**RIGA TECHNICAL UNIVERSITY**

Faculty of Computer Science, Information Technology and Energy

**Report on the second practical assignment**

Study course "Fundamentals of artificial intelligence"

Team number: 8

Students:

Baturalp Bilen Burmaoglu

211ADB042

Monish Ramachandani 211ADB054

Pavlo Nikolaiev 201ADB090

Teaching staff: Alla Anohina-Naumeca

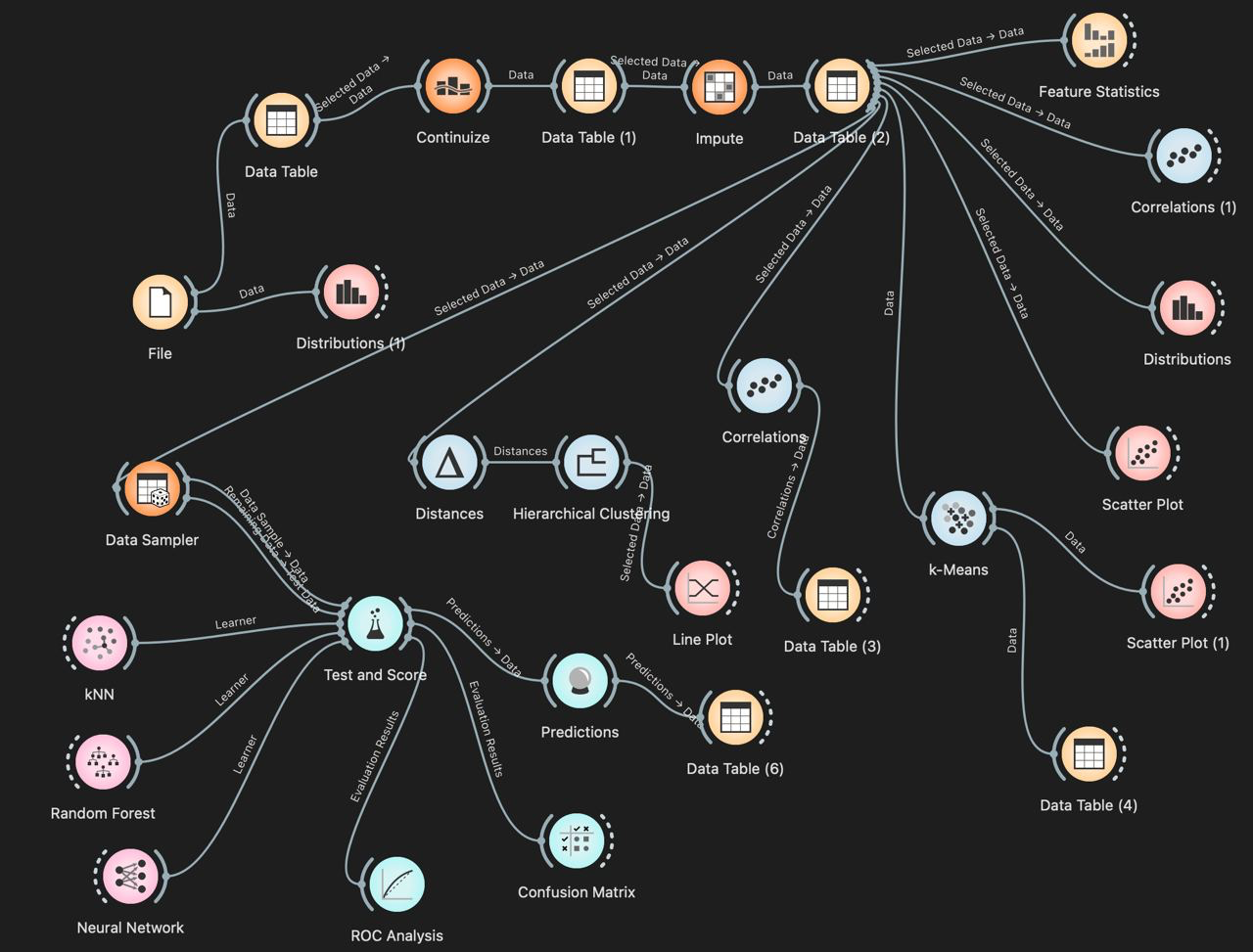
Project link: <https://github.com/Pashost/PW2_Ai_team_8>

Link to dataset:

<https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality>

2023/2024 academic year

# Orange tool workflow



# Part I

<this subsection should provide a general description of the dataset, accompanied by screenshots and references to the information sources used>

## Description of the dataset

### Dataset title: Apple Quality Explore the World of Fruits

### Dataset source: <https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality>

### Creator and/or owner of the dataset: Nidula Elgiriyewithana

*Description of the dataset problem domain:* This dataset contains information about various attributes of a set of fruits, providing insights into their characteristics. The dataset includes details such as fruit ID, size, weight, sweetness, crunchiness, juiciness, ripeness, acidity, and quality.

*Dataset licensing conditions:* It is written as “Other (specified in description)” without any usability restrictions.

*Information about the method or procedure for collecting the dataset:* The dataset was generously provided by an American agriculture company. The data has been scaled and cleaned for ease of use.

## Description of the dataset content

*Number of data objects in the dataset:* **4000**

| **Class label** | **Number of data objects** |
| --- | --- |
| **Quality** | **4000** |

*Representation of features (attributes) of the dataset together with their roles in the Orange tool:*

metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

### Because "A\_id" represents an identity number for the apples, we have designated its role as 'skip' to exclude it from the analysis.

### Number of classes in the dataset:

There are two classes with Quality attribute as Good and Bad.

*Description of classes:*

<labels used to represent classes and the meaning of each class; if the dataset provides several possible data classifications, then the report should clearly identify which classification is being addressed in the assignment>

The quality attribute classifies apples into 'good' and 'bad' categories, representing their quality.

*Number of data objects belonging to each class:*

<add rows to table as needed>

| **Class label** | **Number of data objects** |
| --- | --- |
| Good | 2004 |
| Bad | 1996 |

### Description of features:

| **Feature title** | **Explanation of the feature** | **Value type** | **Range of values** |
| --- | --- | --- | --- |
| Size | Fruit size | float | -7.151703059 - 6.406366899 |
| Weight | Fruit weight | float | -7.149847675 - 5.79071359 |
| Sweetness | Fruit sweetness | float | -6.894485494 - 6.374915513 |
| Crunchiness | Texture indicating the crunchiness of the fruit | float | -6.055057805 - 7.619851801 |
| Juiciness | Level of juiciness of the fruit | float | -5.961897048 - 7.364402864 |
| Ripeness | tage of ripeness of the fruit | float | -5.864598918 - 7.237836684 |
| Acidity | Acidity level of the fruit | float | -7.010538475 - 7.404736238 |
| Quality | Overall quality of the fruit | Boolean | bad - good |

### 

### Data file structure:

<a screenshot showing all the columns in the data file and their values for at least some data objects>

metin, ekran görüntüsü, kalıp, desen, düzen içeren bir resim

Açıklama otomatik olarak oluşturuldu

### Information about missing values or outliers:

<a description of whether values of certain features (attributes) are missing in the dataset or whether there are outlier values and solutions that the students have used to solve the mentioned problems (demonstrated also by screenshots)>

## When the dataset was downloaded, it contained 4,001 rows. The last row (the 4,001st) contained only the dataset creator's name and has been removed in Excel. No other data manipulation techniques were applied. The dataset is complete with no missing data.

## Visual and statistical representation of the dataset

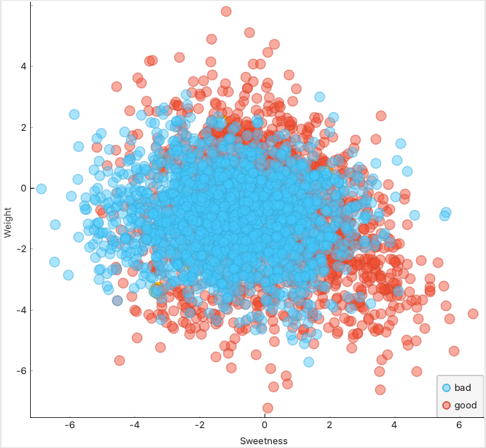
<a scatterplot screenshot>

ekran görüntüsü, renklilik içeren bir resim

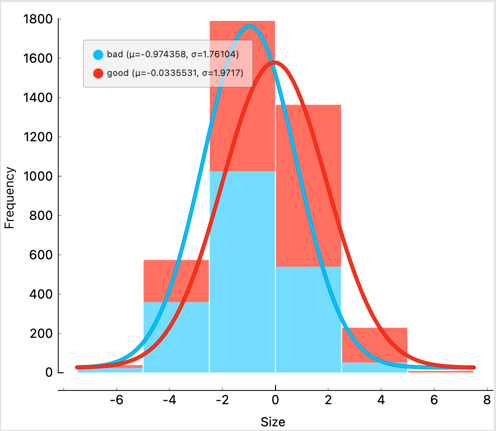
Açıklama otomatik olarak oluşturuldu

<a scatterplot screenshot>

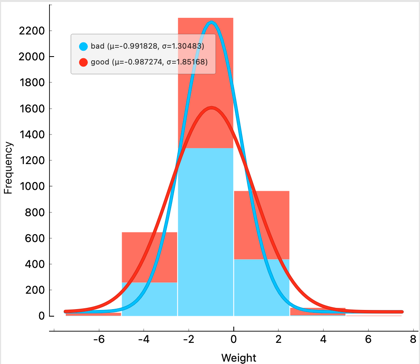
ekran görüntüsü, renklilik, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

