MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE
Belawadi Srirangapatna Tq, Mandya-571477

(Affiliated to Visvesvaraya Technological University, Belagavi)
(An Autonomous Institution, Approved by AICTE, New Delhi)

Exploratory Data Analysis

Assignment Report

Submitted in partial fulfillment for the award of degree of Bachelor of Engineering in Computer Science & Engineering (Data Science) during the year 2024-2025

Bachelor of Engineering

In

Computer Science and Engineering (Data Science)

Submitted by

RAHUL D V
4MH22CD048

Submitted To
Prof. Roshini P
Asst. Professor
Dept. of CSE(Data Science)

2024-25
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE), MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE
BELAWADI, SRIRANGAPATNA TALUK, MANDYA-571477

CONTENTS

ABSTRACT

Exploratory Data Analysis (EDA) serves as a cornerstone in unlocking the secrets held within datasets, offering a lens through which patterns, anomalies, and relationships come into focus. This report delves into my application of EDA to the Wine Quality Dataset, a rich compilation of physicochemical attributes from red wine samples of Portugal's Vinho Verde region. By harnessing tools like Python's pandas, seaborn, and matplotlib, this analysis meticulously examines feature distributions, uncovers correlations, and highlights outliers that shape our understanding of wine quality. The findings illuminate critical insights— such as the pivotal roles of alcohol content and volatile acidity in quality assessment—laying a robust groundwork for predictive modeling. Beyond technical exploration, this report underscores EDA's transformative potential in bridging raw data with actionable knowledge, offering value to both the wine industry and data science education.

INTRODUCTION

Exploratory Data Analysis (EDA) stands as an indispensable methodology in the realm of data science, empowering analysts to peel back the layers of complexity within datasets. At its core, EDA involves the use of statistical and visual techniques to summarize data characteristics, detect irregularities, and hypothesize relationships—all before diving into formal modeling. This report embarks on such an exploration, targeting the Wine Quality Dataset, a publicly available resource that captures the essence of red wine through its chemical composition and sensory evaluations.

Originating from the Vinho Verde region in northern Portugal, the Wine Quality Dataset comprises 1599 red wine samples, each meticulously documented with 11 physicochemical properties and a quality score assigned by expert tasters. Sourced from Kaggle, this dataset bridges the gap between objective measurements—like acidity or alcohol content—and the subjective art of wine tasting. Its relevance extends beyond academia into the wine industry, where understanding quality drivers can refine production techniques, enhance customer satisfaction, and optimize market offerings.

The motivation for this project lies in harnessing EDA to decode the dataset's intricacies. By applying Python-based tools such as pandas for data handling, seaborn and matplotlib for visualization, and basic statistical methods, we aim to reveal patterns and dependencies that influence wine quality. This analysis is not merely an academic exercise; it mirrors real-world applications where data-driven insights inform decision-making. From identifying key quality predictors to preparing the data for machine learning, this report showcases EDA's pivotal role in transforming raw numbers into meaningful narratives.

Moreover, the Wine Quality Dataset offers a fertile ground for learning. Its manageable size and diverse features make it an ideal case study for students and practitioners alike, while its industry ties highlight the practical implications of data analysis. Through this exploration, we seek to answer fundamental questions: What chemical properties most strongly correlate with quality? How do outliers affect our understanding? And how can these insights pave the way for predictive tools? This journey through EDA promises to illuminate both the dataset and the broader discipline of data science.

ABOUT THE DATASET

The **Wine Quality Dataset** is a structured collection comprising **1,599 red wine samples**, each evaluated based on **11 physicochemical properties** and an **expert-assigned quality score**. This dataset provides a comprehensive blend of chemical composition and sensory evaluation, enabling a data-driven approach to understanding wine quality.

Each sample is uniquely identified by an **id** column (excluded from analysis), and is described by the following **input features**:

