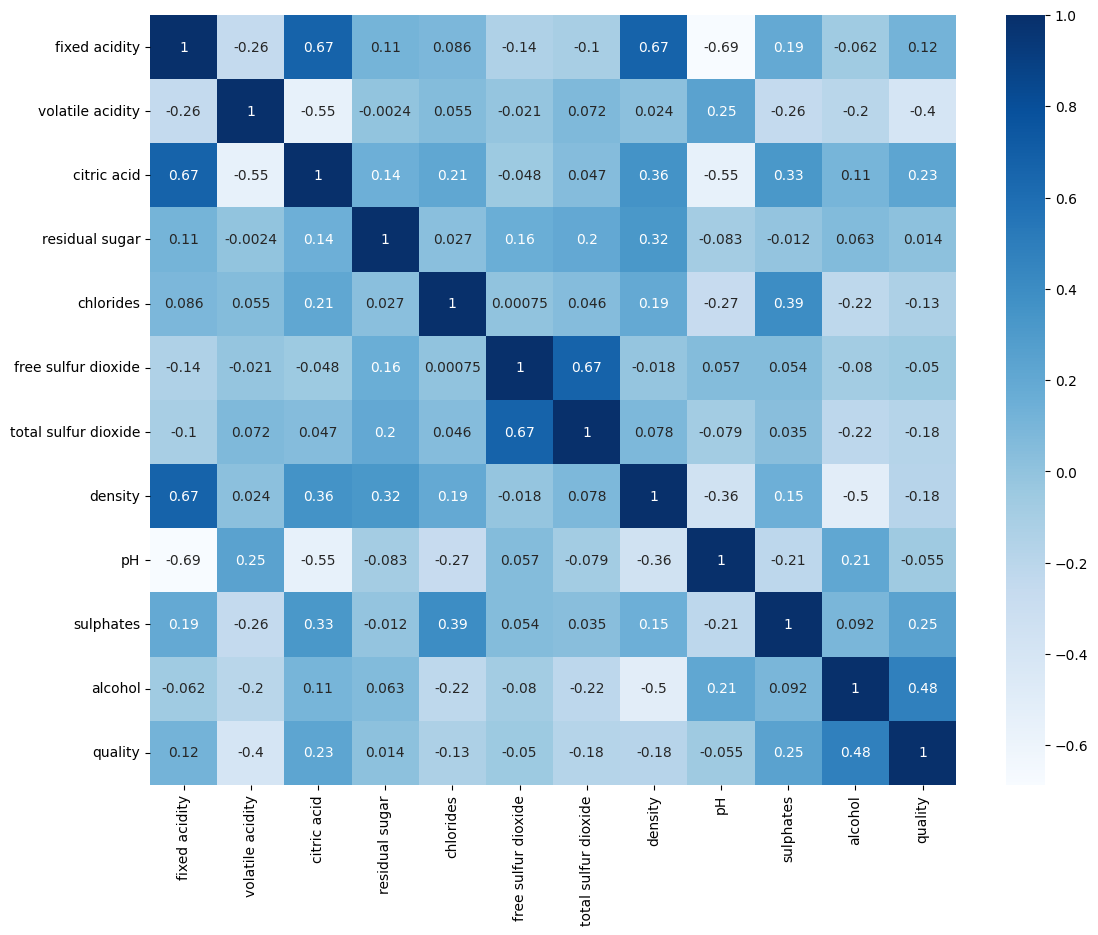
* + 1. Data Correlation



**Result:** The **df.corr()** computes pairwise correlation of all numerical columns.

* + 1. Heatmap

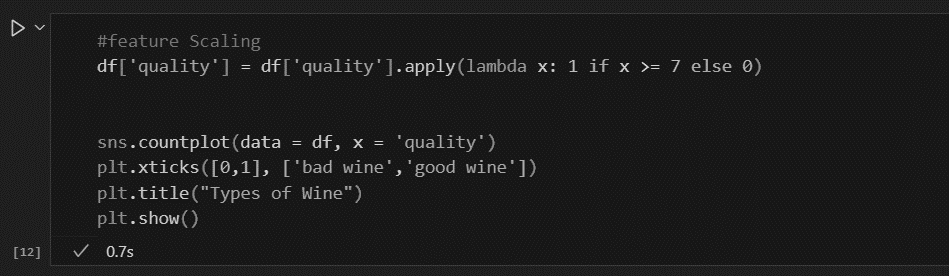




**Result**: The **heatmap** provides visual representation of these correlations. High correlations are shown in dark colors, while low correlations are in light colors. For example: Alcohol has a positive correlation (+0.48) with quality, suggesting that higher alcohol content is associated with better wine quality. Volatile acidity has a strong negative correlation (-0.40) with quality, implying that higher volatile acidity tends to lower wine quality.

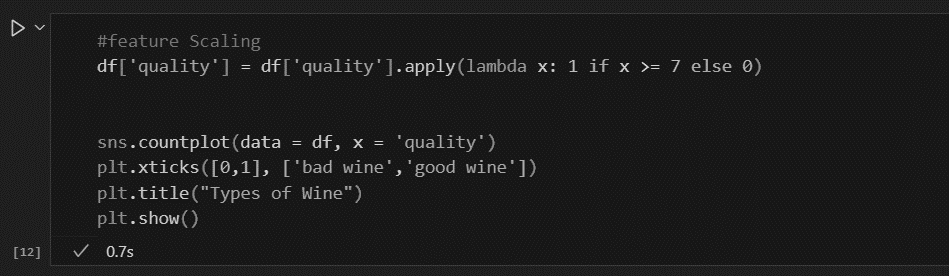
## Preprocessing Data

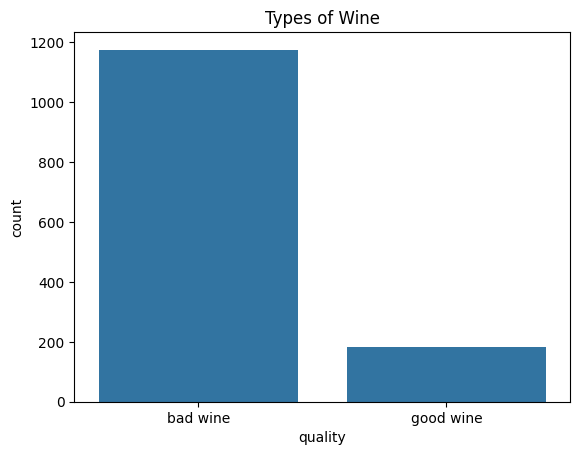
* + 1. Feature Scaling



**Result:** This converts the wine quality to a binary classification:

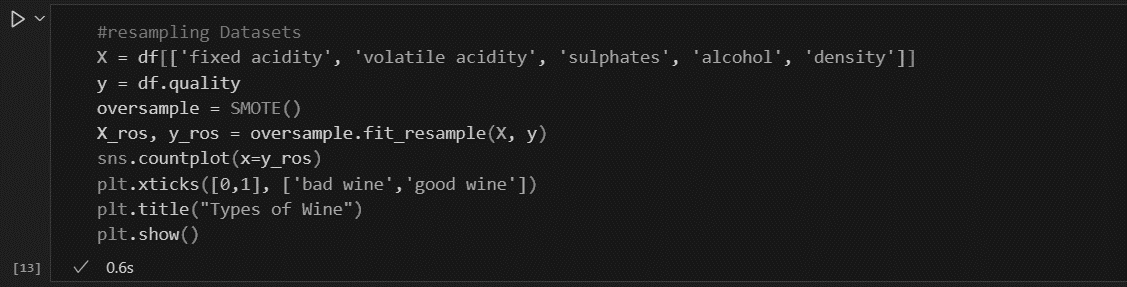
* 1: Good wine (quality >= 7)
* 0: Bad wine (quality < 7)

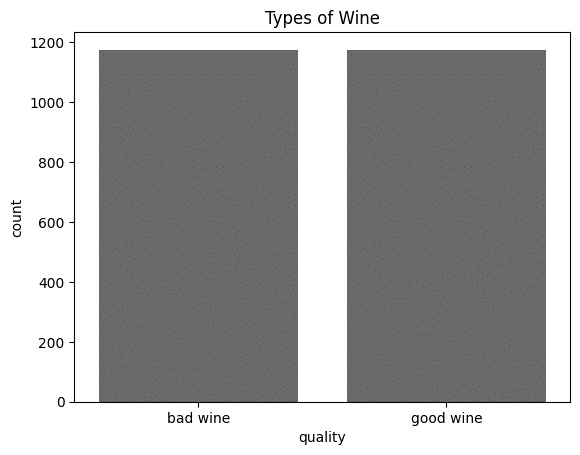




**Result**: **countplot** visualizes the distribution of wine quality in the dataset, showing the count of good vs bad wines.

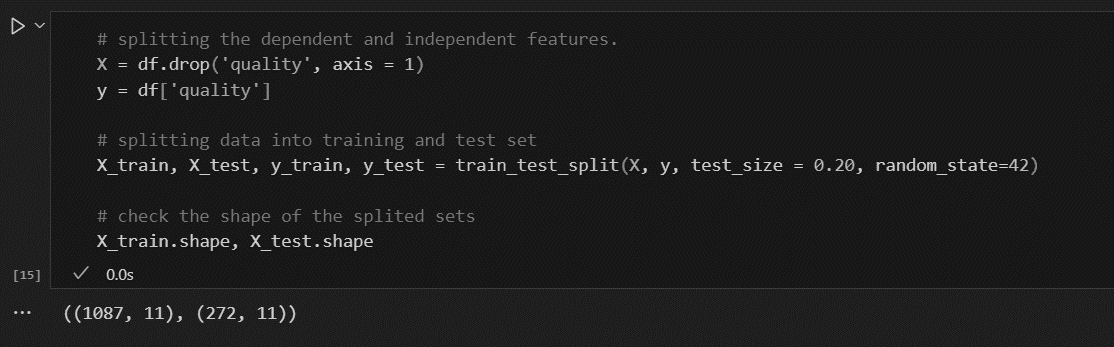
* + 1. Resampling Dataset





**Result: SMOTE** oversamples the minority class (good wine) to balance the dataset, providing a better classification model.

* + 1. Splitting the dataset into train and test

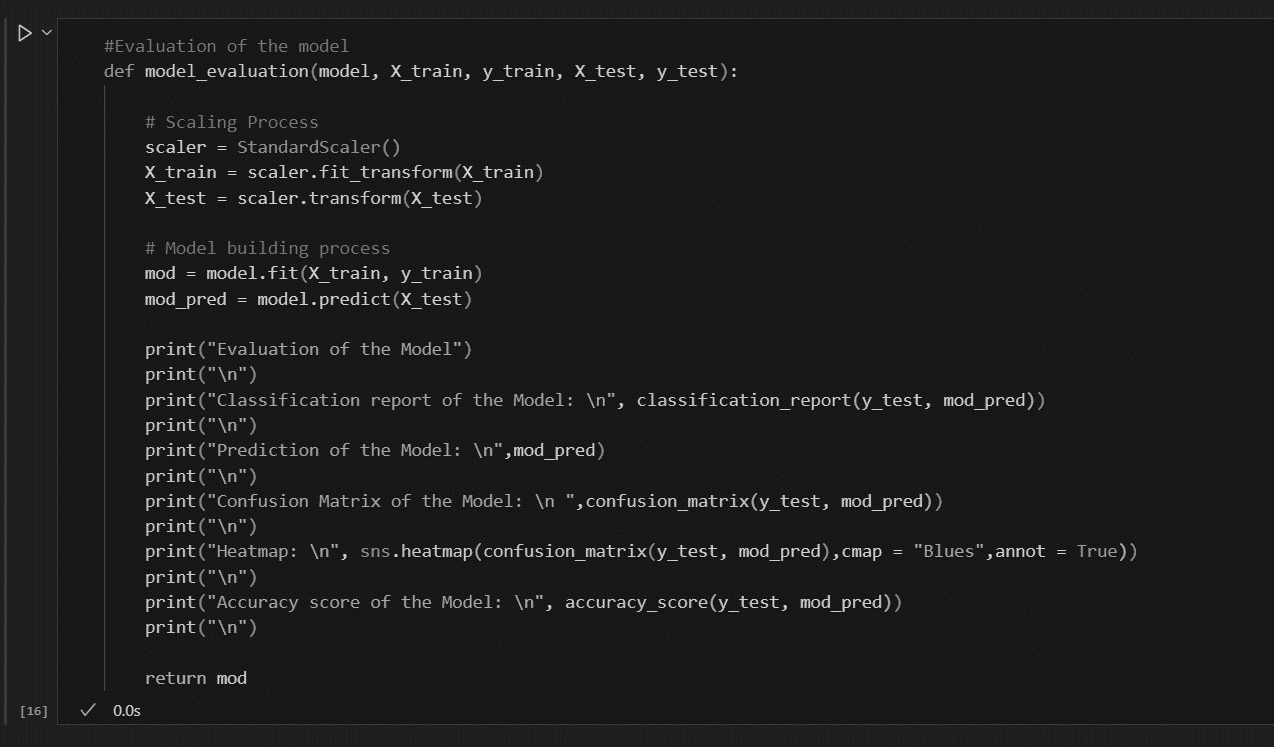


**Result**: **train\_test\_split()** splits the data into two subsets- Training and Testing data. Here in this wine data, training is carried out with 80% and testing with 20%.

