جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering & Technology
Electrical & Computer Engineering Department

Machine Learning and Data Science
ENCS5341

First Semester 2023/2024

Assignment #3

Prepared by: Omar Masalmah & Faris Abu Farha

ID: 1200060 1200546

Instructor's Name: Dr. Yazan Abu Farha

Section: 2

Date: 26-01-2024

Abstract

In this project, we aim to perform a predictive task on a real-world problem using machine learning models. We chose a classification task and picked a dataset from Kaggle. As a baseline model, we evaluated a nearest neighbor baseline using a distance of our choice and reported the performance of this baseline using both k=1 and k=3. We then tried to achieve better performance by evaluating two additional models on the task, namely SVC and ExtraTreeClassifier. We discussed and motivated our model selection and commented on why the performance has improved. We tuned at least one hyper-parameter for each model by testing at least 4 different values. Finally, we analyzed the performance of our best model from examining instances in the test set where our model exhibits errors.

Table of Contents

Table of Figures

List of Tables

Introduction

This project is dedicated to tackling a classification task using machine learning models to address a specific real-world challenge. Classification tasks involve predicting the categorical outcome of data points, making them valuable in scenarios such as disease diagnosis, spam detection, or sentiment analysis. The chosen task provides a practical context for evaluating the predictive capabilities of different machine learning models.

For this project, we have opted to explore the performance of three distinct models:

1. **Nearest Neighbor Baseline:** As a starting point, we employ a nearest neighbor algorithm to establish a baseline for comparison. Two variations are evaluated, where the parameter k (number of neighbors) is set to 1 and 3.

In order to pick 2 more classifiers, we will test some popular classifiers on the validation test and see which classifiers are the best for our dataset, and pick best 2 classifiers.

2. **Support Vector Classifier (SVC):** SVC is a powerful model known for its effectiveness in handling complex decision boundaries. Its ability to work well in high-dimensional spaces and capture intricate patterns makes it an interesting candidate for our classification task.
3. **ExtraTreeClassifier:** The ExtraTree Classifier, based on the concept of extremely randomized trees, provides randomized splits during tree construction, making it suitable for diverse feature sets and potentially enhancing predictive performance. Its ability to handle large datasets aligns with the demands of our classification task.

Evaluation Metrics:

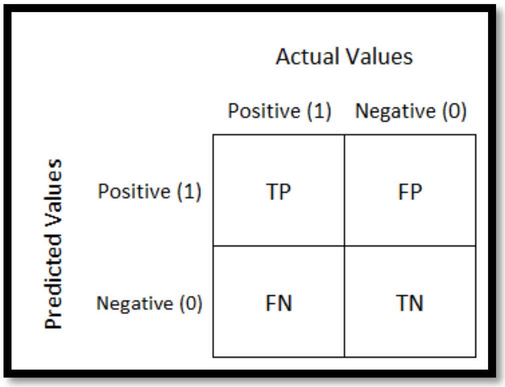
Evaluation metrics are quantitative measures used to assess the performance and effectiveness of a statistical or machine learning model. These metrics provide insights into how well the model is performing and help in comparing different models or algorithms.

When evaluating a machine learning model, it is crucial to assess its predictive ability, generalization capability, and overall quality. Evaluation metrics provide objective criteria to measure these aspects. The choice of evaluation metrics depends on the specific problem domain, the type of data, and the desired outcome.

To gauge the performance of these models, we rely on a set of well-established evaluation metrics tailored for classification tasks:

A. **Accuracy:** This metric measures the overall correctness of predictions, providing a general overview of model performance.

B. **Confusion Matrix:** Is a table that is used to describe the performance of a classification model on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.