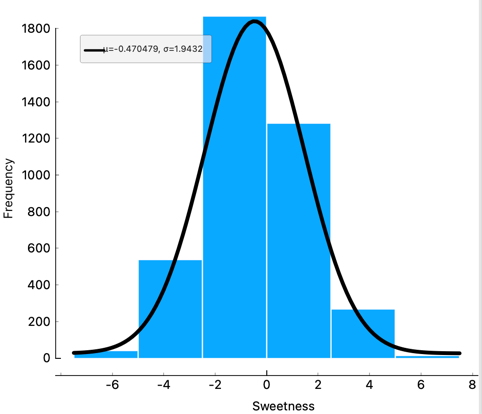
<a histogram screenshot>



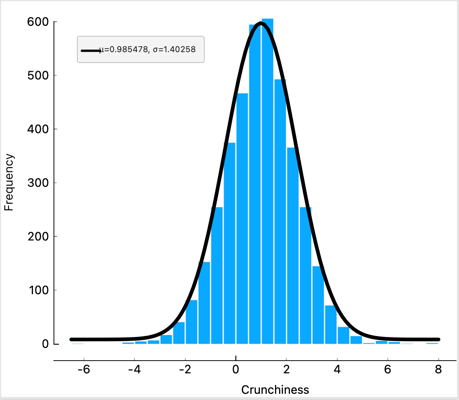
<a histogram screenshot>



<a screenshot of feature distribution>



<a screenshot of feature distribution>



<a screenshot with statistics>

metin, yazılım, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

## Answers to questions

<answers the questions below, referring to the screenshots above and providing an analysis of the results>

### Are the classes in the dataset balanced, or does one class (or several classes) prevail?

### The dataset is balanced. The Good class has 2,004 instances, and the Bad class has 1,996 instances.

### Does the visual representation of the data allow you to see the structure of the data?

### Yes, the distribution graphs clearly represent the attributes across different classes, allowing for easy interpretation. Across all characteristics, the graphs show a significant overlap between the distributions of 'good' and 'bad' quality apples. The curves for both classes are closely aligned, with 'bad' generally on the left side transitioning into 'good' on the right as the value of each characteristic increases.

### How many data groupings can be identified by studying the visual representation of the data?

Two groups have been pre-identified as Good and Bad in the dataset.

### Are the identified data groupings close to each other or far from each other?

## Based on the scatter diagrams that use different variables marked in orange (utilizing the 'find informative projections' feature), it is challenging to identify and classify the groups easily due to the lack of highly discriminative variables.

## Conclusions arising from the analysis of statistical indicators

<analysis of statistical indicators by referencing specific values>

When the features examined it can be seen that; Size has a mean of approximately -0.503 and a median very close to the mean at -0.514, indicating a symmetric distribution. The distribution, however, spans a wide range from -7.152 to 6.406, suggesting substantial variability in apple sizes within the dataset. Weight follows a similar pattern with a mean of -0.989 and a median of -0.985, also suggesting a symmetric distribution. The range from -7.150 to 5.791 indicates significant variation in weight, similar to size.

Sweetness shows a mean of -0.471 with a median of -0.505. The close values of mean and median suggest a fairly symmetric distribution, but with a wide range from -6.894 to 6.375, indicating diverse sweetness levels among the apples.

Crunchiness is notably skewed, as evidenced by a mean of 0.986 and a median closer to 1.000 at 0.999. The positive skew is further indicated by a range extending more significantly on the positive end (-6.056 to 7.619).

Juiciness presents a slightly positive mean of 0.512 with a median of 0.534. The range from -5.962 to 7.364, similar to crunchiness, suggests a broad spectrum of juiciness in the apples.

Ripeness has a mean near the median (0.499 and 0.503 respectively), which indicates a symmetric distribution. Its range from -5.865 to 7.238 shows varying degrees of ripeness across the dataset.

Acidity is different from the other attributes with a very low mean (0.077) compared to its median (0.023), yet it features the widest range from -7.010 to 7.405, which suggests a significant spread in acidity levels, indicative of very diverse apple types with potentially varying flavours.

The Quality attribute is categorical with two classes, good and bad. The mean of 0.693 indicates that approximately 69.3% of the dataset samples are classified as good, showing a slight imbalance but still providing a substantial representation of both classes.

The dataset presents a comprehensive picture of the variability in apple characteristics, with all quantitative attributes showing substantial ranges, indicating a diverse sample set. The balance in class distribution for quality suggests the dataset is fairly representative, although slightly skewed towards good quality apples. Such data is crucial for developing predictive models that aim to classify apples based on quality or for use in agricultural research to understand factors influencing apple characteristics.

# Part II

<this subsection should describe the use of unsupervised machine learning algorithms, accompanied by screenshots and references to the information sources used>

## In this assignment, we applied hierarchical clustering and k-means algorithms. For hierarchical clustering, we experimented with different linkage measures. Initially, distances were calculated using the Distances widget with the 'Rows' and 'Euclidean (normalized)' options selected. Subsequently, we employed three linkage measures: single, average, and complete. The rationale behind this selection was to examine the effects of the shortest (single), farthest (complete), and a compromise approach (average) linkage method, facilitating an easier decision-making process regarding the optimal method.

## Hierarchical clustering

### Hyperparameters available in the Orange tool:

<add rows to table as needed>

| **Hyperparameter** | **Description** |
| --- | --- |
| Linkage | Complete |
| Pruning | None |
| Height Ratio | 80% |

<a screenshot with hyperparameter values set for the algorithm>

metin, yazılım, diyagram, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

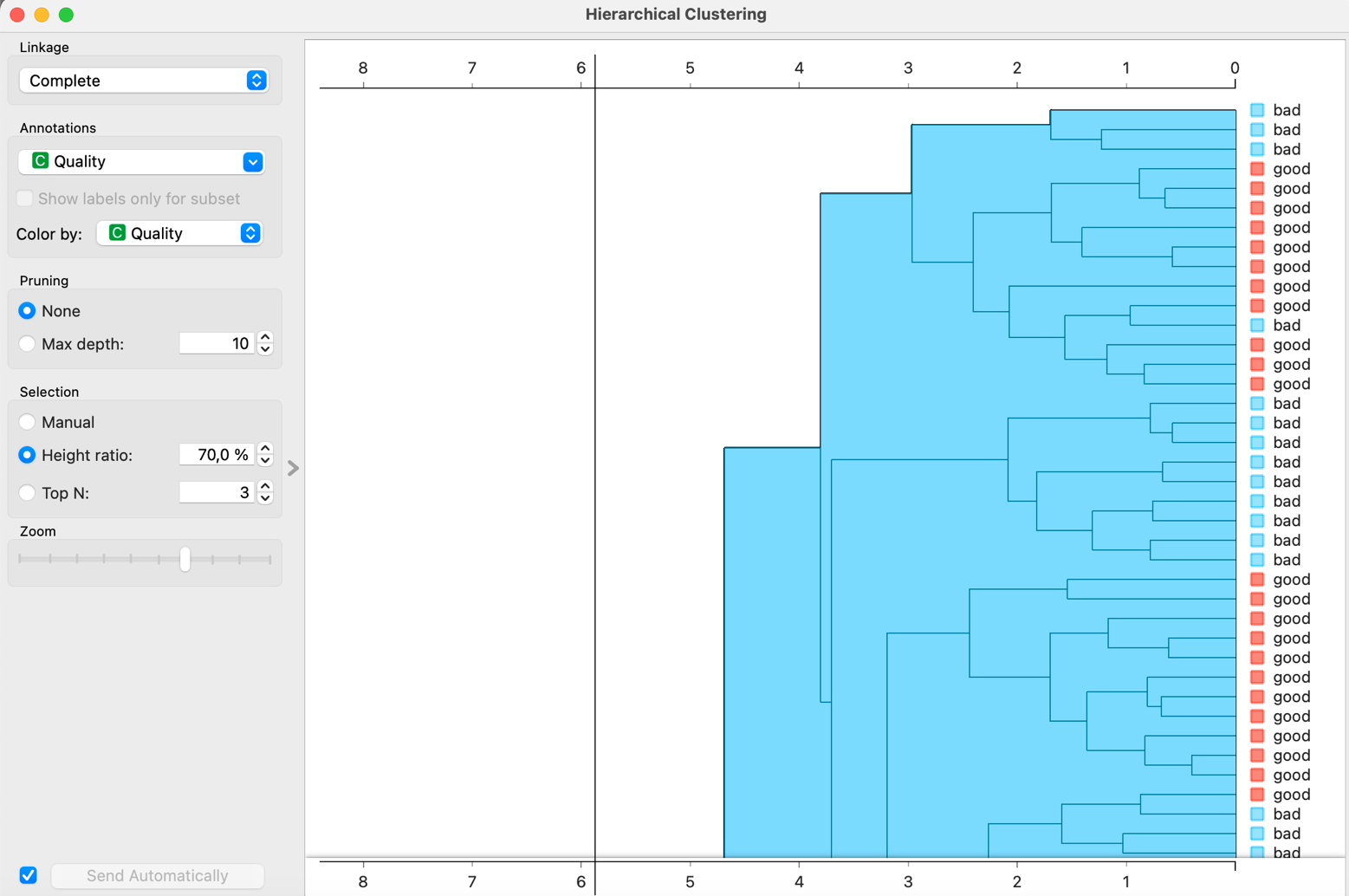
### Description of experiments

<a screenshot for Experiment 1 with a certain position of the horizontal cut-off line>

yazılım, diyagram, multimedya yazılımı, grafik yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

