**Fundamentals of Artificial Intelligence**

**second practical assignment**



Akil Arvinth Malayandipattinam Vasudevan

student id-201ADB095

**Part 1**

**description of the dataset:**

**Name:** Raisin Dataset

**Author:**

1) Ilkay CINAR

Faculty of Technology,

Selcuk University, Konya, TURKEY.

2) Murat KOKLU

Faculty of Technology,

Selcuk University, Konya, TURKEY.

3) Sakir TASDEMIR

Faculty of Technology,

Selcuk University, Konya, TURKEY.

**Domain of the dataset:**

This dataset contains, information about the Kecimen and Besni raisin varieties grown in Turkey. There are total of 900 instances and 7 features.

**List of features:**

1.) Area: Gives the number of pixels within the boundaries of the raisin.

2.) Perimeter: It measures the environment by calculating the distance between the boundaries of the raisin and the pixels around it.

3.) MajorAxisLength: Gives the length of the main axis, which is the longest line that can be drawn on the raisin.

4.) MinorAxisLength: Gives the length of the small axis, which is the shortest line that can be drawn on the raisin.

5.) Eccentricity: It gives a measure of the eccentricity of the ellipse, which has the same moments as raisins.

6.) ConvexArea: Gives the number of pixels of the smallest convex shell of the region formed by the raisin.

7.) Extent: Gives the ratio of the region formed by the raisin to the total pixels in the bounding box.

8.) Class: Kecimen and Besni raisin.

**Git hub repository link to my orange tool file:**

[**https://github.com/akilarvinth/raisins\_machine\_learning\_orange\_tool**](https://github.com/akilarvinth/raisins_machine_learning_orange_tool)

**Data set source:**

<https://archive.ics.uci.edu/ml/datasets/Raisin+Dataset>

**PART 1:**

**Fig 1 – table snippet**

A picture containing text, screenshot, font, number

Description automatically generated

**Fig 2 - feature static's**

A screenshot of a graph

Description automatically generated with medium confidence

**Fig 3 - distribution of Major Axis length splitting with target**

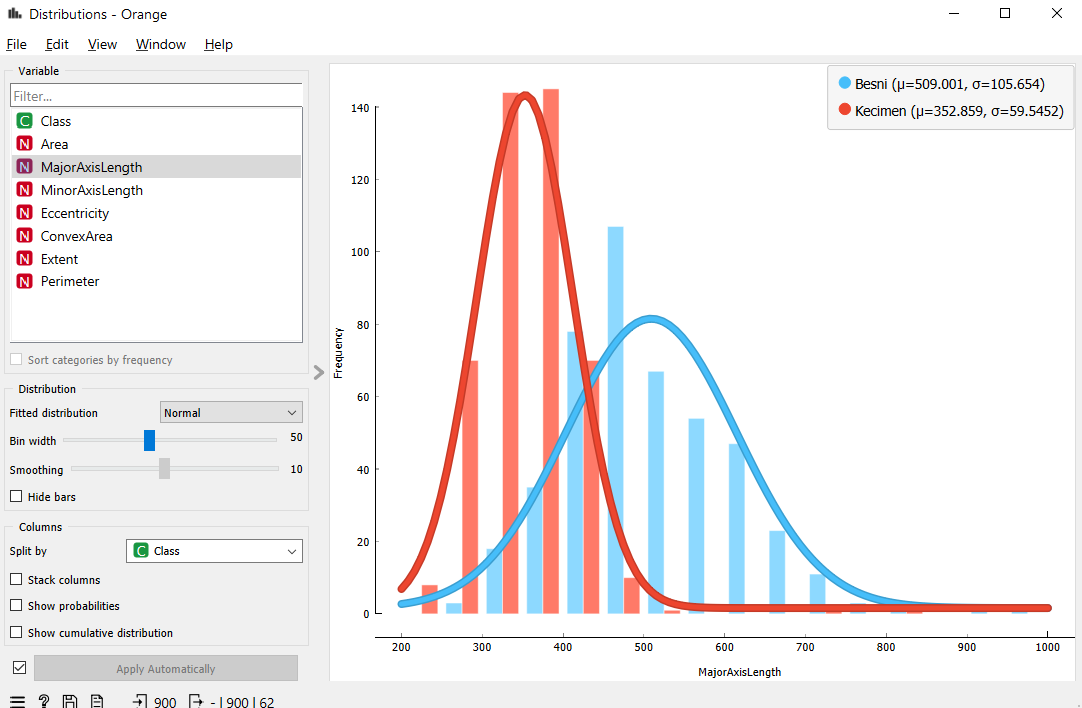


Fig 4 - **distribution of Major Axis length splitting with target** A screenshot of a computer