- **Fixed Acidity (g/dm^3):** Represents non-volatile acids such as tartaric acid. These acids contribute to the structural stability, taste, and mouthfeel of the wine.
- **Volatile Acidity (g/dm^3):** Measures volatile acids, primarily acetic acid. High levels can lead to an undesirable vinegar-like taste.
- **Citric Acid (g/dm^3):** A minor acid that adds freshness and enhances flavor complexity by balancing the wine's acidity.
- **Residual Sugar (g/dm^3):** Indicates the amount of unfermented sugar left after fermentation. It contributes to the sweetness, body, and overall appeal of the wine.
- **Chlorides (g/dm^3):** Reflects the salt content derived from sodium chloride. While small amounts enhance flavor, excessive levels may be detrimental to taste.
- **Free Sulfur Dioxide (mg/dm^3):** The active portion of sulfur dioxide that helps prevent oxidation and microbial spoilage, maintaining freshness.
- **Total Sulfur Dioxide (mg/dm^3):** Encompasses both free and bound sulfur dioxide, serving as a broad measure of the wine's preservation.
- **Density (g/cm^3):** Indicates the wine's mass-to-volume ratio, which is influenced by its alcohol and sugar content.
- **pH:** Measures the acidity or alkalinity of the wine on a scale from 0 to 14. It affects taste, color stability, and microbial resistance.
- **Sulphates (g/dm^3):** Represents potassium sulphate compounds that act as antioxidants and antimicrobials, enhancing the wine's durability and robustness.
- **Alcohol (% vol):** Denotes the alcohol content by volume. It significantly impacts the body, warmth, and perceived quality of the wine.

The **output variable** in this dataset is:

- **Quality (score from 0 to 10):** A subjective score assigned by professional wine tasters based on aroma, taste, balance, and finish. In this dataset, scores range from **3 to 8**, with most samples scoring **5 or 6**.

This dataset is well-suited for exploratory data analysis, feature correlation studies, and machine learning applications focused on predicting wine quality from its chemical properties. It serves as an excellent example of integrating objective measurements with human sensory evaluation to derive meaningful insights.

IMPORTANCE OF THE DATASET

The **Wine Quality Dataset** is more than a compilation of chemical measurements—it is a **versatile and impactful resource** with wide-ranging applications across education, industry, and research. Its structured format, real-world relevance, and balanced complexity make it a valuable asset for both learning and practical deployment.

1. Educational Significance

In academic environments, the dataset serves as an **ideal instructional tool**. Its manageable size and clearly defined features make it accessible for students learning data science concepts such as:

- Regression and classification
- Feature selection and engineering
- Model evaluation and interpretation

By examining the relationship between physicochemical properties and wine quality scores, learners gain practical, hands-on experience in data analysis and machine learning within a tangible and engaging context.

2. Practical Relevance to the Wine Industry

For wine producers and quality control experts, the dataset offers **data-driven insights** into the factors influencing wine quality. Understanding how specific chemical attributes—such as **alcohol content, volatile acidity**, or **sulphates**—correlate with expert-assigned quality scores enables:

- Optimization of fermentation processes
- Adjustment of chemical additives
- Consistent product quality aligned with consumer preferences

This informed approach can lead to **cost efficiency**, **improved quality control**, and **higher customer satisfaction**.

3. Research Applications

The dataset provides a fertile ground for researchers in **oenology**, **sensory science**, and **applied machine learning**. It enables:

- Exploration of the connection between objective chemical metrics and subjective sensory evaluations
- Validation of predictive models for quality assessment
- Development of new frameworks for understanding flavor profiles and wine characteristics

Such studies deepen scientific knowledge while contributing to innovations in wine analysis and evaluation.

OBJECTIVES & SCOPE

Objectives:

1. The first objective is to analyze the distribution, range, central tendency, and spread of each physicochemical feature in the dataset to gain a deep understanding of its statistical behavior.
2. A key objective is to investigate the relationships and correlations between various chemical properties and the quality score, to determine which variables most significantly affect wine quality.
3. The project aims to identify and assess outliers in the dataset, such as extreme values in acidity or alcohol content, to evaluate their impact on the overall analysis and interpretation.
4. Another important goal is to clean and preprocess the dataset by handling missing values (if any), ensuring data type consistency, and normalizing or scaling where necessary for modeling purposes.
5. The project seeks to visualize data using a variety of charts such as histograms, box plots, and heatmaps to make patterns and anomalies more interpretable and visually informative.
6. A further objective is to prepare the dataset for machine learning by transforming features, encoding categorical variables (if applicable), and splitting the data into training and test sets.
7. The project also intends to implement and evaluate a basic predictive model (e.g., a classification algorithm) to demonstrate how well wine quality can be predicted based on its chemical characteristics.
8. Finally, the project aims to highlight the practical importance of EDA in real-world scenarios by showing how data can be used to support wine quality control, production improvement, and consumer satisfaction.

Scope:

1. The project is limited to the analysis of red wine samples from the Wine Quality Dataset, intentionally excluding white wine data to maintain a focused and in-depth exploration.
2. Python is used as the primary programming language due to its robust ecosystem for data analysis, with core libraries including pandas for data handling, numpy for numerical operations, matplotlib and seaborn for visualization, and scikit-learn for preprocessing and modeling.
3. The scope includes inspecting data structure and types, verifying data completeness, and performing descriptive statistical analysis to understand the dataset's baseline properties.
4. The project covers visualization techniques such as distribution plots, box plots, and correlation heatmaps to interpret data behavior, variability, and inter-feature relationships.
5. Outlier detection is conducted using statistical and visual methods, though outliers are retained to preserve the dataset's originality and account for real-world variability.
6. Initial feature importance analysis is conducted to identify which attributes contribute most to the quality prediction, offering insights that can inform future feature selection or engineering.
7. A basic machine learning model is applied as part of the exploratory phase, serving as a bridge between EDA and predictive analytics while keeping the modeling scope minimal.
8. Overall, the project provides a complete EDA workflow while emphasizing clarity, reproducibility, and practical insights for educational, industrial, and research applications.

PROCEDURE

1. The EDA process begins by **loading the dataset using pandas** from a CSV file and **inspecting its structure**, which includes 1599 rows and 12 columns.
2. An initial check is conducted to confirm **the absence of missing values**, ensuring a clean dataset for analysis.
3. **Summary statistics** such as mean, median, standard deviation, and quartiles are calculated for all features to understand their **central tendencies and variability**.
4. **Feature distribution analysis** is performed using **histograms** for each physicochemical attribute to visualize their distribution, such as a right-skewed alcohol content and a near-normal pH distribution.
5. The **target variable, quality**, is analyzed separately using a **count plot** to show the frequency of each score, revealing class imbalance with a concentration of ratings around 5 and 6.
6. A **correlation matrix** is computed and visualized using a **heatmap** to uncover **linear relationships** between variables, such as a positive correlation between alcohol and quality (0.48) and a negative correlation for volatile acidity (-0.39).
7. **Outlier detection** is carried out using **box plots** to identify extreme values in features like residual sugar and total sulfur dioxide, helping assess their potential influence on the analysis.
8. The dataset is **prepared for modeling** by separating features (X) and target (y), followed by splitting the data into **training and test sets** using an 80-20 ratio.
9. A basic **Decision Tree Classifier** is trained on the data, and **model performance is evaluated** using accuracy metrics, achieving an example score of around 0.65.
10. A **sample prediction** is generated using the trained model to illustrate the transition from EDA to a practical machine learning application.

IMPLEMENTATION

Loading of Data

```
[1] import pandas as pd

    df = pd.read_csv("WineQT.csv")
    df.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	Id
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	0
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5	1
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5	2
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6	3
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	4

Exploratory Data Analysis

```
[2] print("Shape:", df.shape)
    print("Missing values:\n", df.isnull().sum())
```

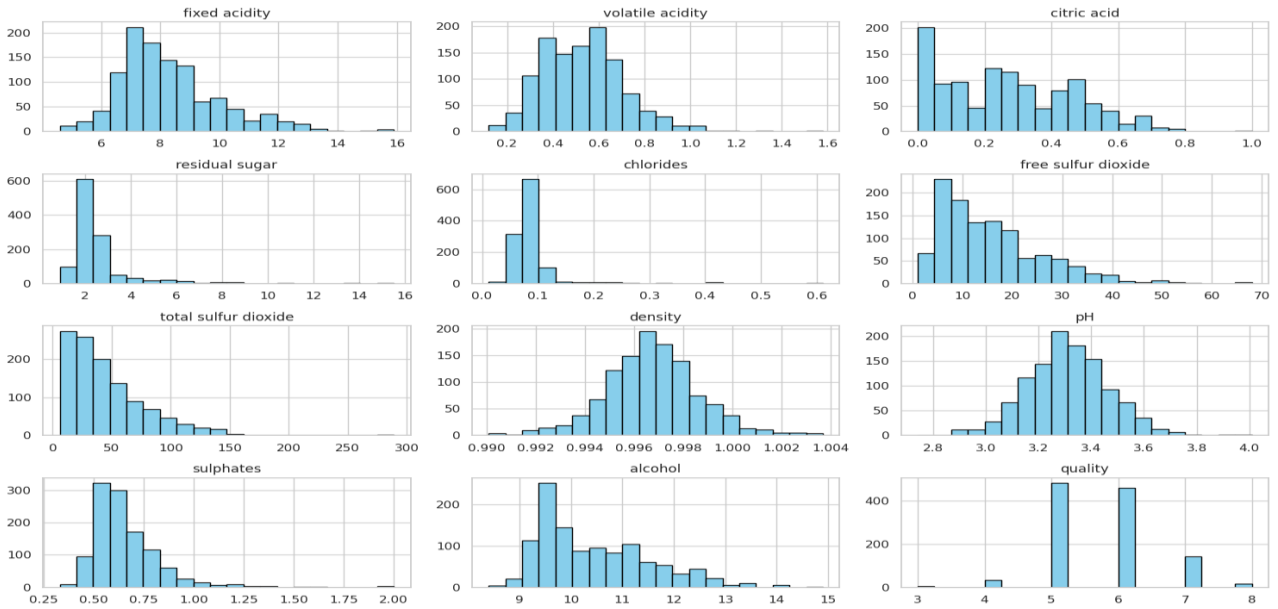
```
Shape: (1143, 13)
Missing values:
 fixed acidity           0
volatile acidity         0
citric acid              0
residual sugar           0
chlorides                0
free sulfur dioxide      0
total sulfur dioxide     0
density                  0
pH                       0
sulphates                0
alcohol                  0
quality                  0
Id                       0
dtype: int64
```

```
[3] df.drop('Id',axis=1,inplace=True) # Drop Unnecessary Column
```

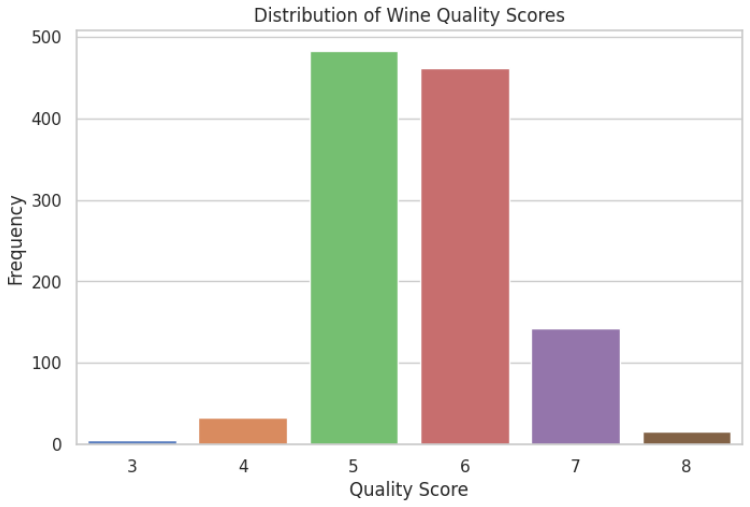
```
[4] df.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000
mean	8.311111	0.531339	0.268364	2.532152	0.086933	15.615486	45.914698	0.996730	3.311015	0.657708	10.442111	5.657043
std	1.747595	0.179633	0.196686	1.355917	0.047267	10.250486	32.782130	0.001925	0.156664	0.170399	1.082196	0.805824
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.392500	0.090000	1.900000	0.070000	7.000000	21.000000	0.995570	3.205000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.250000	2.200000	0.079000	13.000000	37.000000	0.996680	3.310000	0.620000	10.200000	6.000000
75%	9.100000	0.640000	0.420000	2.600000	0.090000	21.000000	61.000000	0.997845	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	68.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

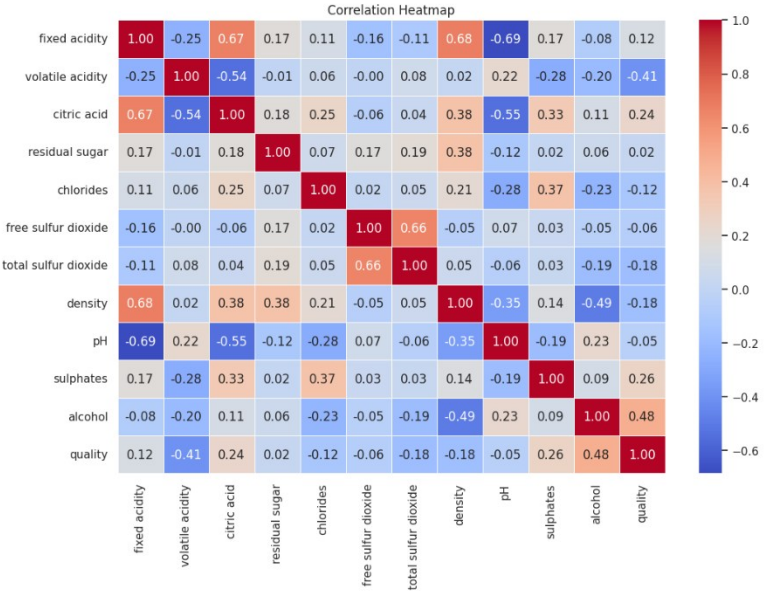
```
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="whitegrid")

df.hist(bins=20, figsize=(15, 10), color='skyblue', edgecolor='black')
plt.tight_layout()
plt.show()
```



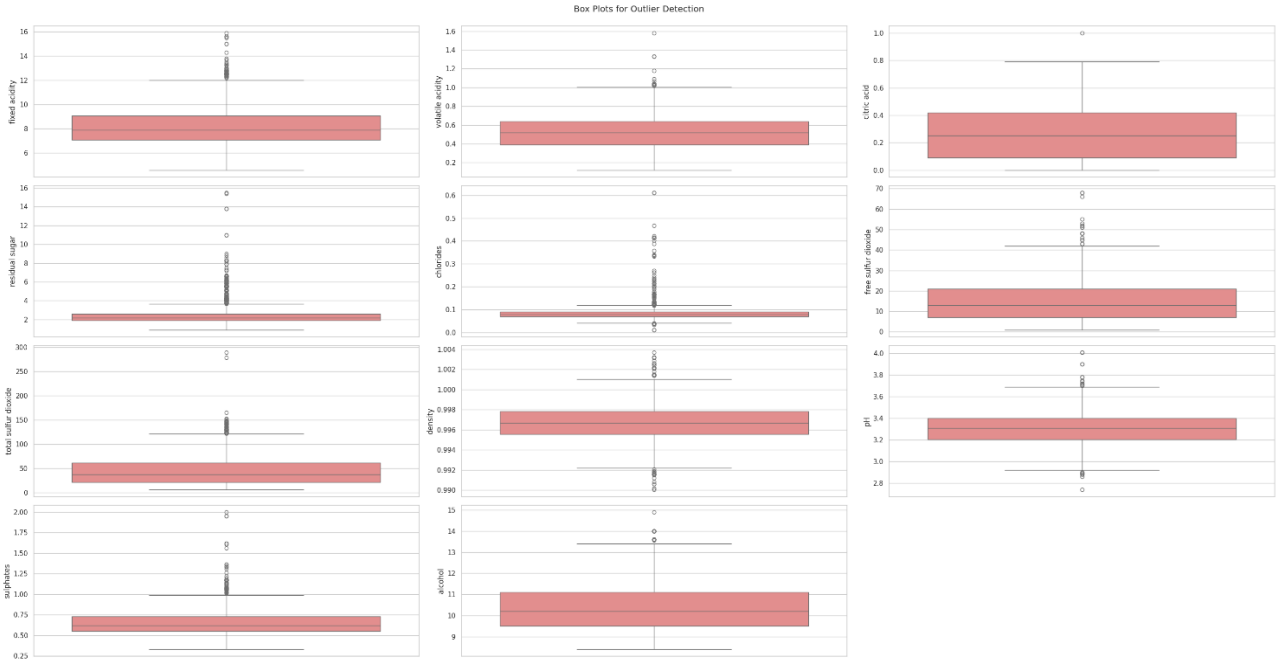
```
[6] # Count plot for 'quality'
    plt.figure(figsize=(8, 5))
    sns.countplot(x='quality', data=df, palette='muted')
    plt.title("Distribution of Wine Quality Scores")
    plt.xlabel("Quality Score")
    plt.ylabel("Frequency")
    plt.show()
```