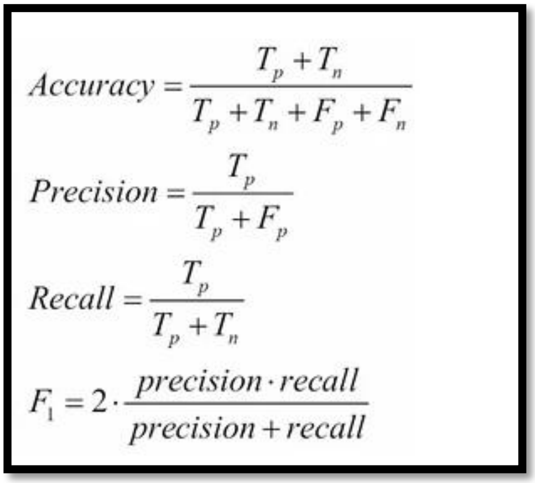
		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 1: Confusion Matrix.

C. **Precision:** Precision quantifies the accuracy of positive predictions, indicating how well the model identifies relevant instances.

D. **Recall:** Recall, or sensitivity, evaluates the ability of the model to capture all relevant instances within the dataset.

E. **F1 Score:** The F1 score, a harmonic mean of precision and recall, offers a balanced assessment of a model's performance.



$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}$$

$$Precision = \frac{T_p}{T_p + F_p}$$

$$Recall = \frac{T_p}{T_p + T_n}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Figure 2: Computing Evaluation Metrics.

These metrics collectively provide a comprehensive understanding of the models' predictive capabilities, allowing us to discern their strengths and weaknesses in the context of the chosen classification task.

Dataset

The dataset used in this project, titled "Apple Quality," is sourced from Kaggle. The dataset comprises various features associated with apples, offering valuable insights into factors influencing apple quality. By the way the data has been scaled and cleaned. Here is a brief overview of the dataset:

Features

1. **Size:** The size of the apple, representing one of the physical dimensions.
2. **Weight:** The weight of the apple, measured in a suitable unit such as grams or ounces.
3. **Sweetness:** A quantitative measure indicating the degree of sweetness in the apple.
4. **Crunchiness:** A quantitative measure representing the texture and crunchiness of the apple.
5. **Juiciness:** A quantitative measure indicating the level of juiciness in the apple.
6. **Ripeness:** A quantitative measure representing the stage of ripeness of the apple.
7. **Acidity:** A quantitative measure representing the acidity level of the apple.
8. **Quality:** The target variable, indicating the overall quality of the apple. This is the label to be predicted and may be categorized into classes such as "good," and "bad."

Statistics

- The dataset comprises a total of 4000 instances, each representing a unique apple.

- **Quantitative Measures:**

Table 1: Dataset Quantitative Measures.

Name	Mean	Standard Deviation
Size	-0.503015	1.928059
Weight	-0.989547	1.602507
Sweetness	-0.470479	1.943441
Crunchiness	0.985478	1.402757
Juiciness	0.512118	1.930286
Ripeness	0.498277	1.874427
Acidity	0.076877	2.110270

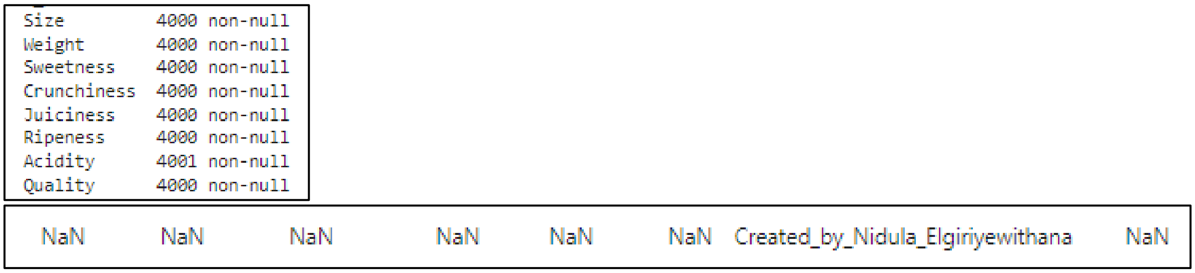
- **Categorical Measures:**
    - **Quality:** Value Counts: 'Good' ➔ 2004 ‖ 'Bad' ➔ 1996.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis will be conducted to gain deeper insights into the dataset's characteristics. Descriptive statistics will be calculated for numerical features, providing measures such as mean and standard deviation. Categorical features, such as color and texture, will be analyzed using frequency distributions and visualizations to discern patterns within the dataset.

The EDA process aims to enhance our understanding of the dataset, identify potential challenges, and inform preprocessing steps as needed before applying machine learning models. This thorough examination of the dataset lays the groundwork for subsequent model evaluations and analyses.

We noticed that every attribute has 4000 values except for the last one which has 4001.



```
Size         4000 non-null
Weight       4000 non-null
Sweetness    4000 non-null
Crunchiness  4000 non-null
Juiciness    4000 non-null
Ripeness     4000 non-null
Acidity      4001 non-null
Quality      4000 non-null
```

```
NaN    NaN    NaN    NaN    NaN    NaN    Created_by_Nidula_Elgiriyewithana    NaN
```

Figure 3: EDA Processing.

We noticed only one row contains Null values, it turned out to be the last row, tis row is for author rights, so we will drop it.

Preprocessing Steps

1. **Handling Missing Values:**

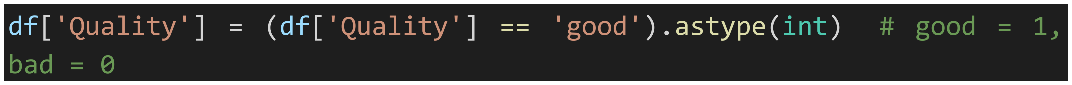
   - Check for any missing values in the dataset through imputation or removal.

2. **Outlier Detection and Removal:**

   - Identify and handle outliers in the quantitative features (if exists) that might adversely affect model performance.

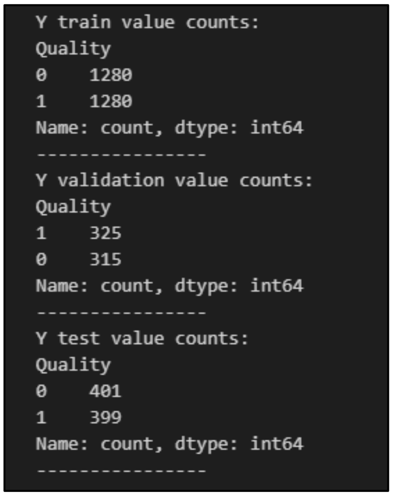
3. **Encoding Categorical Labels:**

   - Encode the **Quality labels** ('good' and 'bad') into numerical values, ensuring compatibility with machine learning algorithms.



```
df['Quality'] = (df['Quality'] == 'good').astype(int)  # good = 1,
bad = 0
```

4. **Splitting the data into train, validation, and test sets:**
   - Splitting the dataset into training, validation, and test sets is a crucial step to assess the performance of machine learning models effectively. This process ensures that models are trained on one subset of the data, validated on another subset to fine-tune parameters, and ultimately tested on a third independent subset to evaluate generalization performance.



```
Y train value counts:
Quality
0    1280
1    1280
Name: count, dtype: int64
----------------
Y validation value counts:
Quality
1    325
0    315
Name: count, dtype: int64
----------------
Y test value counts:
Quality
0    401
1    399
Name: count, dtype: int64
----------------
```

Figure 4: Splitting the data into train, validation, and test sets.

Correlation Analysis

Correlation analysis is a fundamental statistical technique employed to evaluate the strength and direction of the linear relationship between two quantitative variables within a dataset.

I. **Calculate Correlation Coefficients:**

- **Pearson Correlation Coefficient:** Measures the linear relationship between two variables. It ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no linear correlation. This coefficient is well-suited for linear relationships.

II. **Visualize Correlations:**

- Create a correlation matrix heatmap to visually represent the correlation coefficients between pairs of quantitative features.
- Positive correlations are depicted in warmer colors (closer to 1), usually shades of red.
- Negative correlations are depicted in cooler colors (closer to -1), typically shades of blue.
- No correlation is represented in neutral colors, close to 0.

III. **Interpretation:**

- Analyze the correlation matrix to identify significant relationships between quantitative features.
- High positive correlations suggest that as one variable increases, the other tends to increase.
- High negative correlations suggest that as one variable increases, the other tends to decrease.
- Near-zero correlations indicate a lack of a linear relationship.

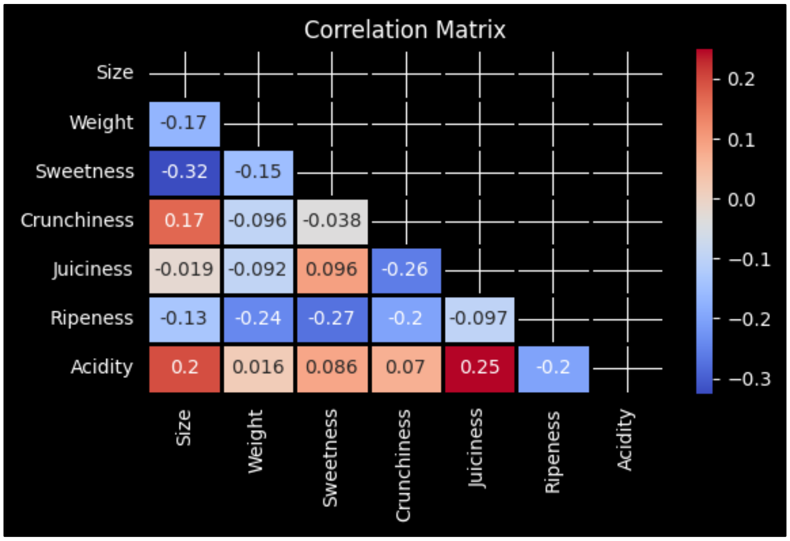


Figure 5: Correlation Matrix.

We can see that there are no strong correlations between the attributes.

Experiments and Results

1. Baseline Model - Nearest Neighbor

As a starting point, a Nearest Neighbor algorithm was employed to establish a baseline for comparison. Two variations were evaluated: k=1 and k=3. The performance metrics, such as accuracy, precision, recall, and F1 score, were computed for each variation.

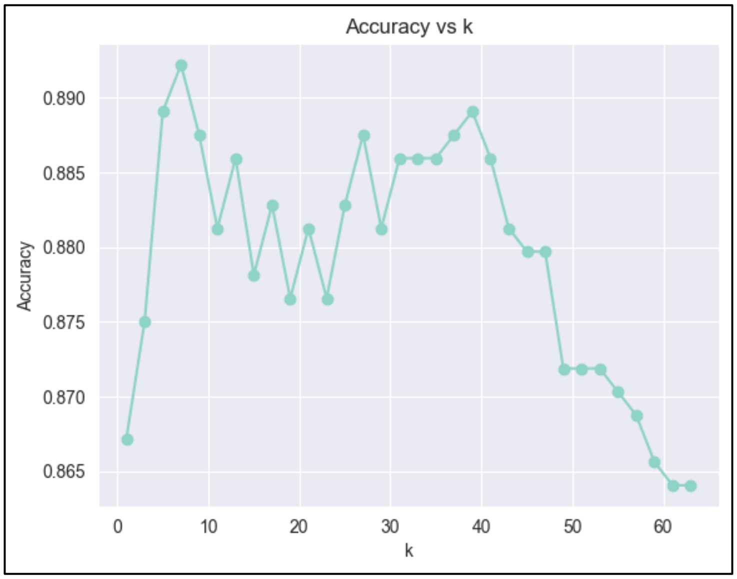


Figure 6: Baseline Model.

- **Results**

✸ **K = 1:**

Confusion Matrix: [352 49]
[40 359]

Accuracy: **0.88875**

Classification Report:

Table 2: K=1 Classification Report.

Class	Precision	Recall	F1-score
0	0.90	0.88	0.89
1	0.88	0.90	0.89

✸ **K = 3:**

Confusion Matrix: [355 46]
[43 356]

Accuracy: **0.88875**

Classification Report:

Table 3: K=3 Classification Report.

Class	Precision	Recall	F1-score
0	0.89	0.89	0.89
1	0.89	0.89	0.89

We notices from Baseline Model that the best K Nearest Neighbor was 7.

✸ **K = 7:**

Confusion Matrix: [360 41]
[39 360]

Accuracy: **0.9**

Classification Report:

Table 4: K=7 Classification Report.

Class	Precision	Recall	F1-score
0	0.90	0.90	0.90
1	0.90	0.90	0.90

2. Support Vector Classifier (SVC)

The Support Vector Classifier (SVC) was chosen for its effectiveness in handling complex decision boundaries. Various hyper-parameters, such as C and kernel type, were explored through grid search. The model was evaluated using cross-validation.

- **Hyper-parameter Selection**
  - Grid search explored C values and different kernel types.
  - After testing different values for C, we found that **C=100** is the best value.
- **Results**

Confusion Matrix: [371 30]
[37 362]

Accuracy: **0.91625**

Classification Report:

Table 5: SVC Classification Report.

Class	Precision	Recall	F1-score
0	0.91	0.93	0.92
1	0.92	0.91	0.92

3. Extra Trees Classifier

The ExtraTree Classifier, known for its efficiency and robustness. Hyper-parameter tuning involved varying parameters like the number of trees, maximum depth, and feature splits.

- **Hyper-parameter Selection**
  - Parameters tuned include [50, 100, 200, 500, 1000, 1500, 2100, 3000]
  - We found that the best parameter is 1500.
- **Results**

Confusion Matrix: [358 43]
[36 363]

Accuracy: **0.90125**

Classification Report:

Table 6: Extra Trees Classification Report.

Class	Precision	Recall	F1-score
0	0.91	0.89	0.90
1	0.89	0.91	0.90

Analysis

I. Model Selection

Support Vector Classifier (SVC) was identified as the best model based on a comprehensive evaluation using metrics such as accuracy, precision, recall, and F1 score.

II. Performance Metrics

- The **accuracy** of the SVC on the test set was **0.91625**, highlighting its overall correctness in predicting Apple classifier.
- **Precision**, indicating the model's ability to correctly identify 'Good' quality apple, achieved **0.92**.
- **Recall**, representing the model's ability to capture all 'Good' quality apple, reached **0.91**.
- The **F1 score**, balancing precision and recall, demonstrated a robust **0.92**.

III. ROC Curve and AUC

- ROC it is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the ‘signal’ from the ‘noise.’ In other words, it shows the performance of a classification model at all classification thresholds.

- AUC It is the measure of the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve.

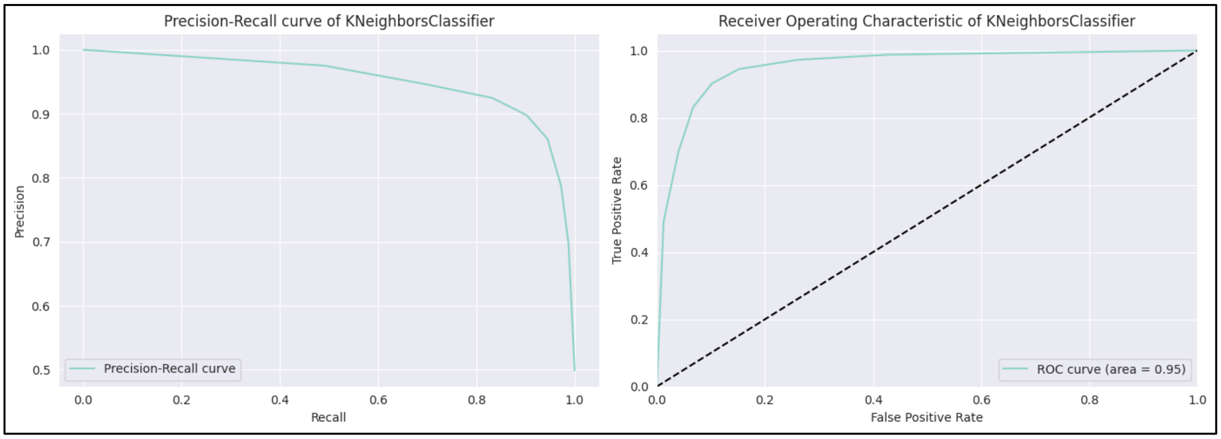


Figure 7: KNN AUC & ROC Curves.

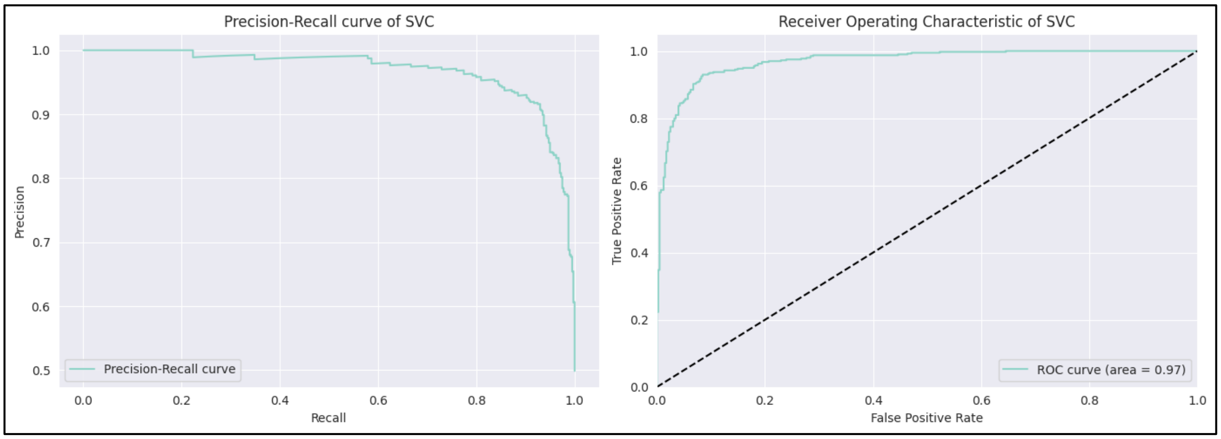


Figure 8: SVC AUC & ROC Curves.

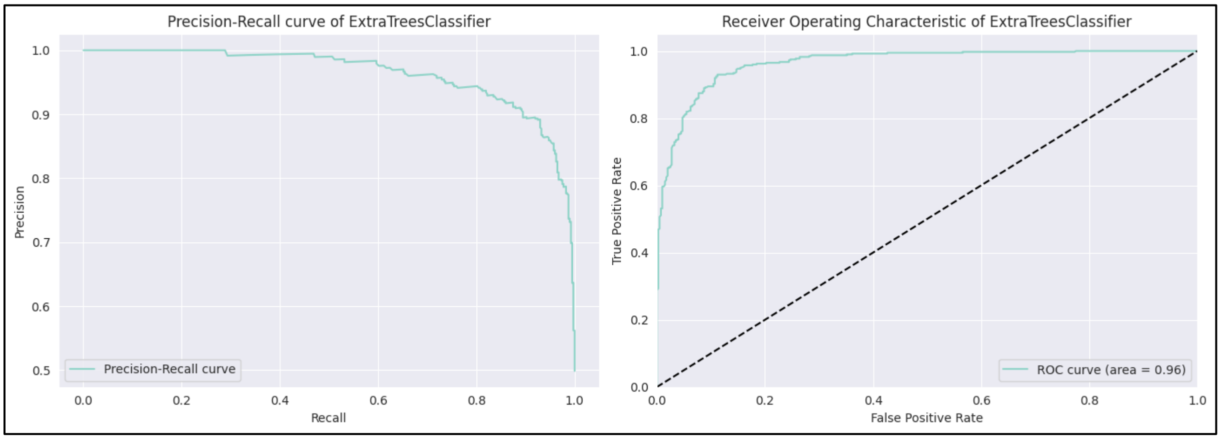


Figure 9: ExtraTrees AUC & ROC Curves.

