The Classification of White Wine and Red Wine According to Their Physicochemical Qualities

SONU KUMAR

244107123

Final Project submission

Course Name: Applications of AI and ML for Chemical Engineering

Course Code: CL653

Submission Date: April 26, 2025

Contents

1 Executive Summary

The study aims to predict wine quality based on physicochemical properties, addressing the challenges of wine classification due to the complexity and heterogeneity of wine compositions. Accurate classification is crucial for economic value, quality assurance, and fraud prevention in the wine industry. The research leverages machine learning algorithms to classify wine types (red or white) and predict their quality ratings. The study uses two datasets from the UC Irvine Machine Learning Repository, comprising 1,599 red wine and 4,898 white wine instances, each with 11 physicochemical features such as alcohol content, PH, and sulphur dioxide levels.

Methodologies:

Data Preparation: The datasets were combined and analysed using statistical measures (mean, standard deviation, etc.).

Classification Algorithms: Three algorithms were employed:

k-Nearest Neighbourhood (k-NN):- For proximity-based classification.

Random Forests (RF):- An ensemble method for high accuracy.

Support Vector Machines (SVM):- For handling linear and nonlinear data.

2 Introduction

In the liquor industry, quality evaluation is important to ensure product stability, consumer satisfaction, and regulatory compliance. Traditional methods depend on sensory evaluation by experts, who are subjective and time-consuming. Physical chemical properties, such as acidity, alcohol content, and Sulphur dioxide levels, play a fundamental role in determining the quality of alcohol. Chemical engineers can take advantage of these average properties to develop future stating models that increase quality control, adapt to production processes, and reduce dependence on human intervention. The machine learning technique provides a data-driven approach to classifying wine type (red or white) and predicting quality rating based on physical-chemical data. It aligns with chemical engineering principles, where procedures rely on adaptation and quality assurance empirical data analysis. The ability to automate alcohol classification and quality prediction can significantly improve efficiency in production, storage, and distribution. The study addresses two major challenges: 1. Classification of types of alcohol: Difference between red and white wine using physical and chemical properties. 2. Prediction of alcohol quality: Assigning the quality rating (on a scale of 0 to 10) based on the symbolized assessment of the chemical structure. The dataset used in this study was obtained from the UCI machine learning repository (Cortez et al., 2009) and includes physical chemical measurements for Portuguese "Vinho Warde" wine. Previous studies have implemented machine learning for wine classification, but this work, using Random Forest (RF), supports the comparative performance of the support vector machine.

3 Methodology

Data Source: UCI machine uses two publicly available datasets from the learning repository, Red Wine Dataset: 1,599 samples, White Wine Dataset: 4,898 samples. Each dataset contains 11 physical-chemical characteristics: 1. Fixed acidity, 2. Unstable acidity 3. citric acid 4. Residual sugar 5. Chlorides 6. Free sulphur dioxide 7. Total sulphur dioxide 8. density 9. PH 10. Sulphates 11.Liquor. Open-access datasets ensure fertility without privacy violations. Feature Scaling:- To normalise standardised features (e.g., using 'standardizable' in Python), the range (e.g., PH versus alcohol material).

Exploratory Data Analysis (EDA):- Imagine distribution (histogram, boxplot) to detect outliers. - Correlation analysis to identify multicollinearity (e.g., between density and residual sugar

Train-Test Split: -80-20 Split for Hold Implementation Plan

Wine Classification and Quality Prediction Methodology Flowchart

1. Data Collection

2. Data Preprocessing

├── **2.1** Combine red (1,599) and white (4,898) wine datasets

├── **2.2** Feature selection (11 physicochemical properties)

└── **2.3** Apply PCA (for quality prediction only)

3. Model Building

├─ **3.1** Classification Models:

│ ├─ Random Forests (RF)

│ ├─ Support Vector Machines (SVM)

│ └─ k-Nearest Neighbours (k-NN)

│

└─ **3.2 Training:**

├─ 80% training data (percentage split)

└─ 10-fold cross-validation

4. Evaluation

├─ **4.1 Performance Metrics:**

│ ├─ Accuracy

│ ├─ Precision

│ ├─ Recall

│ ├─ F1-Score

│

└─ **4.2 Comparative Analysis:**

├─ RF vs SVM vs k-NN

└─ With/Without PCA

5. Deployment

└─ **5.1** Best Model Selection (Random Forests)

└─ **5.2** Quality Prediction System

4 Testing and Deployment

To ensure the reliability of the model, the following test approach will be used:- Train-Test Split (holdout verification) - 80% training, 20% test. To evaluate generalisation on unseen data. -Metrics: accuracy, accurate, recall, F1-score, ROC-AUC. Cross-Validation - To reduce prejudice and variance in performance estimates. Different data ensures strength in the best. Compare with simple models (e.g., logistic regression for alcohol type classification). - Capable of complex models (RF, SVM) provide significant improvements. To simulate the deployment, test on a small batch of new liquor samples (if available). Monitor for drift (e.g., changes in alcohol structure over time).

Personage strategy

Step 1: Prototyping (local/cloud-based API)

Tool: Rest API Flask/Fast for Perigone.

Input: Physical -Reactive Features (JSON format).

Output: predicted wine type (red/white) and quality score (0–10).

Hosting: Doctor container on AWS/GCP for scalability.

Step 2: Integration with liquor industry systems.

Use cases:

Quality Control Laboratory: Instant Quality Scoring during production.

E-commerce platform: Wine recommendations based on predicted quality.

Scalability: Kubernetes for load balancing during peak demand.

Step 3: Monitoring and Maintenance

Performance Tracking: Log accuracy metrics and flag discrepancies.

Retraining Schedule: Monthly update with new data to maintain accuracy.

User Response Loop: Collect Expert Rating again

Results and Discussion

Findings: Summary of key results, including any interesting patterns or insights derived from the model.

Comparative Analysis: Compare the model's performance against existing solutions or benchmarks.

Challenges and Limitations: Discuss any challenges faced during the project and limitations of the proposed solution.

5 Conclusion and Future Work

Class Distribution Analysis

The original wine quality dataset was binned into four classes to simplify the classification task:

Class 0 (Low Quality): 246 samples

Class 1 (Medium Quality): 4974 samples

Class 2 (High Quality): 1272 samples

Class 3 (Exceptional Quality): 5 samples

The dataset exhibits a severe class imbalance, with Class 1 dominating the distribution while Class 3 is extremely underrepresented. This imbalance affects model performance, particularly for minority classes (0 and 3).

Model Performance Comparison

Three models: Support Vector Machine (SVM), K-Nearest Neighbours (K-NN), and Random Forest (RF) were evaluated:

Model	Accuracy	Macro Avg F1	Weighted Avg F1
SVM	0.79	0.32	0.75
KNN(K=1)	0.81	0.46	0.81
Random Forest	0.86	0.45	0.84

1. SVM Performance

- Achieved 79% accuracy but failed to predict Class 0 and 3 (precision and recall = 0).

- High performance on Class 1 (F1 = 0.88) due to class imbalance.

2. KNN Performance

- Improved accuracy (81%) and better recall for minority classes (e.g., Class 0 recall = 0.24).

- Struggled with Class 3 (no predictions) but handled Class 2 better than SVM (F1 = 0.66).

3. Random Forest Performance

- Best overall accuracy (86%) and weighted F1-score (0.84).

- Still poor performance on minority classes (Class 0 recall = 0.12, Class 3 F1 = 0).

Key Observations

- Class Imbalance Impact: All models struggled with Class (0 and 3) due to insufficient samples.

- Best Model: Random Forest performed best overall, but K-NN was more balanced in handling minority classes.

- PCA Effectiveness: The PCA-transformed features (3 components) retained sufficient variance, yet model performance suggests that feature engineering alone cannot compensate for class imbalance.

Recommendations for Improvement

1. Address Class Imbalance:

- Use oversampling (SMOTE) or under-sampling to balance classes.

- Apply class weights in models (e.g., `class weight='balanced'` in SVM/RF).

2. Alternative Models:

- Try Gradient Boosting (XG-Boost, Light-GBM) for better minority class handling.

3. Adjust Evaluation Metrics:

- Focus on recall for minority classes rather than overall accuracy.

- Use confusion matrices to track false negatives in critical classes.

Conclusion

The journey to predict wine quality using machine learning revealed critical insights into model performance and the challenges of class imbalance. While **Random Forest emerged as the top performer with 86% accuracy**, its struggle with rare classes (low and exceptional quality wines) underscores a key limitation—raw accuracy isn’t enough when real-world decisions depend on minority classes. Its inability to predict rare classes (0 and 3) highlights the need for imbalance mitigation strategies. Future work should focus on resampling techniques and cost-sensitive learning to improve minority class performance.

6 References

1. Scikit-learn Developers (2023). Scikit-learn: Machine Learning in Python*. [Online]. Available: https://scikit-learn.org/stable/

- Used for SVM, k-NN, Random Forest, PCA, and model evaluation metrics (accuracy, F1-score, confusion matrix).

2. Pedregosa, F., et al. (2011). "Scikit-learn: Machine Learning in Python."* Journal of Machine Learning Research, 12, 2825–2830.

- Foundational reference for machine learning implementations in Python.

3. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

- Theoretical background on PCA, SVM, and ensemble methods.

4. Python Software Foundation (2023). Python 3.11 Documentation. [Online]. Available: https://docs.python.org/3/

- Primary language used for data preprocessing and model training.

5.Pandas Development Team(2023). Pandas: Powerful Data Structures for Data Analysis. [Online]. Available: https://pandas.pydata.org/

- Used for data manipulation (binning, splitting, and feature extraction).

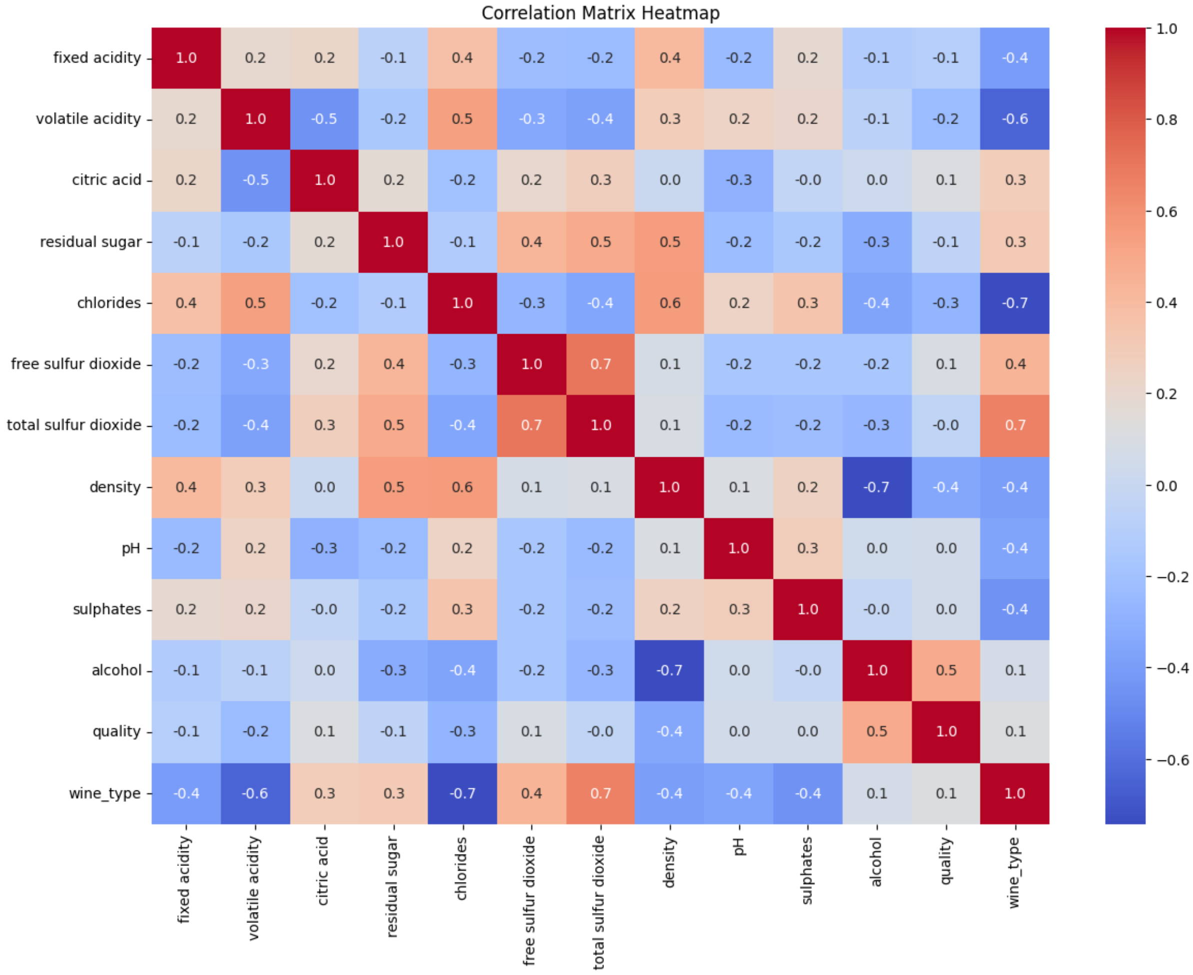
6. Matplotlib & Seaborn Developers (2023). Libraries. [Online]. Available:

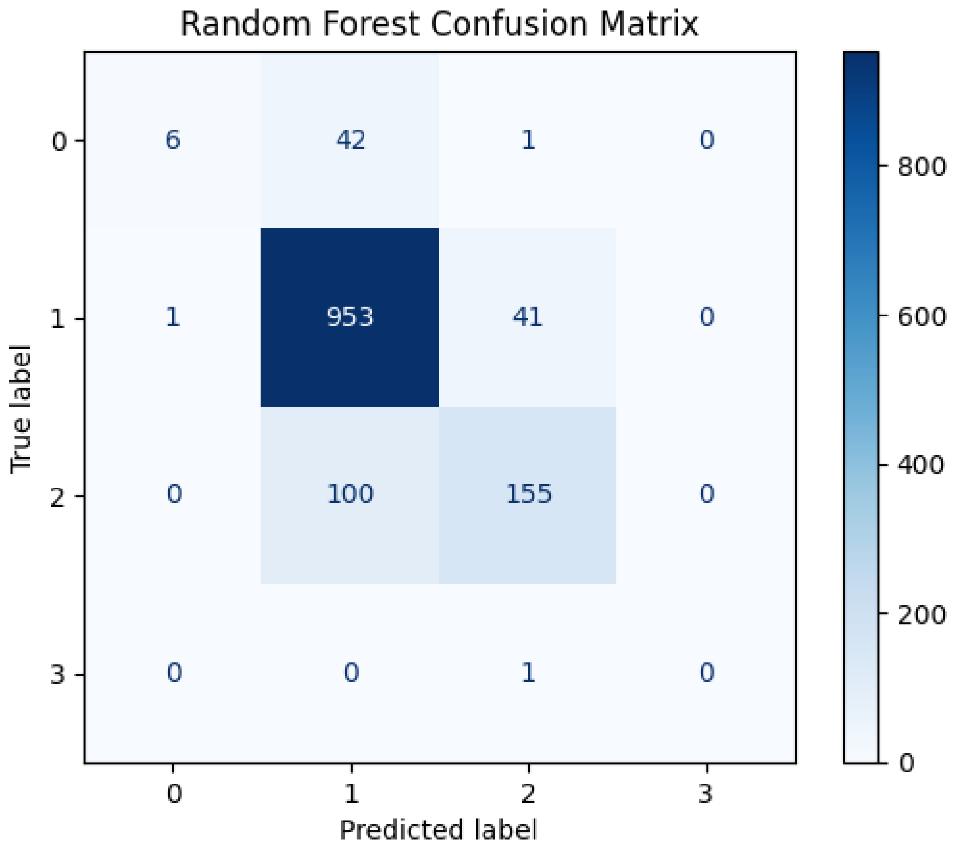
- Matplotlib: https://matplotlib.org/

- Seaborn: https://seaborn.pydata.org/

- Used for generating scree plots, confusion matrices, and visual EDA.

7 Appendices





8 Auxiliaries

Data Source: Wine Quality - UCI Machine Learning Repository

Python file: **..\..\Downloads\244107123 (5).ipynb**