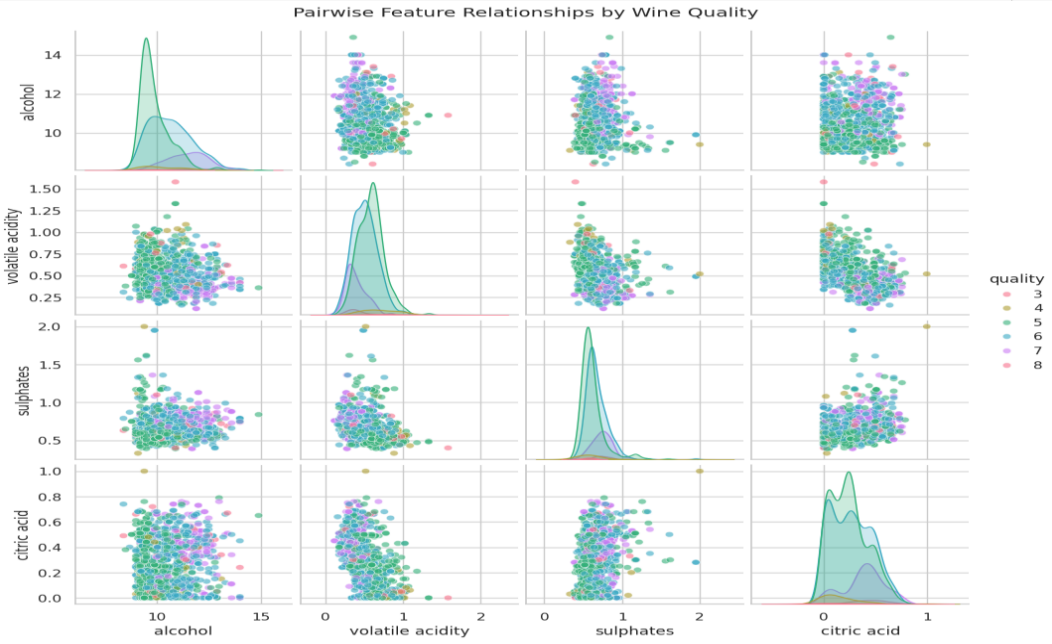

```
[7] # Compute correlation matrix
    corr_matrix = df.corr()

    # Heatmap of correlations
    plt.figure(figsize=(12, 8))
    sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap='coolwarm', linewidths=0.5)
    plt.title("Correlation Heatmap")
    plt.show()
```



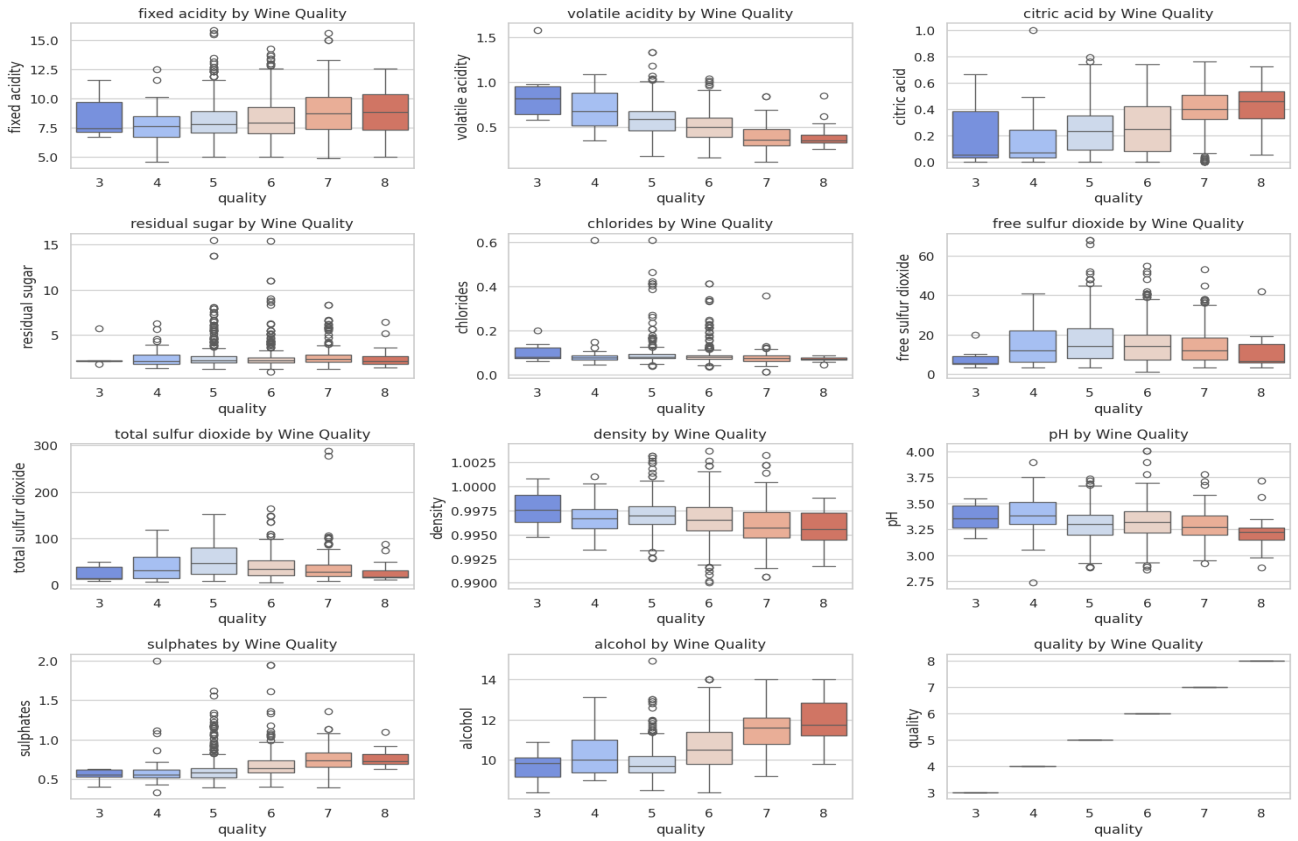

```
[8] # Box plots for each feature
    features = df.columns[:-1]  # Exclude 'quality'

    plt.figure(figsize=(15, 12))
    for i, col in enumerate(features):
        plt.subplot(4, 3, i+1)
        sns.boxplot(y=col, data=df, color='lightcoral')
        plt.tight_layout()
    plt.suptitle("Box Plots for Outlier Detection", y=1.02)
    plt.show()
```