Description automatically generated with medium confidence

**Fig 5 - scatter plot of Minor axis length vs Perimeter**

A screen shot of a graph

Description automatically generated with medium confidence

**Fig 6 - scatter plot of Eccentricity vs Perimeter**

A screen shot of a computer

Description automatically generated with low confidence

**Fig 7 - scatter plot of convex Area vs Extent**

A screenshot of a computer

Description automatically generated with medium confidence

**Fig 8 – Correlations**A screenshot of a computer

Description automatically generated

**Analysis and Concussion:**

In this data set there is no missing instances, and the data set was continuous, so I did not use Continuize and Impute. My data set is equal distributed. When we see the distribution of the class it is equally distributed, it means of equal number of instances in each feature with respective two class (target) and value is been distributed into mix in particular features and clearly separated in specific range other features.

Yes, the data visualization allowed to see the data structure. I used box plot to find the structure and range of distribution of the feature.

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated

**Fig 10 – Convex Area**

**Fig 9 – perimeter**

A screenshot of a computer

Description automatically generated

**Fig 11 – Major Axis**

**Length**

A screenshot of a computer

Description automatically generated

**Fig 12 – Area**

In the above shown images, we can see that feature range is separated well, so I decided to use these features, in my next task and I have mentioned rest of the features range below.

And from scatter pot, it was visible the even in these separate features there were, some point where some instances where merging at some specific points. So, from based on these data visualizations. I can conclude that, we can say that while we take these four features and train the model, then we can be able to predict the class type of the raisin. But there might be some false predictions as there some interceptions.

A screenshot of a computer

Description automatically generated

**Fig 13 –**

**Minor Axis Length**

A screenshot of a computer

Description automatically generated with medium confidence

**Fig 14 –**

**Eccentricity**

A screenshot of a computer

Description automatically generated with medium confidence

**Fig 15 –**

**Eccentricity**

**Fig 17 – My data fallow for part 1:**

A picture containing text, diagram

Description automatically generated

**PART-2:**

**Note:** I made several experiments by changing the linkage and cutting off the cluster class. But only for single linkage, it was possible for me to capture the entire dendrogram in single screen. So, I have attached other dendrogram images are not visible.

**Fig 17 – hierarchy clusters with single linkage**

A screenshot of a computer

Description automatically generated

**Fig 18 – hierarchy clusters with single linkage** **A screenshot of a computer

Description automatically generated**

**Fig 19 – hierarchy clusters with weighted linkage**

**A screenshot of a computer

Description automatically generated with medium confidence**

**(Note:** this is not complete image)

**A screenshot of a computer

Description automatically generated with medium confidence**

**Fig 20.1 – K-means 1**

**Fig 20.2 – K-means 2**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Fig 20.3 – K-means 3**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Fig 20.4 – K-means 4**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Fig 20.5 – K-means 5**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Analysis and Concussion:**

The hyper parameters in hierarchical clustering and Distances are as fallow:

**1) Linkage**: The distance between clusters

**2)** **Pruning**: for changing, the depth of the class

**3)** **Selection: for making cut of cluster class in dendrogram**

**4)** **Metric**: how the distance between two points is calculated. There are 3 types of metrics (Euclidean distance, Manhattan distance, Cosine distance)

I used three different hierarchical clusters (single, average, and weighted.) For all three methods, the maximum depth was set kept at 10. Single linkage considers the minimum distance between clusters. In my experiment for single linkage, at height ratios decreasing from 100% to 32.0%, for these 10 clusters were created and all the cluster class was well separated, I made cut of this point because this only point in dendrogram where all the cluster were not. Average linkage considers the average distance between clusters. In my experiment for Average linkage resulted in 8 clusters with mixed class, when the height ratio decreased from 100% to 26.8%. Weighted linkage is a compromise between single and complete linkage. In my experiment for weighted linkage, also resulting in 8 clusters with mixed class, the height ratio decreased from 100% to 43.6%

The hyper parameters in K-means are as fallow:

**1)** **Number of Clusters (k)**: number of cluster and its range

**2)** **Re-runs:** The number of times the k-means algorithm will run with different centroid

**3)** **Maximum Iterations**: The maximum number of iterations

While working k means, I experimented with different ranges of cluster numbers: 2-4, 4-7, 3-6, 4-9, and 8-15. The best Silhouette was score (0.654) from all the combinations used. It was quite visible as I increased the cluster range, the Silhouette score was becoming bad, other than these 5-k values. I tried few more combinations, but clusters set between 2 and 4 had best score. Consistent Re-runs (10 times for each setup) and I experiment with different values but the time taken to calculate was significantly high and the values provided stability to the results, minimizing the impact of random initialization. The maximum iterations were initially set to 500 and later increased to 600 and as I mentioned for Re-runs, I experimented with some more values for Max iterations but Silhouette remined same and the computation time got increased exponentially.

Snippet of the other experiments I done:

**Fig 21– K-means 5**

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

K-mean with different number of Re-runs (with different initial centroid) and Max iterations values. Based on these experiments, I can conclude that when k value is change from 2 – 4, the Silhouette score is getting worst.

**Fig 22 – My data fallow for part 2:**

**A picture containing text, screenshot, diagram, design

Description automatically generated**

**PART-3:**

I have used **k-Nearest Neighbors (kNN) and Support Vector Machines (SVM) algorithms for supervised machine learning task. I preferred to use kNN, because of the ability of the algorithm to learn to classify new instances to appropriate classes based on the training data. And I was doing these experiments, I found some important details such as one of my k values was keeping producing the best prediction by still using different metrics and keeping weight as by distances. I choose SVM because to** find the best hyperplane that separates different classes in the feature space. As I have binary type target, I thought it will be good to use this algorithm.

**The hyper parameters in Neural Networks are as fallow:**

**1)** **Neurons in Hidden Layer**: the number of neurons in the hidden layers

**2)** **Activation**: activation function determines the output of a neuron given an input

**3)** **Solver:** determines the numerical optimization algorithm, to minimize the loss function

**4)** **Regularization**: a technique used to prevent overfitting by adding a penalty to the loss function

**5)** **Maximal number of iterations:** the maximum number of passes over the training data

**The hyper parameters in kNN are as fallow:**

**1)** **Number of Neighbors**: The number of neighbors (k) to consider when making a prediction

**2)** **Metric:** metric used to calculate the distance between points.

**3)** **Weight:** weight function is tool used in predicting Uniform weights

**The hyper parameters in SVM are as fallow:**

**1)** **Cost: To** determines the trade-off between achieving a low training error and a low testing error.

**2)** **Regression of Loss Epsilon:** The epsilon-tube within which no penalty is associated with the training loss function.

**3)** **Regression Cost**: To determines the trade-off between the model's complexity and the amount up to which deviations larger than epsilon are tolerated.

**4)** **Kernel: This** function is used to transform the data into a higher-dimensional space

**5)** **Numerical Tolerance: T**he tolerance for stopping criterion in the iterative training process

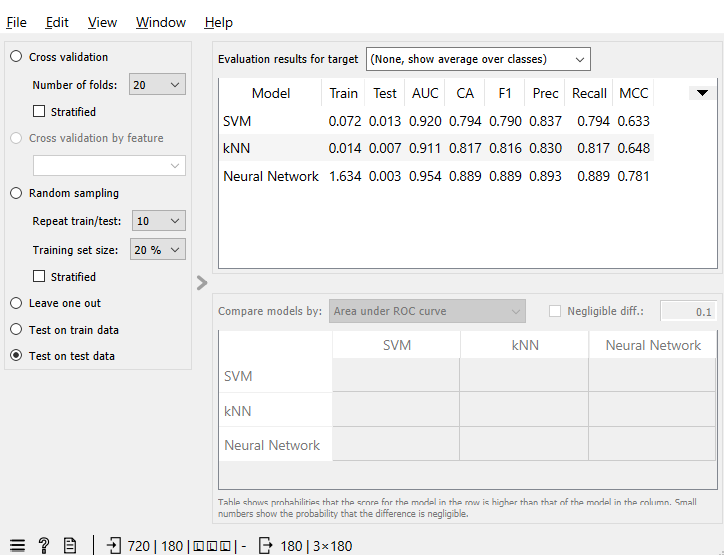
**6)** **Iteration Limit**: the maximum number of iterations for the solver,

