K-NN vs Decision tree: classification of apples based on the quality

by Vasyl Yarmolenko, 210029807

Problem Statement

The central research question addressed in this project is the identification of the **most suitable classification algorithm** for analyzing structured tabular data with both **continuous features** and a **categorical target**. Specifically, the aim is to evaluate and compare the predictive performance of two supervised learning algorithms—**K-Nearest Neighbors (K-NN) and Decision Tree Classifier**—in the context of fruit quality classification. This comparative analysis provides insights into their effectiveness when applied to real-world datasets with multivariate characteristics.

Objectives

- **Exploratory Data Analysis and Cleaning**: To assess the structure and integrity of the dataset by identifying missing values, outliers, and data types, and applying necessary preprocessing steps.

- **Feature Scaling and Normalization**: To apply appropriate scaling and transformation techniques (e.g., **Min-Max normalization, Z-score standardization**) for ensuring that numerical features contribute proportionately to model training.

- **Model Development**: To implement and train K-NN and Decision Tree classifiers, optimizing hyperparameters to improve predictive **accuracy** and **reduce overfitting**.

- **Model Evaluation and Comparison**: To assess model performance using standard classification metrics such as precision, recall, F1-score, and error-based metrics (MAE, MSE, RMSE), and determine which model generalizes better to **unseen data**.

Data Source

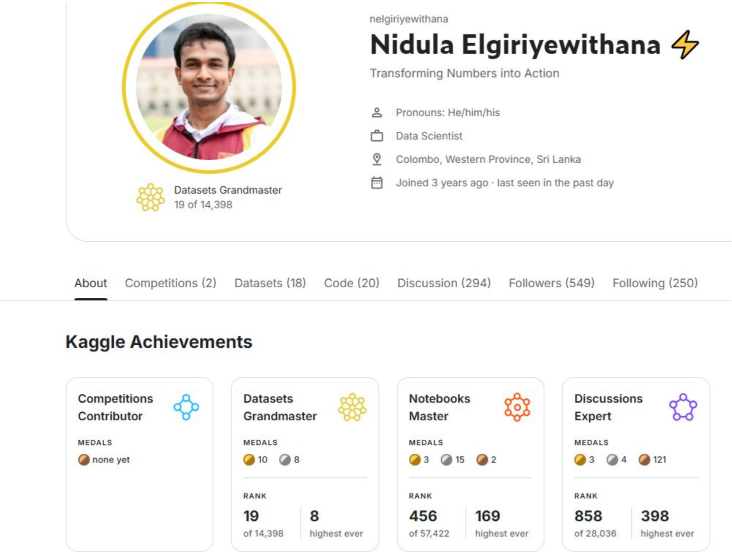
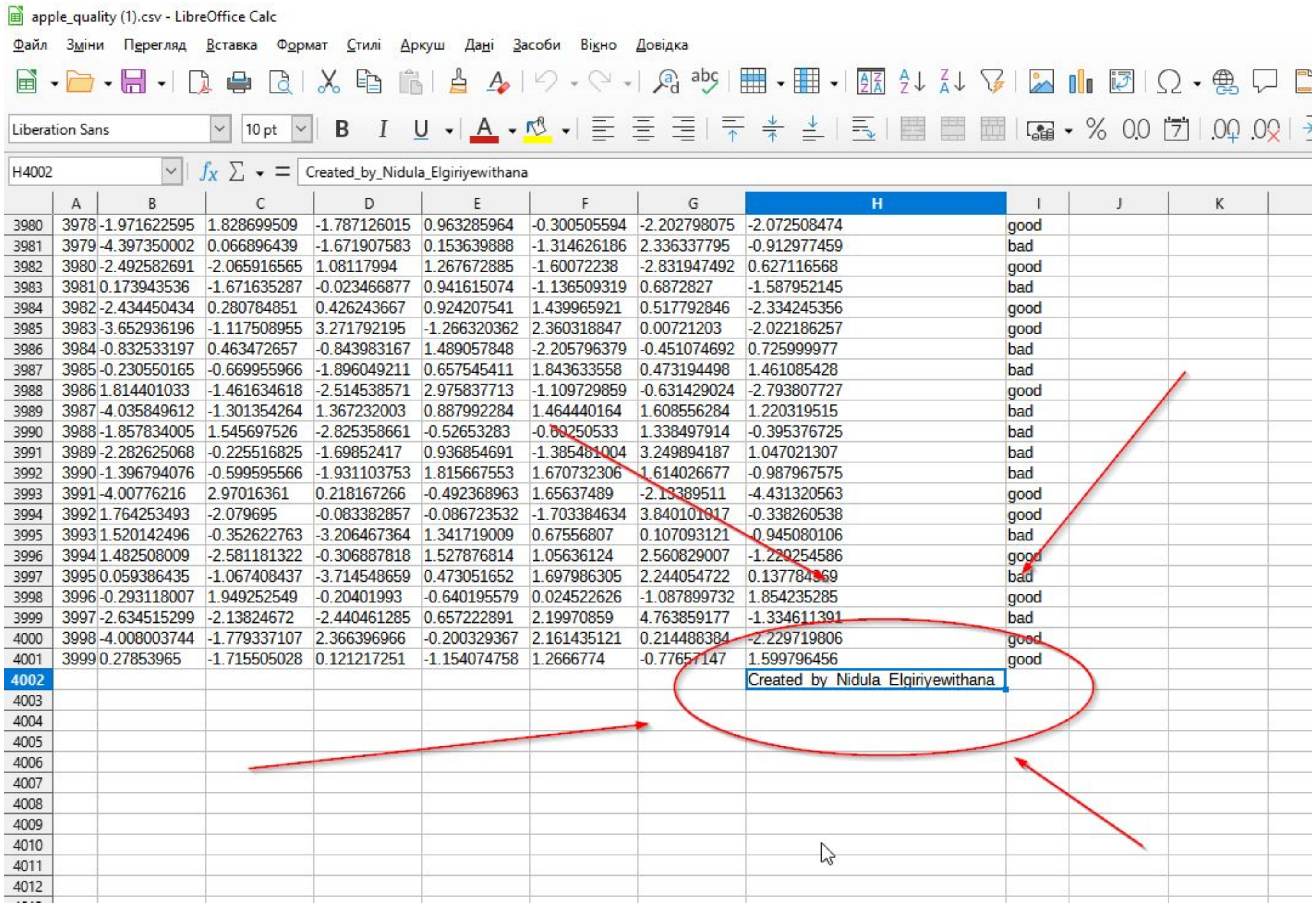


