Algorithms for AI 3

Task 3 Report

Group: 4
Group Members:
Abdullah Zeeshan, Rohan Raj, Angel Lopez Hortelano

Wine Quality Classification Report

Introduction

The objective of this project is to classify wines into three quality categories based on their physicochemical properties using machine learning models. The classes are:

- **Class 0**: Bad
- **Class 1**: Medium
- **Class 2**: Good

The analysis involves exploring the dataset, training two classifiers (Gaussian Process and KNN), handling class imbalance using SMOTE, and evaluating model performance.

Exploratory Data Analysis

Feature Distribution

- Histograms reveal near-normal distribution in most features.
- **Alcohol**, **volatile acidity**, and **sulphates** are better class separators.
- Features like **pH** and **residual sugar** show significant overlap.

Key Observations

- **Class 1 (Medium)** dominates the dataset.
- High class overlap makes accurate classification of **Class 0 (Bad)** challenging.

Figures:

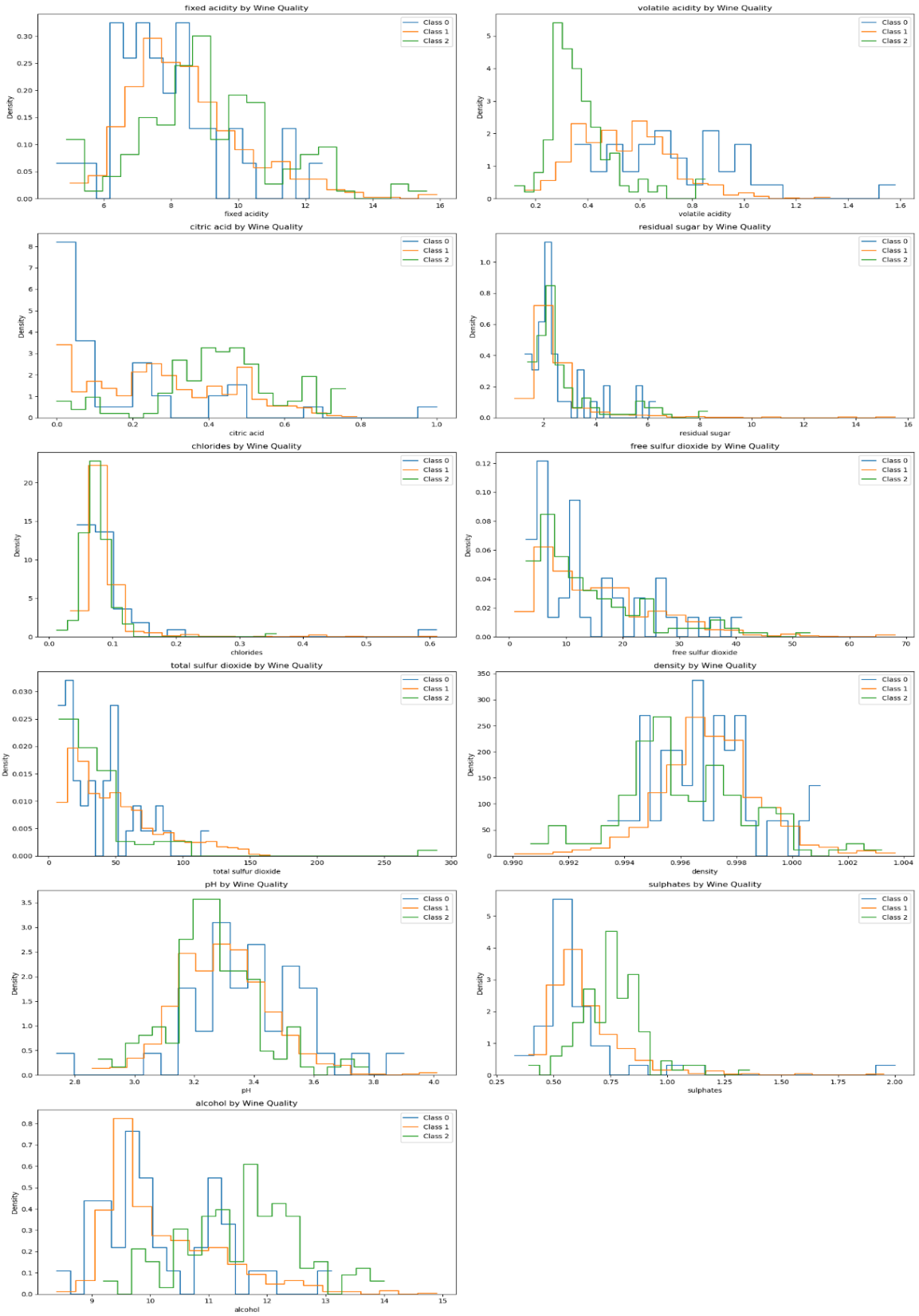

Fig 1.1 Histograms by feature and class showing distribution overlaps.

Task 1. Gaussian Process Classifier (One-vs-Rest)

Without SMOTE

- **Accuracy:** ~82–87%
- **Class 0 Performance:** Very poor (near-zero precision and recall).
- High class imbalance leads to strong bias toward **Class 1**.

Figures:

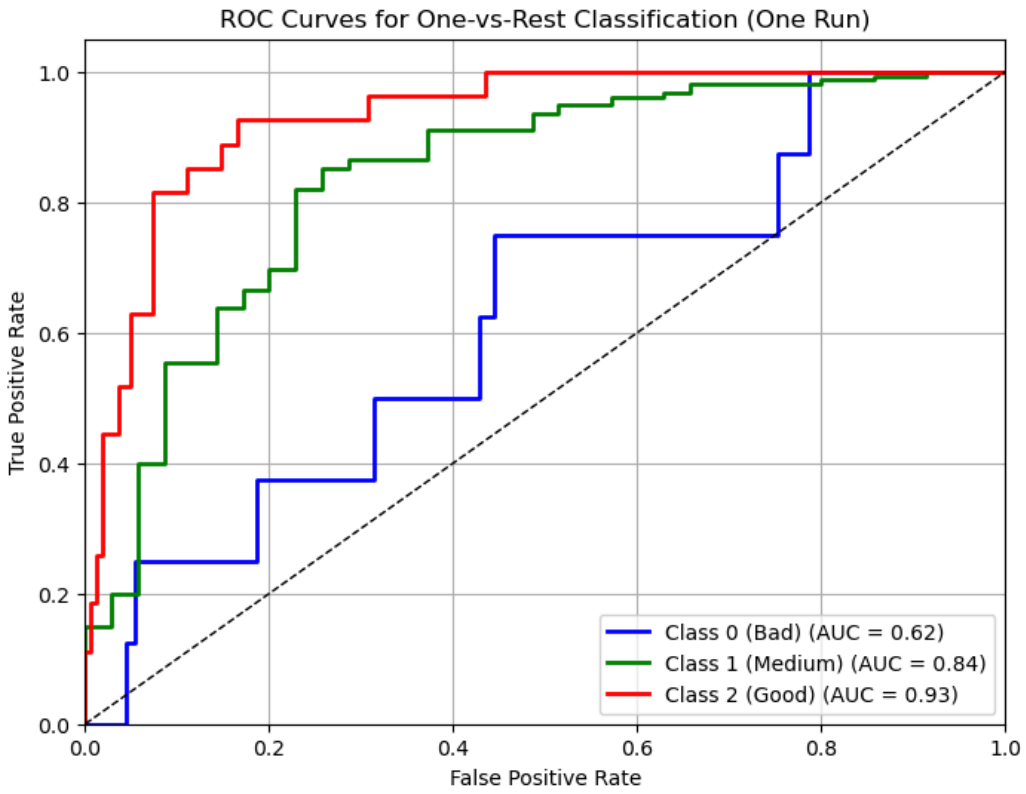

Fig 2.1 ROC Curve before SMOTE.

With SMOTE

- **Accuracy Improvement:** Up to ~88%
- Balanced training improves performance slightly but **Class 0** still underperforms.
- ROC AUC remains low for **Class 0**, better for **Class 1** and **Class 2**.

Figures:

Fig 2.2 Histogram of Features Classified by Quality (After SMOTE).

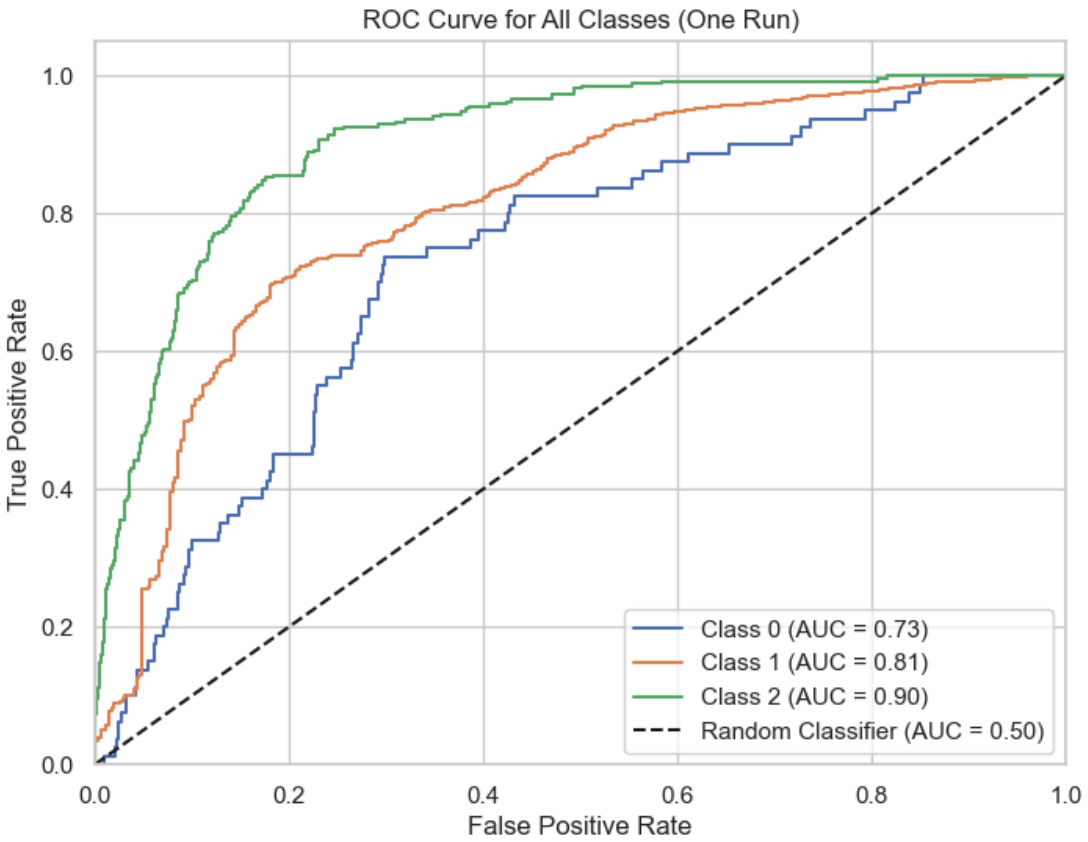

Fig 2.3 ROC Curve after SMOTE.

Task 2. K-Nearest Neighbors (KNN)

Optimal K Selection

- Error rate analysis shows **K=1** gives lowest misclassification rate.

Figure:

