**CST4060 – Visualisation Analytics.**

**Knime – Course Work IV.**

**YAHAYA, Suleyman Olushola.**

**M00736597**

[**Sy423@live.mdx.ac.uk**](mailto:Sy423@live.mdx.ac.uk)

**Problem Definition**

This project aims to design a Knime workflow of algorithms that correctly classify the seven classes of dry beans using the other sixteen features.

**Data**

The study used seven different types of dry beans, taking into account market-specific characteristics such as form, shape, type, and structure. A computer vision system was designed to differentiate between these seven different registered varieties of dry beans with similar features to achieve standardised seed classification.

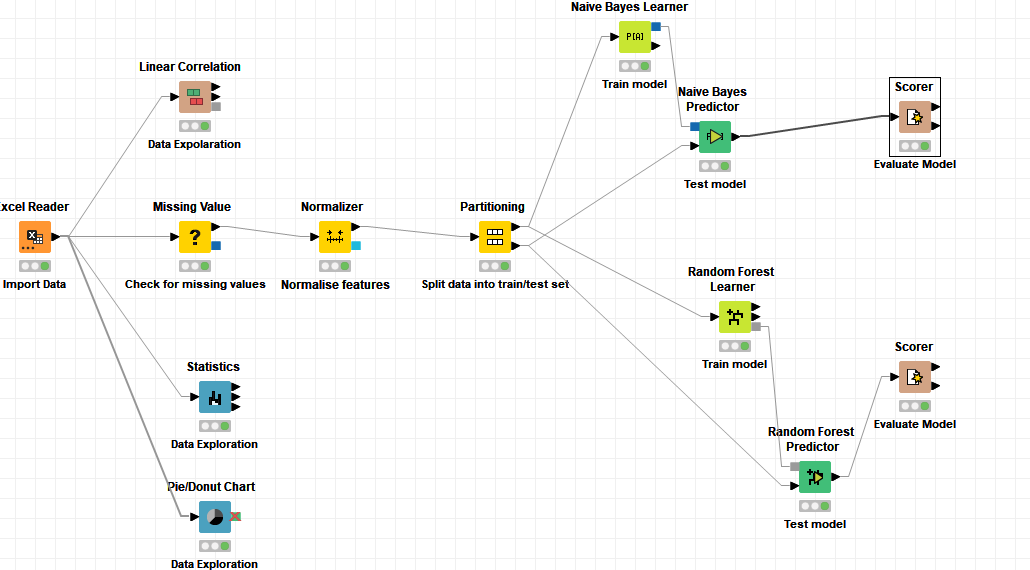
A high-resolution camera was used to capture 13,611 grains from seven different types of dry beans for the dataset. There were 16 various features, 12 different lengths, and four different shape shapes among the grains.

These features include:

* **Area (A**): The size of a bean zone and the number of pixels it contains.
* **Perimeter (P):** The length of a bean's boundary is its circumference.
* Primary axis length (L): This is the distance between the ends of the longest line a bean can draw.
* **Minor axis length (l):** The feature contains the longest line drawn from the bean while standing perpendicular to the central axis.
* **Aspect ratio (K):** Defines how L and l are related.
* **Eccentricity (Ec)**: The eccentricity of an ellipse that has the same moments as the field.
* **Convex area (C):** The number of pixels in the smallest convex polygon that can contain a bean seed's area.
* **Equivalent diameter (Ed):** A circle's diameter equal to the area of a bean seed.
* **Extent (Ex):** The percentage of pixels in the bounding box corresponds to the bean region.
* **Solidity (S)**: This is another word for convexity. The proportion of pixels in the convex shell compared to those in beans.
* **Roundness (R):** Calculated with the following formula: (4piA)/(P^2)
* **Compactness (CO):** Measures the roundness of an object: Ed/L
* **ShapeFactor1 (SF1)**
* **ShapeFactor2 (SF2)**
* **ShapeFactor3 (SF3)**
* **hapeFactor4 (SF4)**
* **Class:** The classes of beans are Seker, Barbunya, Bombay, Cali, Dermosan, Horoz and Sira.

**Workflow Design**

The KNIME analytics framework is an ETL tool used to automate data science operations and selectively examine the findings and models using interactive widgets and views.