Figure 1. Dataset’s author on Kaggle.com

The dataset employed in this study comprises detailed measurements of fruit attributes, particularly apples. Each instance in the dataset represents an individual fruit, described by a range of **physicochemical characteristics**. The available features include a unique identifier (A_id), physical dimensions (**Size, Weight**), organoleptic qualities (**Sweetness, Crunchiness, Juiciness, Ripeness**), and chemical composition (**Acidity**). The target variable, **Quality**, indicates the categorical classification of the fruit as either "good" or "bad." The dataset was generously provided by an **American agriculture company**.

Data Preprocessing Steps



apple_quality (1).csv - LibreOffice Calc

Файл Зміни Перегляд Вставка Формат Стилі Аркуш Дані Засоби Вікно Довідка

Liberation Sans 10 pt

H4002 Created_by_Nidula_Elgiriyewithana

	A	B	C	D	E	F	G	H	I	J	K
3980	3978	-1.971622595	1.828699509	-1.787126015	0.963285964	-0.300505594	-2.202798075	-2.072508474	good		
3981	3979	-4.397350002	0.066896439	-1.671907583	0.153639888	-1.314626186	2.336337795	-0.912977459	bad		
3982	3980	-2.492582691	-2.065916565	1.08117994	1.267672885	-1.60072238	-2.831947492	0.627116568	good		
3983	3981	0.173943536	-1.671635287	-0.023466877	0.941615074	-1.136509319	0.6872827	-1.587952145	bad		
3984	3982	-2.434450434	0.280784851	0.426243667	0.924207541	1.439965921	0.517792846	-2.334245356	good		
3985	3983	-3.652936196	-1.117508955	3.271792195	-1.266320362	2.360318847	0.00721203	-2.022186257	good		
3986	3984	-0.832533197	0.463472657	-0.843983167	1.489057848	-2.205796379	-0.451074692	0.725999977	bad		
3987	3985	-0.230550165	-0.669955966	-1.896049211	0.657545411	1.843633558	0.473194498	1.461085428	bad		
3988	3986	1.814401033	-1.461634618	-2.514538571	2.975837713	-1.109729859	-0.631429024	-2.793807727	good		
3989	3987	-4.035849612	-1.301354264	1.367232003	0.887992284	1.464440164	1.608556284	1.220319515	bad		
3990	3988	-1.857834005	1.545697526	-2.825358661	-0.52653283	-0.60250533	1.338497914	-0.395376725	bad		
3991	3989	-2.282625068	-0.225516825	-1.69852417	0.936854691	-1.385481004	3.249894187	1.047021307	bad		
3992	3990	-1.396794076	-0.599595566	-1.931103753	1.815667553	1.670732306	1.614026677	-0.987967575	bad		
3993	3991	-4.00776216	2.97016361	0.218167266	-0.492368963	1.65637489	-2.13389511	-4.431320563	good		
3994	3992	1.764253493	-2.079695	-0.083382857	-0.086723532	-1.703384634	3.840101017	-0.338260538	good		
3995	3993	1.520142496	-0.352622763	-3.206467364	1.341719009	0.67556807	0.107093121	-0.945080106	bad		
3996	3994	1.482508009	-2.581181322	-0.306887818	1.527876814	1.05636124	2.560829007	-1.220254586	good		
3997	3995	0.059386435	-1.067408437	-3.714548659	0.473051652	1.697986305	2.244054722	0.137784369	bad		
3998	3996	-0.293118007	1.949252549	-0.20401993	-0.640195579	0.024522626	-1.087899732	1.854235285	good		
3999	3997	-2.634515299	-2.13824672	-2.440461285	0.657222891	2.19970859	4.763859177	-1.334611391	bad		
4000	3998	-4.008003744	-1.779337107	2.366396966	-0.200329367	2.161435121	0.214488384	-2.229719806	good		
4001	3999	0.27853965	-1.715505028	0.121217251	-1.154074758	1.2666774	-0.77657147	1.599796456	good		
4002								Created by Nidula Elgiriyewithana			
4003											
4004											
4005											
4006											
4007											
4008											
4009											
4010											
4011											
4012											

