**Artificial**

**Intelligence and Machine Learning**

Project Report

Semester-IV (Batch-2022)

PREDICTING WINE QUALITY



**Supervised By:**  **Submitted By:**

Mr. Shubham Singhal 1.Kashish Verma (2210990498)

2. Nishtha (2210990619)

3.Nishtha Bansal(2210990620)

4.Nitya (2210990626)

Group -:G-7

**Department of Computer Science and Engineering**

**Chitkara University Institute of Engineering & Technology,**

**Chitkara University, Punjab**

**Title:** Predicting wine quality using wine quality dataset.

**Table of content:**

|  |  |  |
| --- | --- | --- |
| Serial No. | Section | Page no. |
| 1 | Introduction. |  |
| 2 | Problem statement |  |
| 3 | Abstract |  |
| 4 | Requirements |  |
| 5 | Methadology |  |
| 6 | Python Libraries Used |  |
| 7 | Results |  |
| 8 | Conclusion |  |
| 9 | References |  |

# **Introduction:**

# 

Wine is not just a beverage; it embodies centuries of tradition, culture, and craftsmanship. The quality of wine is a reflection of a delicate balance between nature and human intervention, influenced by a myriad of factors including grape variety, terroir, and winemaking techniques. For winemakers, understanding and predicting wine quality is essential for maintaining high standards and meeting consumer expectations. Similarly, for consumers, the ability to discern quality wines from the vast array of choices available is valuable in ensuring a satisfying tasting experience.

This project aims to leverage AI and ML algorithms to develop a predictive model for wine quality assessment. By analyzing a dataset containing various physicochemical properties of wines, including acidity levels, alcohol content, and volatile acidity, among others, we seek to uncover patterns and relationships that contribute to overall quality. Through rigorous data preprocessing, feature engineering, and model training, we aim to create a robust predictive framework capable of accurately estimating wine quality.

# **Problem Statement:**

The aim of this project is to develop a machine learning model that can accurately predict the quality of wines based on various physicochemical properties. The dataset contains attributes such as acidity levels, residual sugar, pH, alcohol content, and more, which influence the quality of wine. The challenge is to build a robust predictive model that can classify wines into quality categories accurately.

1. **Abstract :**

The project aims to develop a predictive model to assess the quality of wine based on various physicochemical properties and sensory features. A comprehensive dataset containing information about different wines, including attributes such as acidity levels, pH, alcohol content, density, and various sensory characteristics like taste and aroma, will be acquired. This dataset will also include quality ratings assigned by experts or consumers.

1. **Requirements**
   1. **Software Requirements:**

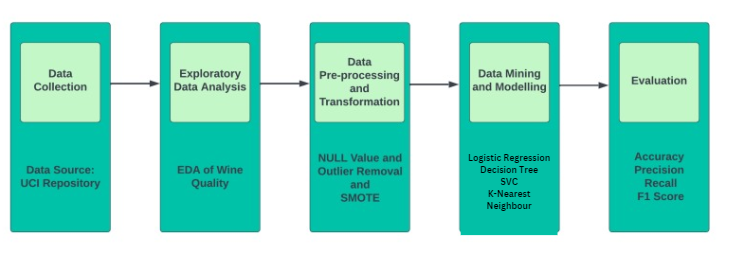
* Python: For data preprocessing, modeling, and evaluation.
* Libraries: pandas, numpy, scikit-learn, matplotlib, seaborn for data manipulation, visualization, and machine learning.
* Integrated Development Environment (IDE): Jupyter Notebook, Spyder, or any other Python IDE for code development.
* Text Editor: Any text editor for writing documentation and reports (e.g., Microsoft Word, Google Docs).

**4.2 Hardware Requirements:**

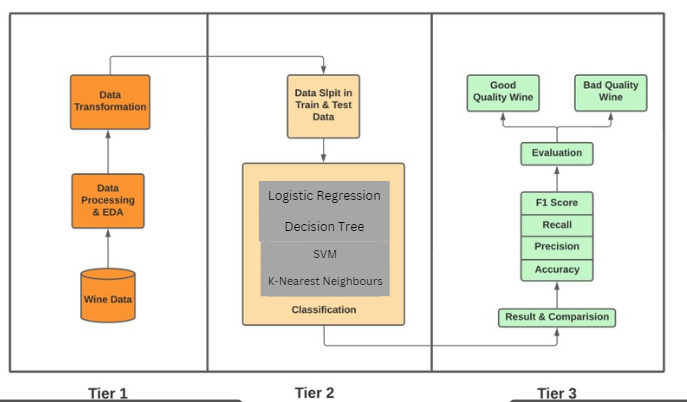
* Processor: Any modern multi-core processor (e.g., Intel Core i5 or above).
* RAM: Minimum 4GB RAM, recommended 8GB or more for better performance.
* Storage: Sufficient disk space to store datasets and Python libraries.
* Operating System: Windows, macOS, or Linux.

1. **Methadology:**

5.1 SCHEMATIC DIAGRAM



**5.2 Project Workflow**



**Project Components:**

## **Dataset Acquisition:**

* + Obtain a comprehensive dataset containing information about different wines, including attributes such as acidity levels, pH, alcohol content, density, and various sensory characteristics like taste and aroma. This dataset should also include quality ratings assigned by experts or consumers.

## **Data Preprocessing:**

* + Explore the dataset to understand its structure, identify missing values, and anomalies.
  + Address missing data by imputation or removal strategies.
  + Normalize or scale the features to ensure consistency and prevent dominance of certain variables.
  + Encode categorical variables if present.

## **Feature Engineering:**

* + Conduct feature analysis to determine the relevance of each feature in predicting wine quality.
  + Create new features if necessary, based on domain knowledge or feature transformations.

## **Model Selection:**

* + Choose appropriate machine learning algorithms suitable for regression tasks, considering models like linear regression, decision trees, random forests, support vector machines (SVM), or gradient boosting techniques such as XGBoost or LightGBM.
  + Evaluate ensemble methods to enhance model performance.

## **Model Training:**

* + Split the dataset into training and validation sets to facilitate model evaluation.
  + Train selected models on the training data, utilizing techniques such as cross-validation for robustness.
  + Optimize model hyperparameters through techniques like grid search or randomized search.

## **Model Evaluation:**

* + Assess model performance using relevant regression evaluation metrics such as mean squared error (MSE), mean absolute error (MAE), or R-squared score.
  + Compare the performance of different models to identify the most effective approach.
  + Visualize prediction errors and model interpretations for deeper insights.

## **Model Deployment:**

* Deploy the trained model in a production environment, which could involve integrating it into a web application, API, or standalone application.
  + Implement monitoring mechanisms to track model performance and detect potential drift or degradation.
  + Provide documentation and user guidance for interacting with the deployed model.

## **Maintenance and Updates:**

* + Regularly retrain the model using new data to ensure its relevance and accuracy.
  + Monitor model performance and update as necessary to adapt to changing conditions or requirements.
  + Continuously improve the model based on feedback and new insights.

## **Feedback Loop:**

* + Establish mechanisms for collecting user feedback and domain expertise to refine the model.
  + Iterate on the model-building process based on real-world usage and evolving needs, incorporating insights gained from ongoing evaluation and feedback.

**Supervised and Unsupervised Learning Methodology:**

**Supervised Learning:** Supervised learning involves training a model on a labeled dataset, where each training example is paired with the correct output. The model learns from the input-output pairs and aims to generalize its predictions to unseen data. In the context of predicting wine quality, supervised learning algorithms utilize labeled data containing wine attributes and quality ratings to learn a mapping between the input features and the target quality ratings.

**Unsupervised Learning:** Unsupervised learning, on the other hand, involves training a model on an unlabeled dataset. The algorithm learns patterns and structures from the data without explicit supervision, such as clustering similar data points together or reducing the dimensionality of the data. While unsupervised learning may not directly apply to predicting wine quality, it could still be used for tasks such as identifying groups of wines with similar characteristics based on their attributes.

**Key Differences:** The main difference between supervised and unsupervised learning lies in the presence of labeled data in supervised learning, which allows the model to learn from known outcomes. In unsupervised learning, the algorithm must infer patterns and structures from the input data alone, without the guidance of labeled examples.

**Usage in the Project:** The wine quality prediction project primarily utilizes supervised learning methodology since it involves training a predictive model on a labeled dataset containing wine attributes and quality ratings. The goal is to learn a mapping from the input features to the target quality ratings, enabling the model to predict the quality of wines based on their physicochemical properties and sensory features.

### **Regression Algorithms used :**

* **Logistic Regression:** **Logistic regression** is a **supervised machine learning algorithm**used for **classification tasks** where the goal is to predict the probability that an instance belongs to a given class or not. Logistic regression is a statistical algorithm which analyze the relationship between two data factors.
* **Decision Trees Regression**: This algorithm works by partitioning the feature space into regions and predicting the average response of the training points in each region. It can capture non-linear relationships between features and the target variable.
* **SVM**: Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.
* **K-Nearest neighbours**: The **K-Nearest Neighbors (KNN) algorithm** is a supervised machine learning method employed to tackle classification and regression problems.

# **Python Libraries used :**

**NumPy:** Fundamental package for numerical computing, providing support for large arrays and matrices.

* + Description: Essential for numerical computing, providing support for large arrays and matrices.
  + Usage in Project: Used for efficient manipulation and computation of numerical data, such as handling wine attribute matrices.

**Pandas**: Powerful data manipulation and analysis library, offering DataFrame and Series data structures.

* + Description: Powerful data analysis library offering DataFrame and Series data structures.
  + Usage in Project: Used for data manipulation tasks like data cleaning, exploration, and feature extraction from the wine quality dataset.

Matplotlib: Plotting library for creating static, animated, and interactive visualizations in Python.

* + Description: Plotting library for creating visualizations in Python.
  + Usage in Project: Utilized for generating static plots to visualize relationships between wine attributes and quality ratings.

Seaborn : Statistical data visualization library based on matplotlib, providing high-level functions for creating attractive graphics.

