An Apple a Day: Predicting Apple Quality with Machine Learning Models

Mahima Haridasan

March 31, 2025

Abstract

This work researched apple quality identification and classification from a variety of factors influencing fruit qualities including size, weight, juiciness, crunchiness, sweetness, ripeness and acidity. By applying machine learning models through R to study the impacts of these attributes.

1 Introduction

We have heard the popular phrase "Eat an apple a day keeps the doctor way" this shows how much importance is this fruit in regards to our health ,Apples are very popular agricultural products with high nutritional value.One of the important reasons affecting the export of apples is that the quality of the apples is rather spotty. With increased attention for fruits of high quality and safety standards, the demand for automatic, accurate and fast quality identification continues to grow . Apple's surface defects have a negative impact on profits and sales. Therefore, defective apples should be precisely detected and automatically weeded out before they are sold in the market.

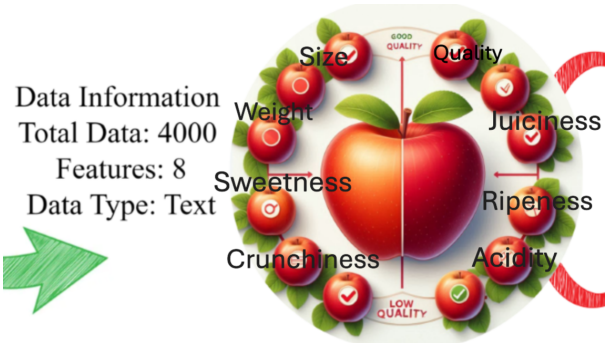


Figure 1: Overview of Apple Quality Attributes

The dataset used in this project is "Apple Quality" is sourced from kaggle (1).There are 4001 entries in the Apple Quality dataset comprises various features associated with apples.

2 Exploratory data analysis

The first image shows the quality of the apple which is differentiated into good and bad ,and their respective counts are 2004 (50.1) and 1996 (49.9) the count of good apples exceeds in number than the bad ones ,it shows a balanced distribution between the two.4a The size of the apples ranged from -7.2mm to 6.4mm with an average diameter of -0.5mm, a median size of -0.5mm and a 1.9% variation in an apple's size from the average size. the size of apples was normally distributed with most of them having a size ranging between -4mm and 0mm. the weight of apples ranged from -7.1g to 5.8g, with an average weight of -1.0g, a median weight of -1.0 and a 1.6% variation in an apples weight compared to the average weight. The weight of apples ranged from -7.1g to 5.8g, with an average weight of -1.0g, a median weight of -1.0 and a 1.6% variation in an apples weight compared to the average weight. The sweetness of an apple fruit ranged from -6.9 degree of sweetness to 6.4 degree of sweetness and -0.5 average degree of sweetness. This further indicated that apples that were above the average degree of sweetness were

Attribute	Methodology	Tools/Standards Used	Key Points
Ripeness	Starch pattern index test; spraying apple flesh with iodine solution to identify starch presence	Iodine solution	Less starch indicates a riper apple as starch breaks down into sugars.
Cruniness	Measure of flesh firmness using a penetrometer which records resistance	Penetrometer, computer	Penetrometer probe measures resistance at a set size, speed, and distance.
Sweetness	Measurement of soluble solids concentration using a refractometer	Refractometer	Soluble solids mostly comprise sugars; refractometer measures light refraction in apple juice.
Acidity	Acid concentration in apple juice measured via titration technique	Titration equipment, juicer	Major acid is malic acid; acidity measured by neutralizing juice sample with a base solution.
Juiciness	Assessed through human sensory evaluation	Human assessors, standards (carrot, celery, watermelon)	Assessors rate attributes on a 0-9 scale, calibrated with standards like carrot and celery.