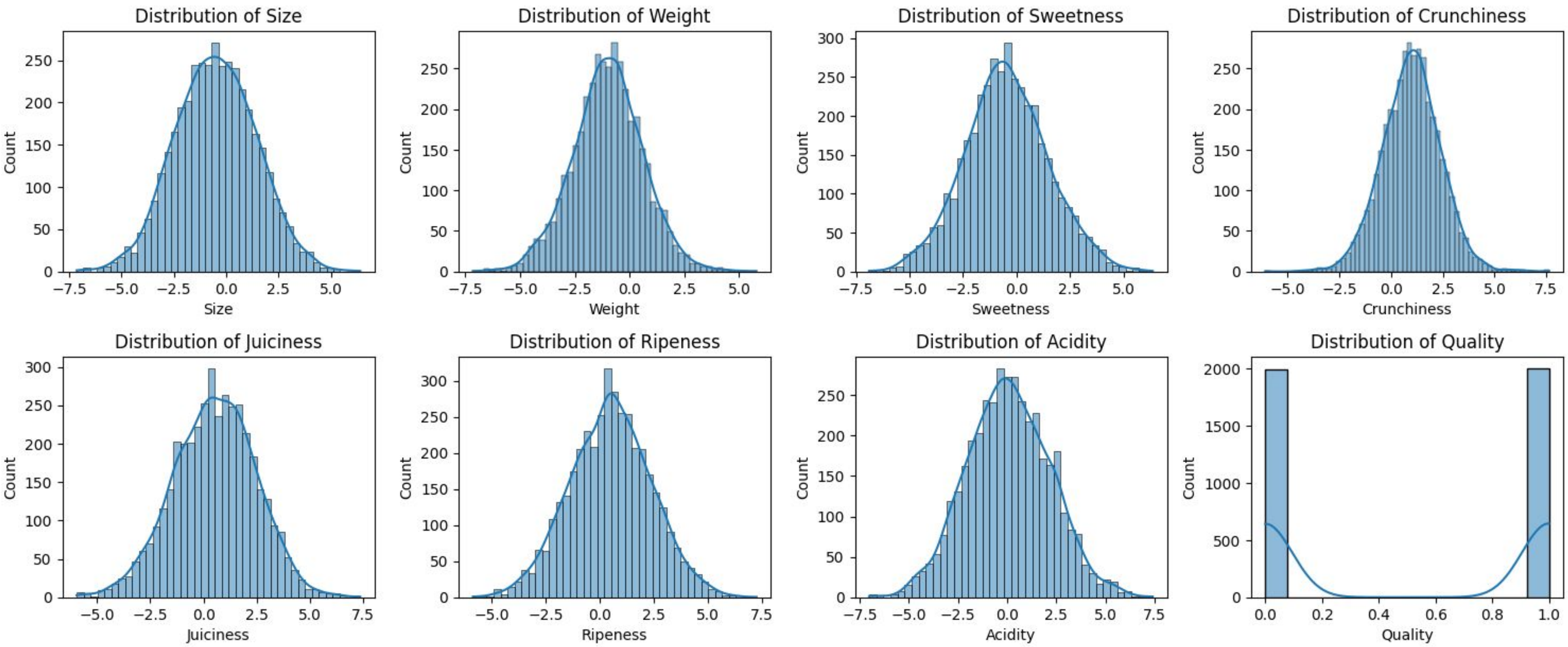
Data Cleaning

- An **extraneous note** present in the final row was identified and **removed**, as it did not constitute a valid data entry
- All feature columns were verified to be **continuous quantitative variables**, while the **target variable (Quality)** was **categorical** and **binary** in nature. The identifier variable *A_id* was later **excluded** due to its lack of predictive utility.
- **No missing values** or **duplicate records** were detected during preliminary analysis. The dataset consisted of 4,000 samples across 9 columns, comprising **1 target, 7 input features, and 1 identifier**.
- Subsequently, the target variable was **label-encoded**, mapping "good" and "bad" to 1 and 0, respectively
- While **most features** exhibited **near-normal distributions**, the **Quality** variable deviated from normality, with an **imbalanced** class distribution.

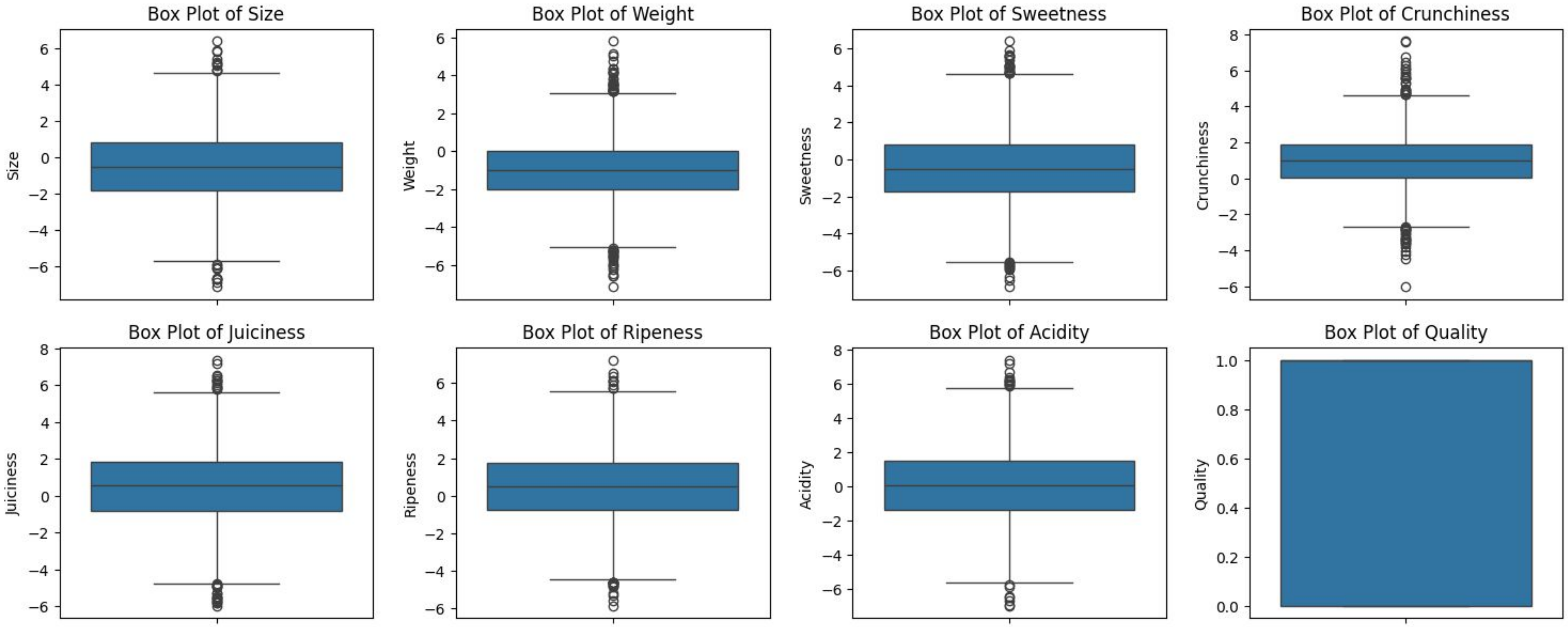
```
Original data dimension:

(4000, 9)

Summary:

              A_id         Size       Weight    Sweetness  Crunchiness  \
count  4000.000000  4000.000000  4000.000000  4000.000000  4000.000000
mean   1999.500000    -0.503015    -0.989547    -0.470479     0.985478
std    1154.844867     1.928059     1.602507     1.943441     1.402757
min       0.000000    -7.151703    -7.149848    -6.894485    -6.055058
25%     999.750000    -1.816765    -2.011770    -1.738425     0.062764
50%    1999.500000    -0.513703    -0.984736    -0.504758     0.998249
75%    2999.250000     0.805526     0.030976     0.801922     1.894234
max    3999.000000     6.406367     5.790714     6.374916     7.619852

         Juiciness     Ripeness      Acidity      Quality
count  4000.000000  4000.000000  4000.000000  4000.000000
mean      0.512118     0.498277     0.076877     0.501000
std       1.930286     1.874427     2.110270     0.500062
min      -5.961897    -5.864599    -7.010538     0.000000
25%      -0.801286    -0.771677    -1.377424     0.000000
50%       0.534219     0.503445     0.022609     1.000000
75%       1.835976     1.766212     1.510493     1.000000
max       7.364403     7.237837     7.404736     1.000000
```

Visualization

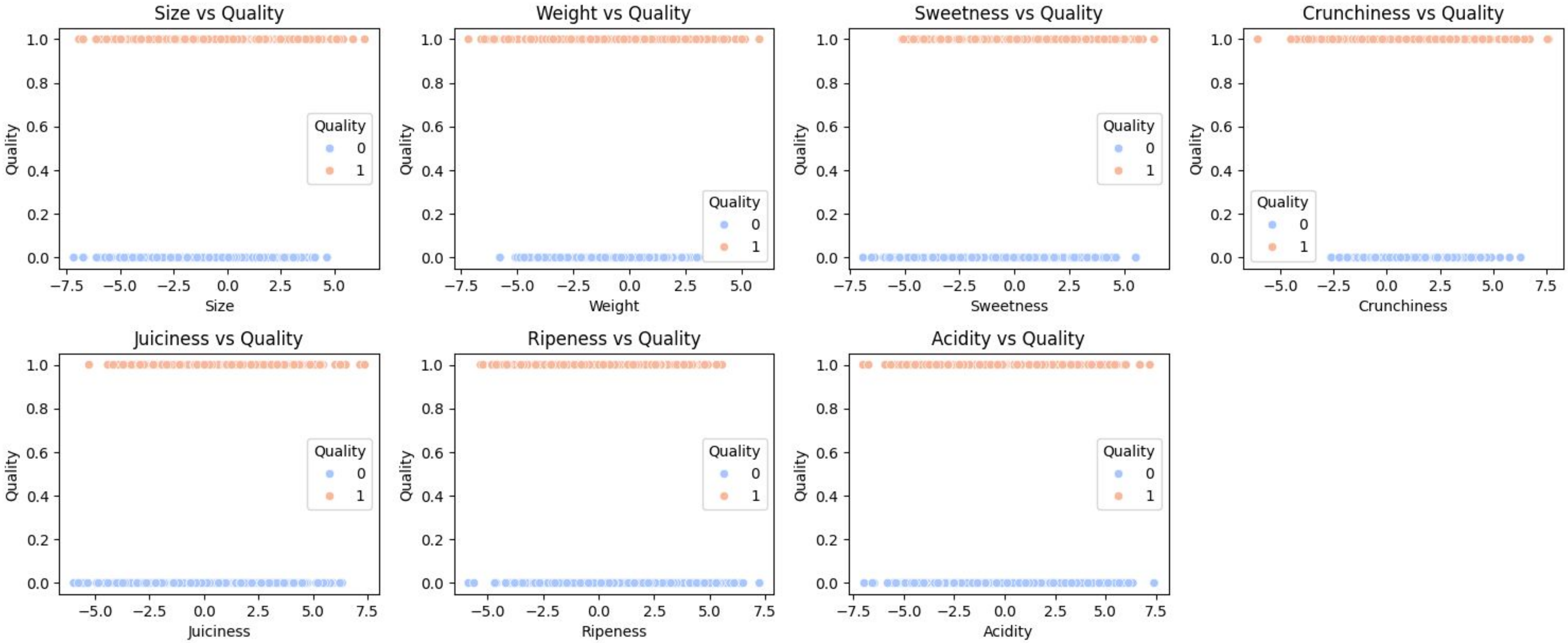
Histograms of numerical variables



Box plots for all numerical variables



Scatter plots of Quality vs. numerical features



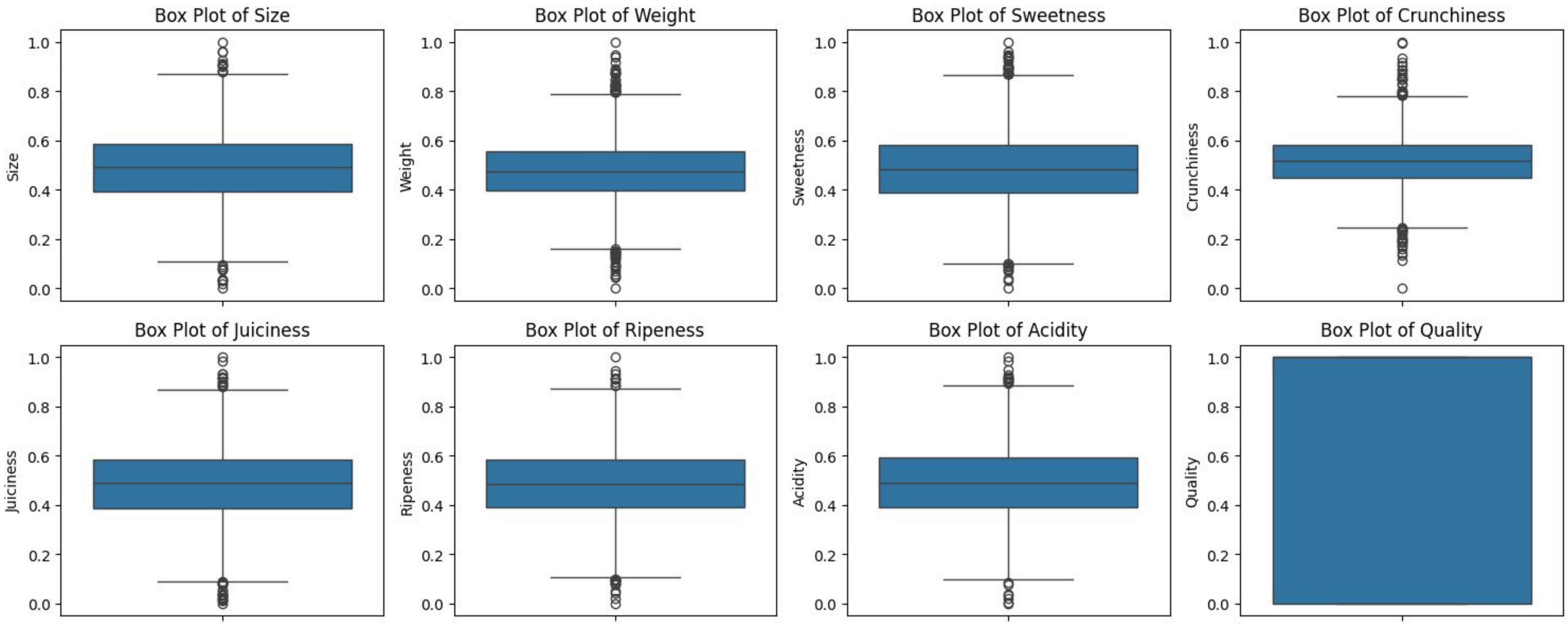
Data Normalization

Three normalization techniques were evaluated to ensure numerical stability in downstream model training:

- **Z-Score Standardization:** Subtracted the mean and scaled by standard deviation
- **Log-Normalization:** Applied to features with skewed distributions
- **Min-Max Scaling:** Scaled features to the [0, 1] interval

Upon empirical comparison, **Min-Max normalization** yielded the most effective standardization, maintaining range consistency across all features while minimizing distortion, and was thus selected for final model training.

Box plots of numerical features after data normalization



Model Training and Evaluation

Following data preprocessing and normalization, the dataset was divided into training and testing subsets using an **80:20** stratified split to preserve class distribution across both sets.

Two supervised classification algorithms were employed in this study:

- **K-Nearest Neighbors (K-NN)**: A non-parametric, instance-based learning algorithm that assigns a class label to a data point based on the majority label of its k nearest neighbors in the feature space.

- **Decision Tree Classifier**: A tree-structured model that uses recursive partitioning to divide the feature space into homogenous regions based on information gain or impurity measures such as Gini index or entropy.

Model performance was assessed using both classification and error-based evaluation metrics, including:

- **Confusion Matrix**
- **Mean Absolute Error (MAE)**
- **Mean Squared Error (MSE)**
- **Root Mean Squared Error (RMSE)**
- **Precision, Recall, and F1-score**

Model	Parameters	Set	MAE	MSE	RMSE	Precision (0 / 1)	Recall (0 / 1)	F1-Score (0 / 1)
K-NN (k=3)	k = 3	**Train**	0.0556	0.0556	0.2358	**0.95 / 0.94**	0.94 / 0.95	0.94 / 0.94
		Test	0.0925	0.0925	0.3041	**0.92 / 0.90**	0.89 / 0.92	0.91 / 0.91
K-NN (k=7)	k = 7	**Train**	0.0766	0.0766	0.2767	**0.93 / 0.92**	0.92 / 0.93	0.92 / 0.92
		Test	0.0950	0.0950	0.3082	**0.92 / 0.89**	0.88 / 0.93	0.90 / 0.91
Decision Tree (Model 1)	max_depth = 9, min_samples_split = 2	**Train**	0.0966	0.0966	0.3107	**0.87 / 0.94**	0.95 / 0.86	0.91 / 0.90
		Test	0.1975	0.1975	0.4444	**0.77 / 0.85**	0.86 / 0.74	0.81 / 0.79
Decision Tree (Model 2)	max_depth = 10, min_samples_split = 5	**Train**	0.0794	0.0794	0.2817	**0.90 / 0.94**	0.94 / 0.90	0.92 / 0.92
		Test	0.1850	0.1850	0.4301	**0.79 / 0.84**	0.85 / 0.78	0.82 / 0.81

Conclusion

The **K-NN classifier** demonstrated strong performance, particularly with k=7, achieving a balance between **training accuracy and generalization to unseen data**. Lower error rates and closely aligned evaluation metrics between training and test sets indicated **minimal overfitting**.

Conversely, the **Decision Tree model**, while initially overfitting the training data, improved after hyperparameter tuning (max depth = 10, min samples split = 5), yet still exhibited **greater variance** between training and test performance **compared to K-NN.**

Overall, the K-NN algorithm outperformed the Decision Tree classifier in terms of **consistent accuracy** and **lower error rates** across multiple configurations. This suggests that instance-based learning may be better suited to this specific classification problem, especially when combined with effective feature scaling.

References

1. Pandas Documentation - https://pandas.pydata.org/docs/reference/frame.html
2. NumPy Documentation - https://numpy.org/devdocs/reference/routines.html
3. Kaggle Dataset Source, 2024, Nidula Elgiriyewithana - https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality
4. Google Colab Notebook Project of Codes, 2025, Vasyl Yarmolenko- https://colab.research.google.com/drive/18LpjabUszoemOSzJ4eUyiwUB6nWgI9TA