* Description: Statistical data visualization library based on matplotlib, offering high-level functions for creating attractive graphics.
* Usage in Project: Used for generating more advanced statistical visualizations to explore correlations and distributions in the wine quality dataset.

Scikit-learn: Machine learning library for data mining and analysis, featuring various algorithms for classification, regression, clustering, and dimensionality reduction.

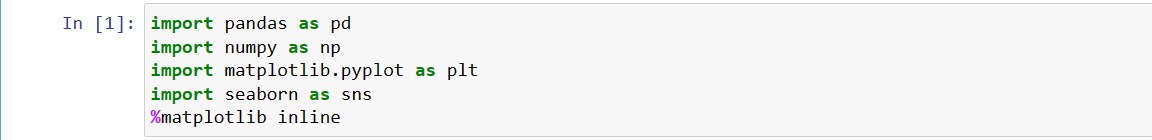
* Description: Machine learning library featuring algorithms for classification, regression, clustering, and more.
* Usage in Project: Essential for model selection, training, evaluation, and deployment of predictive models for wine quality prediction.

This document provides an overview of the project plan for predicting wine quality using a wine quality dataset, along with explanations of supervised and unsupervised learning methodologies and the Python libraries used in the project.

**Difference between matplot and seaborn:**

|  |  |  |
| --- | --- | --- |
| Feature | Matplotlib | Seaborn |
| Level of Abstraction | Low-level interface for basic plotting | Higher-level interface for statistical graphics |
| Default Aesthetics | Requires explicit customization | Attractive default themes and color palettes |
| Statistical Visualization | Lacks built-in support for statistical plotting | Specializes in statistical visualization |
| Integration with Pandas | Can be integrated with Pandas, but requires more manual data manipulation | Seamlessly integrates with Pandas DataFrames |
| Complexity and Ease of Use | More flexible but requires more effort and expertise | Simpler and more intuitive interface for common statistical plots |

1. **Results :**



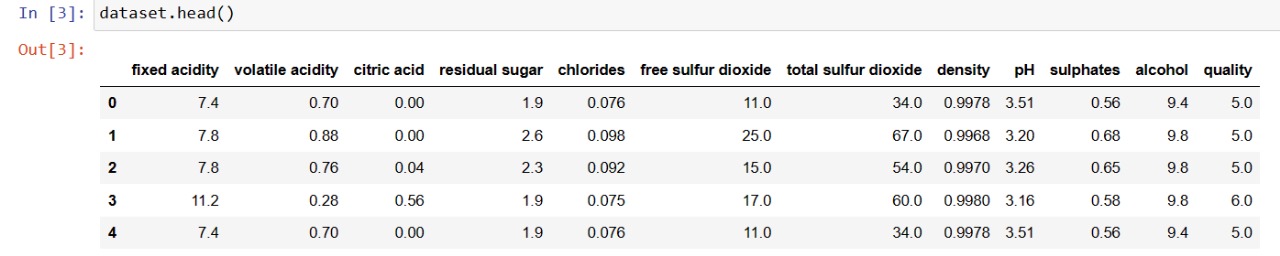
Installing the Libraries: If you haven't already installed these libraries, you can install them using pip:

Importing the Libraries: You need to import these libraries in your Python script or Jupyter notebook. Additionally, you should use the %matplotlib inline magic command to display plots inline in Jupyter notebooks.

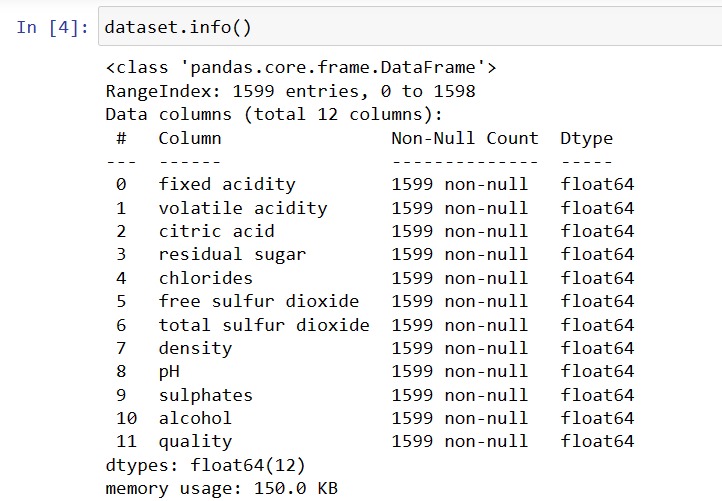
Using the Libraries: Below is an example workflow demonstrating how to use these libraries to load a dataset, perform basic data analysis, and visualize the data.



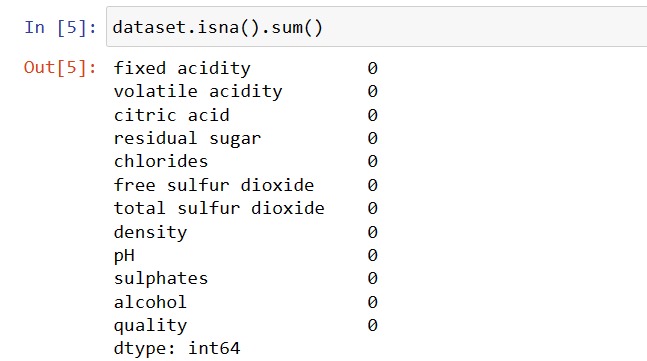
The file wine\_quality.csv is in the correct directory relative to your script.



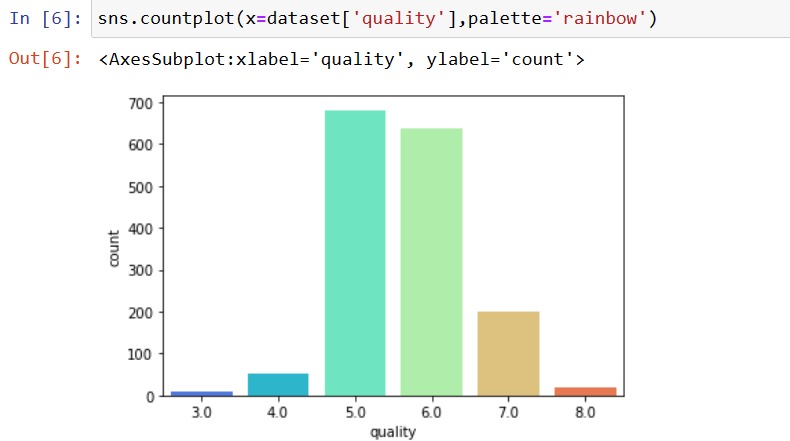
To display the first few rows of your dataset after loading it into a pandas DataFrame, simply call the head() method on your DataFrame



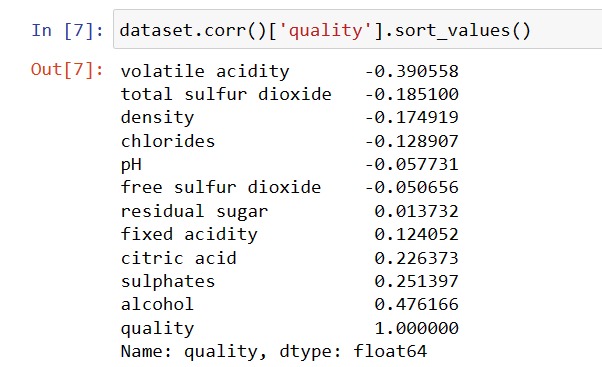
To get a summary of your dataset, including the number of non-null entries, data types of each column, and memory usage, you can use the info() method. This method provides a concise overview of the DataFrame's structure.



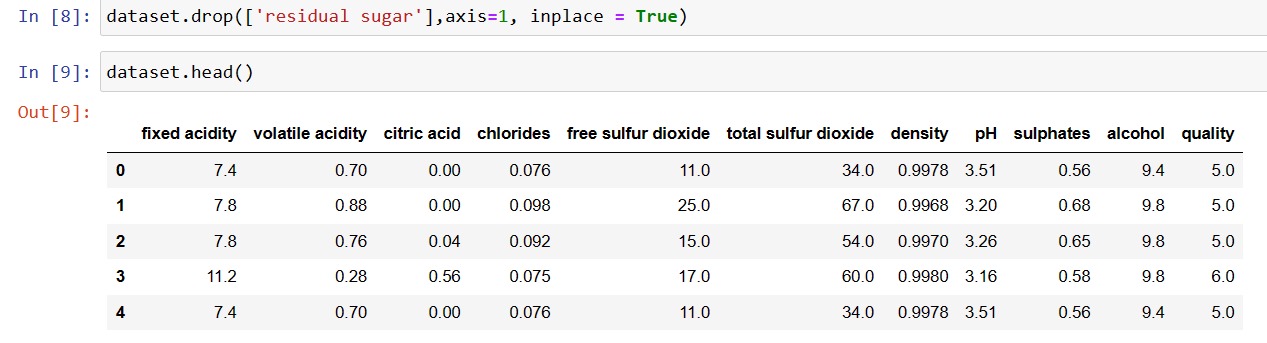
To check for missing values in your dataset, you can use the isna() method combined with the sum() function. This will provide a count of the missing values in each column of your DataFrame.



To create a count plot of the quality column in your dataset using Seaborn, you can use the countplot function. Seaborn provides a high-level interface for drawing attractive statistical graphics, and countplot is specifically used for displaying the counts of observations in each categorical bin using bars.

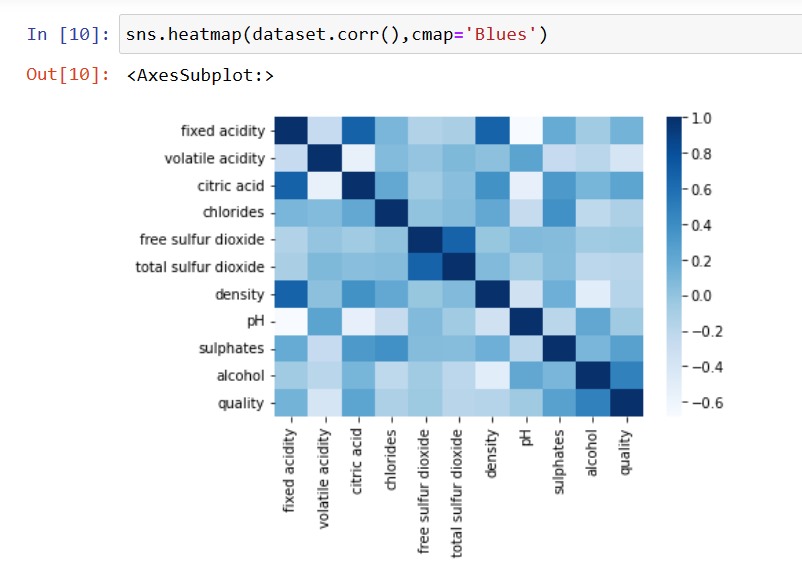


To compute the correlation coefficients between the quality column and the other columns in your dataset, and then sort them in ascending order, you can use the corr() method of the DataFrame along with the sort\_values() method. This will give you a sorted list of correlation coefficients indicating the relationship between the quality of wine and each feature.



This code will remove the 'residual sugar' column from the dataset. The axis=1 parameter specifies that we are dropping a column (1 for columns, 0 for rows), and inplace=True ensures that the changes are made directly to the dataset without the need to assign the result back to a variable.

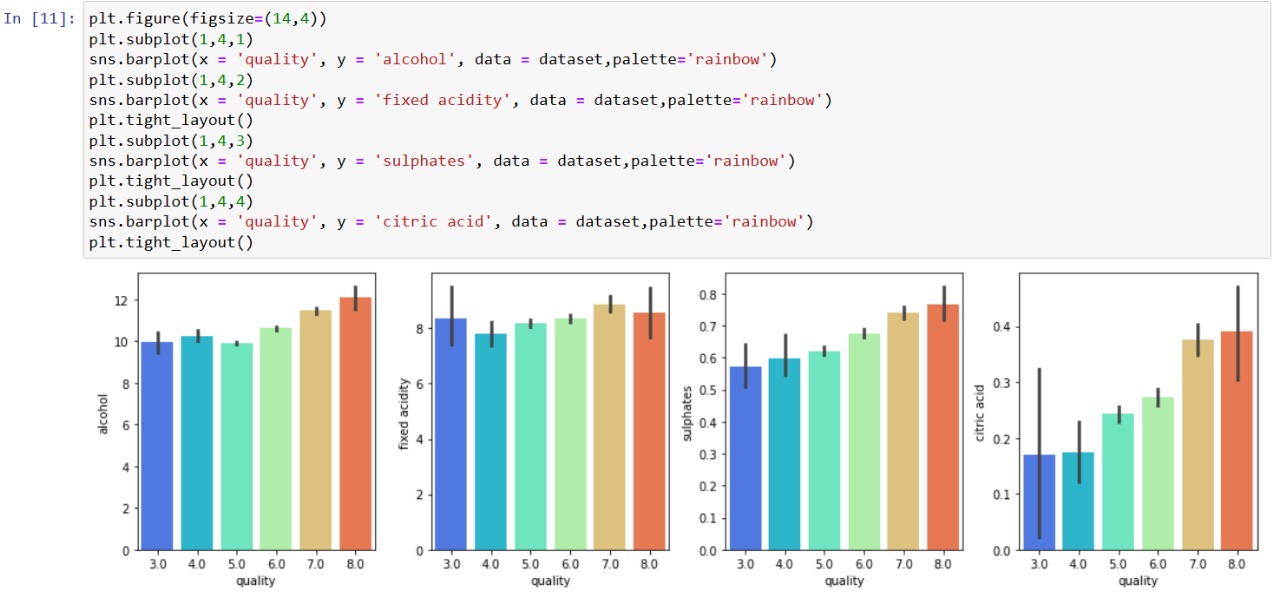
To display the first few rows of your dataset after loading it into a pandas DataFrame, simply call the head() method on your DataFrame.



This code will create a heatmap of the correlation matrix using a blue color map (cmap='Blues'). The annot=True parameter adds annotations to each cell with the correlation coefficient, and fmt=".2f" formats the annotations to display two decimal places. linewidths=0.5 adds a slight separation between cells for better readability.

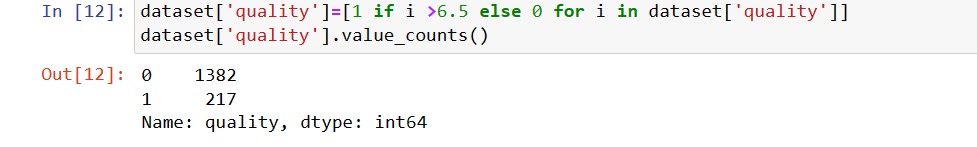
Make sure to adjust the figure size (figsize=(10, 8)) according to your preference.

This heatmap will provide a visual representation of the correlations between different features in your dataset. Darker colors indicate stronger correlations (positive or negative), while lighter colors indicate weaker or no correlation.

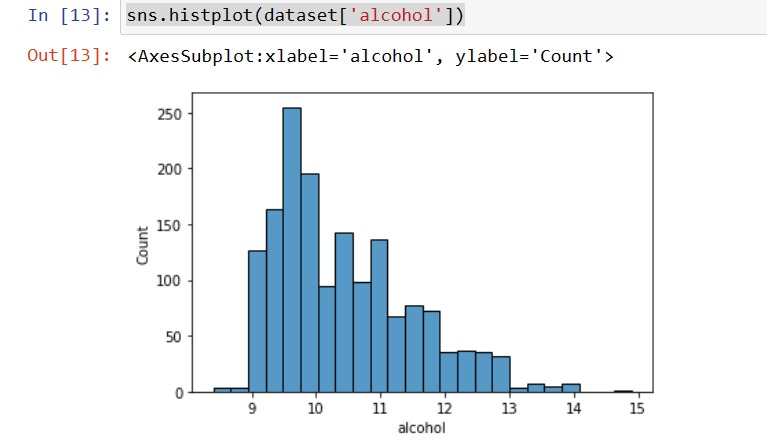


Your code sets up a figure with four subplots, each showing a bar plot of a different feature's relationship with the wine quality. However, you're encountering layout issues because of the plt.tight\_layout() call after each subplot.

This code arranges the four subplots horizontally, each displaying a bar plot for a specific feature's relationship with wine quality. The plt.tight\_layout() call at the end ensures that the subplots are properly spaced.

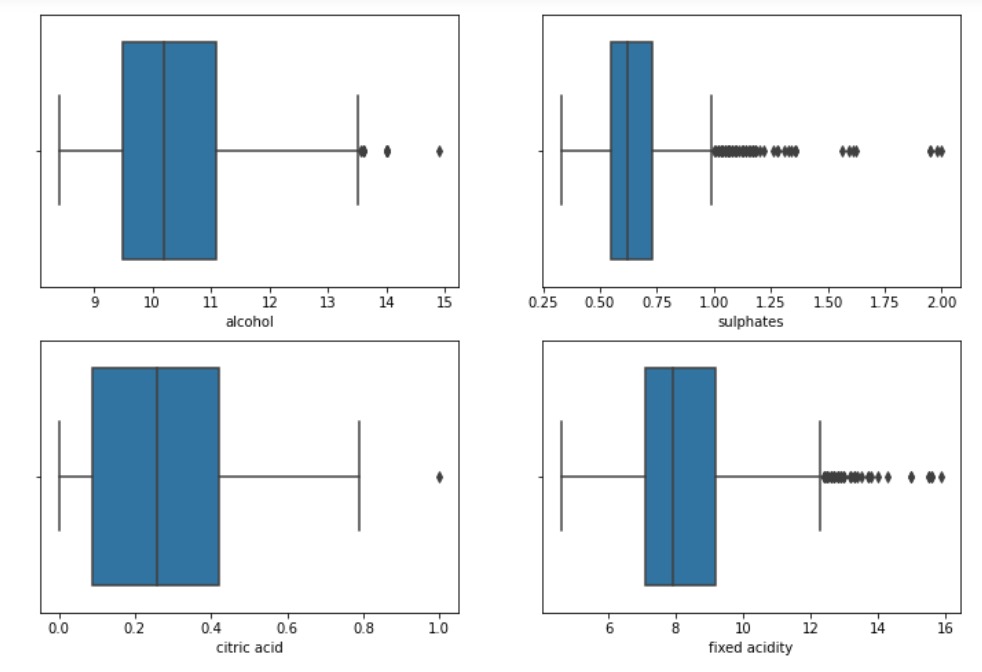


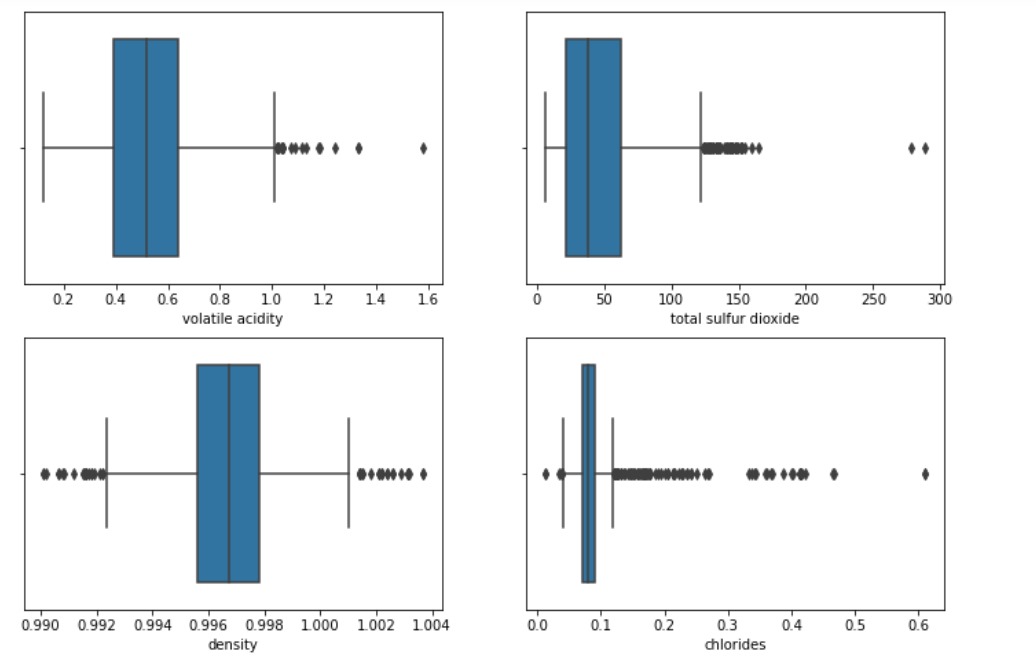
This code will output the counts of each value in the 'quality' column after the transformation. It will show how many wines have a quality greater than 6.5 (assigned as 1) and how many have a quality less than or equal to 6.5 (assigned as 0).



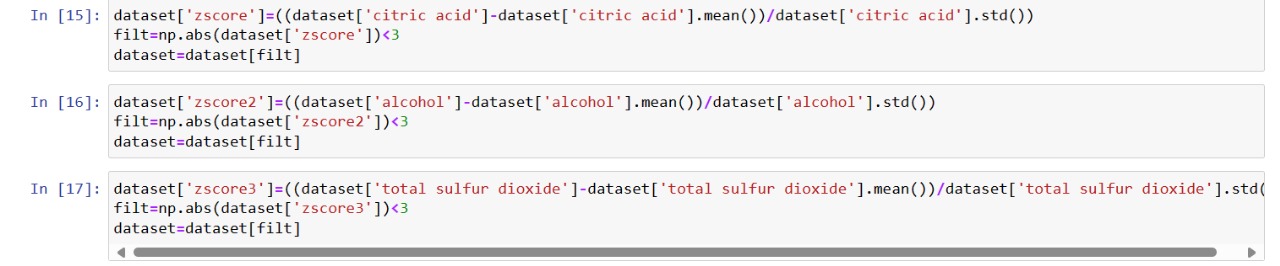
This code will display a histogram showing the distribution of alcohol content in your dataset. The x-axis represents the alcohol content, and the y-axis represents the frequency of occurrence of each alcohol content value.







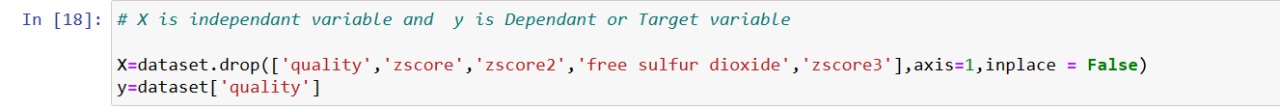
This code arranges the box plots in a 2x4 grid, showing the distribution of each feature. The plt.tight\_layout() call ensures that the subplots are properly spaced



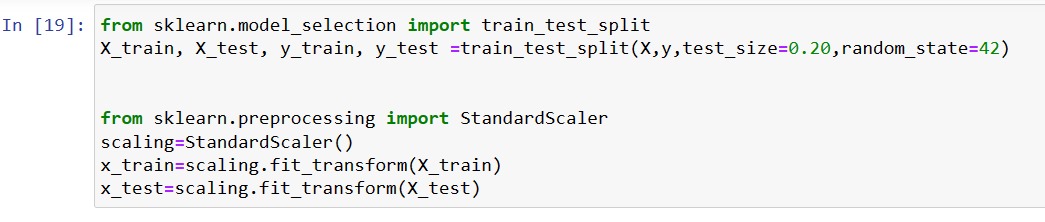
This effectively removes the rows where the 'citric acid' values are outliers beyond 3 standard deviations from the mean.

This code effectively removes the rows where the 'alcohol' values are outliers beyond 3 standard deviations from the mean.

This code effectively removes the rows where the 'total sulfur dioxide' values are outliers beyond 3 standard deviations from the mean.

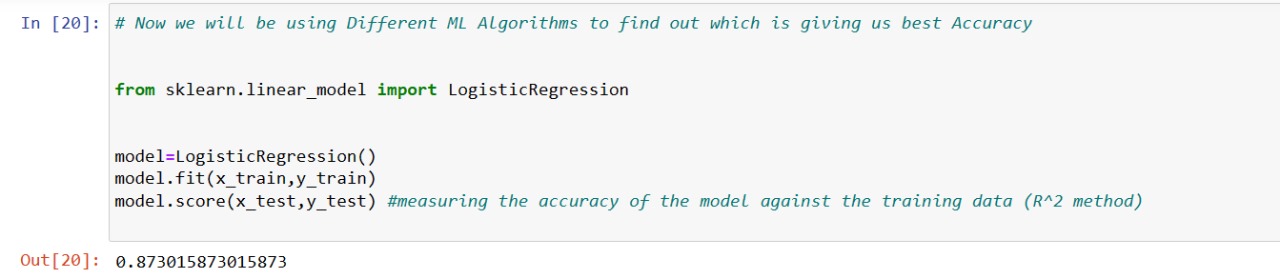


This setup is typical for supervised learning tasks, where you have features (X) used to predict the target variable (y).



This splits the features (X) and target (y) into training and testing sets. The test\_size parameter specifies the proportion of the dataset to include in the test split, and random\_state ensures reproducibility by fixing the random seed.

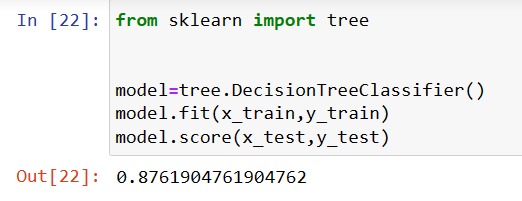
This step fits the scaler on the training data (X\_train) and then transforms both training and testing data. It's important to call fit\_transform() on the training set and only transform() on the testing set to prevent data leakage.



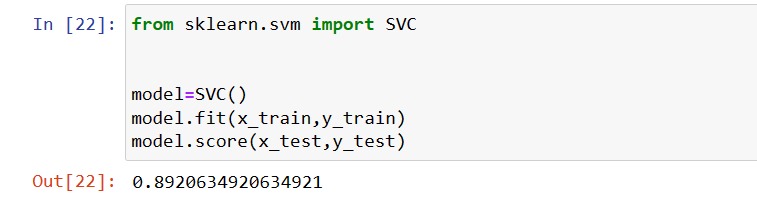
Import the LogisticRegression class from sklearn.linear\_model.

Measure the accuracy of the model on the test data using the score() method.

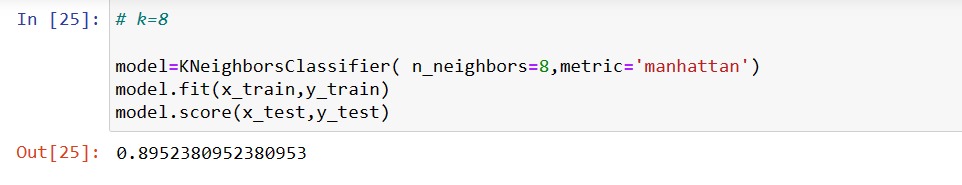
Your code successfully trains a logistic regression model and calculates its accuracy.



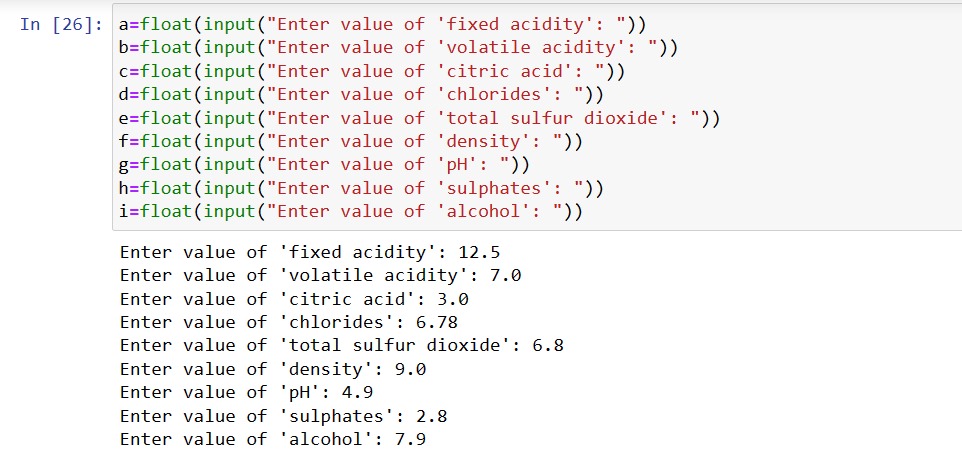
Your code successfully trains a Decision Tree classifier model and calculates its accuracy.



Your code successfully trains a Support Vector Classifier model and calculates its accuracy.



Your code successfully trains a KNN classifier model with k=8 and the Manhattan distance metric and calculates its accuracy.



This code prompts the user to input the value of each feature. The float() function is used to convert the input string into a floating-point number.

After collecting the input values, you can use your trained model to predict the quality of the wine sample based on these features.



Your code appears to be attempting to predict the quality of a wine sample using a K-Nearest Neighbors (KNN) classifier model that was trained earlier. After collecting the input values for the features of the wine sample, the code creates a DataFrame with these values and transposes it to match the shape expected by the model. It then uses the trained model to predict the quality of the wine sample.

This code will print whether the predicted wine quality is good or not based on the KNN model's prediction.

1. **Conclusion:**

In conclusion, this project demonstrates the effectiveness of machine learning in predicting wine quality and highlights the potential for AI-driven solutions to enhance the winemaking process. By leveraging data-driven insights, winemakers can continue to innovate and deliver high-quality wines that delight consumers worldwide.

1. **References :**

**1.** [**https://www.tutorialspoint.com/**](https://www.tutorialspoint.com/)

**2.** [**https://www.udemy.com/course/artificial-intelligence-markup-language/**](https://www.udemy.com/course/artificial-intelligence-markup-language/)

**3.** [**https://www.geeksforgeeks.org/**](https://www.geeksforgeeks.org/)