Algorithms for AI 3
Group # 4

Task 3 Report

Group Members:

Abdullah Zeeshan, Rohan Raj, Angel Lopez Hortelano

Wine Quality Classification Report

Introduction

The aim of this lab task was to study and evaluate the performance of various classifiers for a multiclass classification problem using the Wine Quality Dataset. We used three wine quality categories ("bad", "medium", and "good") encoded as 0, 1, and 2 respectively. The focus was on implementing and assessing the following techniques:

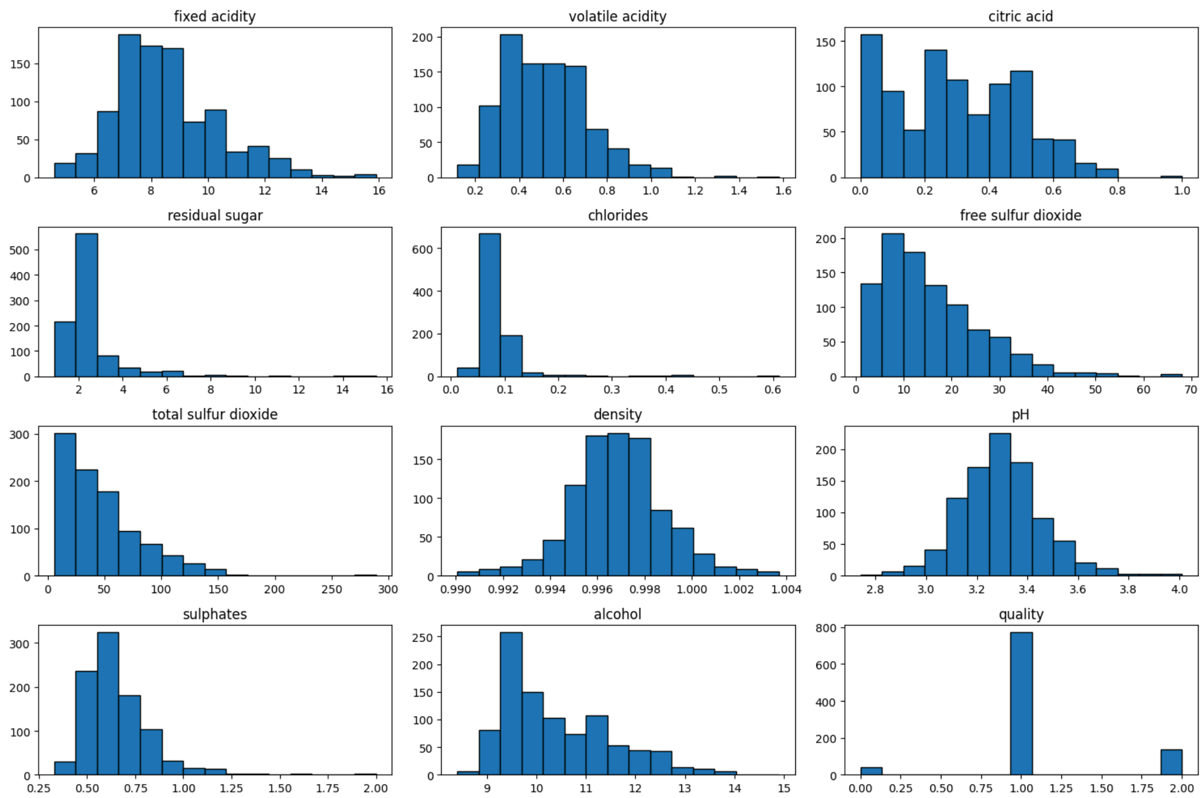
1. One-vs-Rest classification with and without SMOTE.
2. k-Nearest Neighbors classification with and without SMOTE.

Performance was evaluated using metrics like accuracy, precision, and ROC curves.