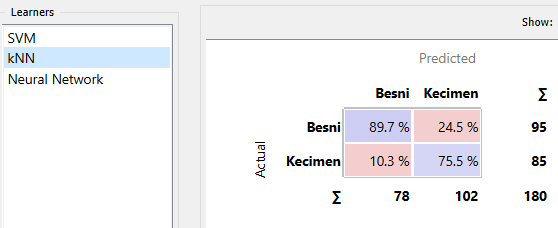
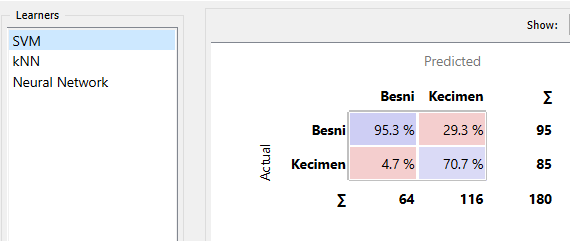
**1St EXPERIMENT:**

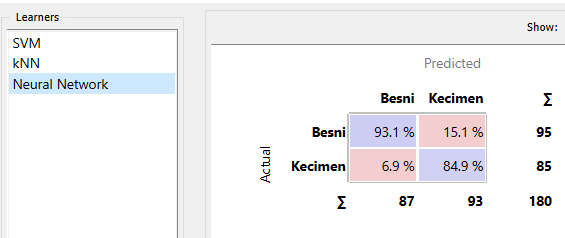
**Test score based on test data:**

**A screenshot of a computer

Description automatically generated with medium confidence**

****

****

****

**A screen shot of a graph

Description automatically generated with medium confidence**

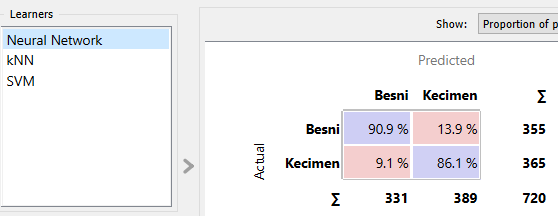
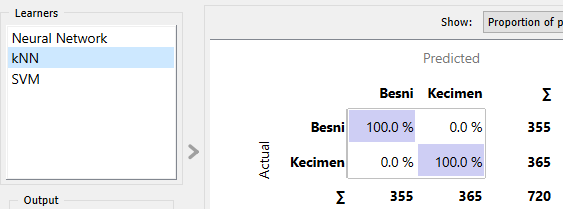
**A screen shot of a graph

Description automatically generated with medium confidence**

**Test score based on train data:**

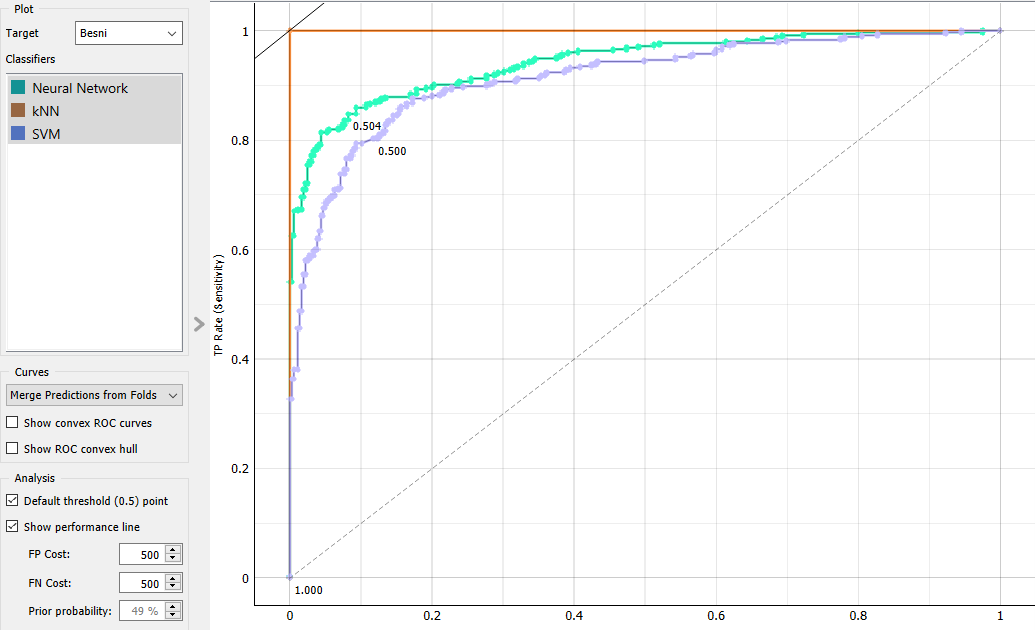
**A screenshot of a computer

Description automatically generated**

**** 

A screenshot of a computer

Description automatically generated with medium confidence

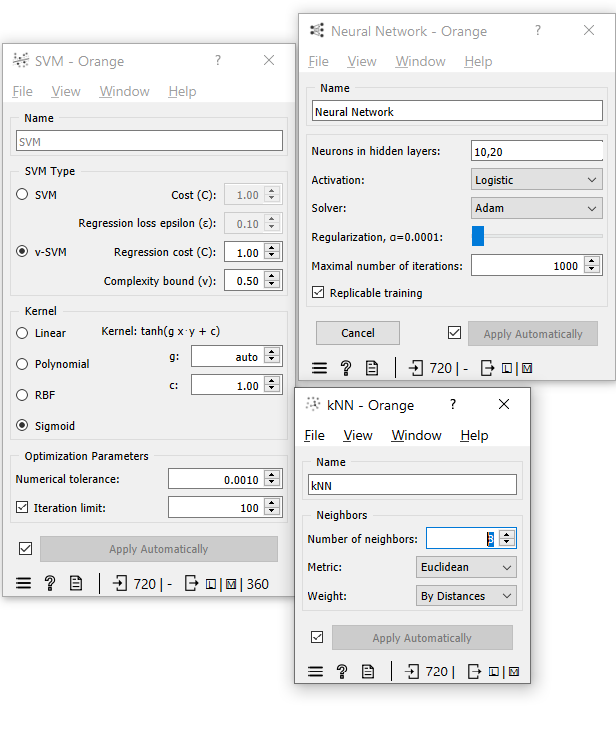


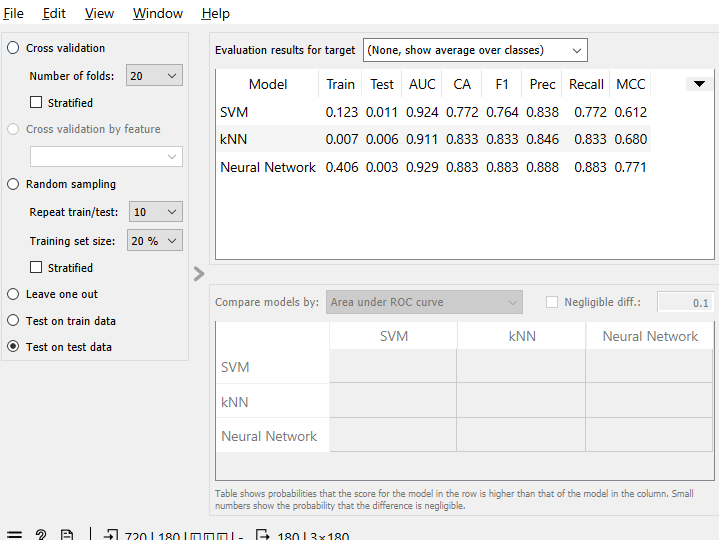
A picture containing text, screenshot, diagram, plot

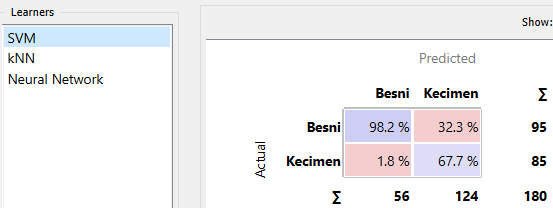
Description automatically generated

**2**nd **EXPERIMENT:**

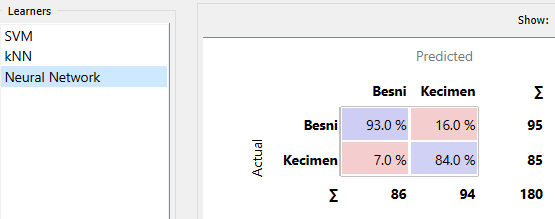
**Test score based on test data:**

****

****

**** **A screenshot of a computer

Description automatically generated**

****

**A picture containing text, screenshot, diagram, plot

Description automatically generated**

**A picture containing text, screenshot, diagram, plot

Description automatically generated**

**Test score based on train data:**

**A screenshot of a computer

Description automatically generated** **A screenshot of a computer

Description automatically generated with medium confidence** A screenshot of a computer

Description automatically generated with medium confidence

**A screenshot of a computer

Description automatically generated with medium confidence**

**A picture containing text, screenshot, diagram, plot

Description automatically generated**

**A picture containing text, diagram, plot, screenshot

Description automatically generated**

**3**rd **EXPERIMENT:**

**Test score based on test data:**

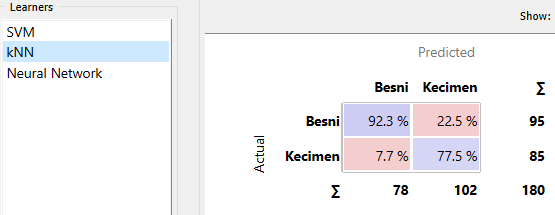
**A screenshot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated with medium confidence**A screenshot of a computer

Description automatically generated

**A picture containing text, plot, diagram, screenshot

Description automatically generated**

**A picture containing text, diagram, plot, line

Description automatically generated**

**Test score based on train data:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated with medium confidence**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

**A picture containing text, diagram, plot, screenshot

Description automatically generated**

**A picture containing text, screenshot, plot, diagram

Description automatically generated**

**My data fallow for part 3:**

**A picture containing text, diagram, line, screenshot

Description automatically generated**

**Analysis and Concussion:**

SVM, kNN, and Neural Network are three algorthims, i used for this data set. From the result I can say that Neural Network was giving the best prediction precentage in all the three experiments. It was able to predicted almost prefect in both the training and testing phases, demonstrating both its ability to comprehend the training data and its ability to function well with new data in future. The kNN model produced excellent results, particularly during the training phase, demonstrating its ability to effectively learn from the supplied data.The SVM model performed well in the first two experiments but it was bad in the third experiment. I was not satisfied this prediction, so apart from these three experiments, I made few more experiments for svm algorithm, I kept changing it is Metrix at v-Svm type with sigmoid type. The prediction of bensi was 98.2 % for SVM and 93.4 % for Neural Networks, but for another target variable kitmen the prediction percentage of SVM was only 67.7 % and for Neural network it was 84 %. Therefore, since it is prediction percentage for another variable is not better. I decided to use Neural Network for classifying the different types of raisins in the provided dataset. So, I conclude that Neural Network is the best algorthim for both training and testing for raisin data set.

# References

<https://youtu.be/vz1RZDIuViA>

<https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/visualize/boxplot.html>

<https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/data/featurestatistics.html>

<https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/visualize/distributions.html>

<https://orangedatamining.com/blog/2021/2021-12-17-visualizations-101/>

<https://orangedatamining.com/blog/2019/1/4/how-to-abuse-p-values-in-correlations/>

<https://orangedatamining.com/widget-catalog/unsupervised/kmeans/>

<https://orangedatamining.com/workflows/Hierarchical-Clustering/>

<https://www.youtube.com/live/ICfii0UVXKk?feature=share>

<https://youtu.be/vkkPkVCuN-U>

<https://archive.ics.uci.edu/ml/datasets/Raisin+Dataset>

data set relevant papers –

Relevant Papers: CINAR I., KOKLU M. and TASDEMIR S., (2020), Classification of Raisin Grains Using Machine Vision and Artificial Intelligence Methods. Gazi Journal of Engineering Sciences, vol. 6, no. 3, pp. 200-209, December, 2020. DOI: https://doi.org/10.30855/gmbd.2020.03.03