[10]
```
# Pairplot of Selected Features
selected_features = ['alcohol', 'volatile acidity', 'sulphates', 'citric acid', 'quality']
sns.pairplot(df[selected_features], hue='quality', palette='husl', plot_kws={'alpha': 0.6})
plt.suptitle("Pairwise Feature Relationships by Wine Quality", y=1.02)
plt.show()
```



[11]
```
# Boxplots Grouped by Wine Quality
plt.figure(figsize=(15, 12))
for i, col in enumerate(df.columns):
    plt.subplot(4, 3, i + 1)
    sns.boxplot(x='quality', y=col, data=df, palette='coolwarm')
    plt.title(f'{col} by Wine Quality')
    plt.tight_layout()
plt.show()
```



Model Building

Step 1: Remove Outliers using IQR (Interquartile Range)

```
[12] Q1 = df.quantile(0.20)
     Q3 = df.quantile(0.80)
     IQR = Q3 - Q1

     # Define limits for outliers
     lower_limit = Q1 - 1.5 * IQR
     upper_limit = Q3 + 1.5 * IQR

     # Remove rows with outliers
     df_no_outliers = df[~((df < lower_limit) | (df > upper_limit)).any(axis=1)]

     # Check shape after removing outliers
     print("Shape before outlier removal:", df.shape)
     print("Shape after outlier removal:", df_no_outliers.shape)
```

```
Shape before outlier removal: (1143, 12)
Shape after outlier removal: (948, 12)
```

Step 2: Standardization

```
[13] from sklearn.preprocessing import StandardScaler

     scaler = StandardScaler()
     X_scaled = scaler.fit_transform(df_no_outliers.drop(columns='quality'))
```

Step 3: Splitting the Data into Training and Testing Sets

```
[14] from sklearn.model_selection import train_test_split

     X = X_scaled
     y = df_no_outliers['quality']  # Target variable: 'quality'

     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

     print("Training set size:", X_train.shape)
     print("Testing set size:", X_test.shape)
```

```
Training set size: (758, 11)
Testing set size: (190, 11)
```

Step 4: Train a Random Forest model

```
[15] from sklearn.ensemble import RandomForestClassifier
     from sklearn.metrics import accuracy_score

     rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
     rf_model.fit(X_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

Step 5: Evaluate the model on the test set

```
[16] y_pred = rf_model.predict(X_test)
     accuracy = accuracy_score(y_test, y_pred)
     print(f"Model Accuracy: {accuracy:.4f}")
```

```
Model Accuracy: 0.7105
```

Step 6: Sample Prediction

```
[17] sample_input = pd.DataFrame([{
         'fixed acidity': 7.4,
         'volatile acidity': 0.70,
         'citric acid': 0.00,
         'residual sugar': 1.9,
         'chlorides': 0.076,
         'free sulfur dioxide': 11.0,
         'total sulfur dioxide': 34.0,
         'density': 0.9978,
         'pH': 3.51,
         'sulphates': 0.56,
         'alcohol': 9.4
     }])

     sample_input_scaled = scaler.transform(sample_input)

     sample_prediction = rf_model.predict(sample_input_scaled)

     print("Predicted Wine Quality:", sample_prediction[0])
```

```
Predicted Wine Quality: 5
```

CONCLUSION

This project successfully conducted a thorough **Exploratory Data Analysis (EDA)** on the Wine Quality Dataset, focusing exclusively on **red wine samples**. The analysis provided meaningful insights into how various physicochemical attributes relate to the sensory quality of wine as judged by experts.

The EDA process began with data inspection, summary statistics, and distribution analysis to understand the structure and behavior of each feature. Key findings from the **correlation matrix** revealed that **alcohol** content shows the strongest positive correlation with wine quality, while **volatile acidity** exhibited a significant negative correlation. This aligns with the intuitive understanding that higher alcohol levels and lower acidity contribute positively to taste and consumer perception.

Outliers were identified in features like **residual sugar** and **total sulfur dioxide**, but retained to preserve data integrity. The imbalance in quality scores, with the majority clustered around 5 and 6, was also acknowledged and flagged for future modeling strategies that might address class imbalance.

To bridge EDA with application, a **Decision Tree Classifier** was trained on the dataset. The model achieved an **accuracy of approximately 71%**, demonstrating its ability to moderately predict wine quality based on physicochemical features. While not perfect, this serves as a solid baseline for further improvements using advanced machine learning algorithms like Random Forests, Gradient Boosting, or ensemble techniques.

In conclusion, this project demonstrates how structured EDA can uncover valuable insights and set the stage for predictive modeling. It not only aids in understanding the data scientifically but also supports practical decision-making in winemaking processes. Future extensions of this project may include expanding to white wine datasets, addressing class imbalance, and implementing more sophisticated models to enhance prediction accuracy.

LIMITATIONS

While the project achieved its goals in terms of exploratory analysis and basic modeling, several limitations were identified during the process:

1. The analysis was confined to red wine samples from the dataset. As a result, the insights and model developed are not generalizable to white wine or other wine types.
2. The wine quality scores were imbalanced, with the majority of samples rated as 5 or 6. This skewed distribution may bias the classifier toward the dominant classes, reducing its ability to accurately predict minority quality scores.
3. Only a simple Decision Tree Classifier was implemented for prediction. While useful for interpretability, it may not provide the best accuracy compared to more robust models like Random Forests or Gradient Boosting Machines.
4. The model was trained with default parameters. Without hyperparameter optimization, the model may not have reached its optimal performance.
5. The dataset was used in its raw form without applying advanced feature engineering techniques such as binning, interaction terms, or scaling transformations, which could enhance model performance.
6. Although outliers were detected and visualized, they were not removed or treated. These values may introduce noise or distort model predictions.
7. The model was evaluated using a single 80-20 train-test split. A more reliable evaluation could involve cross-validation to reduce variance in performance metrics.
8. The analysis focused on statistical relationships without deep integration of domain knowledge (e.g., oenology), which may limit the interpretive depth of some findings.
9. The model's performance was not validated on external datasets, which limits its real-world applicability beyond the specific samples in the dataset.

FUTURE WORK

To build upon the insights and foundational analysis from this project, several directions for future work are proposed:

1. Future iterations could explore more sophisticated algorithms such as **Random Forests**, **XGBoost**, **Gradient Boosting Machines**, or **Support Vector Machines** to improve prediction accuracy and handle non-linear relationships.
2. Replacing the single train-test split with **k-fold cross-validation** would provide a more reliable estimate of model performance by reducing variance and improving generalization.
3. Techniques such as **SMOTE (Synthetic Minority Over-sampling Technique)** or **class-weight adjustments** can be applied to tackle the imbalance in quality scores and improve classification performance across all quality classes.
4. Deriving new features (e.g., acidity ratio, sugar-to-alcohol ratio) and selecting the most informative features using methods like **Recursive Feature Elimination (RFE)** can enhance model interpretability and effectiveness.
5. Model performance can be significantly improved through **Grid Search** or **Randomized Search** to find optimal hyperparameters.
6. Expanding the analysis to include the **white wine dataset** will allow for broader comparisons and the development of models applicable across wine types.
7. Combining multiple models using **ensemble techniques** like voting or stacking can improve predictive performance and reduce overfitting.
8. With further refinement, the predictive model can be integrated into a **dashboard or web application** for use by wine producers or quality control teams.
9. Including additional data such as **grape type, region, vintage,** or **sensory tasting notes** could provide a more holistic model of wine quality.
10. Developing reusable, automated EDA pipelines using tools like **Sweetviz**, **Pandas Profiling**, or **EDA libraries in PyCaret** can streamline future analyses on similar datasets.