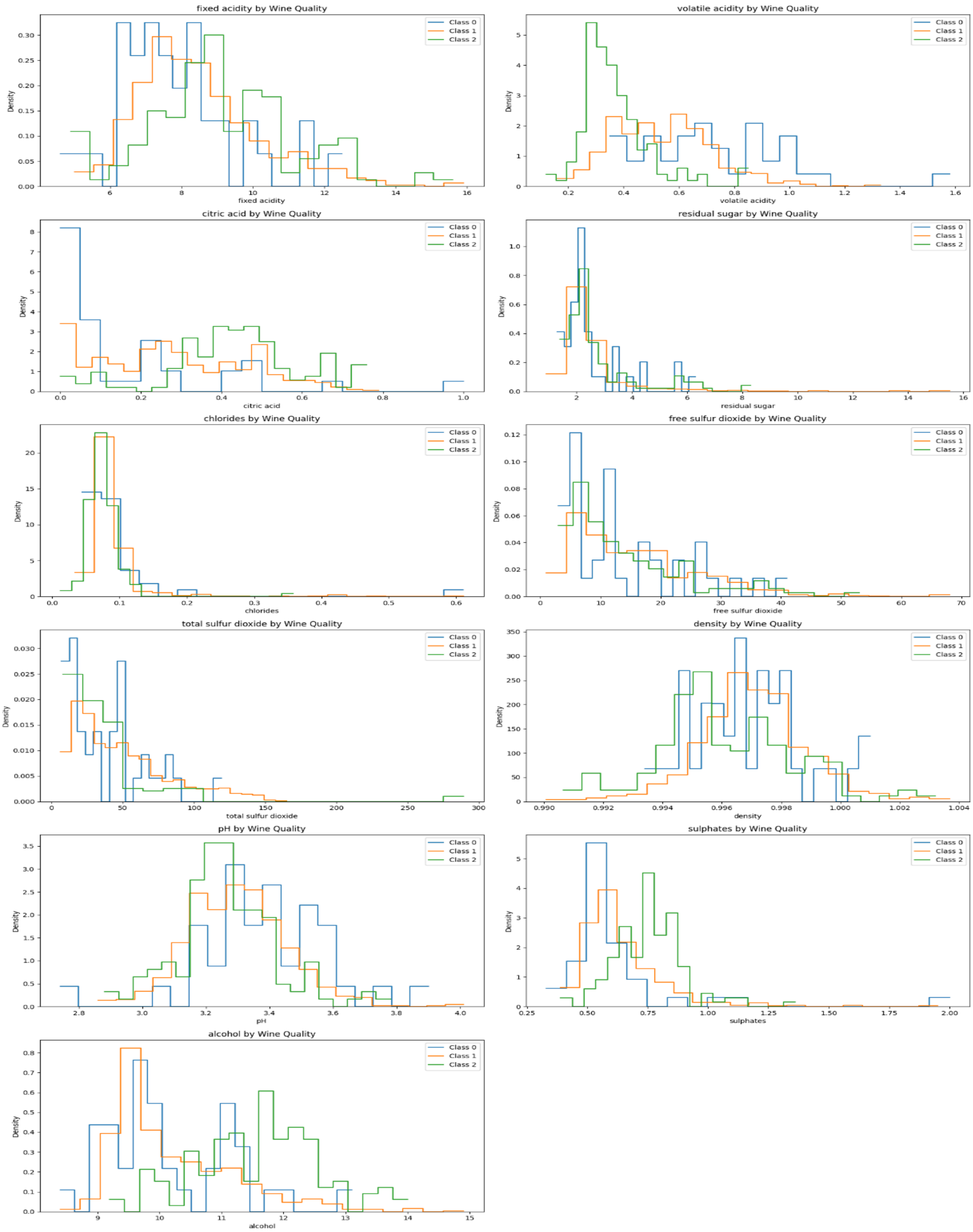
Task 1 – One-vs-Rest Classifier

Task 1.1 – Without SMOTE

We started by visualizing the original dataset. Histograms of each feature and the target label were plotted to understand data distribution.



We noticed that the dataset was imbalanced, with most wine samples classified as “medium” (label 1), followed by “good” (label 2), and the least as “bad” (label 0).



We trained a One-vs-Rest model using a `GaussianProcessClassifier` as the base estimator. This was done over **10 separate runs**, each using a different random split between training and test datasets. For every run, we evaluated the classification report on the test set.

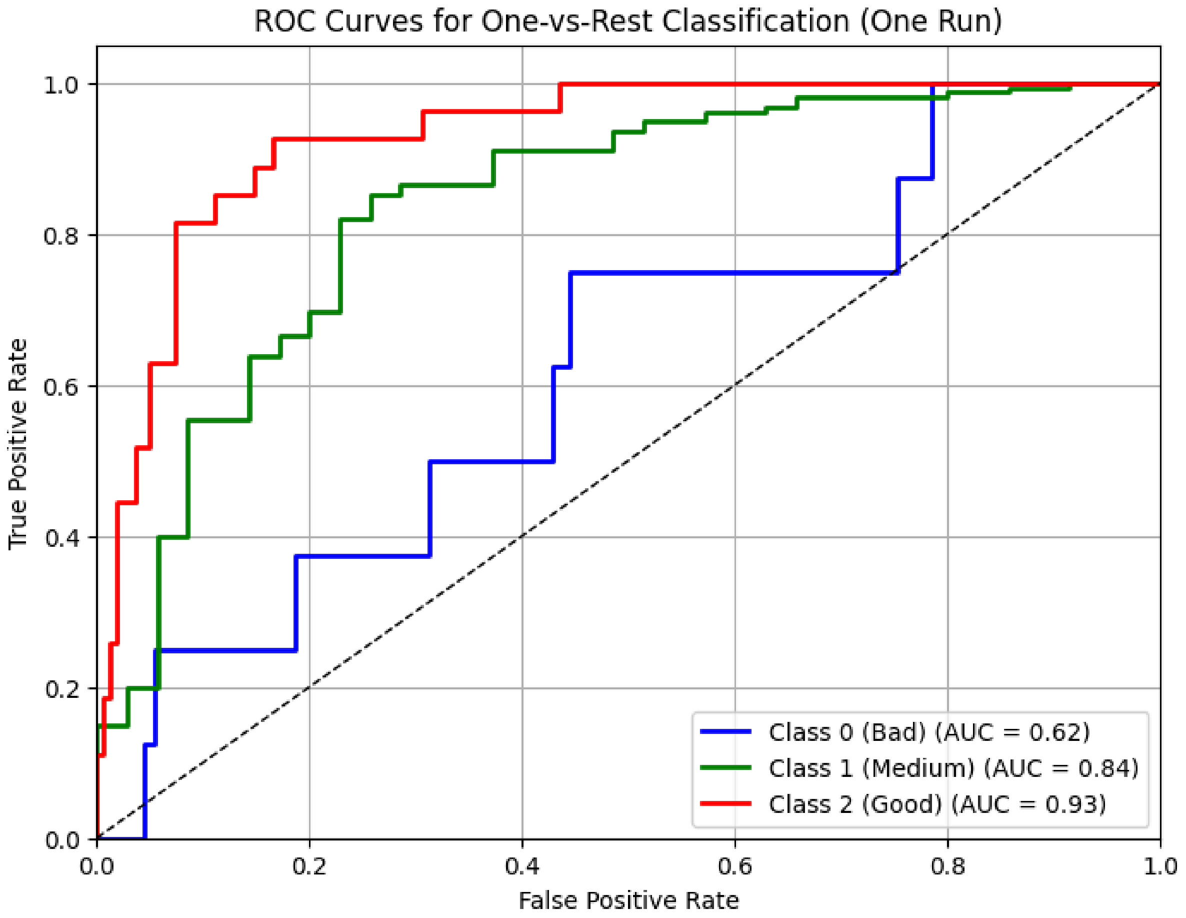
Here is a summary of the **accuracy scores** for the 10 runs (without SMOTE):

```
[0.75, 0.66, 0.63, 0.64, 0.67, 0.63, 0.67, 0.63, 0.65, 0.64]
```

The **mean accuracy** was:

```
Mean Accuracy: 0.657
Standard Deviation: 0.034
```

We also plotted the **ROC curves** for one representative run, which showed relatively lower area under the curve (AUC) for the minority class, consistent with the class imbalance issue.

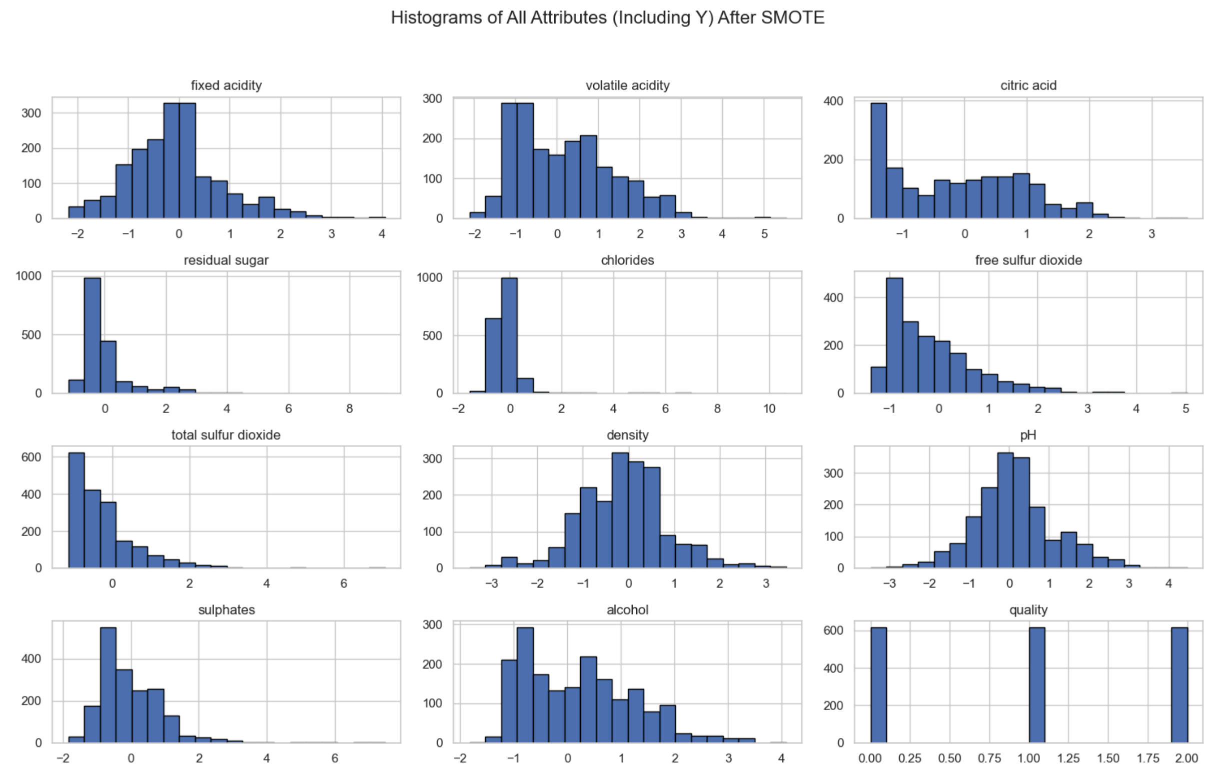


Observations

From these runs, it was evident that the imbalance in class labels had a negative effect on classification performance, particularly for class 0 ("bad" wines). The ROC curves confirmed that classes with fewer samples were harder to classify correctly. These results led us to consider data balancing methods.

Task 1.2 – With SMOTE

We applied the **SMOTE (Synthetic Minority Over-sampling Technique)** method on the training dataset to balance the number of samples in each class before fitting the classifier. After applying SMOTE, we re-plotted the histograms and confirmed that all classes were balanced in the training set.



Again, we repeated the classification process over 10 runs using the same `OneVsRestClassifier`.

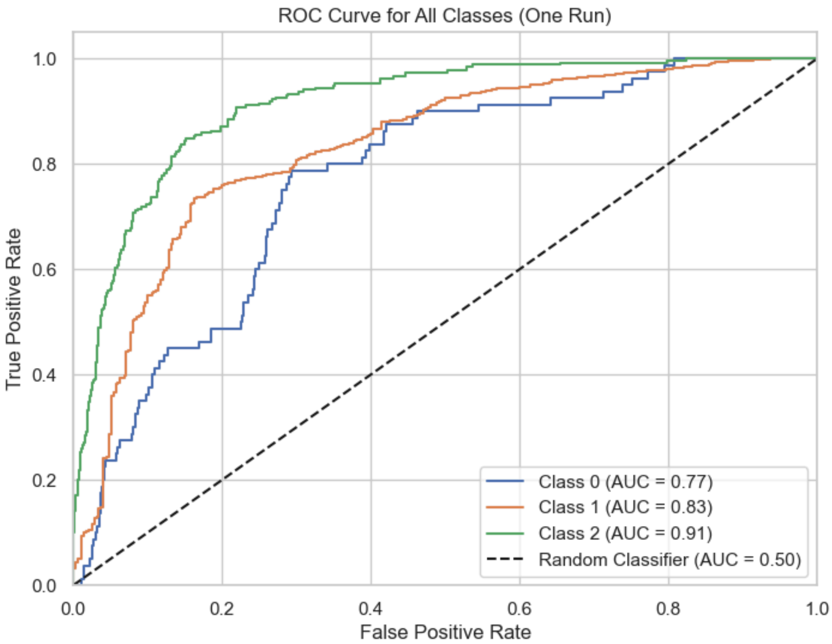
Here are the **accuracy scores** after applying SMOTE:

```
[0.64, 0.66, 0.64, 0.67, 0.67, 0.69, 0.67, 0.67, 0.68, 0.68]
```

The **mean accuracy** and standard deviation were:

```
Mean Accuracy: 0.667
Standard Deviation: 0.014
```

We also plotted the **ROC curves** again. This time, we observed a slight improvement in the curves for class 0 and class 2, suggesting better generalization due to balanced training.



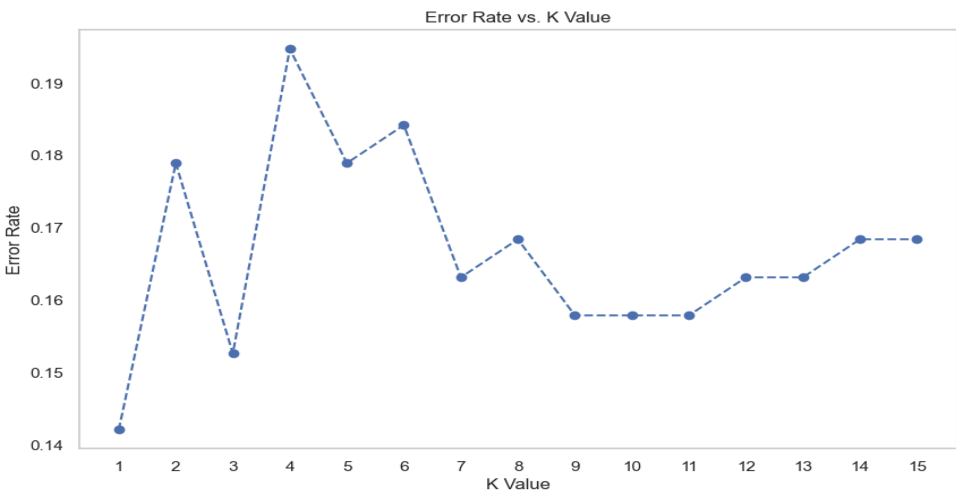
Observations

While the improvement in accuracy was moderate, the standard deviation decreased significantly, indicating more stable performance across runs. The ROC curves for minority classes improved, confirming the effectiveness of SMOTE in balancing the class distributions.

Task 2 – k-Nearest Neighbors Classifier

Task 2.1 – Without SMOTE

We trained **15 k-NN classifiers** with k values ranging from 1 to 15. We then plotted the mean error (1 - accuracy) against each k value to identify the optimal k.



From the plot, we found that **k = 1** yielded the lowest mean error and was thus chosen for further testing. This was evident in the plot where the mean error was at its minimum at k = 1.

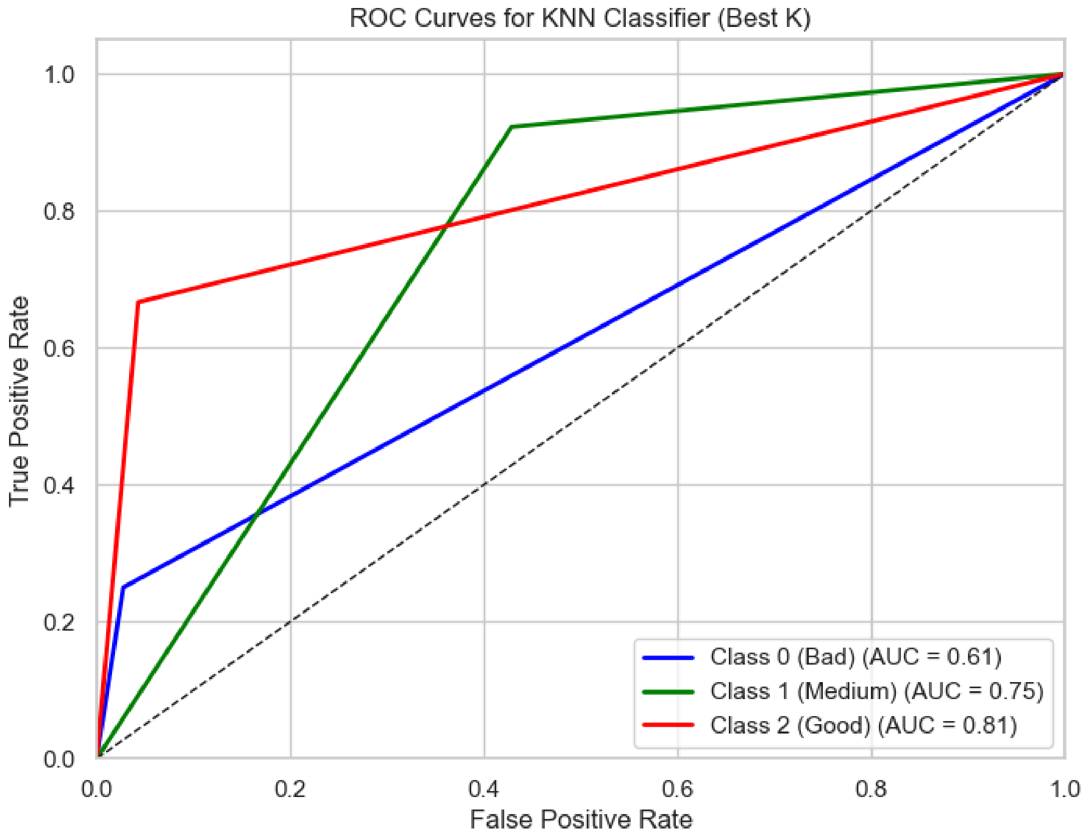
We then performed **five classification runs** using k = 1. Here were the **accuracy scores**:

```
[0.66, 0.68, 0.70, 0.69, 0.68]
```

The **mean accuracy** and standard deviation were:

```
Mean Accuracy: 0.682
Standard Deviation: 0.013
```

We also plotted the **ROC curves** for one of the best-performing runs. These curves showed reasonably strong classification performance across all three classes.

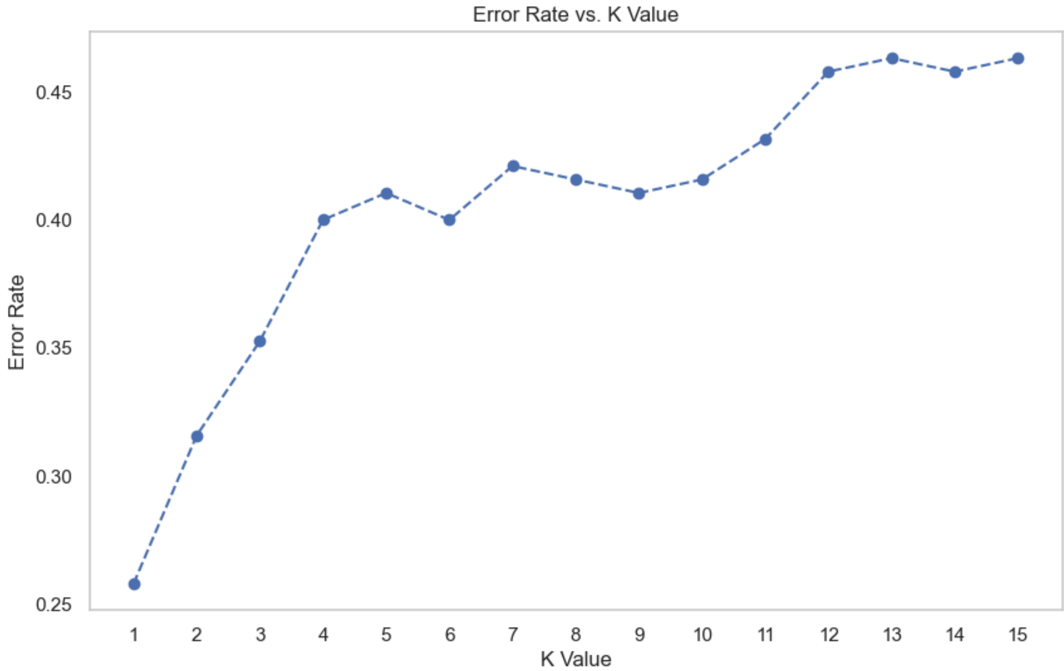


Observations

The k-NN classifier performed slightly better than the One-vs-Rest classifier, especially in the balanced prediction across all three classes. However, class imbalance still affected the sensitivity of predictions for classes 0 and 2.

Task 2.2 – With SMOTE

We applied SMOTE to the training data before fitting the k-NN classifiers. Again, we trained 15 models with different `k` values and identified **k = 1** as the optimal one.



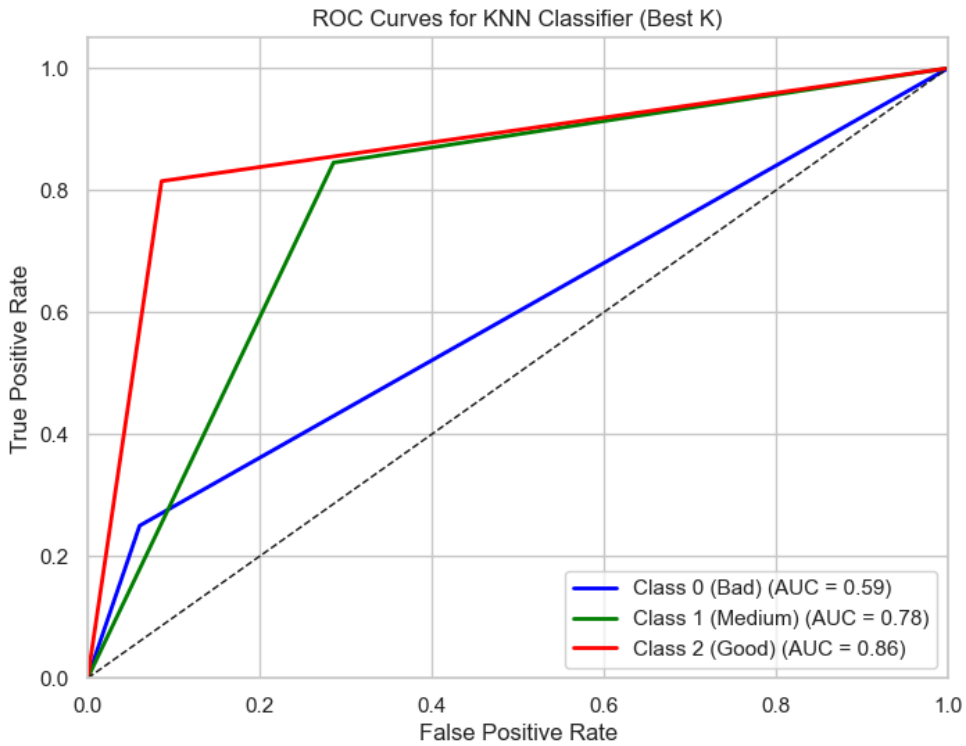
After applying SMOTE, the five classification runs yielded the following **accuracy scores**:

```
[0.68, 0.67, 0.70, 0.69, 0.70]
```

The **mean accuracy** and standard deviation were:

```
Mean Accuracy: 0.688
Standard Deviation: 0.012
```

The **ROC** curves again showed slight improvements for the minority classes, indicating that SMOTE helped balance classification performance.



Observations

SMOTE helped reduce performance variance across classes and improved the sensitivity of the classifier to underrepresented classes. The improvement over the non-SMOTE version was minor but consistent.

Summary of Findings

- **One-vs-Rest without SMOTE** had an average accuracy of 65.7% with high variance and poor performance on minority classes.
- **One-vs-Rest with SMOTE** improved accuracy slightly to 66.7%, but more importantly, it reduced variance and improved ROC for minority classes.
- **k-NN without SMOTE** performed better with an accuracy of 68.2% and showed balanced classification across classes.
- **k-NN with SMOTE** further improved to 68.8% and showed improved ROC curves for underrepresented classes.

SMOTE proved to be beneficial in all settings, especially for the One-vs-Rest classifier which struggled more with imbalanced data.

Comments on the Lab Activity

This lab was our first attempt at applying these models in a practical, iterative setting. It was **challenging**, especially managing the class imbalance and interpreting ROC curves across multiple classes. However, the task was also **interesting**, particularly in experimenting with SMOTE, which was new to us.

The relevance of this lab to real-world machine learning problems was clear. Handling imbalanced data and selecting appropriate models are important parts of classification tasks.

Possible Improvements

- Having built-in plotting for ROC curves for multiclass in scikit-learn would be useful.

- Adding more runs with cross-validation could give a more comprehensive performance measure.

- Future work could test other classifiers or tuning hyperparameters beyond k in k-NN.