****

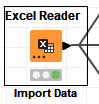
The workflow is an idealised sequence of events in which all of the tasks can be completed in the order shown in the diagram above, as it will frequently be necessary to go back and repeat those actions. The steps involve importing the data, visualising/exploring data, cleaning and scaling data, splitting data into Train and Test sets, modelling using classification algorithms and declaring the findings.

**Data Processing**

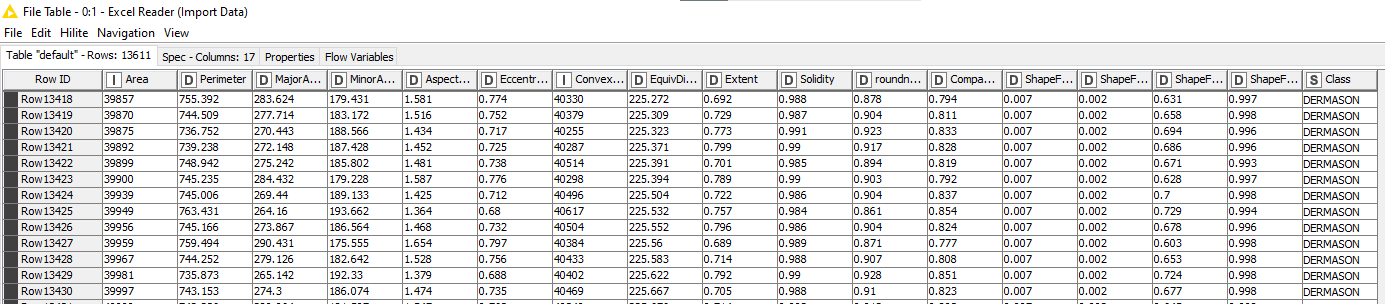
This stage involves uploading the data into Knime, exploring the data and cleaning the data.

**Importing Data**

The data is imported via the Excel Reader node.

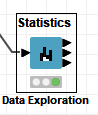


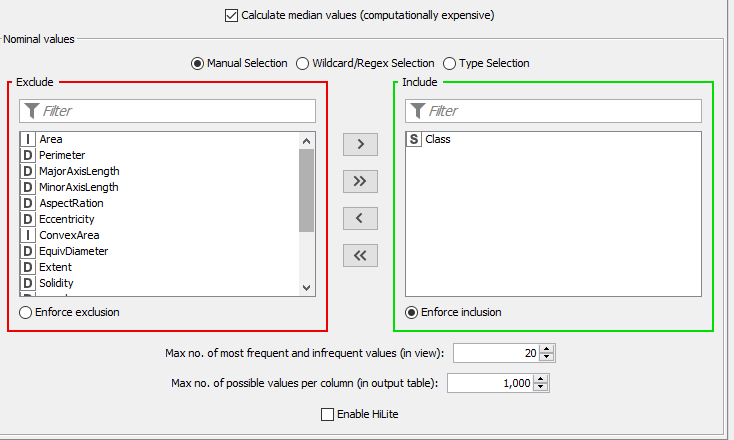
The Excel Reader supports Excel file formats such as XLSX, XLSM, and XLS. The node can read single or multiple files simultaneously, but it only reads one sheet per file. The data has been imported into Knime, as seen in the image below.



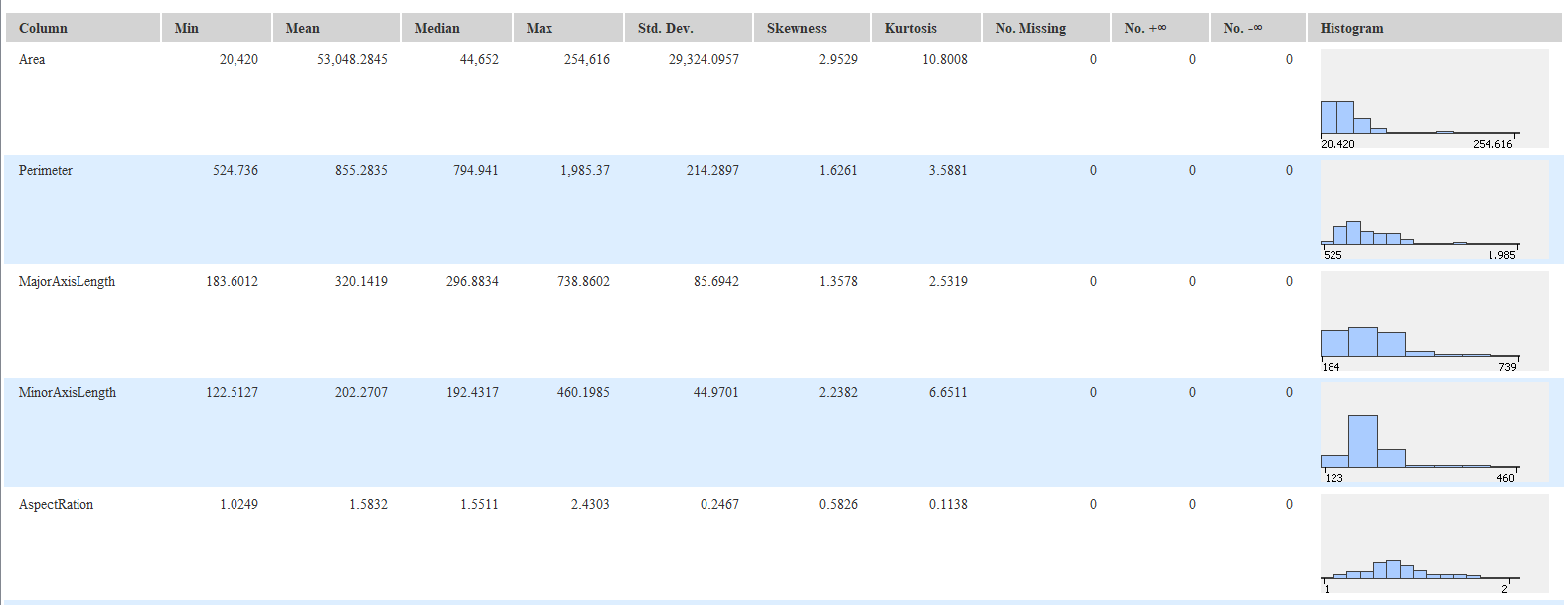
**Data Exploration**

By using visualisation, data exploration aids in understanding and identifying trends and discovering insights in every data analytics mission. In essence, it aids the analyst in grasping viewpoints more quickly. This stage involved the use of the Linear Correlation and Statistics nodes.

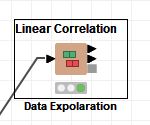


This node counts all nominal values and occurrences and measures statistical moments such as minimum, maximum, mean, standard deviation, variance, median, total sum, number of missing values, and row count across all numeric columns. The node is configured as shown in the image below.

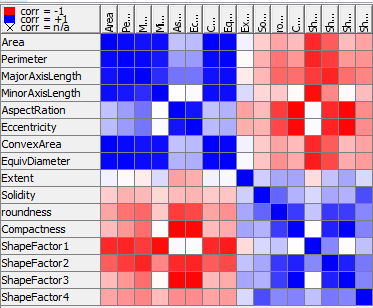
Following the configuration, the node's view displays a statistical overview of all features in relation to all of the dry bean classes present in the dataset with the aid of numbers and histograms, as shown below.



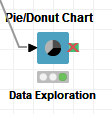
By examining the data, it was discovered that there were no missing values in the dataset.



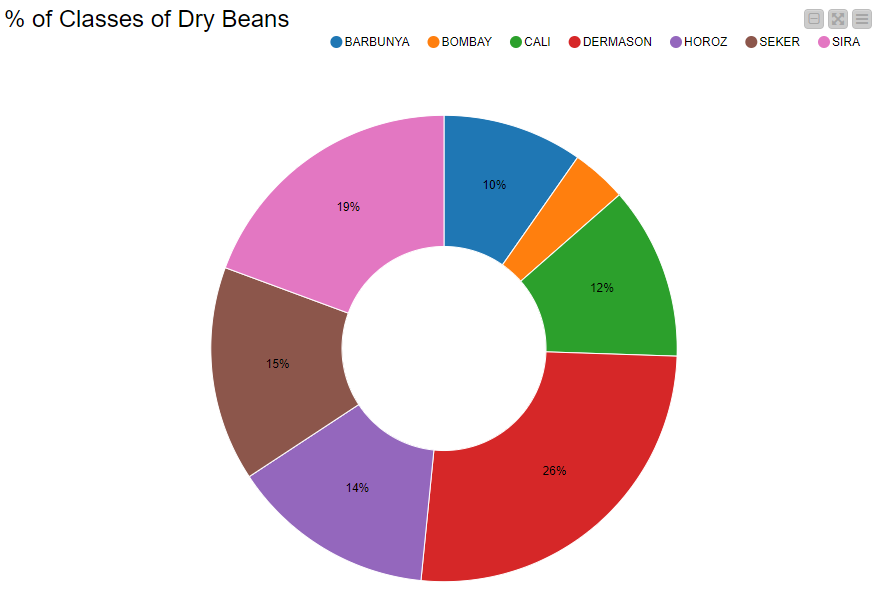
For each pair of selected columns, the Linear correlation node calculates a correlation coefficient, measuring the two variables' correlation. The level correlation between the variables is defined graphically in the image below.



Some attributes, such as 'Perimeter,' 'MajorAxisLength,' 'MinorAxisLength,' 'ConvexArea,' and 'EquivDiameter' all show a correlation with the Area feature, which makes sense given that they are all connected to the number of pixels inside the classes' boundaries.



The Donut chart node displays a percentage count of all the class features.

.

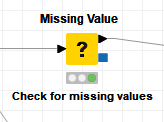
The Dermasson variant makes up 26% of all the types of dry beans present in the dataset.

**Data Mining**

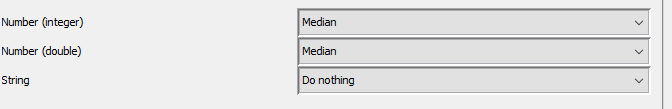
The data mining method entails several techniques for converting raw data into useful information. Cleaning, normalisation, breaking the dataset into training and test sets, and modelling are some of the methods used.

**Data Cleaning**

The Data Cleaning phase involves dealing with missing values and eliminating unrequired features.



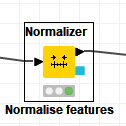
The Missing Value node is set up to append the median values to all continuous features with a missing value. At the same time, attributes with strings are left alone, as shown in the configuration below.



It was discovered that the data had no missing values during the earlier investigation.

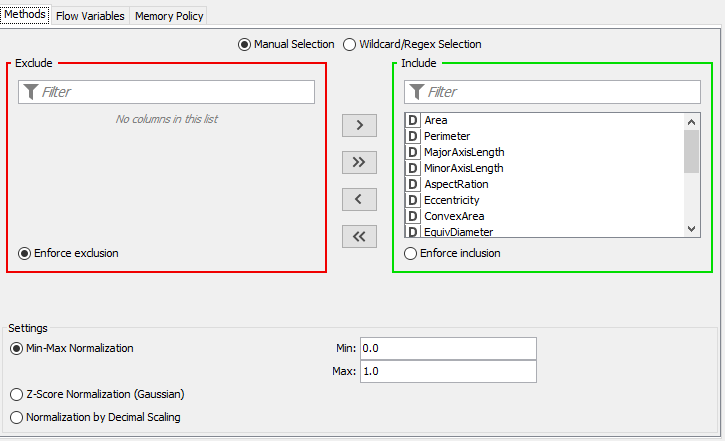
**Feature Scaling**

Since the continuous features of the dataset do not have a specific range and vary from one another, scaling entails normalising all numeric columns.



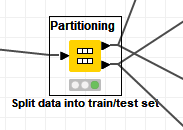
In the Normaliser node, the Min-max scaler was used, and all values were transformed linearly.

As seen in the configuration below, each column's minimum and maximum values are the same, with 1 and 0 as the minimum and maximum values, respectively.

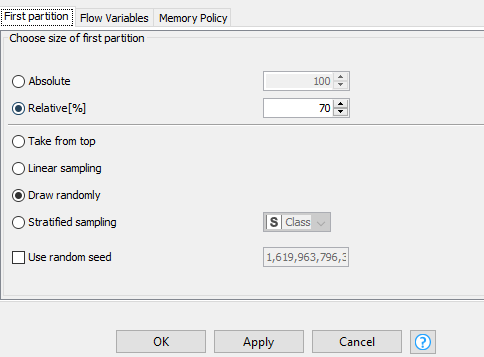


**Feature Split**

The features are divided into two categories (row-wise), such as train and test set, with the train set accounting for 70% of the data and the test set for 30%.



The partitioning node selects rows randomly and allows you to specify a fixed seed, as shown in the configuration below.



**Modelling**

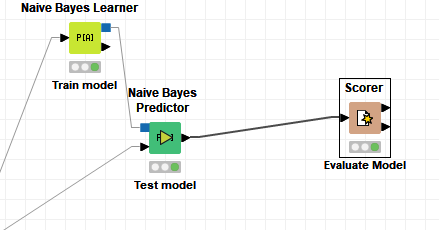
Training a Machine Learning algorithm to predict labels from features, tuning it for the business need, and validating it on holdout data are all part of this process. The result of modelling is a trained model that can infer new data points and make predictions.

Naive Bayes and Random Forrest Classifiers were chosen to perform this task since it is a classification task.

**Naïve Bayes Classifier**

The Naive Bayes classifier is a probabilistic machine learning model for classification problems.

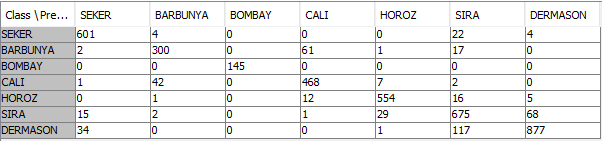
The Naive Bayes Learner node calculates the number of rows per attribute value per class for nominal attributes and the Gaussian distribution for numeric features from the provided training data to construct a Bayesian model.

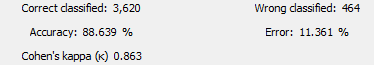


The Naive Bayes Predictor Node predicts the class per row based on the learned model. The product of the probability of each attribute and the probability of the class attribute itself is the class probability.

The Scorer node evaluates the model. When their attribute-value pairs compare two columns, the confusion matrix, or how many rows of which attribute and classification fit, is shown.

**Findings:**

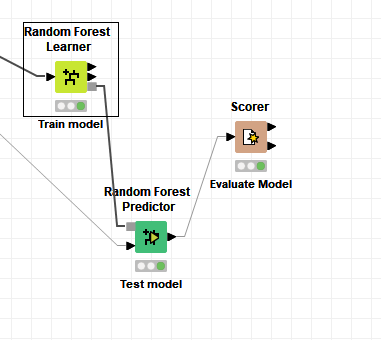




According to the Confusion Matrix, the model correctly classified 3,620 of 4,084 dry bean images with an accuracy score of 88.6% with an error rate of 11.36%, which shows that it is a fairly good model.

**Random Forrest Classifier**

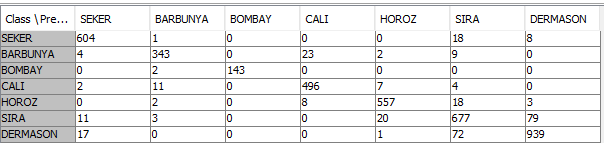
The Random Forest Classifier creates several decision trees and then merges them to derive an accurate and consistent forecast.

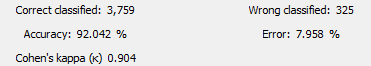


The Random Forest Node learns by picking a few decision trees at random. The decision tree model is trained on different rows (records) and columns (descriptive attributes.

In a random forest model, the Random Forest predictor aggregates the predictions of individual trees to predict trends. T The scorer assesses the model's performance and generates a result.

**Findings:**

****



With an accuracy score of 92% and an attrition rate of 8%, the random forest model correctly predicted 3,759 of 4,084 dry bean images.

**Conclusion**

The project aimed to classify images of dry beans using two classification algorithms correctly. Because of its adaptability and ability to minimise decision tree overfitting, which improves precision, the Random Forest Classifier obtained the best result, correctly classifying the dry beans Images with a minute error rate. The Knime Analytics platform helps streamline the operations and is a codeless environment that makes it easy to use.