<a screenshot for experiment 2 with a certain position of the horizontal cut off line >



<a screenshot for Experiment 3 with a certain position of the horizontal cut off line >

yazılım, diyagram, grafik yazılımı, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

### Conclusions from experiments:

<a description of how the number and content of clusters change according to the position of the horizontal cut off line, and conclusions about whether class separation is achieved referring to and analysing the screenshots above>

Height Ratio at 90%: This setting leads to a relatively smaller number of larger clusters. Most of the clusters at this height consist of mixed classes, but with a predominance of 'good' quality apples. Notably, there are fewer instances where 'bad' quality apples are isolated into individual clusters.

Height Ratio at 80%: Lowering the height ratio increases the number of clusters. The clusters become more defined and slightly smaller in size compared to the 90% setting. The separation between 'good' and 'bad' quality apples improves slightly, suggesting a better granularity in distinguishing between the two categories.

Height Ratio at 70%: Further lowering the height ratio to 70% results in a higher number of smaller clusters. This adjustment leads to a more pronounced separation between 'good' and 'bad' classes, with clusters more likely to be homogeneous in terms of apple quality.

At a higher height ratio (90%), clusters generally contain a mix of both 'good' and 'bad' quality apples, although 'good' apples predominate. The clusters are larger, indicating a less refined classification.

At intermediate (80%) and lower height ratios (70%), the clusters begin to show a more distinct separation of quality classes. Particularly at the 70% height ratio, many clusters become exclusive to one quality class, either entirely 'good' or entirely 'bad'. The complete linkage method, known for its tendency to create more compact clusters, shows a progressive improvement in class separation as the height ratio is decreased. This suggests that lower height ratios may be more effective in distinguishing between the quality classes in this dataset when using complete linkage. From the visual analysis, a height ratio of around 70% appears to offer the best balance between the number of clusters and the purity of class separation. This setting enables more clusters with greater homogeneity in terms of apple quality, which is desirable for analytical purposes.

## K-means algorithm

### Hyperparameters available in the Orange tool:

<add rows to table as needed>

| **Hyperparameter** | **Description** |
| --- | --- |
| From 2 to 6 | It means that from 2 clusters to 6 clusters try clustering. |
| Normalize columns | Normalize the variables to not disturb the dataset based on different ranges of attributes. |
| Initialize with Kmeans++ | K-means++ is the standard K-means algorithm coupled with a smarter initialization of the centroids. This algorithm ensures a smarter initialization of the centroids and improves the quality of the clustering. Apart from initialization, the rest of the algorithm is the same as the standard K-means algorithm (https://www.geeksforgeeks.org/ml-k-means-algorithm/) |
| Re-Runs | 10 |
| Maximum Iterations | 300 |

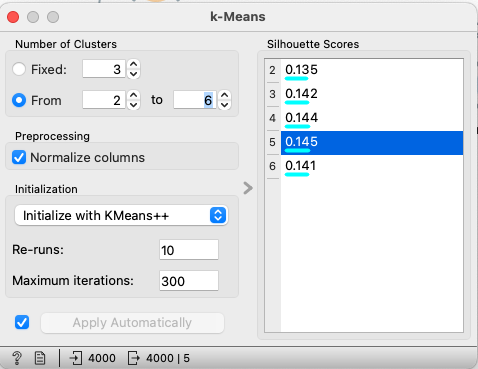
<a screenshot with hyperparameter values set for the algorithm>

metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

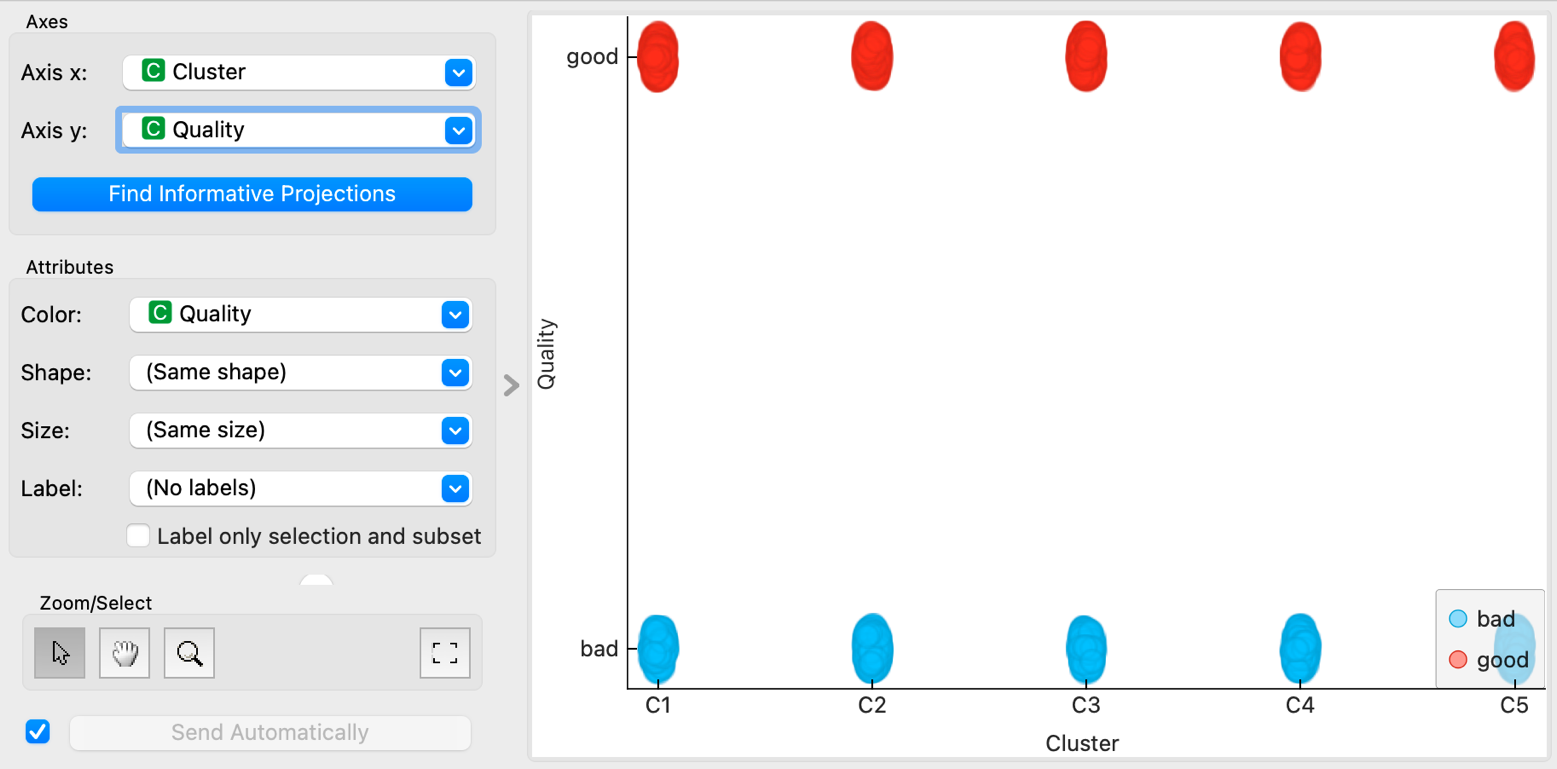
Açıklama otomatik olarak oluşturuldu

### Description of experiments

<a screenshot of Silhouette coefficient with at least 5 different k values>



<a screenshot of the scatterplot according to the best value of the Silhouette coefficient>



### Conclusions from experiments:

<a description of whether the classes are separable referring to and analysing the screenshots above>

The silhouette scores range from 0.135 to 0.145 across different numbers of clusters, with the highest score being 0.145 for a configuration of 5 clusters. This score is moderate, indicating that while there is some structure to the clusters, the separation between them is not particularly distinct.

The scatter plot visualization of clusters (C1 to C5) coloured by apple quality shows that clusters are not perfectly homogeneous in terms of quality. While some clusters predominantly contain apples of one quality (good or bad), the separation is not clear-cut. For instance, clusters C2 and C3 show a mixture of both good and bad qualities, although some clusters like C1, C4, and C5 seem more dominated by one quality category.