**X\_train.shape** and **X\_test.shape** show the size of the training and test sets.

## Training the Model

After completion the data preprocessing steps, we can now train our model. We will focus on using the Logistic Regression and the Random Forest algorithm for this. During training, the model learns from the data and adjusts its internal parameters to make accurate predictions. Once trained, the model can be evaluated and deployed to predict the quality of wines based on their attributes.

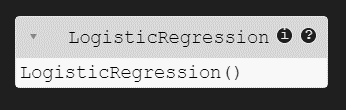


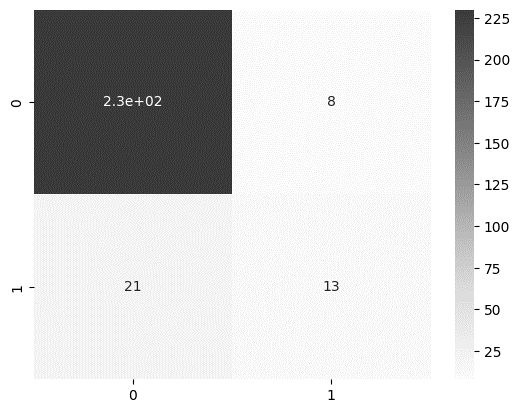
**Result**: This function does the following for any model:

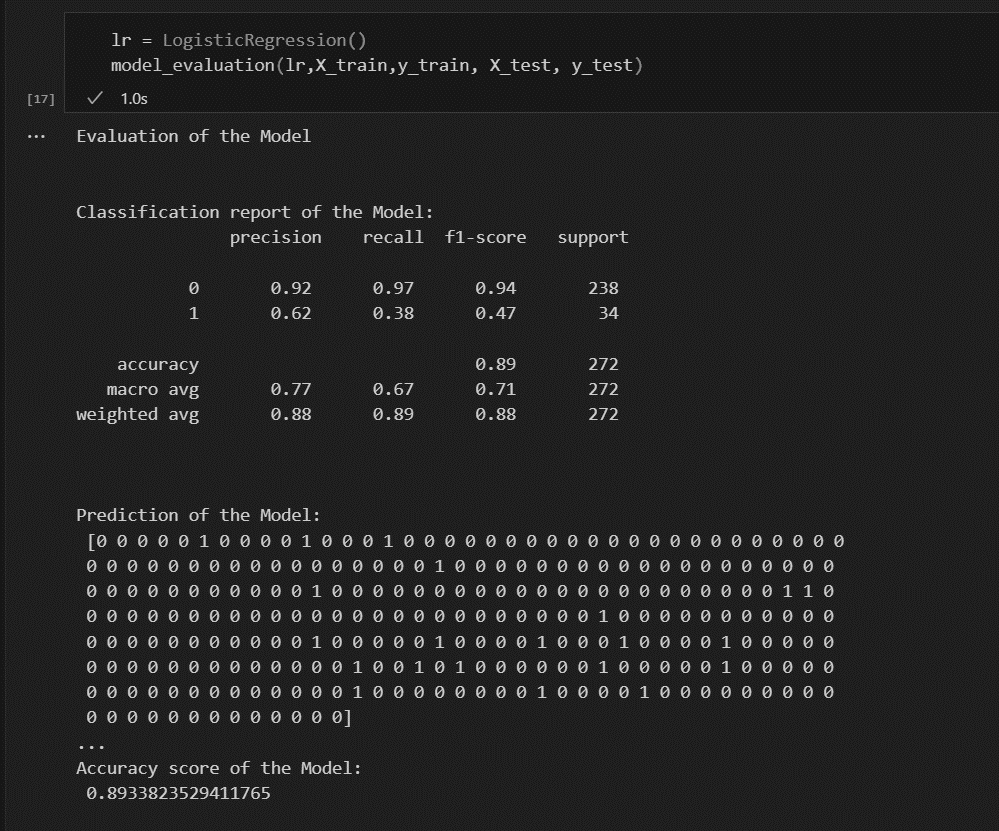
* **Standard Scaler**: Scales the training and test data.
* **Model Training**: Fits the model to the training data.
* **Prediction**: Predicts on the test data.
* **Metrics**: Outputs several evaluation metrics, such as:
  + **Classification Report**: Precision, Recall, F1-Score.
  + **Confusion Matrix**: True Positives, True Negatives, False Positives, False Negatives.
  + **Heatmap**: Visual representation of the confusion matrix.
  + **Accuracy Score**: Overall accuracy of the model on the test data.

### Logistic Regression

The logistic regression statistic modeling technique is used when we have a binary outcome variable.



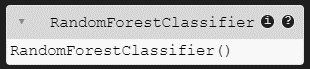


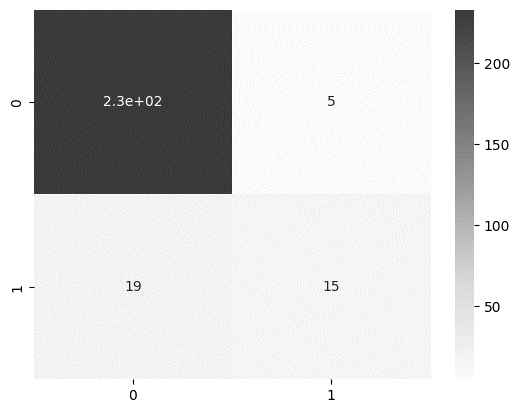


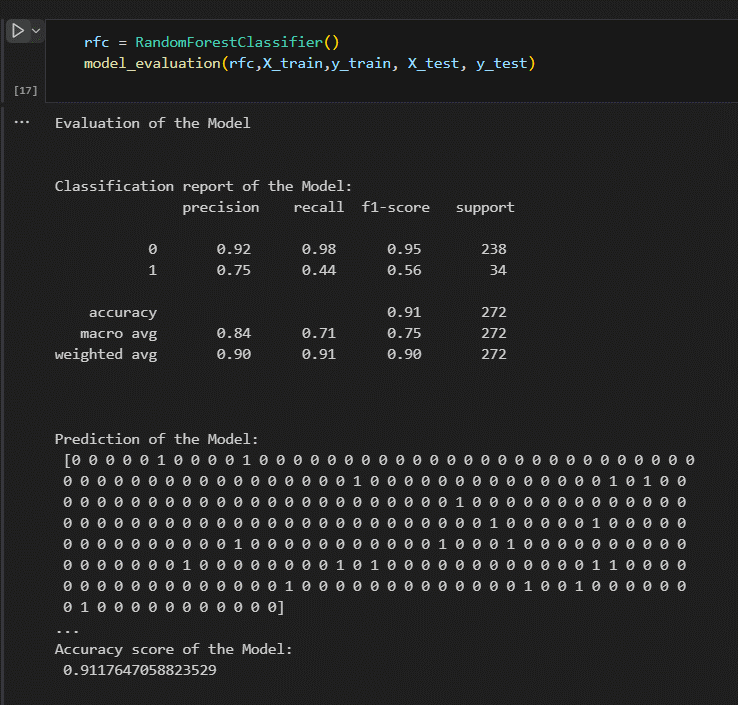
Result: Using the **logistic regression model**, the **accuracy** of the predictions reached approximately **89%**, which indicates that the model performs well overall. However, it was observed that the model struggled with the good wine. While the model's **precision** and **recall** for bad wine were high (**92% and 97%**, respectively), the same metrics for good wine were lower, particularly with a recall of **38%** and precision of **62%**. This highlights that the model is biased toward the bad wine, leading to fewer correct predictions for the good wine.

### Random Forest Classifier

The **Random Forest classifier** is an ensemble machine learning algorithm that builds multiple decision trees during training and merges their outputs to improve predictive accuracy and reduce overfitting.







Result: Using the **Random Forest Classifier**, the model achieved an **accuracy of around** **91%**, showing strong overall performance. The classifier excelled in predicting bad wine with a **precision of 92%** and **recall of 98%**. However, it struggled with the good wine, where recall was significantly lower at **44%**, despite a reasonable precision of **75%**. This suggests that while the model is effective in detecting bad wines, it needs improvement in correctly classifying good wines.

Conclusion

The project aimed to classify red wine quality using various machine learning algorithms. After preprocessing the data (scaling, handling class imbalance), two model was tested:

* Logistic Regression performed well but showed weaknesses in predicting the minority class (good wine).
* Random Forest provided the best results, improving the balance between precision and recall for both classes, making it the most effective model for this problem.

To improve the classification, **hyperparameter tuning** (for example, adjusting tree depth or the number of estimators) or **feature engineering** could further enhance model performance. Moreover, addressing class imbalance using techniques like **SMOTE** or **class weighting** significantly improved results, especially for predicting good wine.

References

|  |  |
| --- | --- |
| [1] | W. Enthusiast, "Red Wine Information & Basics," 10 August 2020. [Online]. Available: https://www.wineenthusiast.com/basics/red-wine-basics/. |
| [2] | N. Yadav, "Red Wine Quality Classification," 20 March 2024. [Online]. Available: https://www.kaggle.com/code/niteshyadav3103/red-wine-quality-classification/notebook. |
| [3] | P. Kumar, "Red Wine Quality Classification," 10 February 2021. [Online]. Available: https://pranavkumar623.medium.com/red-wine-quality-classification-9e91c858771e. |