The complete examination of ROC curves and AUC values gives an improved comprehension of the models' discriminating skills. The Support Vector Classifier's excellent performance, as seen by its steep ROC curve and high AUC, makes it the preferred model for predicting apple quality. This comparison not only helps with model selection, but it also leads to a better understanding of the models' actions in the context of the classification problem.

IV.Error Analysis

The error analysis conducted on the Support Vector Classifier (SVC) model for apple quality classification revealed insightful patterns and characteristics in instances where misclassifications occurred.

Feature Diversity in Misclassifications:

Observation: Misclassifications were diverse across Size, Weight, Sweetness, Crunchiness, Juiciness, Ripeness, and Acidity.

Implication: No single feature overwhelmingly contributed to errors, suggesting a balanced impact across various characteristics.

Statistical Summary:

Observation: Statistical summary provided mean, standard deviation, and range values for Size, Weight, Sweetness, Crunchiness, Juiciness, Ripeness, and Acidity.

Implication: Variability in feature values among misclassified instances, indicating a wide range of characteristics in apples leading to errors.

Distribution of Errors:

Observation: Errors were evenly distributed between 'Good' and 'Bad' quality apples.

Implication: Lack of bias towards a specific class implies a need for overall model improvement rather than specific class-focused enhancements.

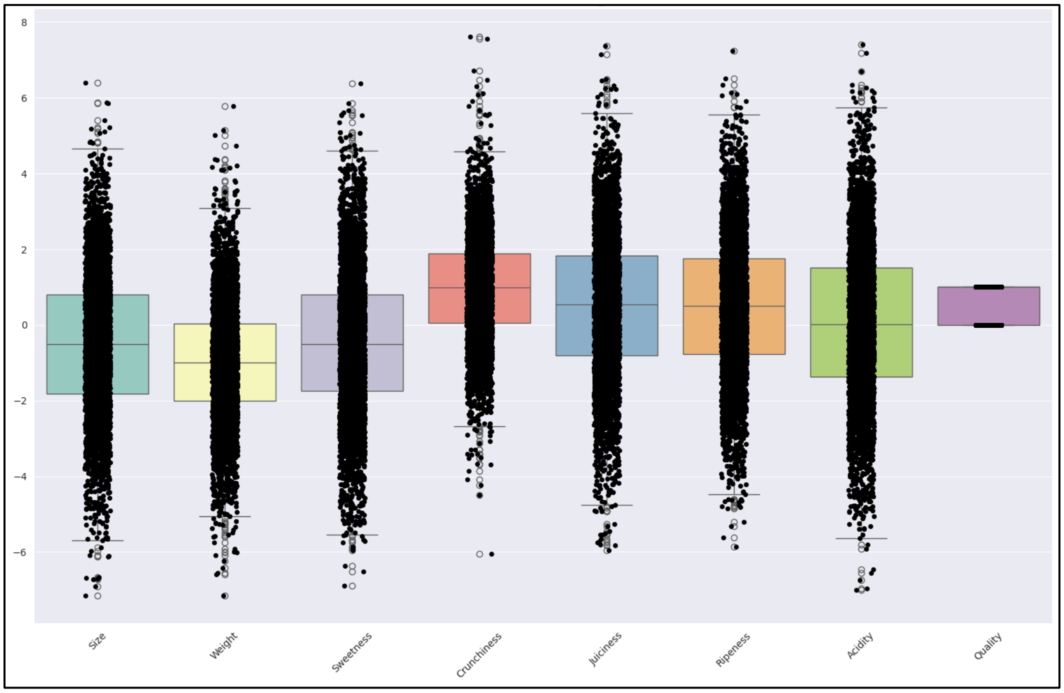


Figure 10: Plot Dataset .

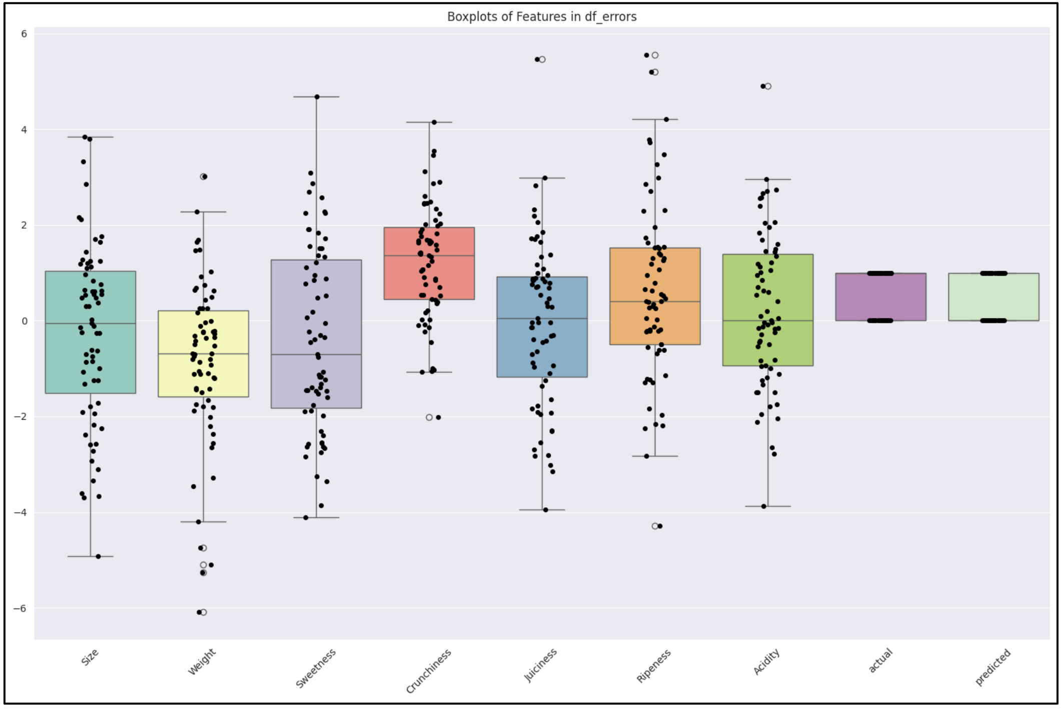


Figure 11: Plot Error of Dataset.

From Comparing dataset with errors dataset, There's no clear pattern in the errors dataset.

Conclusion and Discussion

The project aimed to evaluate the performance of three models—Nearest Neighbor Baseline, Support Vector Classifier (SVC), and ExtraTree Classifier for apple quality classification.

The models exhibited varying degrees of success, with SVC emerging as the best performer based on comprehensive evaluation metrics.

Nearest Neighbor Baseline: The baseline models, particularly k=7 in Nearest Neighbor, provided a solid starting point. However, more sophisticated models surpassed their performance.

Support Vector Classifier (SVC): SVC demonstrated superior performance with an accuracy of 91.625%. Its precision, recall, and F1 score reflected a balanced and robust classification ability.

ExtraTree Classifier: While ExtraTree Classifier performed well, achieving an accuracy of 90.125%, it fell slightly short of SVC in terms of overall metrics.

Evaluation Metrics:

Strengths:

The chosen metrics, including precision, recall, F1 score, ROC curves, and AUC, offered a comprehensive evaluation of model performance.

The ROC curves and AUC provided insights into the models' discriminatory abilities.

Limitations:

The binary classification approach might not fully capture the nuances of apple quality. Expanding to a multi-class classification could provide more detailed assessments.

The evaluation metrics are based on a specific threshold; sensitivity analysis for threshold selection could further enhance understanding.