The classes are not distinctly separable using the k-Means clustering method under the tested parameters. While some clusters tend to favour one quality over another, the overlap in apple quality across clusters suggests that apple quality (good or bad) is not strongly discriminative with the features used or possibly is influenced by non-linear relationships not captured well by k-Means.

The moderate silhouette scores corroborate the visual observation that clusters are not entirely homogeneous or clearly separable by apple quality. This implies that either more sophisticated clustering techniques might be needed, such as those capable of handling complex patterns or non-linear separations, or additional features should be considered to better distinguish between the qualities.

The cluster count of 5, yielding the highest silhouette score, suggests a slight edge over other configurations but does not significantly improve class separability. This observation points to an inherent complexity in the dataset that simple linear separability cannot resolve.

## Final conclusions

<conclusions whether the classes in the dataset are well or poorly separable based on the analysis of the performance of the two algorithms>

Both clustering methods highlight challenges in distinctly separating apple quality classes. Although hierarchical clustering at lower height ratios shows a somewhat better class separation compared to k-means, neither method achieves clear-cut distinction across the dataset.

The silhouette scores from k-means, ranging from 0.135 to 0.145, and the observed overlap of classes in various clusters indicate moderate separability. This suggests that the apple quality classes are not strongly discriminative based on the features used, which could be due to non-linear relationships not well-captured by linear clustering techniques like k-means.

Hierarchical clustering improved in terms of class homogeneity as the height ratio was lowered, suggesting that more refined subdivisions lead to a better grouping of similar qualities. However, the complete linkage method’s tendency to create compact clusters only provided substantial class separation at much lower height ratios.

In k-means clustering, while some clusters predominantly contained apples of one quality, others showed a significant mix, reflecting similar challenges in achieving cluster homogeneity.

*Comparative Effectiveness:*

Hierarchical Clustering: This method showed a gradual improvement in the separation between quality classes with the decrease in height ratio, particularly at a 70% height ratio. This indicates that hierarchical clustering, with appropriate tuning of the height ratio, can somewhat effectively group apples by quality, but still faces limitations in cleanly separating all classes.

k-Means Clustering: The results suggest that k-means struggles more with class separability compared to hierarchical clustering, especially in handling the inherent complexities and mixed quality distributions within the dataset. The best results were obtained with five clusters, yet this did not lead to a significant improvement in class separability.

The dataset likely contains inherent complexities, such as non-linear separations between classes or overlapping features across classes, which are challenging for these traditional clustering algorithms to handle effectively. The moderate success of hierarchical clustering at lower height ratios and the general struggle of k-means underscore the potential need for more sophisticated clustering techniques or machine learning models that can handle complex patterns, such as those based on density or distribution models. Given the limited success in unsupervised class separation, applying supervised learning techniques might be more effective if labels are reliable. These could leverage the nuances in the dataset more effectively by learning from the labelled instances of apple quality.

# Part III

<this subsection should describe the use of supervised machine learning algorithms, accompanied by screenshots and references to the information sources used>

## Description of the selected algorithms

<a description of freely chosen algorithms and the rationale for their choice (except artificial neural network)>

### Title of the first algorithm: Artificial Neural Network

### Description of the first algorithm: A neural network is a type of machine learning model designed to make decisions by simulating how the human brain operates. It uses a structure akin to biological neurons, with layers of artificial neurons or nodes that include an input layer, several hidden layers, and an output layer. Each node has a specific weight and threshold; it activates and passes data to the next layer if its output surpasses this threshold, otherwise, it remains inactive. Neural networks must be trained using data to enhance their accuracy gradually. Once optimised, these networks become robust tools in computer science and artificial intelligence, capable of classifying and clustering data quickly and effectively.

### Title of the second algorithm: Random Forest

### Description of the second algorithm:

### This method creates multiple trees at training time and gives the mean prediction of the individual trees.

### 

### Hyperparameters in Orange Tool: For this algorithm we’ll check the number of trees included in the forest and the variables/ attributes needed to split them

### This is an ensemble learning method that operates by constructing multiple decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. We can also use depth as a hyperparameter too.

### Title of the second algorithm: kNN

### Description of the second algorithm: The k-nearest neighbors (KNN) algorithm is a classification (or regression) method that predicts the classification of a point by aggregating the classifications of the K closest points. This technique is considered supervised because it classifies points based on the established classifications of neighboring points. Unlike KNN, which does not generate cluster information for individual nodes, the K-means algorithm actively forms clusters, assigning each node to a specific group.

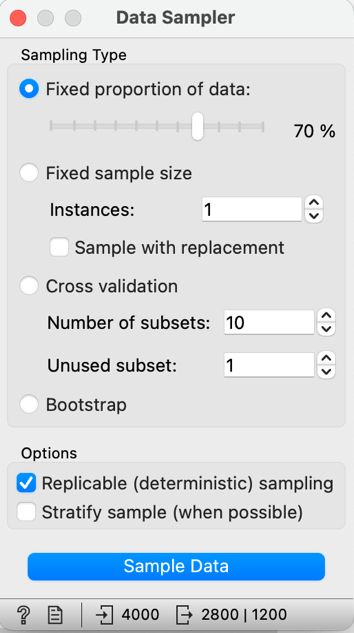
## Description of hyperparameters

<a description of the hyperparameters available in the Orange tool should be given for each of the algorithms, adding rows to the table as necessary>​​​​

| **Hyperparameter** | **Description and values** |
| --- | --- |
| Artificial Neural Networks | |
| Neurons in Hidden Layer | 10 0, |
| Activation | ReLu |
| Solver | Adam |
| Random Forests | |
| Number of Trees | 10 |
| Do not split subsets smaller than | 5 |
|  |  |
| kNN | |
| Number of Neighbours | 5 |
| Metric | Euclidean |
| Weight | Uniform |

## Information about test and training datasets

<a screenshot of splitting dataset into test and training datasets>



### Number of data objects in the training dataset:

2800

### % proportion of data objects in the training dataset:

<add rows to table as needed>

| **Class label** | **Number of data objects in the training dataset** | **% proportion of data objects in the training dataset** |
| --- | --- | --- |
| Good | 1410 | 50.36 |
| Bad | 1390 | 49.64 |

### Number of data objects in the test dataset:

1200

### % proportion of data objects in the test dataset:

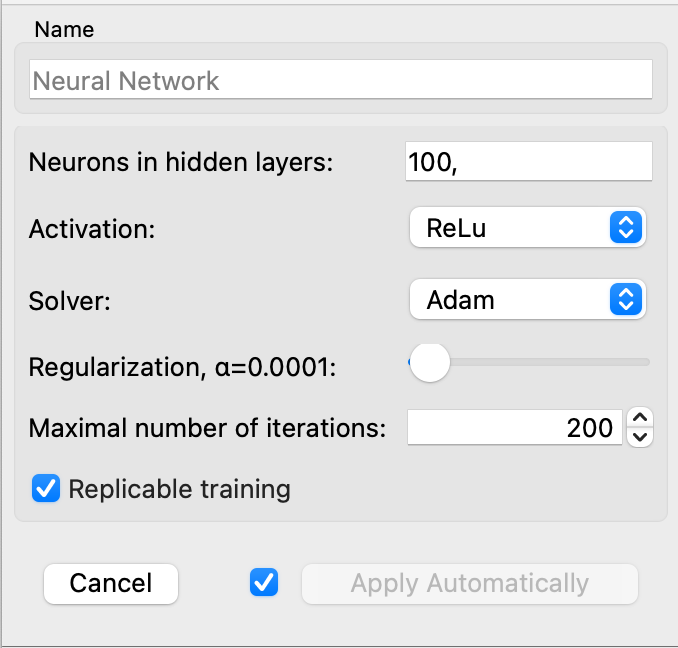
<add rows to table as needed>

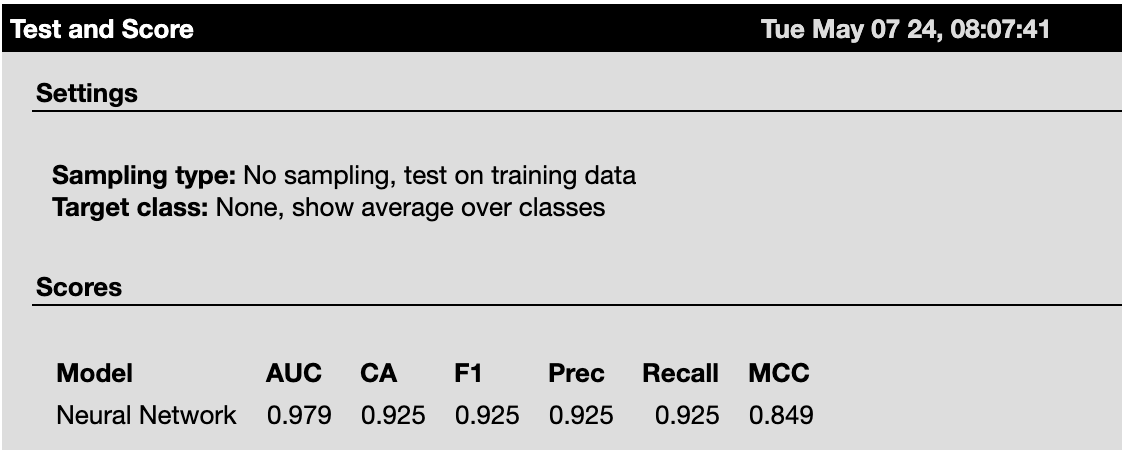
| **Class label** | **Number of data objects in the test dataset** | **% proportion of data objects in the test dataset** |
| --- | --- | --- |
| Good | 594 | 49.5 |
| Bad | 606 | 50.5 |