Figure 2: Apple Quality Measurements

Column Name	Description	Data Type	Potential Use
A_id	Identifier for each apple	Numerical	Indexing apples
Size	Represents the size of the apples	Numerical	Analyzing size-related quality factors
Weight	Indicates the weight of the apples	Numerical	Analyzing weight-related quality factors
Sweetness	Scores reflecting the sweetness of the apples	Numerical	Correlating sweetness with consumer preference
Crunchiness	Scores indicating the crunchiness of the apples	Numerical	Studying texture preferences
Juiciness	Scores assessing the juiciness of the apples	Numerical	Evaluating juiciness in quality assessment
Ripeness	Related to the ripeness of the apples	Numerical	Determining optimal harvest time
Acidity	Values representing the acidity level of the apples	Numerical	Balancing taste and preservability
Quality	Overall quality of the apples ('good' or 'bad')	Categorical	Outcome variable for quality models

Figure 3: Summary of Apple Quality Dataset

better than those below average.5b The histogram indicates that the degree of sweetness in the apples was normally distributed with majority of them having a degree between -2 degree of sweetness and 0 degree of sweetness.

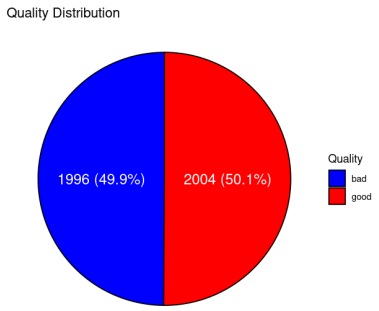
The results presented in the table above indicated that the texture levels of the apples ranged from -6.1 levels to 7.6 levels with an average and median texture levels of 1, 1.4% variation in the apples texture from the average texture level. The histogram indicates that there was a normal distribution in the apples texture levels with most of the apples having textures levels that ranged between 0 to 4 levels of texture.

the juiciness levels ranged from -6.0 to 7.4 levels with an average level of juiciness at 0.5cl.the juiciness of apples was distributed to the right(right skewed) which signified that most of the harvested apples had juice levels that ranged from 1 to -6 levels of juiciness.

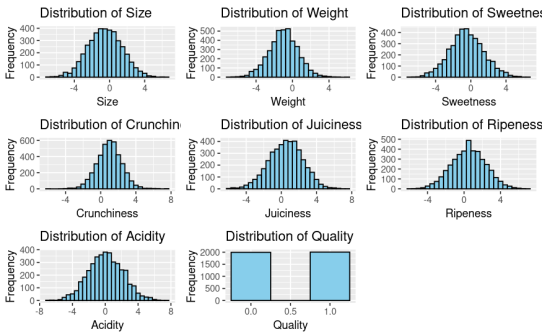
the stage of ripeness in the apples ranged between -5.9 and 7.2, an average stage of ripeness at 0.5 with a 1.9% deviation in each apple's stage of ripeness from the average.there was a normal distribution in the stages of ripeness in the apples with most of them having a ripening stage that laid between -1 and 1 stages of ripeness. The acidity levels in apples ranged between -7 and 7.4 levels with an average of 0.1 acidity levels. the acidity levels in the apples were normally distributed which implied that the acidity levels were equally distributed in these apples.

3 Data Pre-Processing Steps

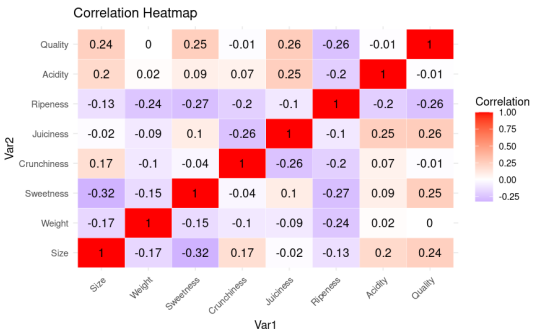
Several Pre-Processing steps were used to see how the attributes are working:



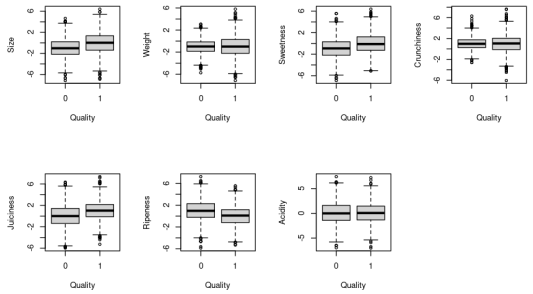
(a) Quality distribution



(b) Distribution of other attributes



(a) Correlation Map



(b) Box plots

Figure 5: Exploratory analysis

3.1 Handling missing values

Checked if there is any missing values in the dataset through imputation and removal.

3.2 Outliers

Using box plot identified and handled outliers in the quantitative features that might affect the performance of the model.

3.3 Feature Scaling

The dataset is already scaled,so no further scaling was done.

3.4 Feature Encoding

Encode the "Quality" label [Good and Bad] with numerical values 1 and 0,ensuring compatibility with the machine learning algorithm.

3.5 Class Balance

The histograms shows the frequencies indicate a balanced distribution between the two quality levels.

3.6 Correlation Analysis

Measuring the relationship between variables,We can see there is no strong correlation between the attributes.

3.7 Splitting the data into train,validation and test split

The train test split is done for 80% train data and 20% test data and the validation set is divided 13% within the train data making training set as 66%.

4 Baseline-KNN Model

A K Nearest Neighbour algorithm was deployed to understand the baseline comparison with other classifiers.Various variation with respect to k was evaluated to check which k value performs best. We can see the accuracies are stable with the values of k=6,7,8,but peaks at k=9 making it 0.8998748.So we can take the optimal value of k=9.

Algorithm 1 Train and Evaluate a kNN Model with 10-Fold Cross-Validation

Require: Train data, Test data
Ensure: Accuracy
train control ← `trainControl(method = "cv", number = 10)` ▷ Setup for 10-fold cross-validation
knn model ← train(Quality ∼ ., data = train_data, method = "knn",
trControl = train control, preProcess = "scale", tuneGrid = data.frame(k = 7)) ▷ Train kNN model with $k = 7$
predictions ← predict(knn model, newdata = test_data) ▷ Predict using trained model
accuracy ← $\frac{\sum(\textbf{predictions}==\textbf{test\_data.Quality})}{\textbf{nrow(test\_data)}}$ ▷ Calculate accuracy

5 Decision Tree and Random forest

We also evaluate our test data using decision tree and random forest. For hyper parameter tuning, I have used random search.(2)

Algorithm 2 Train and Evaluate Multiple Classifiers

Require: Train data, Validation data, Test data
Ensure: Sorted classifiers by validation and test accuracy
sorted_classifiers ← Empty dataframe ▷ Store validation results
test_sorted_classifiers ← Empty dataframe ▷ Store test results
for each *model_name* in classifiers **do**
    model ← classifiers[*model_name*] ▷ Retrieve classifier
    predictions_val ← predict(model, X_{val}) ▷ Predict on validation set
    val_accuracy ← $\frac{\sum(predictions\_val==y_{\text{val}})}{n}$ ▷ Compute validation accuracy
    Append ($model\_name, val\_accuracy$) to sorted_classifiers
    predictions_test ← predict(model, X_{test}) ▷ Predict on test set
    test_accuracy ← $\frac{\sum(predictions\_test==y_{\text{test}})}{n}$ ▷ Compute test accuracy
    Append ($model\_name, test\_accuracy$) to test_sorted_classifiers
end for
Sort sorted_classifiers by validation accuracy in descending order
Sort test_sorted_classifiers by test accuracy in descending order
Output: Display sorted classifiers by validation and test accuracy

6 SVM Model

The Support Vector Classifier 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.

7 Model Performance and Results

The performance of four different machine learning models—Nearest Neighbor baseline , Random Forest, Decision Tree, SVM, and Logistic regression using two metrics: Accuracy and F1 Score were analyzed. Accuracy measures the overall correctness of the model, while the F1 Score is a harmonic mean of precision and recall, providing a balance between them, especially useful when the class distribution is

Algorithm 3 SVM Model Training with Hyperparameter Tuning

param_grid ← expand.grid($C = \{0.1, 1, 10, 100\}, \sigma = \{0.1, 0.5, 1, 2\}$) ▷ Define hyperparameter grid
train_control ← trainControl(method = "cv", number = 5) ▷ Setup 5-fold cross-validation
svm_model ← train(Quality ∼ ., data = train data, method = "svmRadial",
trControl = train_control, preProcess = {"center", "scale"}, tuneGrid = param_grid) ▷ Train SVM model
(C^*, σ^*) ← svm_model.bestTune ▷ Retrieve best hyperparameters
predictions ← predict(svm_model, newdata = test data) ▷ Predict using trained model
accuracy ← $\frac{\sum(predictions == y_{\text{test}})}{n}$ ▷ Calculate accuracy
conf_matrix ← confusionMatrix(predictions, y_test) ▷ Compute confusion matrix

uneven. The models exhibited varying degrees of success, with SVC emerging as the best performer based on comprehensive evaluation metrics.(3)

- **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% after hyperparameter tuning. Its precision, recall, and F1 score reflected a balanced and robust classification ability.
- **Random Forest:**The random forest Algorithm gave 100 % accuracy in training set but did not perform well than SVM in test set,this because random forest has learned pattern too specific to validation set it leads to overfitting.
- **Decision Tree:**While Decision Tree Classifier not performed well, achieving an accuracy of 0.67%, it fell short of every other models in terms of overall metrics.
- **Logistic Regression :**A slightly similar performance as compared to decision tree can be seen in Logistic regression predicting classification approach but not fully capture the pattern of apple quality.

Table 1: Performance Summary of Machine Learning Models

Model	**Accuracy**	**F1 Score**
Random Forest	0.8798498	0.8766067
Decision Tree	0.6720901	0.6561680
Logistic Regression	0.7259074	0.7195903
SVM	0.8811014	0.8795944

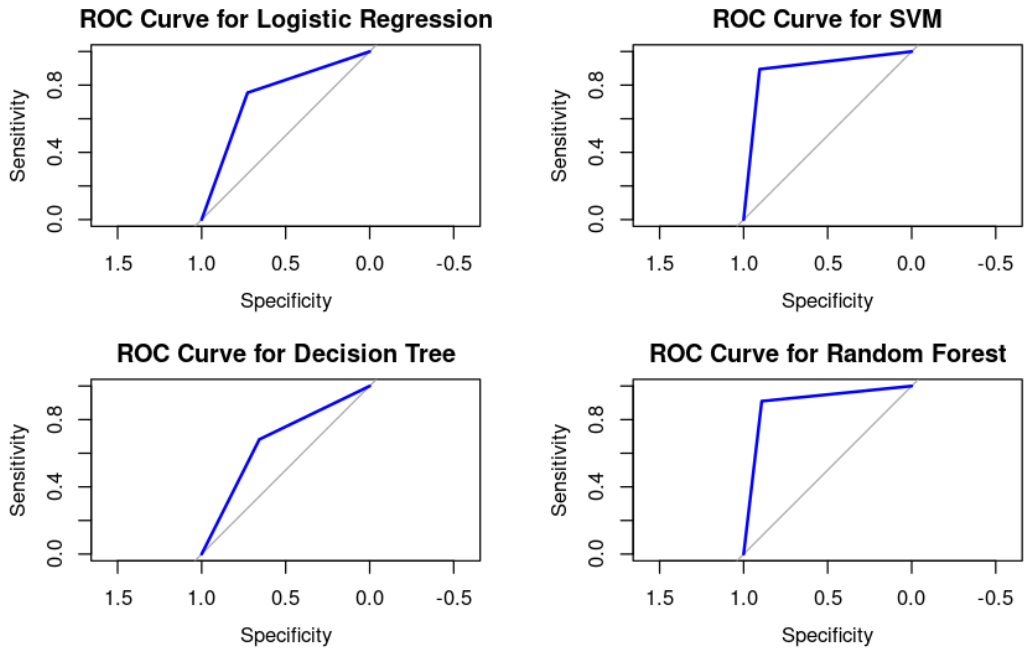


Figure 6: ROC Curve for all Classifier models

8 Acknowledgments

I would like to express my sincere thanks to Professor Fabrice Muhlenbach for his supervision throughout this project. I would also like to thank to Professor Jeudy Baptiste for his teaching of the theoretical part of the course.

References

[1] K. U. Nelgiriyewithana, "Apple quality dataset," Kaggle, 2024, available at https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality.

[2] J. Brownlee, "Tune machine learning algorithms in r (random forest case study)," *Machine Learning Mastery*, 2016.

[3] T. A. Cengel, B. Gencturk, E. T. Yasin, M. B. Yildiz, I. Cinar, and M. Koklu, "Apple (malus domestica) quality evaluation based on analysis of features using machine learning techniques," *Applied Fruit Science*, vol. 66, no. 6, pp. 2123–2133, 2024.

9 Links

Github Link-"https://github.com/mahimahari/Machine-Learning-models-for-Apple-Quality-Dataset-using-R".

Kaggledataset"https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality."