# Bosch Project Report

# Machine Learning Exercise for Job Application

**Handed in by:** Javier Vente

**Date:** 17/01/2024

## Introduction

The objective of this project is to develop a classifier using machine learning that can predict the correct fruit type between bananas, apples and grapes, based on the given value of its features. These features include weight, size and color, being size and color both categorical features, and weight, continuous. All the necessary base dataset for training such a classifier model was provided in an excel file.

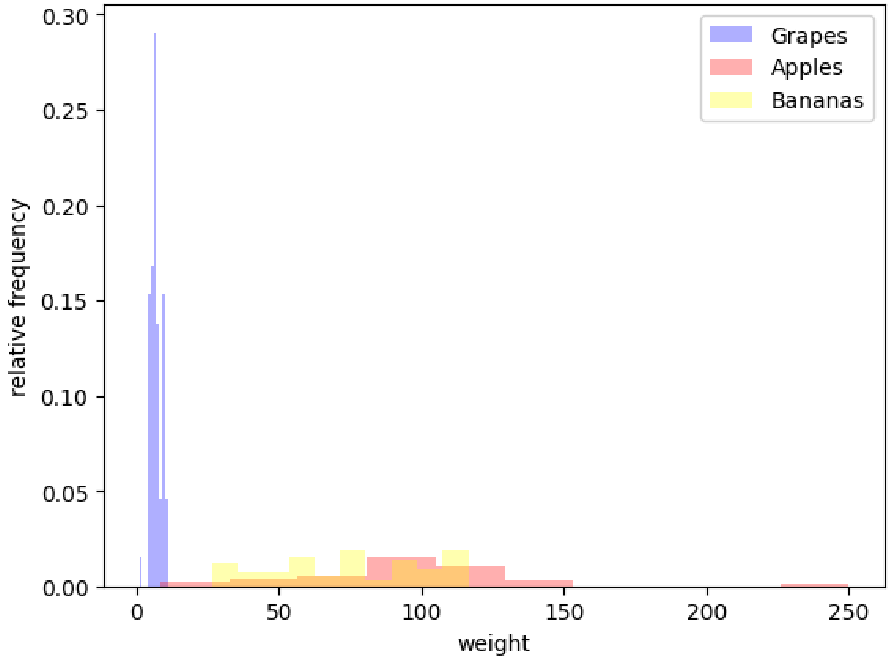
The program was written using python and its machine learning libraries inside the Google Colab platform. The methodology used can be resumed in the following steps: **1)** Data understanding and preparation, **2)** modeling and analysis, **3)** model performance and interpretation. The goal is to come up with a trained model of both the Logistic Regression and Tree Decision, being these the most appropriate for a multi-class classifier within the supervised learning context of the project.

## Data understanding and preparation

Inside the python platform of google colab, the library *pandas* was used for managing the dataset in the form of a *pandas dataframe*. For the necessary machine learning libraries, *sklearn* was enough to provide the necessary classification models.

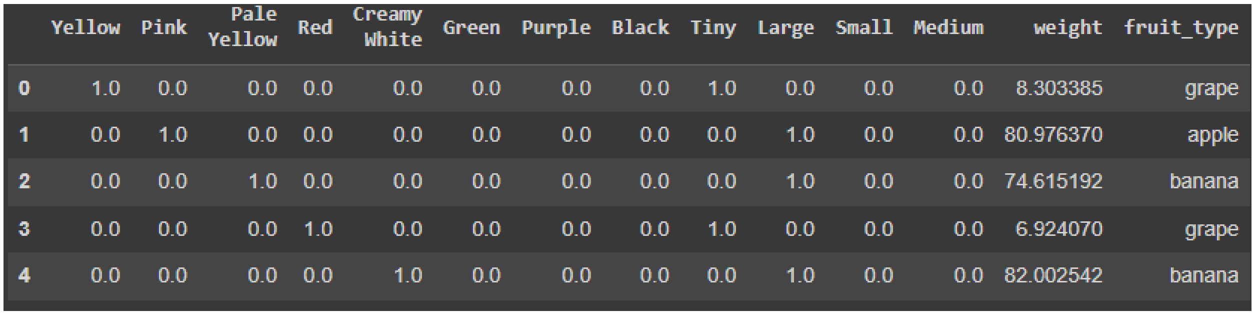
From the data understanding and preparation step, issues regarding data errors, as well as outliers were found. Data errors found were spelling mistakes in some of the values of size and color of the inputs. This problem was identified by listing the unique values from each of these two columns and later on visually analyzing these. After identifying these misspelled values, the solution was to use the .loc[] property of the dataframe and replacing the wrong values with the right ones.

Another type of problem found during the data preparation was after plotting the histogram of the weight of the inputs, separated by the fruit type (also shown in different colors for each of the fruits), here it was easy to visually identify the presence of data outliers.



Such was the case for both low-end and high-end of the graph. In the lower end however, the proximity to the rest of the data was enough to not consider the necessity of excluding them. However, on the higher end, two outliers were found being in both cases the weight of an apple. For simplicity reasons, the method for dealing with these two outliers was to simply run a filter that excluded any input with a weight higher than 200.

Another important part of the data preparation process, was the one-hot-encoding of all the categorical features in the dataframe, these being both “color” and “size”. This was done through a block of code that created a new column for each category value from both features, and later set 1 as the value if the category was the same as the new column, or 0 if it wasn’t.



| | Yellow | Pink | Pale Yellow | Red | Creamy White | Green | Purple | Black | Tiny | Large | Small | Medium | weight | fruit_type |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 8.303385 | grape |
| 1 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 80.976370 | apple |
| 2 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 74.615192 | banana |
| 3 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 6.924070 | grape |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 82.002542 | banana |

Another important part of the data preparation process was the segmentation of the dataframe into the feature matrix and target vector for both the training and testing of the models. This segmentation was done using the “train_test_split” function from the sklearn model_selection object. The parameter to be used was the test_size with the value 0.2, meaning that the test section will be 20% of the length of the total inputs, and the training was kept at 80%.

Finally, it was also necessary to run the feature matrix from both sections through a scaler, since the scale of the features needed to be normalized. This was done using the Standard Scaler object and its fit_transform function.

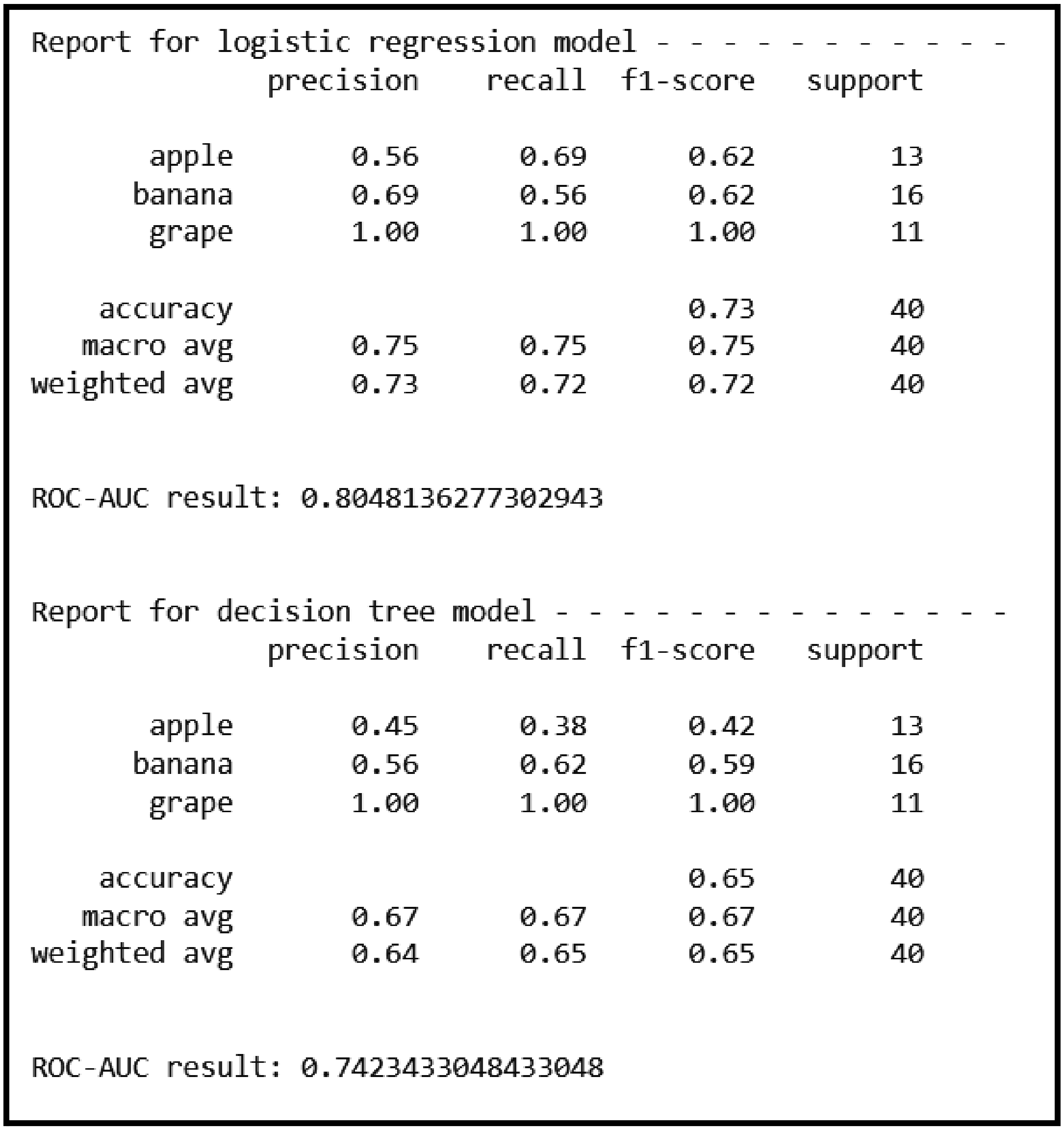
### Modeling and Analysis

This step consisted basically of the training from both models utilized: Logistic Regression and Decision Tree. Thanks to the prior preparation, the data modeling process itself was very straightforward, being only required to create an object for both models and feeding them the training data. In both cases the random_state hyperparameter was set to 42 to ensure the replicability of the training and test results.

**Model performance and interpretation**

Having created and trained both models, the chosen way of measuring the performance was to run their predictions and compare these to the actual test target values, with this a classification report was possible by using metric functions from the sklearn library. This report shows the f1 score of each target value and the overall accuracy of the model among some other information. Additionally to this, the ROC-AUC score was also obtained by using other functions from the same library, for this however a further binarization of the target values was needed.

The final report shows the following results:



```
Report for logistic regression model - - - - - - - - - - - -
              precision    recall  f1-score   support

       apple       0.56      0.69      0.62        13
      banana       0.69      0.56      0.62        16
       grape       1.00      1.00      1.00        11

    accuracy                           0.73        40
   macro avg       0.75      0.75      0.75        40
weighted avg       0.73      0.72      0.72        40


ROC-AUC result: 0.8048136277302943


Report for decision tree model - - - - - - - - - - - - - - -
              precision    recall  f1-score   support

       apple       0.45      0.38      0.42        13
      banana       0.56      0.62      0.59        16
       grape       1.00      1.00      1.00        11

    accuracy                           0.65        40
   macro avg       0.67      0.67      0.67        40
weighted avg       0.64      0.65      0.65        40


ROC-AUC result: 0.7423433048433048
```

From the results, we can conclude that overall the Logistic Regression model showed a better performance, this based on the f1 score, accuracy and ROC-AUC value of both models.

Another interesting intake from this report is that both models performed well in identifying the actual grapes, but on the other hand they both had problems classifying the fruit as a banana or apple. The most likely explanation for this is that the feature values of many inputs of bananas and apples overlap, or in other words, many bananas and apples share similar values in color (e.g. Green bananas and green apples),  size (e.g. Medium, large), and weight. Hence the increase in the difficulty for differentiating one from the other.

One advisable improvement, although out of the scope of this project, could be to introduce more relevant features to be obtained from the samples, in order to increase the classifier overall performance in both models.