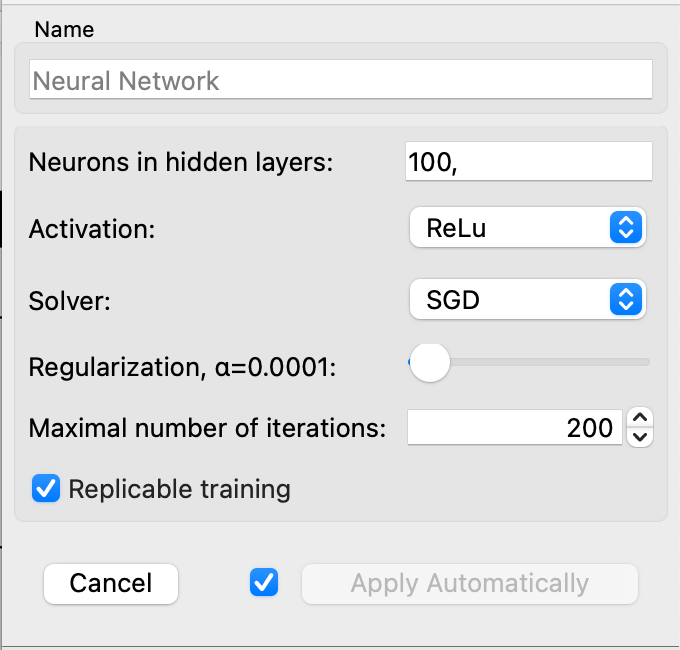
## Experiments with artificial neural network

<add rows to table as needed>

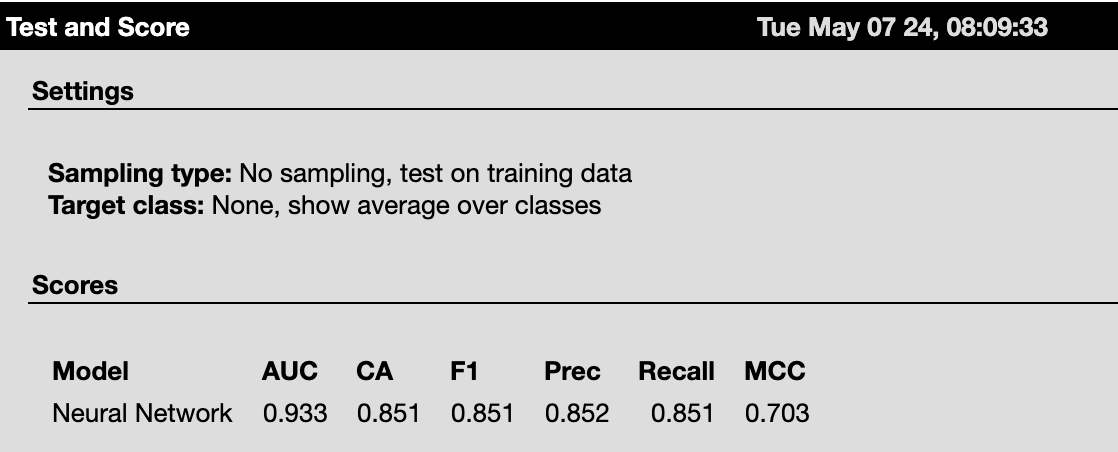
| **Experiment** | **Hyperparameter values** |
| --- | --- |
| Experiment 1 | Activation: ReLu Solver: Adam |
| Experiment 2 | Activation: ReLu Solver: SGD |
| Experiment 3 | Activation: ReLu Solver: L-BFGS-B |

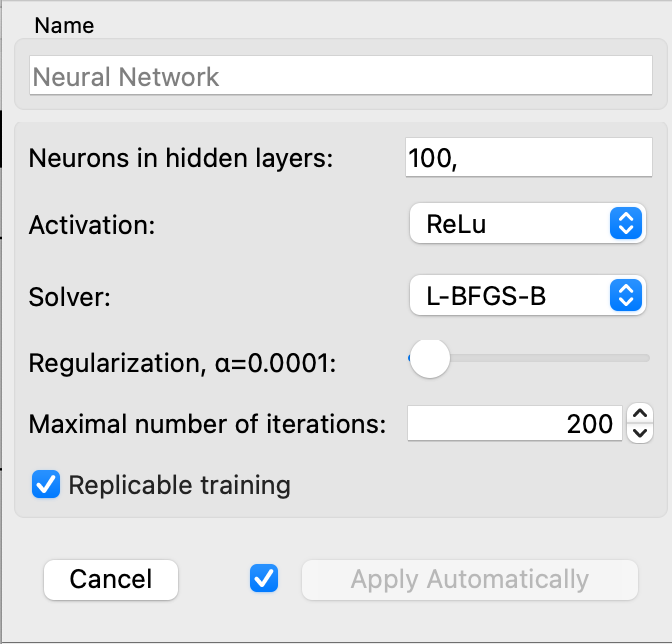
<a screenshot of hyperparameter values for Experiment 1>

<a screenshot of performance metrics for Experiment 1> 

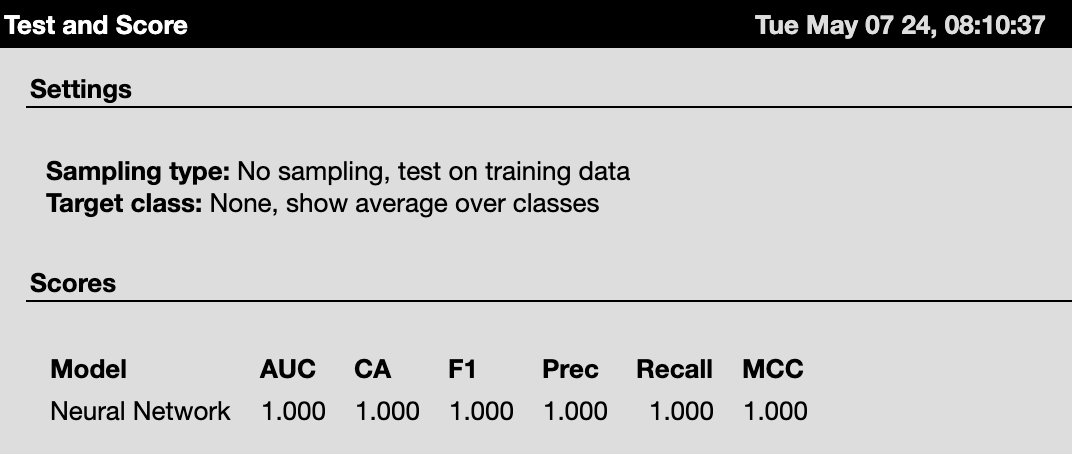
<a screenshot of hyperparameter values for Experiment 2>

<a screenshot of performance metrics for Experiment 2>



<a screenshot of hyperparameter values for Experiment 3>

<a screenshot of performance metrics for Experiment 3>



*Conclusions from experiments:*

<conclusions about the performance of the models in the conducted experiments referring to and analysing the screenshots above>

### Experiment 3, using the L-BFGS-B solver, shows perfect performance across all metrics, which might indicate a very strong fit to the training data. While this could suggest excellent model training, it might also be prudent to check for overfitting by validating this model on a separate test dataset. Comparing the three experiments, it is clear that the L-BFGS-B solver dramatically outperforms the other solvers in terms of all metrics. The Adam solver (Experiment 1) provides robust performance across the board but does not reach the perfection of the L-BFGS-B solver. Experiment 2, using the SGD solver, shows the weakest performance, which might be indicative of the need for more epochs, a different learning rate, or other SGD-specific hyperparameters adjustments. The high scores in Experiment 3 require a cautious approach to ensure that the model's performance is due to genuine learning and not just memorizing the training data. Experiment 1's results suggest a more balanced and potentially more generalizable model.

### Model selected for testing:

<indication of which experimental model is selected for the testing process>

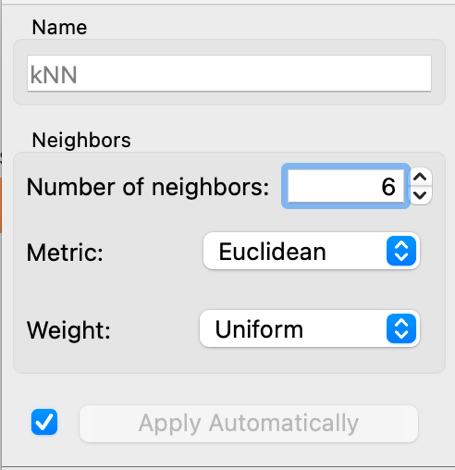
## While Experiment 3's results are impressive, it is recommended to validate these results further to ensure they are not due to overfitting. For further testing on a separate dataset, Experiment 1 with the Adam solver is suggested as a safer choice due to its robustness and good generalization capabilities shown by high but not perfect scores. This makes it likely to perform well on new, unseen data while maintaining good accuracy and reliability.

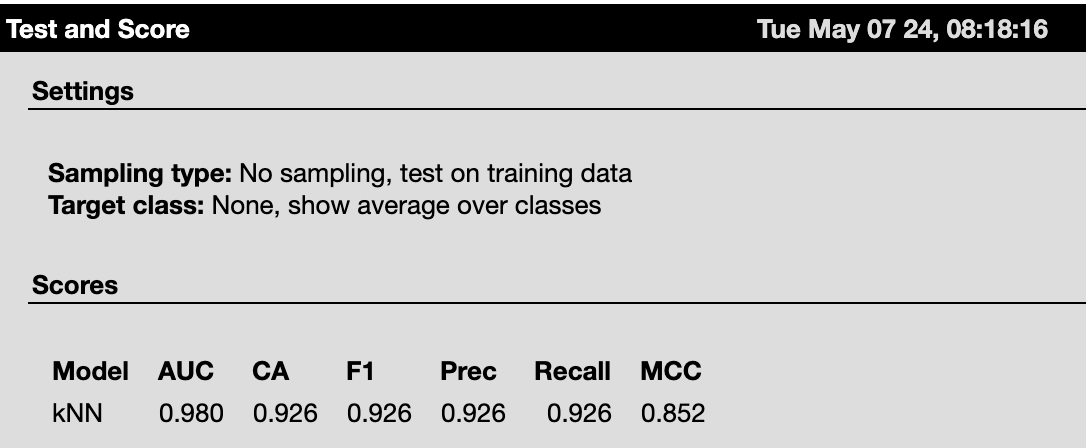
## Experiments with /kNN/

<add rows to table as needed>

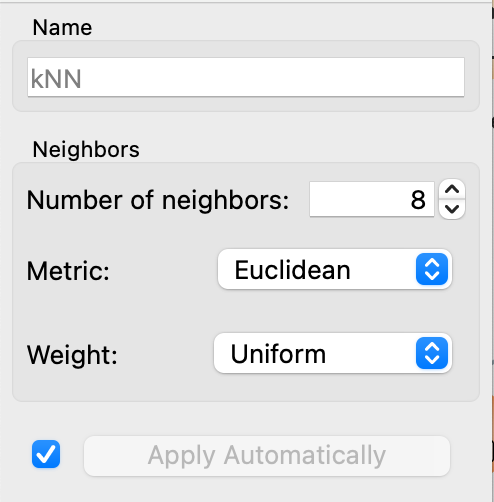
| **Experiment** | **Hyperparameter values** |
| --- | --- |
| Experiment 1 | Number of Nighbours:6 Metric: Euclidean Weight: Uniform |
| Experiment 2 | Number of Nighbours:8 Metric: Euclidean Weight: Uniform |
| Experiment 3 | Number of Nighbours:4 Metric: Euclidean Weight: Uniform |

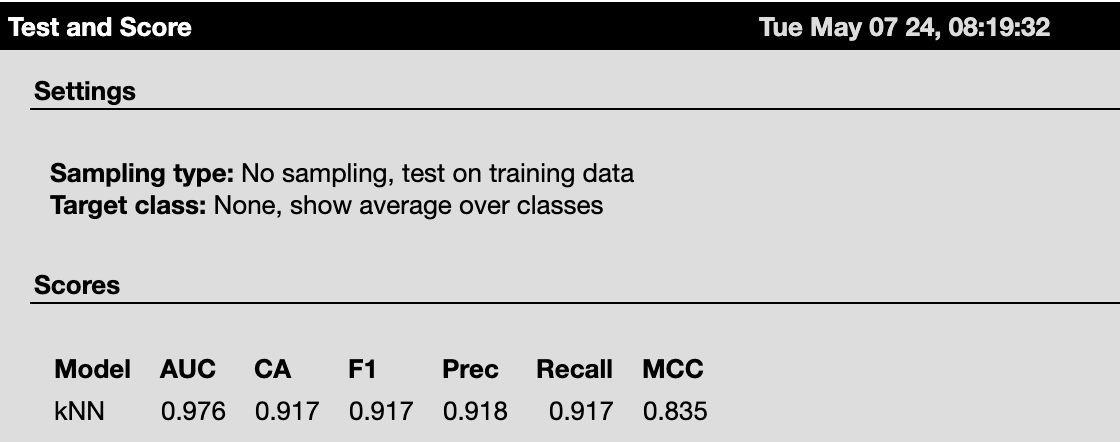
<a screenshot of hyperparameter values for Experiment 1>



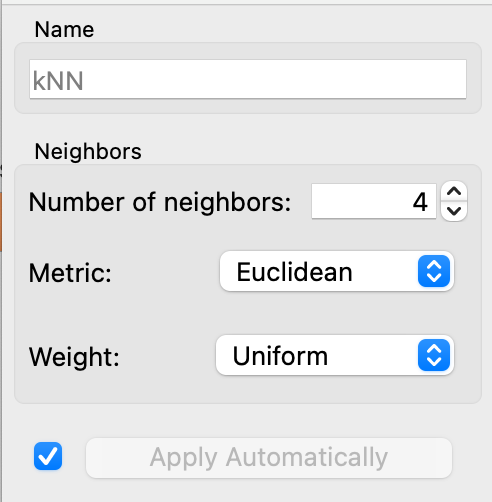
<a screenshot of performance metrics for Experiment 1>

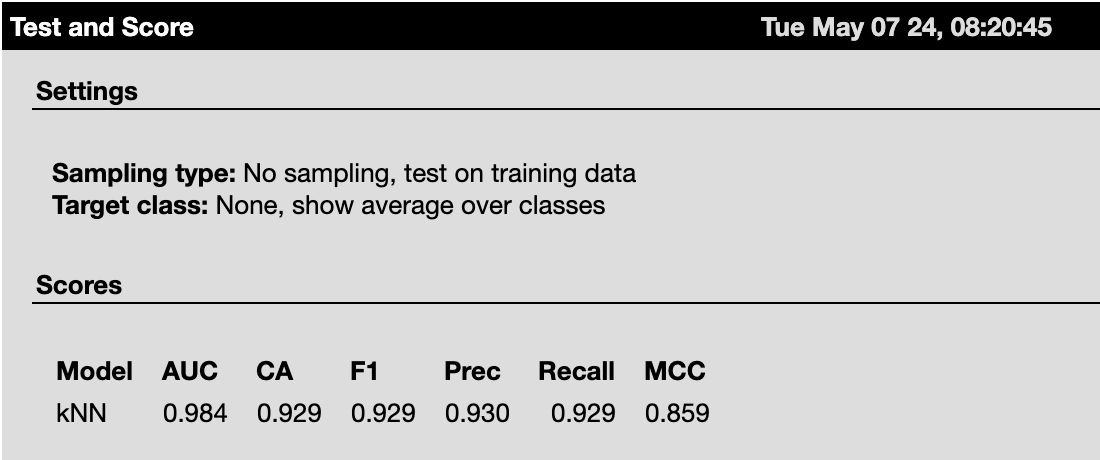
<a screenshot of hyperparameter values for Experiment 2>



<a screenshot of performance metrics for Experiment 2>

<a screenshot of hyperparameter values for Experiment 3>



<a screenshot of performance metrics for Experiment 3>

*Conclusions from experiments:*

<conclusions about the performance of the models in the conducted experiments referring to and analysing the screenshots above>

### The third experiment, where the number of neighbours was set to 4, shows the highest scores across most metrics including AUC, Classification Accuracy, F1 Score, and MCC. This suggests that a smaller number of neighbours results in a more accurate and reliable classification for this specific dataset and settings. Experiment 1 and 3 have better overall performance metrics compared to Experiment 2. It appears that increasing the number of neighbours beyond a certain point (from 6 to 8) has slightly diminished performance, particularly in terms of MCC and overall accuracy. While all three experiments have relatively high precision, Experiment 3 provides the best balance between precision and recall, leading to the highest F1 Score and MCC, which are critical for assessing the overall effectiveness of a classification model, especially when the class distribution is imbalanced.

### Model selected for testing:

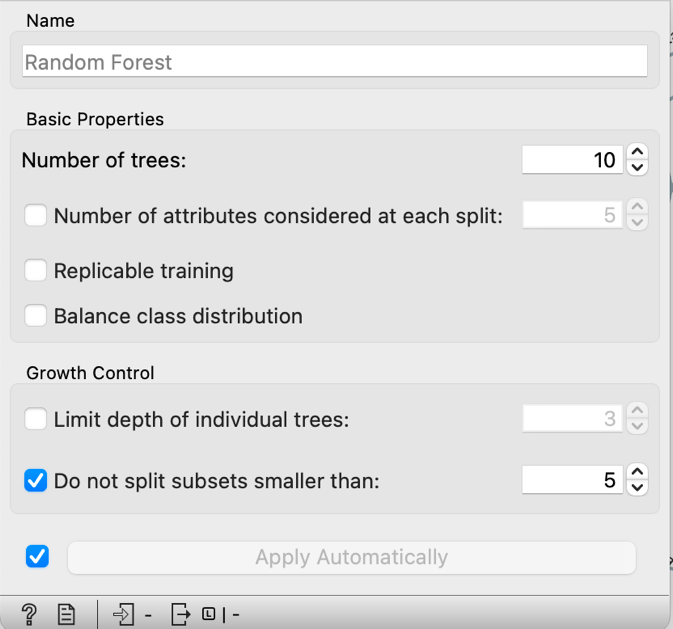
<indication of which experimental model is selected for the testing process>

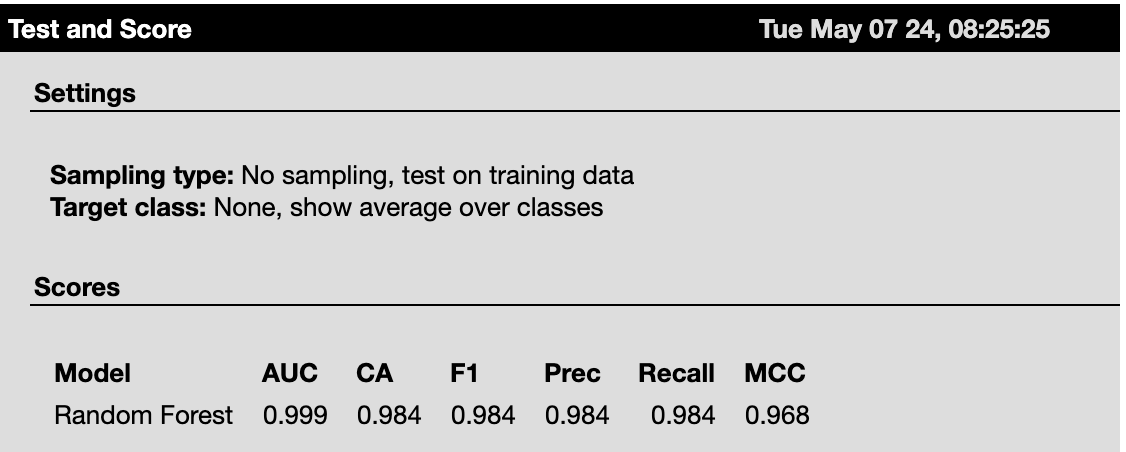
## Based on the analysis, Experiment 3 with 4 neighbours is the most suitable model for further testing. This model not only yielded the highest metrics across the board but also suggests that a tighter neighbourhood results in more precise classifications. It would be advisable to test this model configuration on a separate test dataset to validate its effectiveness in a different or more challenging data environment.

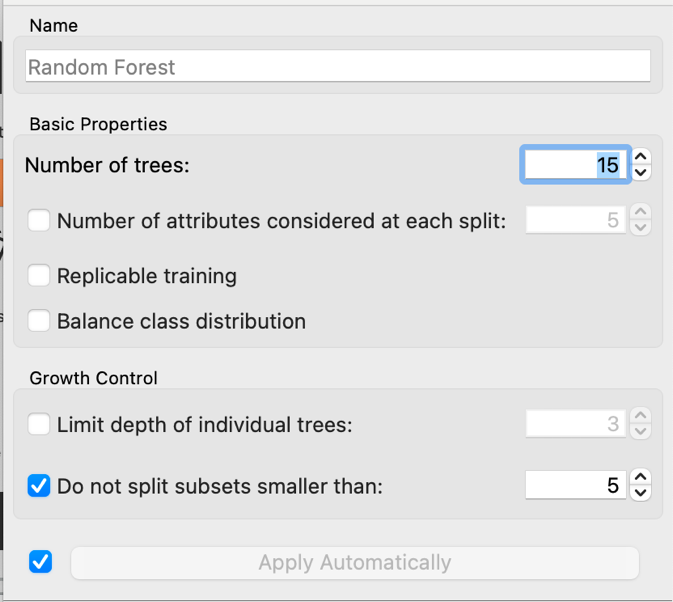
## Experiments with /Random Forests/

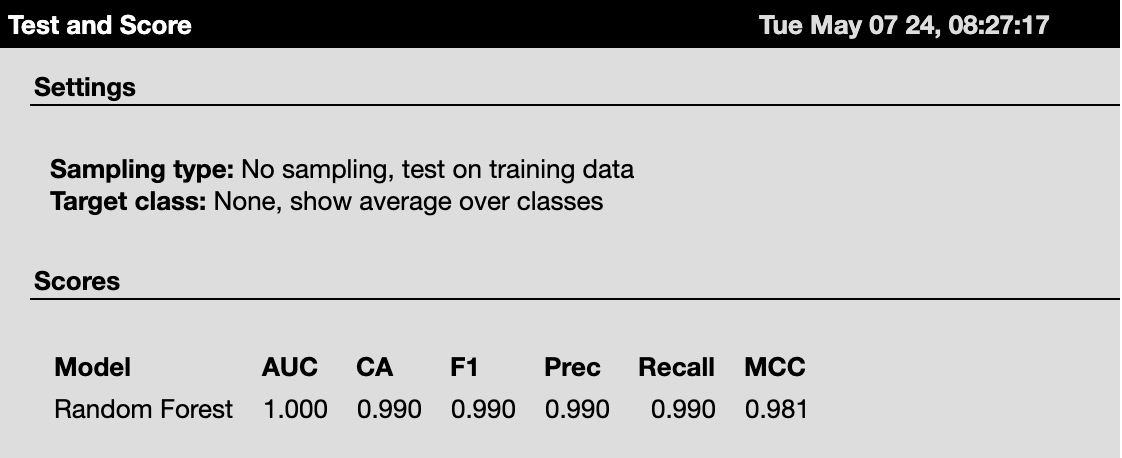
<add rows to table as needed>

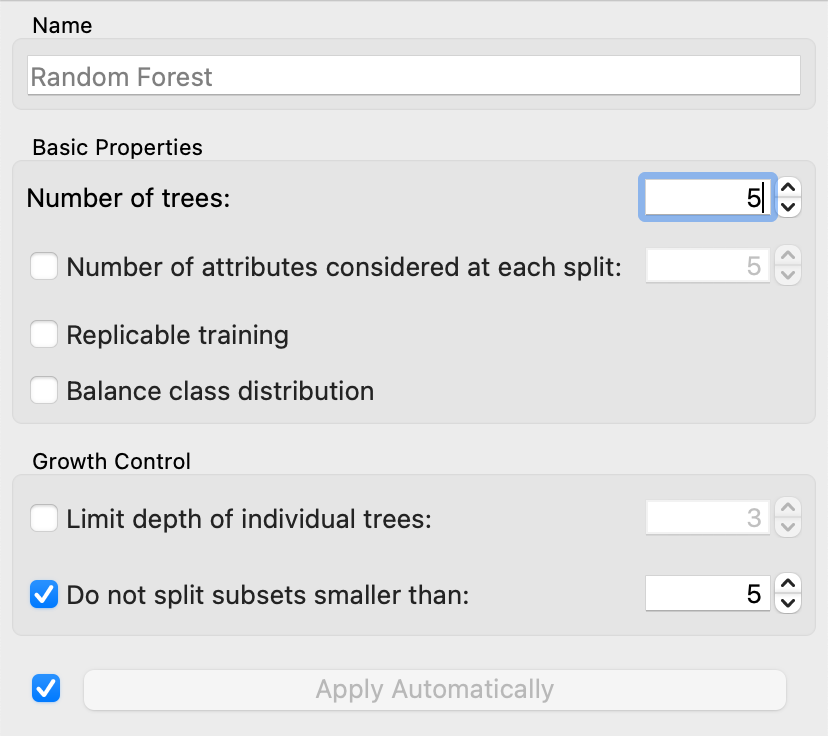
| **Experiment** | **Hyperparameter values** |
| --- | --- |
| Experiment 1 | Number of trees:10 Do not split subset smaller than: 5 |
| Experiment 2 | Number of trees:15 Do not split subset smaller than: 5 |
| Experiment 3 | Number of trees:5 Do not split subset smaller than: 5 |