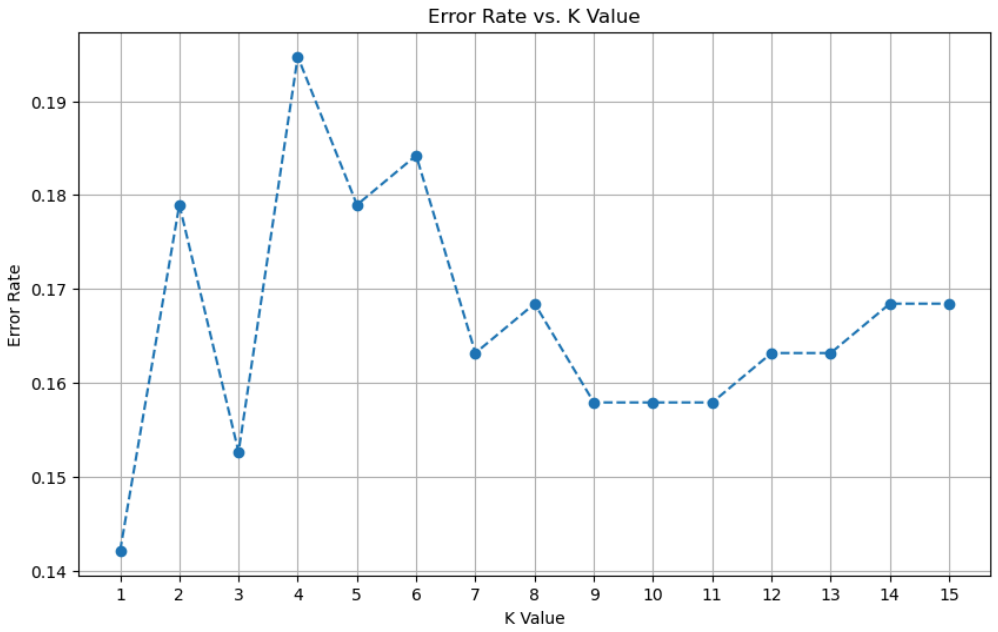

Fig 3.1 Plot of K values vs. error rate.

Performance Summary

- **Accuracy:** ~84%
- **Stability:** Low variance in accuracy across multiple runs.
- **Class 1** again dominates predictions.

ROC and Confusion Matrix

- Best AUC for **Class 1**, lowest for **Class 0**.
- Consistent with Gaussian Process classifier.

Figure:

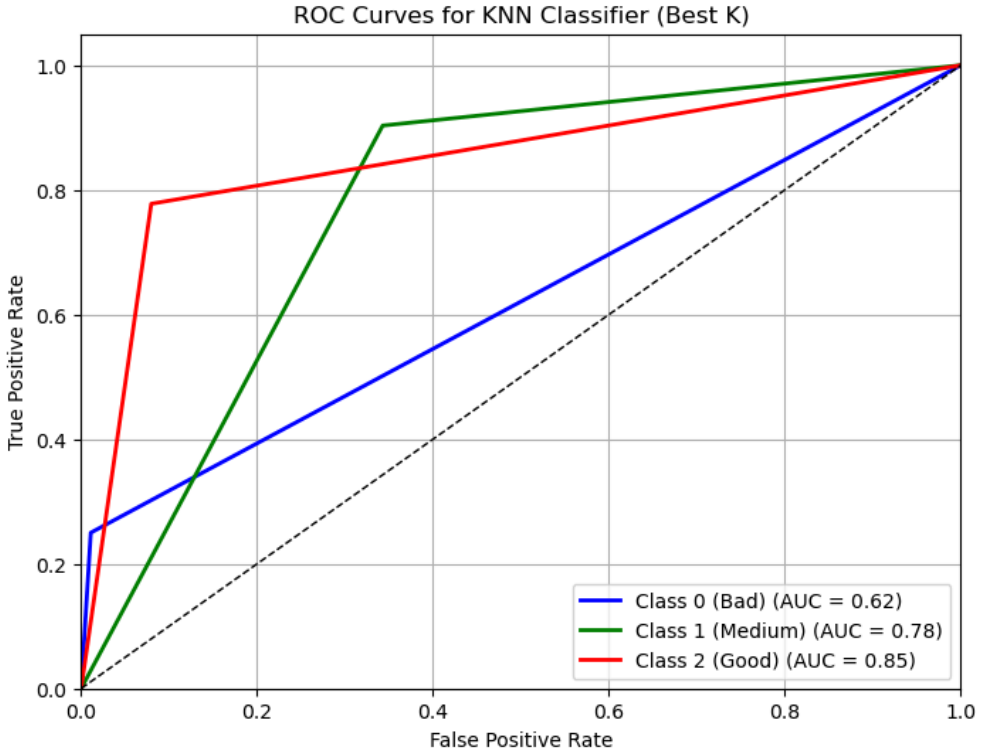

Fig 3.2 ROC curve for K=1 model.

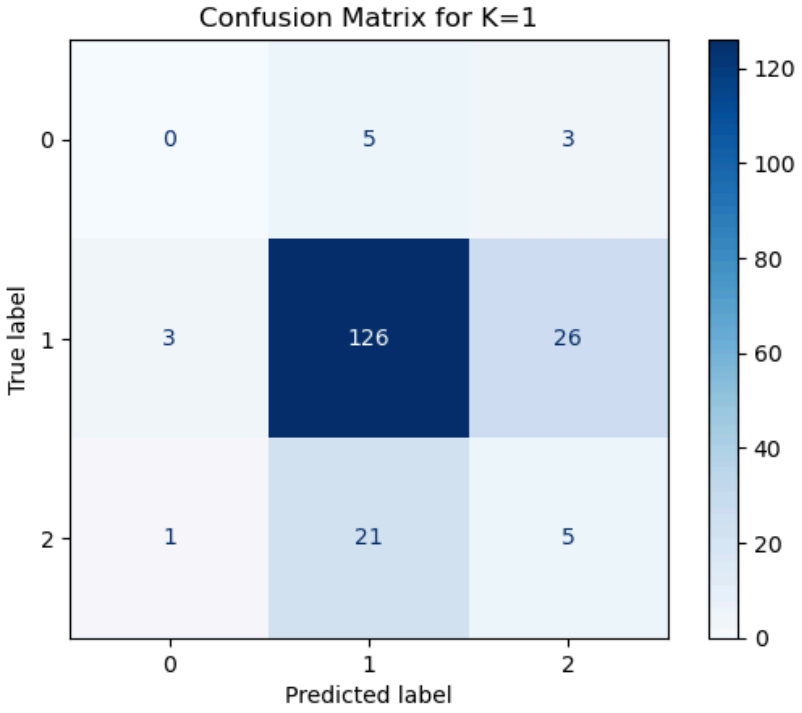

Fig 3.3 Confusion matrix highlighting misclassifications.

Evaluation and Discussion

Findings

- Both models perform well overall but **struggle significantly with Class 0**.
- SMOTE helped slightly but was not sufficient alone.
- KNN is faster and slightly more stable than Gaussian Process.
- Gaussian Process is more sensitive to class imbalance.

Possible Improvements

Data-Level Enhancements

- Apply **undersampling**, **ensemble SMOTE**, or **class weighting**.
- Conduct **feature selection** or apply **PCA** to improve model focus.

Model-Level Strategies

- Experiment with **Random Forest**, **XGBoost**, or **LightGBM**.
- Combine models using **ensemble learning** (bagging/boosting).
- Perform **hyperparameter tuning** (GridSearchCV).

Evaluation Tactics

- Use **Stratified K-Fold Cross-Validation** for more reliable results.
- Focus on **F1-score, precision, recall per class**, not just accuracy.

Summary

This project demonstrated that both Gaussian Process and KNN classifiers can reasonably predict wine quality, with average accuracies between **84% and 88%**. However, due to class

imbalance, **bad wines (Class 0)** are poorly detected. SMOTE improves model fairness but doesn't fully resolve the issue. Future work should focus on better balancing techniques, more powerful classifiers, and comprehensive evaluation metrics to improve minority class detection.

Comments on the Lab Activity

This lab activity was an interesting and valuable learning experience for our group. As it was our first attempt at approaching a task in this structured way, we found it both engaging and somewhat challenging. Implementing SMOTE (Synthetic Minority Oversampling Technique) for the first time was particularly insightful, as it allowed us to understand the importance of handling imbalanced datasets in machine learning. The activity was highly relevant to the course content and helped us apply theoretical knowledge in a practical setting. While the task did require careful thought and collaboration, it ultimately strengthened our understanding of model development and evaluation. For future improvements, a bit more guidance on interpreting model performance metrics or comparing different balancing techniques could make the experience even more enriching.