# *Golden Harvesting: A Predictive For Apple Quality Assurance*

## 1. Introduction

### 1.1. Project overviews

This project aims to build a machine learning model that can predict the quality of apples based on several features such as size, weight, sweetness, crunchiness, juiciness, ripeness, and acidity. By analysing these features, the model will classify apples into different quality categories, helping in automating the process of quality assessment.

### 1.2. Objectives

- To collect and prepare a dataset of apple characteristics.
- To perform exploratory data analysis (EDA) and visualize the data.
- To build and evaluate various machine learning models.
- To optimize the best-performing model using hyperparameter tuning.
- To deploy the final model for practical use.

## 2. Project Initialization and Planning Phase

### 2.1. Define Problem Statement

The problem at hand is to classify apples into different quality categories based on their characteristics. Accurate prediction can help automate quality assessment in the agricultural sector.

### 2.2. Project Proposal (Proposed Solution)

The proposed solution involves building several machine learning models to classify apple quality. The best model will be selected based on performance metrics and optimized using hyperparameter tuning.

### 2.3. Initial Project Planning

Initial planning involved setting up the project environment, defining objectives, and outlining the workflow for data collection, preprocessing, model development, and evaluation.

# 3. Data Collection and Preprocessing Phase

## 3.1. Data Collection Plan and Raw Data Sources Identified

The dataset was sourced from 'apple_quality.csv', which includes features relevant to apple quality assessment.(Kaggle)

## 3.2. Data Quality Report

- **Data Shape**: Initial dataset had (number of rows, number of columns) rows and columns.
- **Missing Values**: Identified and handled by dropping rows with missing values.

## 3.3. Data Exploration and Preprocessing

- **Univariate Analysis**: Histograms of numerical features were plotted.
- **Bivariate Analysis**: Scatter plots and pair plots were used to explore relationships between features.
- **Outlier Handling**: Outliers were detected and managed using the IQR method.

# 4. Model Development Phase

## 4.1. Feature Selection Report

Features were selected based on their relevance to predicting apple quality, and scaling was applied to standardize the data.

## 4.2. Model Selection Report

- **Models Tested**: Decision Tree, Random Forest, XGBoost, Logistic Regression, SVM, KNN, Naive Bayes.
- **Evaluation Metrics**: Accuracy, Confusion Matrix, Classification Report.

## 4.3. Initial Model Training Code, Model Validation and Evaluation Report

- **Code**: Included model training and evaluation steps for each algorithm.
- **Validation**: Models were validated using a test set, and performance metrics were recorded.

# 5. Model Optimization and Tuning Phase

## 5.1. Hyperparameter Tuning Documentation

- **XGBoost**: GridSearchCV was used to tune hyperparameters, resulting in improved accuracy.
- **Random Forest**: GridSearchCV was used to find optimal hyperparameters.

## 5.2. Performance Metrics Comparison Report

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Decision Tree | 81.00% | 0.82 | 0.82 | 0.82 |
| Random Forest | 90.88% | 0.91 | 0.91 | 0.91 |
| XGBoost | 90.50% | 0.90 | 0.90 | 0.90 |
| Logistic Regression | 75.00% | 0.75 | 0.75 | 0.75 |
| SVM | 90.00% | 0.90 | 0.90 | 0.90 |
| KNN | 90.00% | 0.90 | 0.90 | 0.90 |
| Naive Bayes | 76.00% | 0.76 | 0.76 | 0.76 |

## 5.3. Final Model Selection Justification

Random Forest was chosen as the final optimized model because it demonstrated the highest accuracy of 90.88%. The confusion matrix of Random Forest also showed fewer misclassifications, with lower False Positive and False Negative counts, indicating better overall performance in predicting the correct classes. Additionally, Random Forest is known for its robustness and efficiency, especially with structured data, making it a more suitable and reliable choice for predicting apple quality based on the given dataset.

# 6. Results

## 6.1. Output Screenshots

The source code for the project is provided in the accompanying files.

# 7. Advantages & Disadvantages

**Advantages**: High accuracy, effective handling of outliers, scalable for larger datasets.

**Disadvantages**: Computationally intensive, requires careful tuning.

# 8. Conclusion

The project successfully developed a machine learning model to predict apple quality with high accuracy. The Random Forest model, after hyperparameter tuning, provided the best performance.

# 9. Future Scope

- Further data collection to include more features and increase the dataset size.
- Explore additional features and engineering techniques.
- Experimenting with deep learning models to see if they can outperform traditional machine learning models.
- Integration with a real-time quality assessment system for practical deployment.

# 10. Appendix

## 10.1. Source Code

**Code File**: Apple_Quality.ipynb

## 10.2. GitHub & Project Demo Link

**GitHub Repository**: GitHub Link