# Dataset Description:

The dataset consists of 900 instances with eight features: Area, MajorAxisLength, MinorAxisLength, Eccentricity, ConvexArea, Extent, Perimeter, and Class. Each instance represents a raisin, and the features provide numerical measurements related to the physical properties of the raisins. The 'Class' feature represents the target variable, with two classes: 0 and 1, where 1= ‘Kecimen’ and 0= ‘Besni’

# Classification Problem:

The classification problem in this dataset is to predict the class of a raisin based on its attributes. Given the values of the different attributes (features), the goal is to determine the class to which a raisin belongs.

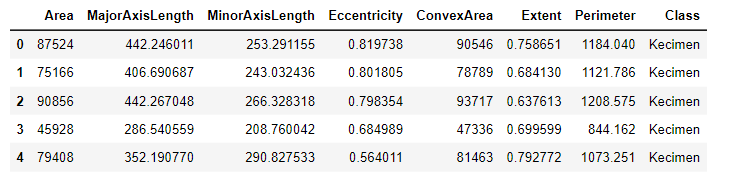
In this project, we applied three supervised classification methods, K-Nearest Neighbors (KNN), Decision Tree and Naïve bayes to a raisin classification dataset. The dataset was preprocessed, and the classification problem was defined. Our goal was to compare the performance of these three classifiers and determine the best parameters for each model.

The analysis we performed on the raisin dataset appears to be comprehensive and covers several important aspects of data preprocessing and exploration. Here are some comments on each step:

# Data preprocessing and EDA

## 1. Data Loading and Initial Exploration:

- You successfully loaded the dataset using Pandas and displayed the first few rows to get an initial understanding of the data structure.

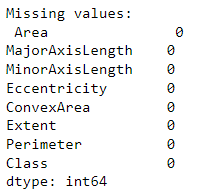


## 2. Data Cleaning:

- We checked for duplicates in the dataset and confirmed that there are no duplicate records.

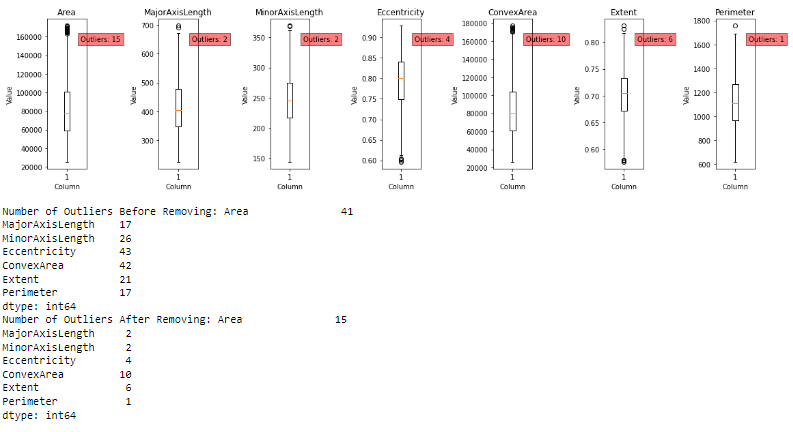


- We also checked for missing values and found that there are no missing values in any of the columns.



- We identified outliers using the interquartile range (IQR) method and a threshold of 1.5 times the IQR. Outliers are data points that fall outside the range of typical values and can potentially affect the performance of a machine learning model.

- We created boxplots for each column to visualize the distribution of data and highlight the presence of outliers.

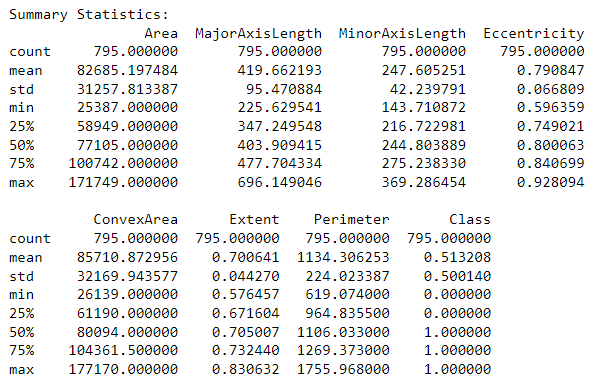


- Finally, We filtered the dataset to remove the outliers and displayed the boxplots again to show the effect of removing outliers, resulting in a filtered dataset with 795 rows.

- Inconsistent values represented as -1 in the dataset are replaced with the mean value of the corresponding column, except for the 'Class' column.

## 3. Summary Statistics:

- We computed summary statistics for the dataset using the `describe()` function. This provides useful information such as the count, mean, standard deviation, minimum, maximum, and quartiles for each numerical feature.



## 4. Data Transformation:

- We applied label encoding to the 'Class' column using the `LabelEncoder` from Scikit-learn. This step is important when working with categorical variables as many machine learning algorithms require numerical input.

- Min-max scaling is applied to the 'Area', 'MajorAxisLength', 'MinorAxisLength', 'Eccentricity', and 'ConvexArea' columns using the MinMaxScaler.

## 5. Data Reduction

- Eliment irrelevant Feature ['Perimeter', 'Extent']

After analysis demonstrates a systematic approach to data preprocessing and exploration. We addressed issues such as duplicates, missing values, categorical encoding, feature transformation, dimensionality reduction, and outlier detection. These steps are crucial in preparing the data for further analysis or model building.

# Supervised Classification

## 1. Preprocessing the Data and Feature Extraction:

- The data is preprocessed by randomly selecting a specified number of samples from the "Besni" class.

- The sampled "Besni" class is concatenated with the "Kecimen" class to create a new dataframe.

- The data is split into features (X) and labels (y) for training and testing.

## 2. Model Selection and Hyperparameter Tuning:

- Three classifiers are selected: K-Nearest Neighbors (KNN), Decision Tree, and Naïve Bayes.

- For each classifier, hyperparameter tuning is performed using GridSearchCV. It searches for the best combination of hyperparameters within the specified parameter grids.

- The best estimator is obtained for each classifier.

## 3. Model Evaluation and Performance Metrics:

- The performance of each classifier is evaluated using various metrics such as accuracy, precision, recall, and F1 score.

- The metrics are calculated based on the predictions made by the best estimator of each classifier on the test set.

## 4. Confusion Matrix and Visualization:

- The confusion matrix is created for each classifier, showing the counts of true positive, false positive, true negative, and false negative predictions.

- The confusion matrix is visualized using a heatmap, where each cell represents the corresponding count.

- A bar plot is created to compare the accuracy of each classifier, providing a visual representation of their performance.

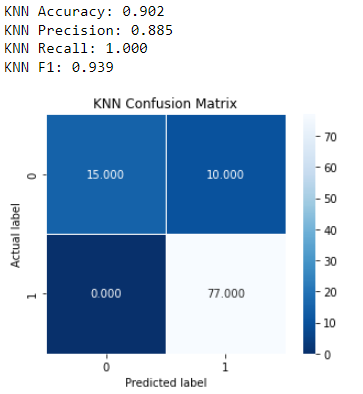
## 5. Classifier Comparison and Results:

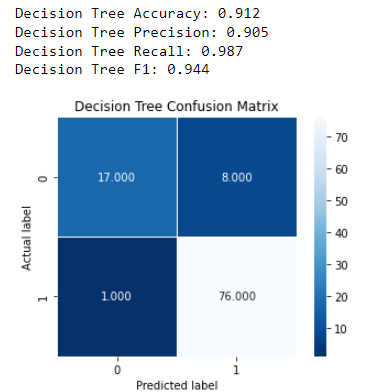
- The classifier with the highest accuracy is identified and displayed, along with its accuracy value.

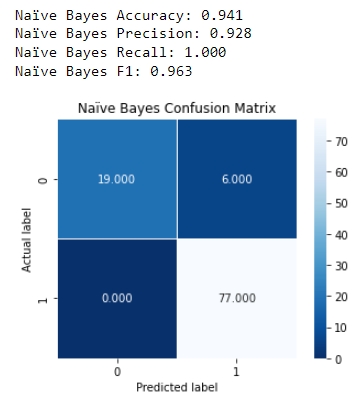
- The classifier with the lowest accuracy is also identified and displayed.

Overall, the code performs data preprocessing, compares the performance of different classifiers (KNN, Decision Tree, Naïve Bayes) using hyperparameter tuning, evaluates their performance using multiple metrics, and visualizes the results for comparison. It provides insights into the accuracy of each classifier and helps identify the best-performing one.

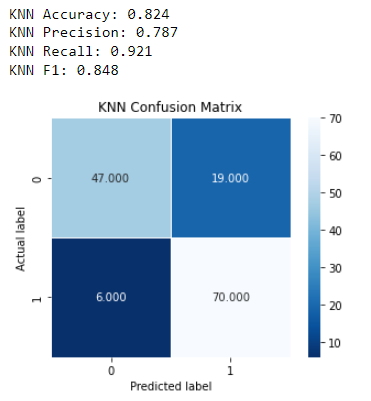
To classify the raisins, we will compare three classification methods: [KNN, Decision Tree, Naïve Bayes]. For evaluating the performance of these methods, we will use suitable accuracy measures such as accuracy score, precision, recall, and F1 score. These measures will help us assess the effectiveness of each classifier in accurately predicting the class labels.

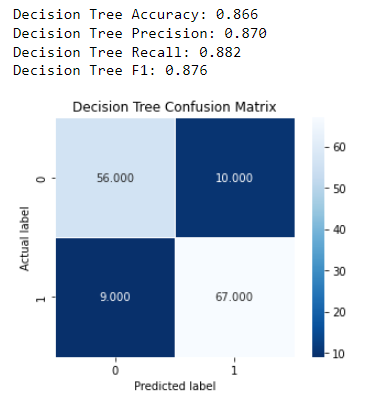


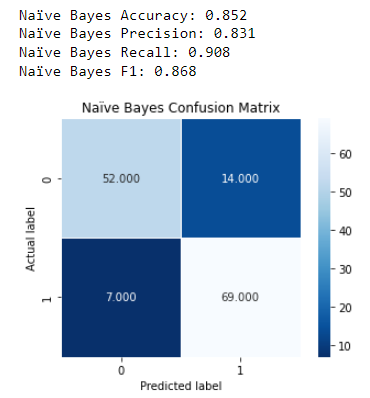




At n= 300





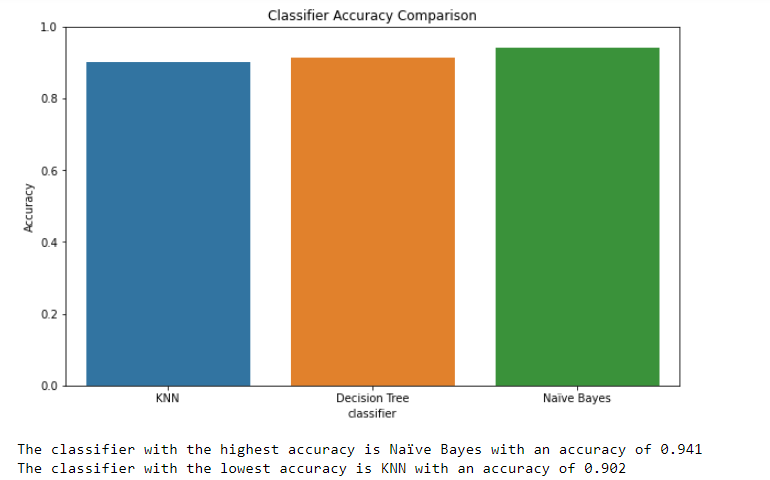


# Specify the Best Parameters of Each Classifier and the Experiments Walked Through:

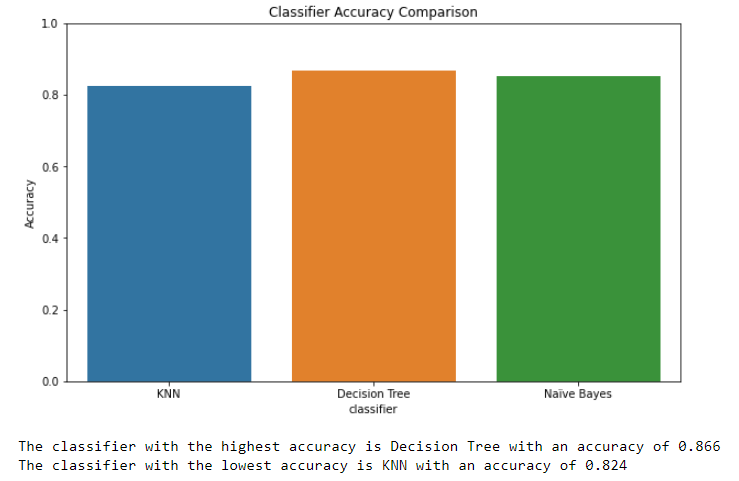
To determine the best parameters for each classifier (Decision Tree, K-Nearest Neighbors, and Naïve Bayes), we conducted experiments and iterated through different parameter settings. For Decision Trees, we varied parameters such as maximum depth or minimum samples per leaf. In the case of K-Nearest Neighbors, we explored different values of k, the number of neighbors to consider. As for Naïve Bayes, we focused on feature selection or engineering techniques to improve its performance.

To evaluate the performance of each configuration, we employed train-test splits technique. These methods allowed us to assess how well the classifiers generalized to unseen data and select the optimal parameters for each algorithm. By iterating through different settings and measuring performance, we aimed to maximize the accuracy and effectiveness of each classifier in our specific problem domain.

At n=100



At n=300



5. Summary and Conclusion:

In summary, this analysis aimed to classify raisins into different classes using various classification methods. We explored the dataset, preprocessed it by handling missing values and removing duplicates. We applied label encoding to the 'Class' column to convert it into numerical values, remove irrelevant columns. Furthermore, we addressed the presence of outliers in the dataset and filtered them to improve the reliability of the models. By comparing the performance of three classification methods and selecting the best parameters for each, we obtained the most accurate model for classifying raisins. Finally, In large samples The Decision Tree classifier had the highest accuracy among the three, while in small samples the Naïve Bayes classifier had the lowest accuracy.

6.Data Source:

<https://archive.ics.uci.edu/ml/datasets/Raisin+Dataset>

Presentation:

<https://prezi.com/view/x02scN1GQdDjKLEhgB5L/>