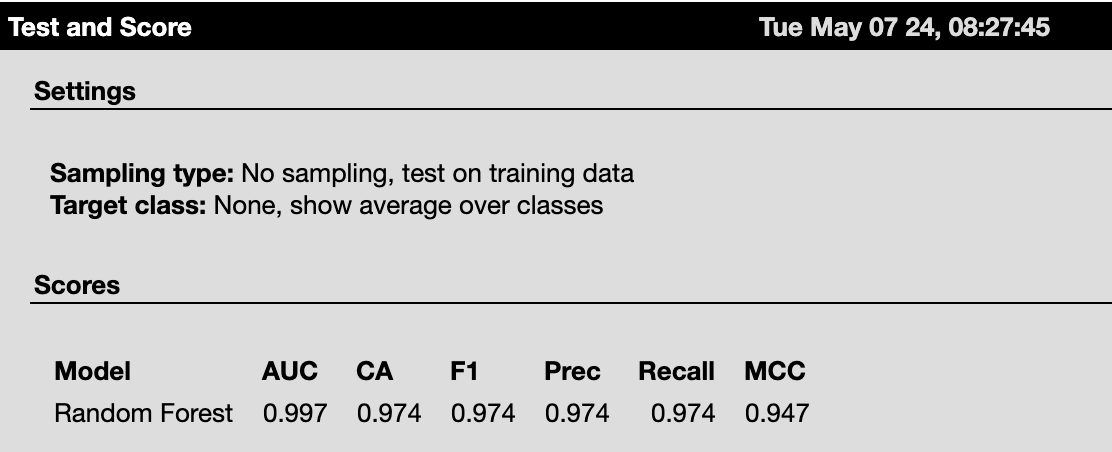
<a screenshot of hyperparameter values for Experiment 1> 

<a screenshot of performance metrics for Experiment 1>

<a screenshot of hyperparameter values for Experiment 2>

<a screenshot of performance metrics for Experiment 2>

<a screenshot of hyperparameter values for Experiment 3>

<a screenshot of performance metrics for Experiment 3>

*Conclusions from experiments:*

<conclusions about the performance of the models in the conducted experiments referring to and analysing the screenshots above >

### Experiment 2, which used 15 trees, demonstrates the highest performance across all metrics, achieving perfect scores in AUC and nearly perfect scores in other metrics. This suggests that a slightly larger number of trees in the Random Forest model enhances its prediction ability and stability without overfitting, as indicated by the high MCC. Experiment 1 and 3, though using fewer trees, also show high performance, particularly Experiment 1 with 10 trees, which balances performance and computational efficiency well. Experiment 3, with the fewest trees, shows a slight drop in all metrics, indicating that reducing the number of trees to 5 may slightly compromise model accuracy and reliability. Increasing the number of trees from 10 to 15 improves the model's accuracy and the consistency of its predictions as evidenced by the improved metrics in Experiment 2 compared to Experiment 1.

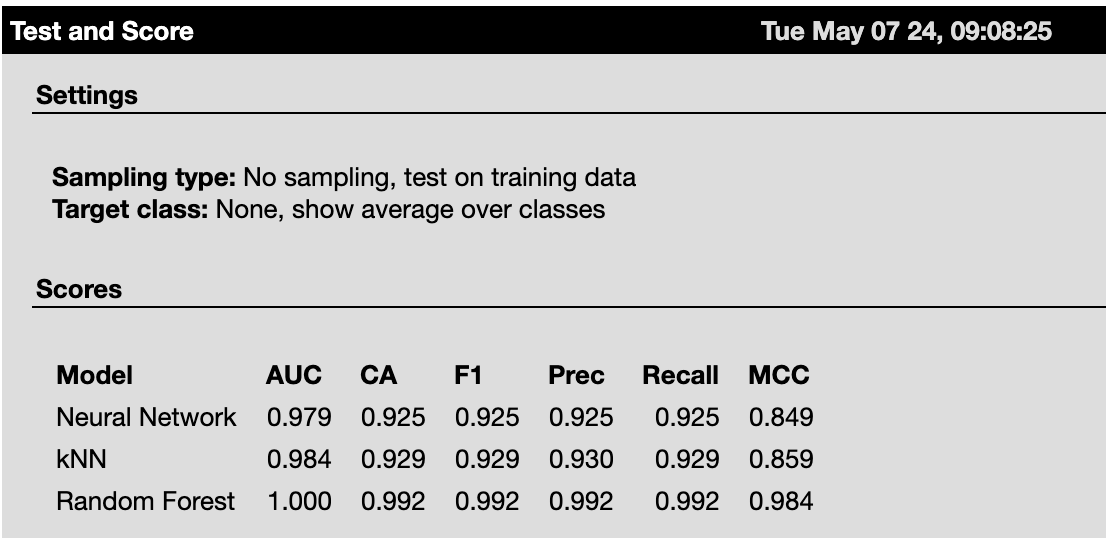
### Model selected for testing:

<indication of which experimental model is selected for the testing process>

Based on the analysis, Experiment 2 with 15 trees is recommended for further testing. This model configuration not only provided the highest metrics but also suggests robust performance, making it a promising candidate for validation on a separate test dataset to confirm its effectiveness and generalizability.

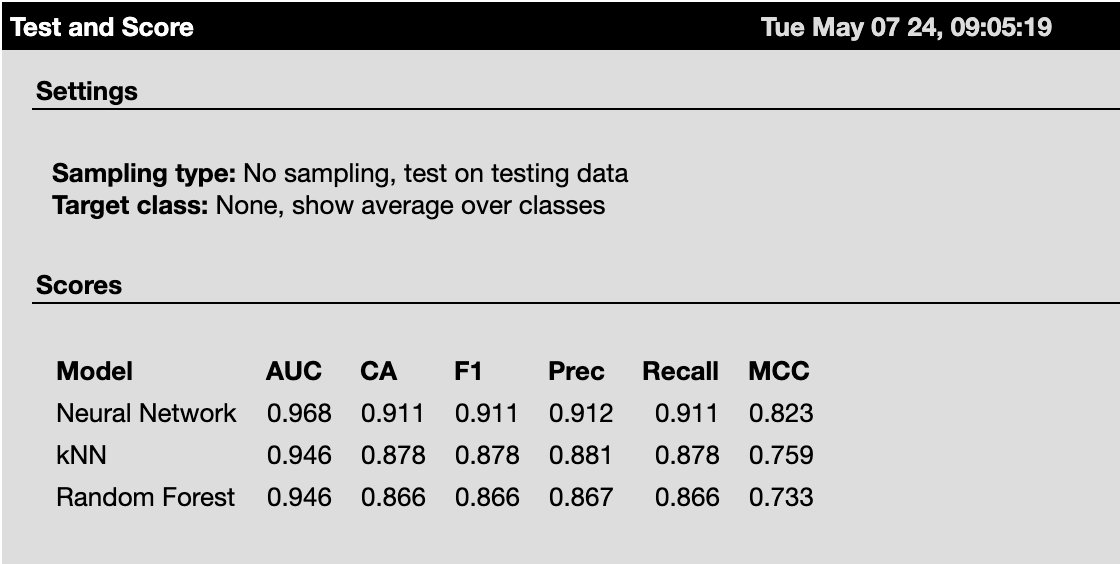
## Testing results of the trained models

<a screenshot of performance metrics of models selected for testing>



*Conclusions after testing:*

<comparison of performance of the trained models in the testing process referring to and analysing the screenshot above >

**

The Neural Network model demonstrates superior performance across all key metrics when compared to kNN and Random Forest. It not only offers the highest Classification Accuracy and F1 Score, but also the highest Matthews Correlation Coefficient (MCC), suggesting that it is more capable of producing reliable and consistent predictions across varied class distributions. The Neural Network maintains a balance between precision and recall, which is crucial for maintaining a low rate of false positives and false negatives. Its precision and recall scores are closely matched, indicating a robust model performance. The AUC of the Neural Network being the highest suggests its superior ability to discriminate between the classes over a range of thresholds. Furthermore, its higher MCC indicates better quality of classification and a higher correlation between the observed and predicted classifications. The kNN model, while outperforming the Random Forest in terms of accuracy and MCC, still lags behind the Neural Network. The Random Forest shows the lowest performance among the three models on the testing data, which could be due to overfitting during training or its inability to generalize as effectively as the other models.

# Conclusion and learning-

After completing this practical we came to the conclusion that Supervised machine learning algorithms are better choice for classification task as in unsupervised ML we need to do a lot of hit and trial methods which means the experimentation with data is increased. especially when data objects are high in number then difficulty of the classification will increase with unsupervised algorithms.

The backlash which comes from the Supervised algorithm is that it requires high computational resources.

# Workflow of orange tool

# 

# 1) Importing data and dataset into an orange tool.

# 2) Examining and changing the variables

# 3) Examining for missing values

# 4) Examining correlation between the different variables

# 5) Scatterplot creation

# 6) Making of a sample size for bar plot

# 7) Bar plot creation.

# 8) Using distance for hierarchical clustering and distance matrix

# 9) Making distributions and feature statistics

# 10)Configure the test score and integrate it with the confusion matrix, forecasts, and machine learning algorithms..

# 

# References

Adams, Ryan P. (2024) Hierarchical Clustering Course Notes, <https://www.cs.princeton.edu/courses/archive/spring19/cos324/files/hierarchical-clustering.pdf> (Access Date: May 6th 2024).

Shahapure, K. R., & Nicholas, C. (2020, October). Cluster quality analysis using silhouette score. In *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)* (pp. 747-748). IEEE.

<https://stats.stackexchange.com/questions/56500/what-are-the-main-differences-between-k-means-and-k-nearest-neighbours>

Bicego, M., & Escolano, F. (2021, January). On learning random forests for random forest-clustering. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 3451-3458). IEEE.

<https://www.ibm.com/topics/neural-networks>

<https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality>