Theoretical task1

1.1. What is Machine Learning (ML) approach in your understanding? Give specific sub-stages of Training/Testing phases. Draw a taxonomy of Machine Learning methods that you know, include also specific sub-categories describing particular tasks that these methods solve.

Machine Learning (ML) is a field of computer science and subfield of Artificial Intelligence (AI). Its purpose is to teach computers how to learn and as a result solve difficult tasks for human beings without the need of programming computers on a regular basis or when needed. Machine Learning can be considered as one of the pillars of 21st century.

Training and testing phase are the two crucial steps in the achievement of machine learning. The training phase should be accomplished in an efficient way to acquire high quality system. Training and testing sets are obtained by dividing the dataset into two parts based on specific rules or "Sampling methods". To achieve a reasonable success rate, the amount of training and testing sets plays an important role and it depends on the correlation of the dependent and independent (target variable) variables. For instance, if the correlation between the features and the target variable is high, as a result training and test sets can be divided into equal parts, however, when there is doubt of achieving success, the ratio of training set can be enlarged. When sampling it should be taken care of, that the training and test datasets represents the whole dataset because this process has a huge impact on the performance of the training and testing results. (Muhammed Kursad Ucar, 2020)

1.2. Give your understanding of a "Variable" (also called "Feature" or "Attribute") in datascience and describe different types/properties of Variables that you know. Why is it important to have good quality variables for ML tasks? Use examples to support your answer.

An attribute is basically a characteristic measured for each record and can be different in different observations. In machine learning the two terms "attribute" and "feature" are used interchangeably, while "variable" is common in statistics (statistics.com , 2021). Examples, attributes of hair can be color, texture, length and so on. In data science there are six types of attributes. These are, Nominal attributes, Binary attributes, Ordinal attributes, Discrete attributes, Continuous attributes, and Numeric attributes (Ratio and Interval) (GeeksforGeeks, 2020). In any dataset the quality of features impacts the accuracy of the training dataset and

the quality of the insight obtained for decision making. Using processes such as “feature selection” and “feature engineering” the quality of the variables can be elevated leading to an optimal dataset containing vital features. It is also important to know which variables to choose and which variables to avoid depending on the project goal.

1.3. What is the difference between Classification, Regression and Association Rules? Present examples of Rules/Trees that can be used for such tasks and define the meaning of nodes/leaves if applicable.

To be able to select the right algorithm for a machine learning problem, it is essential first to know the difference between different algorithms. Supervised and Unsupervised learning are two machine learning approaches and there exits one basic difference between them, that is on uses labeled data and other uses unlabeled data for the prediction of outputs. Now Classification and Regression are the two types of problems that come under supervised learning. By using classification technique, it is possible to predict the response value and separate data into “classes”. For instance, predicting the weather or if a mail is spam or not. While Regression technique reproduces the output value, it tries to understand the relationship between dependent and independent variables. Instances can be predicting housing prices or stock values (Delua, 2021).

Association Rule is a technique used in unsupervised learning problems and its technique is to check whether one data item is dependent of another data item or not. More clearly, it tries to find relationship between the variables present in the datasets. Used cases are medical diagnosis, customer market analysis etc. (javatpoint, 2021).

1.4. Describe briefly the main procedures in the Genetic Algorithm method and their mainpurpose towards Global Optimization (like minimization and maximization)?

The idea of Genetic Algorithm (GA) was given by John Holland which is basically based on real biological evolution, *genetics and natural selection*. The study of genes and chromosomes that how best genes are passed to the new generations. The same idea is used in Artificial Intelligence (AI) and Machine Learning (ML) where we it says that there are already so many solutions that exists, but it is main goal is to find the “optimal” or “best” solution. Genetic Algorithm, a search-based algorithm, specially focuses on optimization and solves complex problems. In terms of mathematics “optimal” or “best” indicates maximizing or minimizing one or more objective functions by changing the input parameters.

For a given problem there exists a pool of possible solutions and these endure reproduction and alteration, as a result generating new generation. This process is occurs repetitively over several generations and then every candidate solution is given a fitness value depending on their fitness the candidates gets higher chance to regenerate and produce even better and fitter individuals until it reaches the goal.

Genetic Algorithm is considered to give an optimal solution within a reasonable time frame which is the reason of its popularity and attractiveness for problems requiring optimization. GA is mainly used when problems are complex enough that even highly powerful computers are unable to solve or takes long time to complete. Another area where GA is useful is when functions like linear regression is unable to find the global optima because it gets stuck at the local optima but with GA, global optima can be achieved as it mimics the principles of biological evolution. In this way, it improves the search criteria to find global optima and is appropriate for continuous and discrete optimization problems (tutorialspoint, 2021).

1.5. Imagine you have the following features: irrelevant, redundant, constant, high InformationGain value, low Information Gain value. Which of those can be removed to reduce the complexityof the dataset?

In dimensionality reduction process the features that are of no use or take the machine learning model in the wrong way resulting in wrong accuracy or inefficient model are removed. Numerous methods can be applied depending on situations in order identify useful features. Here, features that are irrelevant to the target is certainly of no use thus can be removed. Redundant features also do not add any value to the solution rather adds to the complexity of the dataset and in addition occupies extra space thus can be removed.

Information gain (IG) shows the importance of the feature, and a high IG describes high entropy (level of variance in a dataset/impurity of variable) and low IG is low entropy. So, the highest IG values indicates the feature to be the best and yet to be chose for split (Tyagi, 2021). Knowing the importance of Information Gain, concludes that these features need to remain in the dataset.

1.6. What is the K-means method? Explain briefly how does the method works and what are the necessary parameters to make the final mode reliably trained? What can be the reason for having an "empty" cluster (without any data points assigned to it)?

When it comes to classifying of unlabeled data, K-means clustering plays a vital role. K-means is a method which refers to unsupervised learning and it groups the data features. Here, the

term K refers to the number of categories or groups formed. The best feature of this techniques is that it is the machine that generates clusters which is based on observed proofs and not suppositions and there is no involvement of person biasness (DeepAI, 2021).

The way K-means clustering algorithm works is that it divides the given dataset into specified number of clusters and randomly selects first centroids (location whether real or imaginary that shows that center of the cluster) and repeats the calculations to optimize the status of the centroids. The stopping point in k-means clustering reaches when firstly, there exists no change in the values of centroids and second, specific number of iterations is completed (Garbade, 2018). Empty clusters basically are the local minimums which should be avoided. Reason of attaining empty clusters could be because there are no points that are assigned to a cluster in the process of assigning.

Practical Task

Data Analysis & Knowledge Representation

Data: Dry Bean Dataset

Seven different types of dry beans were used in this research, taking into account the features such as form, shape, type, and structure by the market situation. A computer vision system was developed to distinguish seven different registered varieties of dry beans with similar features in order to obtain uniform seed classification. For the classification model, images of 13,611 grains of 7 different registered dry beans were taken with a high-resolution camera. Bean images obtained by computer vision system were subjected to segmentation and feature extraction stages, and a total of 16 features; 12 dimensions and 4 shape forms, were obtained from the grains.

Attribute Information:

1.) Area (A): The area of a bean zone and the number of pixels within its boundaries.

2.) Perimeter (P): Bean circumference is defined as the length of its border.

3.) Major axis length (L): The distance between the ends of the longest line that can be drawn from a bean.

4.) Minor axis length (l): The longest line that can be drawn from the bean while standing perpendicular to the main axis.

5.) Aspect ratio (K): Defines the relationship between L and l.

6.) Eccentricity (Ec): Eccentricity of the ellipse having the same moments as the region.

7.) Convex area (C): Number of pixels in the smallest convex polygon that can contain the area of a bean seed.

8.) Equivalent diameter (Ed): The diameter of a circle having the same area as a bean seed area.

9.) Extent (Ex): The ratio of the pixels in the bounding box to the bean area.

10.)Solidity (S): Also known as convexity. The ratio of the pixels in the convex shell to those found in beans.

11.)Roundness (R): Calculated with the following formula: (4piA)/(P^2)

12.)Compactness (CO): Measures the roundness of an object: Ed/L

13.)ShapeFactor1 (SF1)

14.)ShapeFactor2 (SF2)

15.)ShapeFactor3 (SF3)

16.)ShapeFactor4 (SF4)

17.)Class (Seker, Barbunya, Bombay, Cali, Dermosan, Horoz and Sira)

Relevant Papers:

KOKLU, M. and OZKAN, I.A., (2020), â€œMulticlass Classification of Dry Beans Using Computer Vision and Machine Learning Techniques.â€• Computers and Electronics in Agriculture, 174, 105507.

DOI: [Web Link]

Task 1.1. Select two attributes with the highest merit using Information Gain measure (in Weka, for example) and represent a dataset in the Cartesian coordinate system (x-y plot) with colors assigned to different classes. Evaluate the plot that you have obtained according to the following criteria: possibilities to apply classification models, and potential uncertainties in classification due to overlapping of samples in different classes.

InfoGain attribute evaluation method is applied to the dataset at the first step of this task. Here are the results:

Attribute Evaluator (supervised, Class (nominal): 17 Class):

Information Gain Ranking Filter

Ranked attributes:

1.524 2 Perimeter

1.49 7 ConvexArea

1.484 1 Area

1.484 8 EquivDiameter

1.437 3 MajorAxisLength

1.376 14 ShapeFactor2

1.329 13 ShapeFactor1

1.328 4 MinorAxisLength

1.192 15 ShapeFactor3

1.192 12 Compactness

1.175 5 AspectRation

1.175 6 Eccentricity

1.147 11 roundness

0.533 16 ShapeFactor4

0.34 10 Solidity

0.284 9 Extent

Selected attributes: 2,7,1,8,3,14,13,4,15,12,5,6,11,16,10, 9: 16

According to InfoGain method results Perimeter and ConvexArea attributes have the highest two ranks in attribute selection. That means these two attributes would be the most helpful attributes in distinction of 7 classes in any classification method. All seven types of dry beans can be identified on X (Perimeter) – Y (ConvexArea) axes. Bombay (green) stays as a completely different group, but it points are dispersed a lot. The other six groups are more compact. There are some overlapping areas between Barbunya and Cali. There are also some dark blue points (Seker) overlapping with Horoz. At the end Perimeter and ConvexArea are very might be very helpful in classification (figure 1).

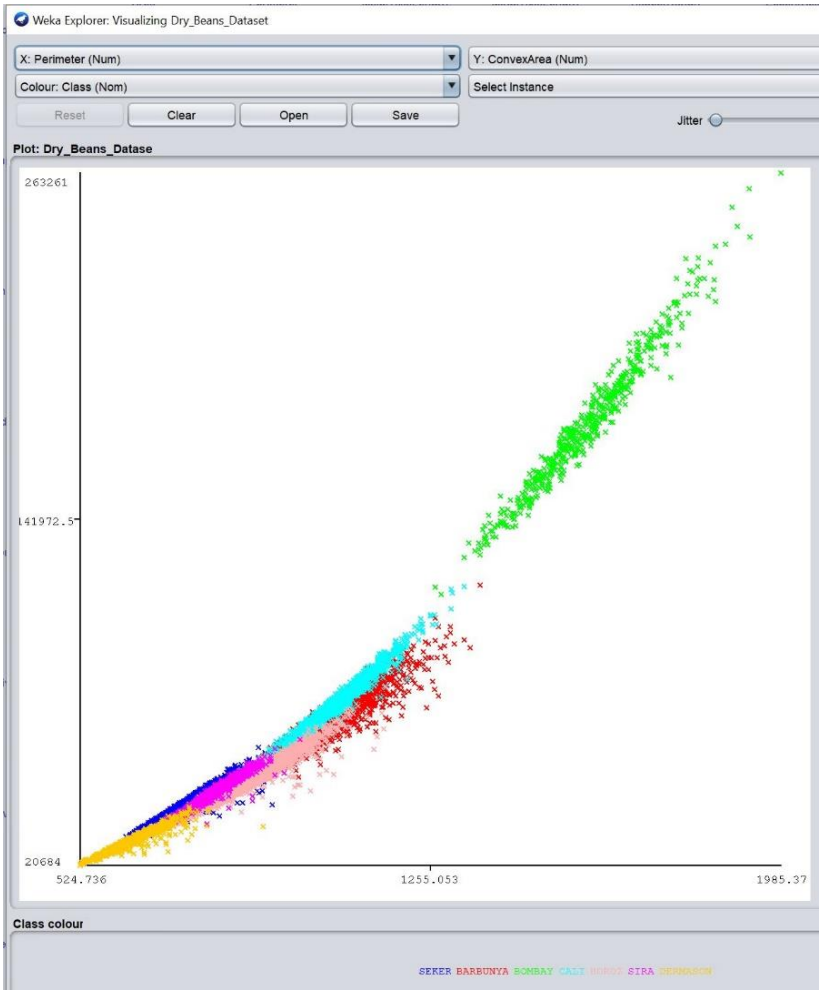


Figure 1: Perimeter – ConvexArea correlation plot by Information Gain method.

Area and EquivDiameter are the third and fourth ranks in the InfoGain attribute evaluator result. There is not much difference between grades of these four attributes, but classes on Area – EquivDiameter figure are not clearly identified as they can be on the first figure. There are 5 classes on the figure and the other two have been overlapped by the others. Hardly we can see some blue dots (Seker), but it is nearly impossible to see red dots (Barbunya). These are the uncertainties on the which can decrease accuracy of classification. Therefore, Area and EquivDiameter attributes are not helpful as much as Perimeter and ConvexArea are (figure 2).

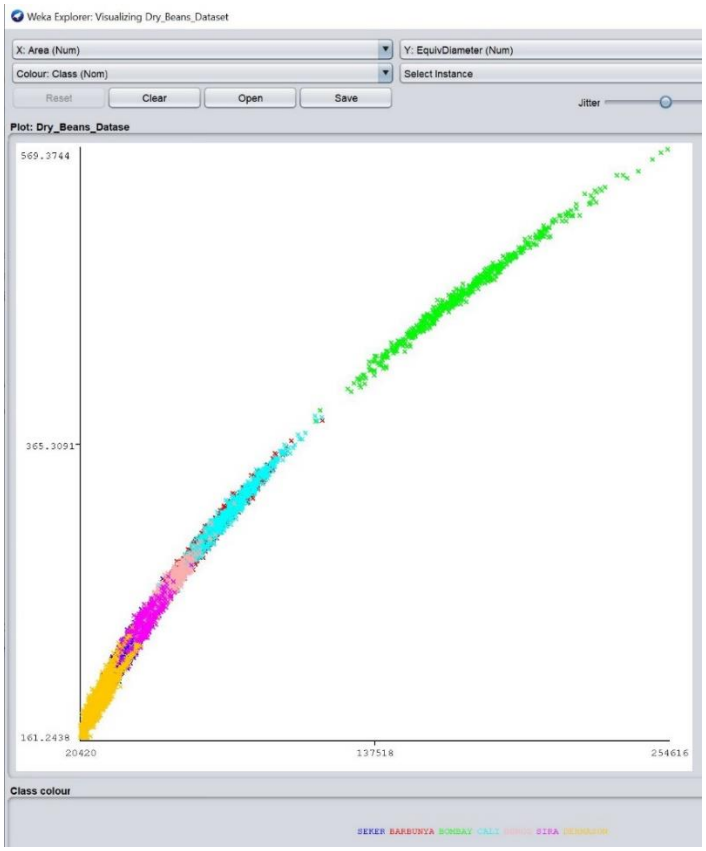


Figure 2: Area – EquivDiameter correlation plot by Information Gain method.

In this part the attributes which have the two lowest ranks were analyzed. Extent and Solidity attribute coordinate map is like a huge dispersed cluster and all classes are dispersed everywhere. There is no clustering of classes and all classes overlap each other. Only pink dots (Horoz) clustered on the left side, but it is possible to find pink dots all around the figure. Thus, these two attributes cannot be helpful of classifying dry bean types (figure 3).

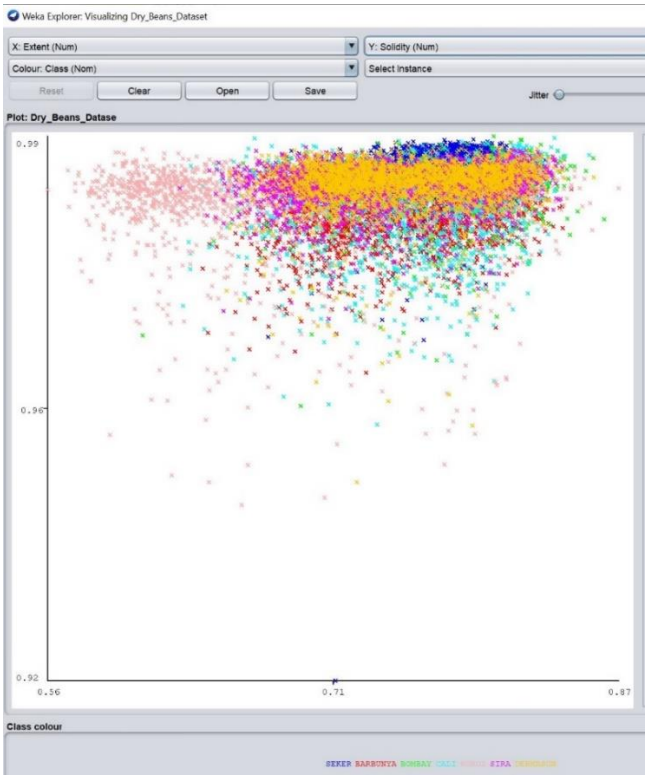


Figure 3: Extent – Solidity correlation plot by Information Gain method.

Task 1.2. Sketch possible decision boundaries of the classes and indicate whether you have

some challenges to do so. Why?

Sketching boundaries is very easy on figure 1, because there are clear clustering and no overlapping areas this figure. 5 classes can be identified on the second figure and border lines can be drawn between these clusters. It is not easy to identify borders of overlapped clusters. There is no way to sketch boundaries of classes on the third figure, because there is no cluster on the figure.

Task 1.3. Looking at the Cartesian plot, which one of the two attributes do you think is more

useful for the classification (which attribute can discriminate classes more easily)? Apply other attributes evaluators from Weka. Compare the results to the Information Gain. Are there any differences and why?

As it is mentioned above Perimeter and ConvexArea attributes are most useful in classification according to InfoGain attribute selection method.

Here is the result of **Gain Ratio** attribute selection evaluator:

Gain Ratio feature evaluator

Ranked attributes:

0.3829 7 ConvexArea

0.3804 1 Area

0.3804 8 EquivDiameter

0.3708 2 Perimeter

0.3573 3 MajorAxisLength

0.3425 14 ShapeFactor2

0.3377 4 MinorAxisLength

0.3361 13 ShapeFactor1

0.3275 15 ShapeFactor3

0.3275 12 Compactness

0.3253 6 Eccentricity

0.3253 5 AspectRation

0.2832 11 roundness

0.1671 16 ShapeFactor4

0.1084 10 Solidity

0.0993 9 Extent

Selected attributes: 7,1,8,2,3,14,4,13,15,12,6,5,11,16,10,9: 16

Due to these results, an X-Y coordinate figure was made between ConvexArea and Area attributes. There are five distinctive clusters on the figure, but the other three are overlapped. It is hard to distinguish red dots (Barbunya) and very little dark blue (Seker) dots. In that case it can be claimed that InfoGain method made a better job than Gain Ratio method (figure 4).

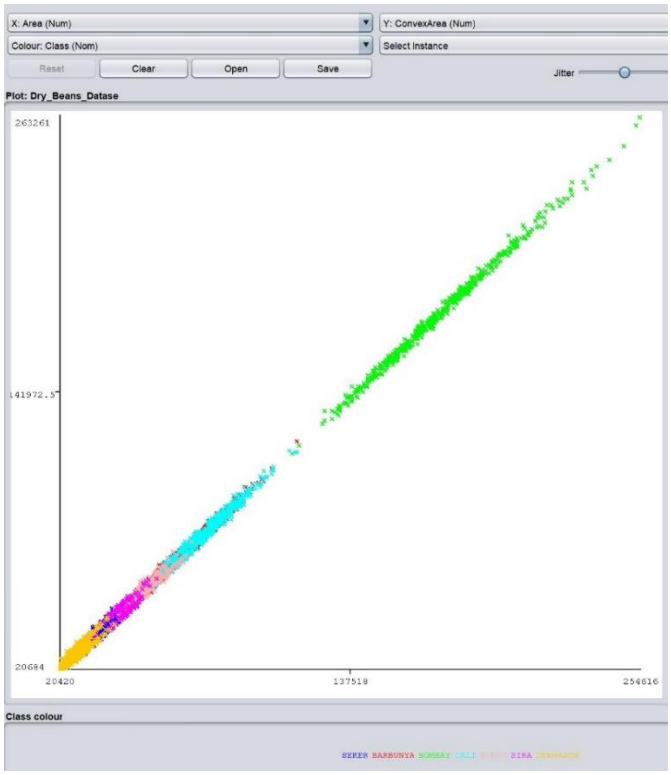


Figure 4: Area – ConvexArea correlation plot by Gain Ratio method.

CFS Subset Evaluator results:

Including locally predictive attributes

Selected attributes: 2,3,4,5,7,9,11,12,13,14,16: 11

Perimeter

MajorAxisLength

MinorAxisLength

AspectRation

ConvexArea

Extent

roundness

Compactness

ShapeFactor1

ShapeFactor2

ShapeFactor4

7 classes can be seen as clusters in this figure and CFS subset method did a very good job. All classes clearly identifiable on the figure except some overlapping areas on the borders. Green, red and dark blue dots show some dispersion either which may decrease accuracy of classification a little (figure 5).

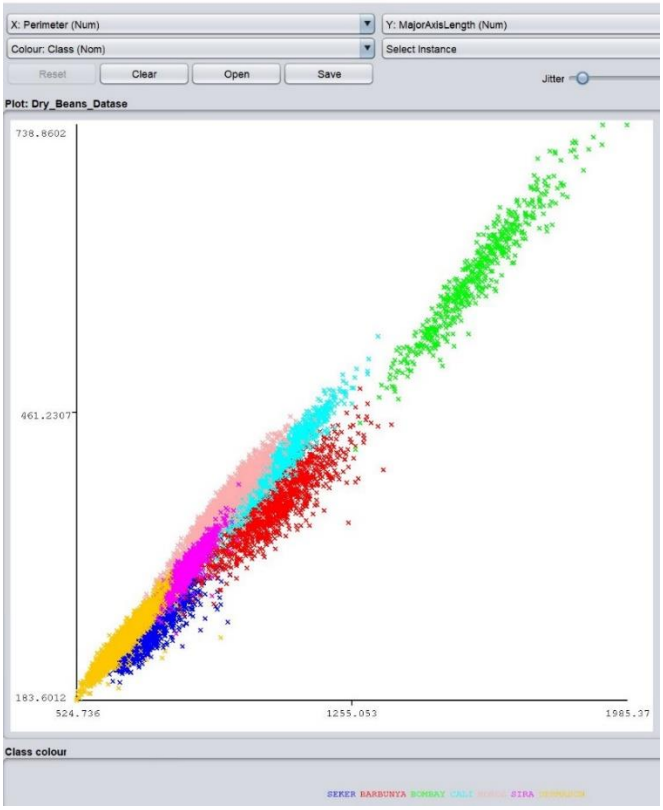


Figure 5: Perimeter – MajorAxisLength correlation plot by CFS Subset method.

These methods gave different results because they are using different **search methods**.

Practical Task 2

2. Classification – Simple Decision Rules

2.1. See Jupyter Notebook Code

2.2.

Input

0 Area 13611 non-null int64

1 Perimeter 13611 non-null float64

2 MajorAxisLength 13611 non-null float64

3 MinorAxisLength 13611 non-null float64

4 AspectRation 13611 non-null float64

5 Eccentricity 13611 non-null float64

6 ConvexArea 13611 non-null int64

7 EquivDiameter 13611 non-null float64

8 Extent 13611 non-null float64

9 Solidity 13611 non-null float64

10 roundness 13611 non-null float64

11 Compactness 13611 non-null float64

12 ShapeFactor1 13611 non-null float64

13 ShapeFactor2 13611 non-null float64

14 ShapeFactor3 13611 non-null float64

15 ShapeFactor4 13611 non-null float64

Output

16 Class 13611 non-null object

2.3.

Based on Infomation Gain from Weka, we select the 5 most important features as followed:

´ 0.3829 7 ConvexArea

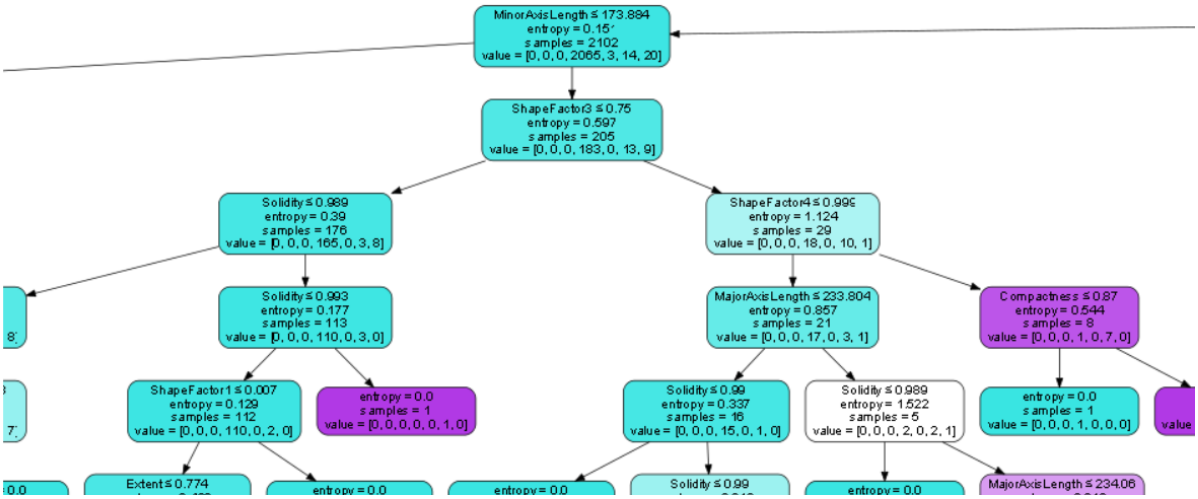
´ 0.3804 1 Area

´ 0.3804 8 EquivDiameter

´ 0.3708 2 Perimeter

´ 0.3573 3 MajorAxisLength

0.3425 14 ShapeFactor2



2.4.

This part is explained as a separate ‘.ipynb’ file. Please see the attachment.

Practical Task 3

3.2. Feature Selection: Evaluate features of the dataset using Information Gain and CFS methods. What are the most useful and least useful features?

Here is the result of InfoGain Attribute Evaluation method with 10 folds cross-validation:

average merit	average rank	attribute
1.524 +- 0.002	1 +- 0	2 Perimeter
1.49 +- 0.003	2 +- 0	7 ConvexArea
1.485 +- 0.002	3 +- 0	1 Area
1.485 +- 0.002	4 +- 0	8 EquivDiameter
1.437 +- 0.002	5 +- 0	3 MajorAxisLength
1.377 +- 0.002	6 +- 0	14 ShapeFactor2
1.328 +- 0.003	7.4 +- 0.49	4 MinorAxisLength
1.328 +- 0.002	7.6 +- 0.49	13 ShapeFactor1
1.193 +- 0.004	9 +- 0	15 ShapeFactor3
1.193 +- 0.004	10 +- 0	12 Compactness
1.175 +- 0.004	11 +- 0	5 AspectRation
1.175 +- 0.004	12 +- 0	6 Eccentricity
1.145 +- 0.006	13 +- 0	11 roundness
0.532 +- 0.002	14 +- 0	16 ShapeFactor4
0.337 +- 0.003	15 +- 0	10 Solidity
0.284 +- 0.003	16 +- 0	9 Extent

According to InfoGain method Perimeter, ConvexArea, and Area are the most useful attributes and Extent, Solidity, and ShapeFactor4 are the least useful attributes.

And here is the result of CFS Attribute Evaluation method with 10 folds cross-validation:

number of folds	(%)	attribute
1	(10 %)	1 Area
10	(100 %)	2 Perimeter

10	(100 %)	3 MajorAxisLength
10	(100 %)	4 MinorAxisLength
10	(100 %)	5 AspectRation
0	(0 %)	6 Eccentricity
9	(90 %)	7 ConvexArea
0	(0 %)	8 EquivDiameter
10	(100 %)	9 Extent
0	(0 %)	10 Solidity
10	(100 %)	11 roundness
10	(100 %)	12 Compactness
10	(100 %)	13 ShapeFactor1
10	(100 %)	14 ShapeFactor2
0	(0 %)	15 ShapeFactor3
10	(100 %)	16 ShapeFactor4

CFS method finds no relation between 5 attributes and classes and these attributes are Area, Eccentricity, EquivDiameter, Solidity, and ShapeFactor3. According to CFS method the other five attributes are 100% relevant to classes.

3.3. *Correlation:* Utilize Principal Component Analysis in Weka and find features with the highest and lowest Pearson correlation. Build a pair-wise figure of the relation between such attributes to demonstrated such correlation.

The table below gives the correlation values between attributes. The attributes are (1) Area, (2) Perimeter, (3) MajorAxisLength, (4) MinorAxisLength, (5) AspectRation, (6) Eccentricity, (7) ConvexArea, (8) EquivDiameter, (9) Extent, (10) Solidity, (11) roundness, (12) Compactness, (13) ShapeFactor1, (14) ShapeFactor2, (15) ShapeFactor3, (16) ShapeFactor4 respectively. According to the table there is a very high positive correlation between Perimeter and EquivDiameter (0.99), ConvexArea and EquivDiameter (0.99), Area and EquivDiameter (0.98) and Perimeter and MajorAxisLength (0.98). This high correlation is a strong sign of multi-collinearity and some of the attributes should be eliminated in a real-world study. In addition, there is a very high negative correlation between AspectRation and Compactness (-

0.99), AspectRation and ShapeFactor3 (-0.98) and Eccentricity and ShapeFactor3 (-0.99). There are is very low positive correlation between AspectRation and ShapeFactor1 (0.02), MinorAxisLength and Eccentricity (0.02), and Eccentricity and ShapeFactor1 (0.02). There is also some very low correlation between MinorAxisLength and AspectRaion (-0.01), Compactness and ShapeFactor1 (-0.01), ShapeFactor1 and ShapeFactor3 (-0.01), Perimeter and Extent (-0.02), MinorAxisLength and Compactness (-0.02) and MinorAxixLength and ShapeFactor3 (-0.02) (table 1).

Table 1: Attribute Evaluator for Principal Component Analysis.

Attribute Evaluator (unsupervised):
Principal Components Attribute Transformer

Correlation matrix

1	0.97	0.93	0.95	0.24	0.27	1	0.98	0.05	-0.2	-0.36	-0.27	-0.85	-0.64	-0.27	-0.36
0.97	1	0.98	0.91	0.39	0.39	0.97	0.99	-0.02	-0.3	-0.55	-0.41	-0.86	-0.77	-0.41	-0.43
0.93	0.98	1	0.83	0.55	0.54	0.93	0.96	-0.08	-0.28	-0.6	-0.57	-0.77	-0.86	-0.57	-0.48
0.95	0.91	0.83	1	-0.01	0.02	0.95	0.95	0.15	-0.16	-0.21	-0.02	-0.95	-0.47	-0.02	-0.26
0.24	0.39	0.55	-0.01	1	0.92	0.24	0.3	-0.37	-0.27	-0.77	-0.99	0.02	-0.84	-0.98	-0.45
0.27	0.39	0.54	0.02	0.92	1	0.27	0.32	-0.32	-0.3	-0.72	-0.97	0.02	-0.86	-0.98	-0.45
1	0.97	0.93	0.95	0.24	0.27	1	0.99	0.05	-0.21	-0.36	-0.27	-0.85	-0.64	-0.27	-0.36
0.98	0.99	0.96	0.95	0.3	0.32	0.99	1	0.03	-0.23	-0.44	-0.33	-0.89	-0.71	-0.33	-0.39
0.05	-0.02	-0.08	0.15	-0.37	-0.32	0.05	0.03	1	0.19	0.34	0.35	-0.14	0.24	0.35	0.15
-0.2	-0.3	-0.28	-0.16	-0.27	-0.3	-0.21	-0.23	0.19	1	0.61	0.3	0.15	0.34	0.31	0.7
-0.36	-0.55	-0.6	-0.21	-0.77	-0.72	-0.36	-0.44	0.34	0.61	1	0.77	0.23	0.78	0.76	0.47
-0.27	-0.41	-0.57	-0.02	-0.99	-0.97	-0.27	-0.33	0.35	0.3	0.77	1	-0.01	0.87	1	0.48
-0.85	-0.86	-0.77	-0.95	0.02	0.02	-0.85	-0.89	-0.14	0.15	0.23	-0.01	1	0.47	-0.01	0.25
-0.64	-0.77	-0.86	-0.47	-0.84	-0.86	-0.64	-0.71	0.24	0.34	0.78	0.87	0.47	1	0.87	0.53
-0.27	-0.41	-0.57	-0.02	-0.98	-0.98	-0.27	-0.33	0.35	0.31	0.76	1	-0.01	0.87	1	0.48
-0.36	-0.43	-0.48	-0.26	-0.45	-0.45	-0.36	-0.39	0.15	0.7	0.47	0.48	0.25	0.53	0.48	1

The figure below shows class distribution on EquivDiameter (X-axis) – Perimeter (Y-axis) plot. All classes except Barbunya (red dots) show good clustering on the plot and it can be assumed that both EquivDiameter and Perimeter attributes are useful for all six attributes. Red dots (Barbunya) show relatively large dispersion on both X and Y axis and they mix with pink, cyan, even dark blue dots sometimes. There are also some dark blue dots (Seker) mixed with pink (Horoz) dots. Generally, these two attributes are very useful in classification of dry beans (figure 6).

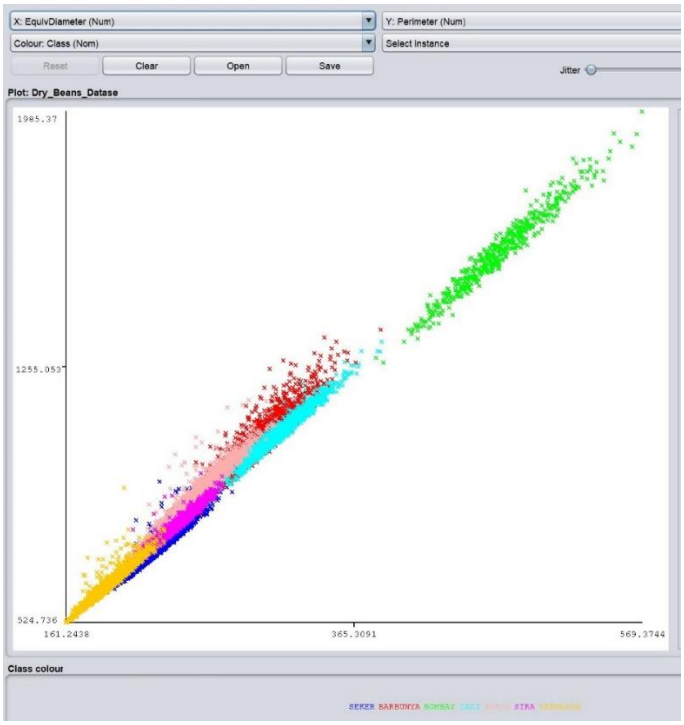


Figure 6: EquivDiameter – Perimeter correlation plot by PCA method.

There is 0.99 correlation value between EquivDiameter and ConvexArea. In this plot, red dots (Barbunya) dispersed a lot on both X and Y axis, and mixed with cyan (Cali), pink (Horoz), and dark blue (Seker) dots. That mixture is a sign that ConvexArea is not a good attribute in classification of Barbunya (figure 7).

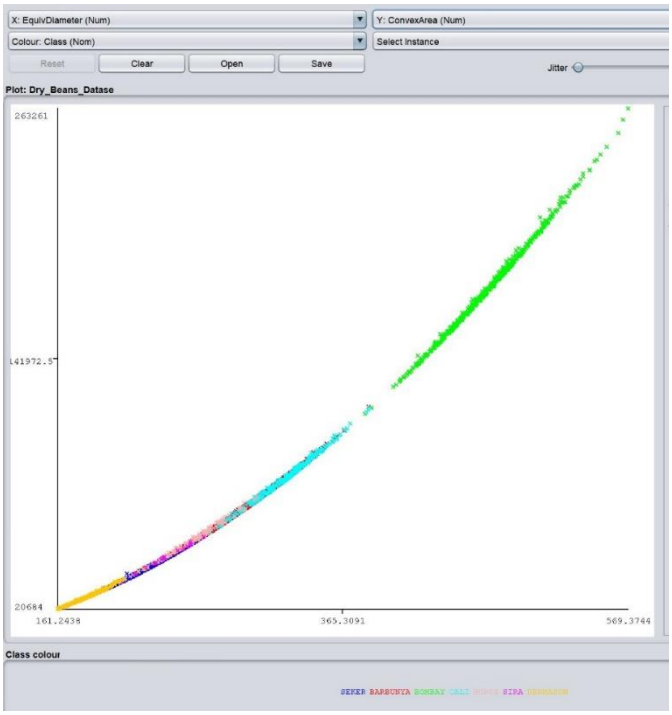


Figure 7: EquivDiameter – ConvexArea correlation plot by PCA method.

There is -0.99 correlation between AspectRation (X-Axis) and Compactness (Y-Axis). In this plot, almost all classes show large dispersions on both X and Y axes. Only dark blue (Seker) and pink (Horoz) dots show good clustering and it is very hard to detect any green dot (Bombay) on the figure. Therefore, it is hard to say that these attributes are useful in classification of dry bean classes (figure 8).

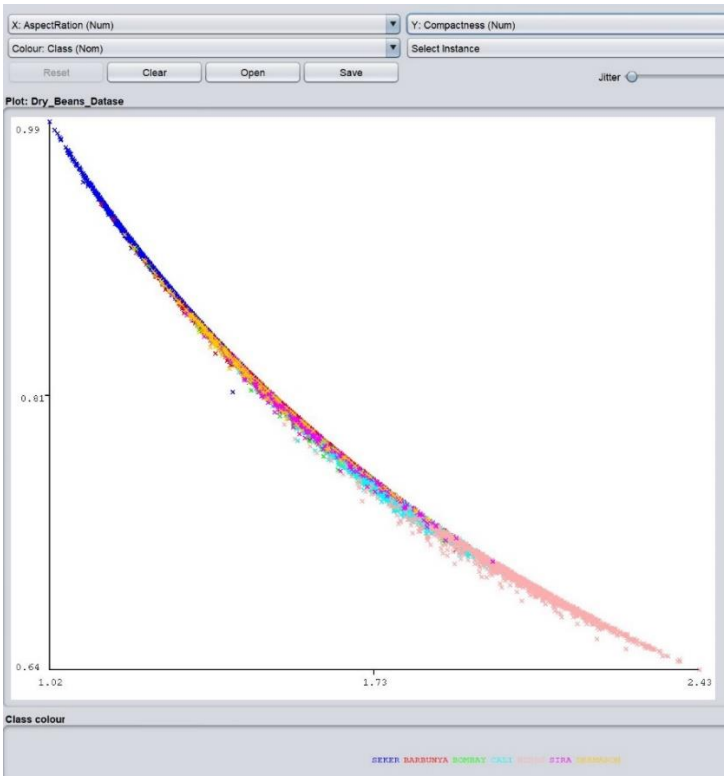


Figure 8: AspectRation – Compactness correlation plot by PCA method.

There is a very low positive correlation - 0.02 - between AspectRation and ShapeFactor1. Green dots (Bombay) cluster separately in the plot but the other classes cluster with mixing. Magenta dots (Sira) stay in the center and they mix with all other 5 classes, especially with yellow (Dermason). There is also a big mixture between red (Barbunya) and cyan (Cali). It can be said that AspectRation and ShapeFactor1 cannot be very helpful in classification of dry beans due to their large dispersion on the plot which is created by principal component analysis (figure 9).

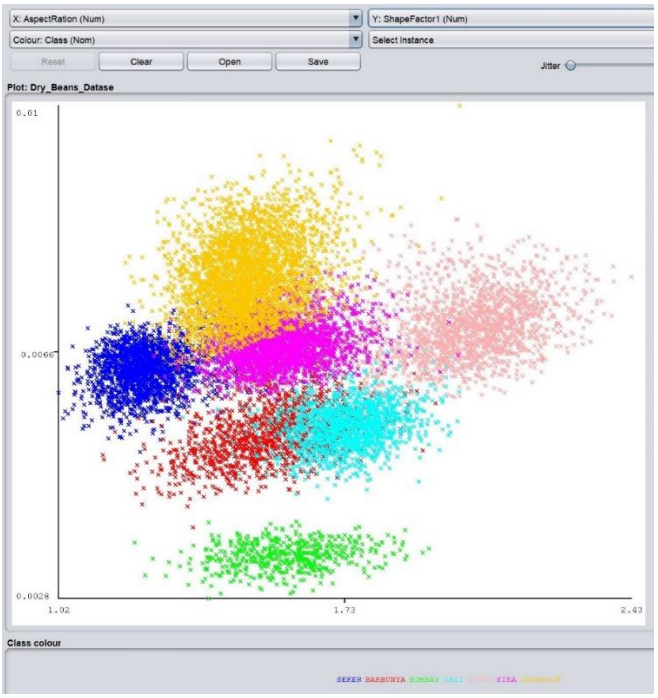


Figure 9: AspectRation – ShapeFactor1 correlation plot by PCA.

There is a very low correlation between MinorAxisLength and AspectRation with value of -0.01. The dispersion of dots is very similar to previous plot and it can be assumed that MinorAxisLength may not be very useful for classification. Because there is a big dispersion of dots on both axes (figure 10).

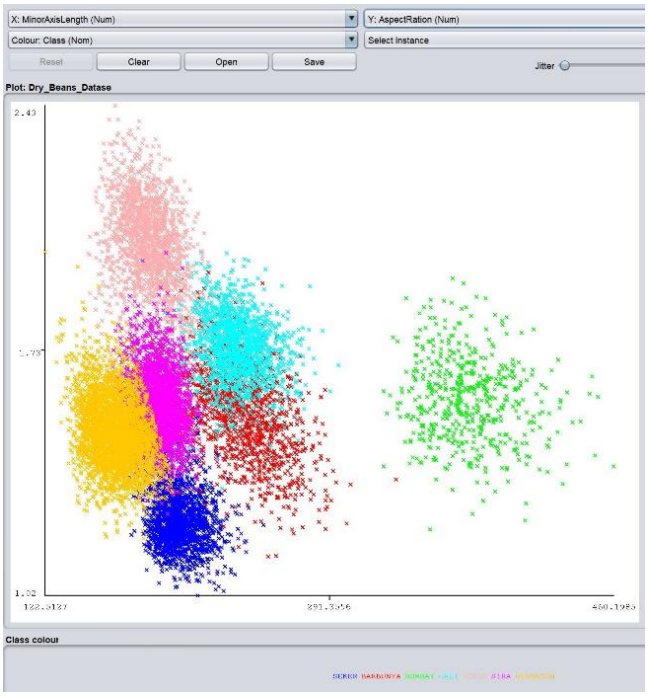


Figure 10: MinorAxisLength – AspectRation correlation plot by PCA.

The figure below is correlation plot between the first and the second attributes generated by PCA method. There are clusters in the plot there are also mixing and large dispersion of dots. Therefore, it can be said that these two attributes may not be good enough despite the fact that they are the best PCA created (figure 11):

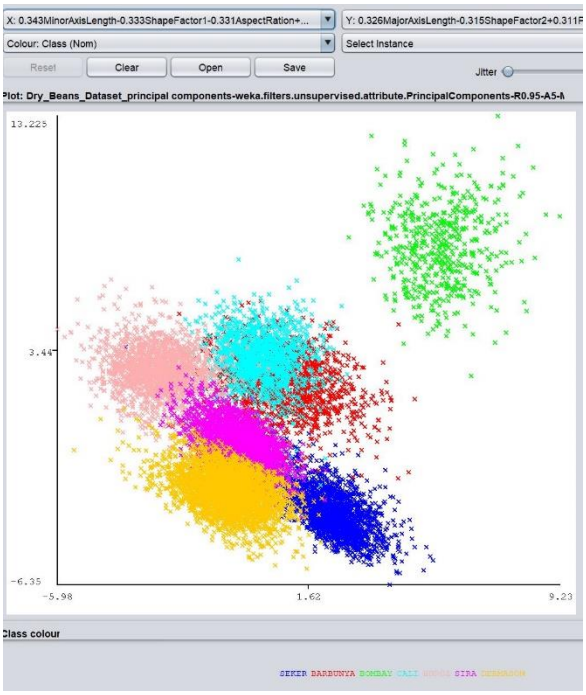


Figure 11: Correlation plot between the first two attributes created by PCA.

3.4. *Clustering:* Use K-means and Expectation-Maximization (EM) methods. Evaluate the optimal number of clusters and if it corresponds to classes in the dataset.

This task is completed by python programming. After scaling continuous features (all attributes except class attribute), MinMaxScaler is used for mix of categorical and continuous features.

```
mms = MinMaxScaler()

mms.fit(data)

data_transformed = mms.transform(data)
```

In the next part the inertia attribute is used for finding the sum of squared distances of samples to the nearest cluster centre.

```
Sum_of_squared_distances = []

K = range(1,10)

for k in K:
```

```
km = KMeans(n_clusters=k)
km = km.fit(data_transformed)
Sum_of_squared_distances.append(km.inertia_)
```

At last, a plot generated which shows sum of squared distances for k in the range specified. If the plot looks like an arm, then the elbow on the arm is optimal k.

Plot generation:

```
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(12, 6), dpi=80)
ax = fig.add_subplot(121)
ax.set_xlabel('Number of Clusters')
ax.set_ylabel('Sum of Squared Distance')
ax.set_title('Elbow Method for Optimal k')
ax.title.set_color('red')
ax.xaxis.label.set_color('red')
ax.yaxis.label.set_color('red')
ax.tick_params(axis='x', colors='red')
ax.tick_params(axis='y', colors='red')
plt.plot(K, Sum_of_squared_distances, 'bx-')
plt.show()
```

The result: The figure shows an elbow in value of 7 and this number is compatible with the dataset. There are 7 different classes in our dataset and it seems elbow method gave us an accurate result.

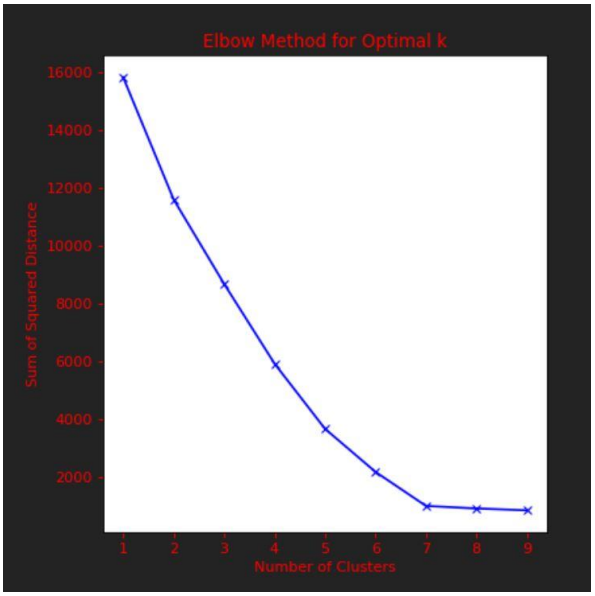


Figure 12: Elbow Method for Optimal Clusters by KNN method.

Another test was applied by Weka software. Clusters were created by both KMeans and EM methods. KMeans method creates two metrics: incorrectly classified instances and sum of squared errors. Besides EM model creates incorrectly classified instances and log likelihood. Both methods created the lowest incorrectly classified instances value with 6 attributes although the dataset has 7 different classes (figure 13). That means these methods assumes there are 6 different classes in the dataset.

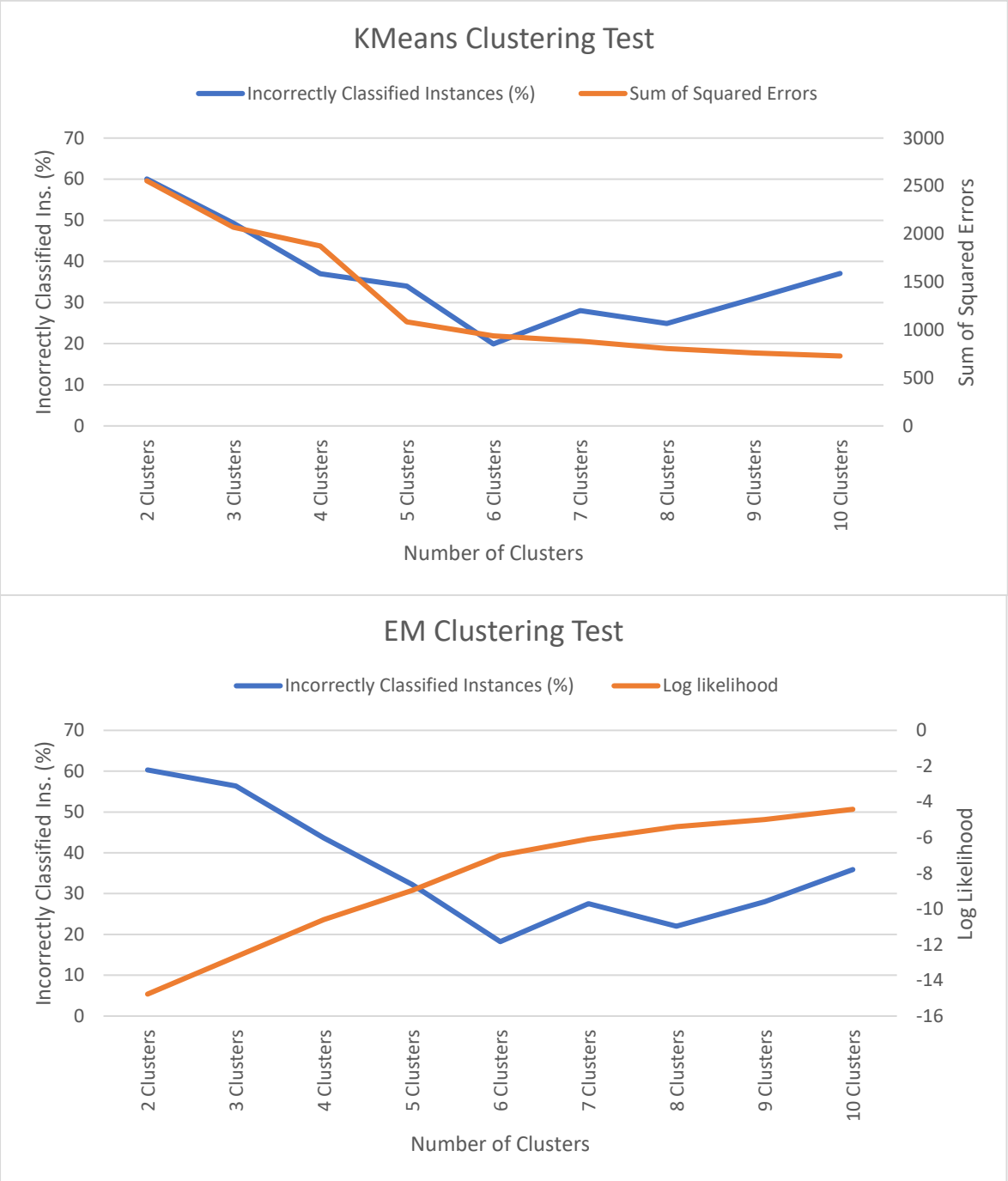


Figure 13: Optimum cluster test bey KMeans and EM methods.

This may be a result of a little unbalanced dataset of us. Dry bean number are 3546 for Dermason, 2636 for Sira, 2027 for Seker, 1928 for Horoz, 1630 for Cali, 1322 for Barbunya and 522 for Bombay. The difference between the highest and the lowest might have caused this little shift.

3.5. Classification: Build a table with accuracies for the following methods: SVM, k-NN (k=1,3

and 5), BayesNet, ANN (Multilayer perception with learning rate: 0.01, 0.1, 1 and number of epochs: 1, 10, 100, 1000). Please, use cross-validation for this task. Which set of the parameters and method gives you the best accuracy? Is there a set of parameters that gives you a significant error? Elaborate on your answers.

Different attribute sets were used for classification task. First all attributes were used without any discard for classification and here is the result (table 2):

Table 2: Classification metrics table for all attributes.

Classification	Correctly Classified Instances	Incorrectly Classified Instances	Kappa Statistics	MAE	RMSE
J48	12429 (91.3158%)	1182 (8.6842%)	0.8949	0.0309	0.1466
BayesNet	12189 (89.5526%)	1422 (10.4474%)	0.8737	0.03	0.1635
Random Forest	12602 (92.5869%)	1009 (7.4131)	0.9103	0.0315	0.1252
Support Vector Machine	12553 (92.2269%)	1058 (7.7731%)	0.906	0.2054	0.3303
KNN (k = 1)	12291 (90.302%)	1320 (9.698%)	0.8827	0.0278	0.1664
KNN (k = 3)	12466 (91.5877%)	1145 (8.4123%)	0.8983	0.029	0.1381
KNN (k = 5)	12526 (92.0285%)	1085 (7.9715%)	0.9036	0.0297	0.1321
ANN (Learning Rate=0.01, Epochs= 1)	3546 (26.0525%)	10065 (73.9475%)	0	0.2356	0.3426
ANN (Learning Rate=0.01, Epochs= 10)	11459 (84.1893%)	2152 (15.8107%)	0.8076	0.1322	0.2206
ANN (Learning Rate=0.01, Epochs= 100)	12535 (92.0946%)	1076 (7.9054%)	0.9044	0.0428	0.1306
ANN (Learning Rate=0.01, Epochs= 1000)	12634 (92.822%)	977 (7.178%)	0.9132	0.0317	0.123
ANN (Learning Rate=0.1, Epochs= 1)	11415 (83.866%)	2196 (16.134%)	0.8038	0.1319	0.2211
ANN (Learning Rate=0.1, Epochs= 10)	12483 (91.7126%)	1128 (8.2874%)	0.8997	0.0437	0.1339
ANN (Learning Rate=0.1, Epochs= 100)	12570 (92.3518%)	1041 (7.6482%)	0.9075	0.0334	0.1262
ANN (Learning Rate=0.1, Epochs= 1000)	12629 (92.7852%)	982 (7.2148%)	0.9127	0.0304	0.1239
ANN (Learning Rate=1, Epochs= 1)	12063 (88.6268%)	1548 (11.3732%)	0.8622	0.0515	0.1544
ANN (Learning Rate=1, Epochs= 10)	12454 (91.4995%)	1157 (8.5005%)	0.8971	0.0353	0.1362
ANN (Learning Rate=1, Epochs= 100	12482 (91.7052%)	1129 (8.2948%)	0.8997	0.0325	0.1352
ANN (Learning Rate=1, Epochs= 1000)	12508 (91.8963%)	1103 (8.1037%)	0.9019	0.03	0.1344

Artificial Neural Network (Multilayer Perceptron) method with 0.01 learning rate and 1000 epochs gave the highest accuracy which is 92.822%. Although Artificial Neural Network (Multilayer Perceptron) method with 0.01 learning rate and 1 epoch gave the lowest accuracy which is 26.0525%.

Another classification test is applied based on Cfs attribute selection method. According to Cfs method ten attributes, Perimeter, MajorAxisLength, MinorAxisLength, Eccentricity, Extent, roundness, ShapeFactor1, ShapeFactor2, ShapeFactor3, ShapeFactor4. The same methods with the previous one gave the highest and lowest percentages as 92.6457% and 26.0525% respectively (table 3).

Table 3: Classification metrics table for 10 best attributes according to Cfs method.

Classification	Correctly Classified Instances	Incorrectly Classified Instances	Kappa Statistics	MAE	RMSE
J48	12399 (91.0954%)	1212 (8.9046%)	0.8922	0.032	0.1469
BayesNet	12287 (90.2726%)	1324 (9.7274%)	0.8824	0.0289	0.1548
Random Forest	12555 (92.2416%)	1056 (7.7584%)	0.9061	0.0316	0.1261
Support Vector Machine	12545 (92.1681%)	1066 (7.8319%)	0.9053	0.2054	0.3303
KNN (k = 1)	12255 (90.0375%)	1356 (9.9625%)	0.8796	0.0286	0.1687
KNN (k = 3)	12478 (91.6759%)	1133 (8.3241%)	0.8993	0.0292	0.1385
KNN (k = 5)	12544 (92.1608%)	1067 (7.8392%)	0.9052	0.0295	0.1316
ANN (Learning Rate=0.01, Epochs= 1)	3546 (26.0525%)	10065 (73.9475%)	0	0.2361	0.3432
ANN (Learning Rate=0.01, Epochs= 10)	10737 (78.8847%)	2874 (21.1153%)	0.7425	0.153	0.2453
ANN (Learning Rate=0.01, Epochs= 100)	12528 (92.0432%)	1083 (7.9568%)	0.90337	0.0459	0.1322
ANN (Learning Rate=0.01, Epochs= 1000)	12610 (92.6457%)	1001 (7.3543%)	0.911	0.0339	0.126
ANN (Learning Rate=0.1, Epochs= 1)	10638 (78.1574%)	2973(21.8426%)	0.7335	0.1534	0.2464
ANN (Learning Rate=0.1, Epochs= 10)	12474 (91.6465%)	1137 (8.3535%)	0.8989	0.0461	0.1345
ANN (Learning Rate=0.1, Epochs= 100)	12546 (92.1754%)	1065 (7.8246%)	0.9053	0.0349	0.1285
ANN (Learning Rate=0.1, Epochs= 1000)	12595 (92.5354%)	1016 (7.4646%)	0.9097	0.0323	0.1264
ANN (Learning Rate=1, Epochs= 1)	12124 (89.075%)	1487 (10.925%)	0.8677	0.0534	0.1538
ANN (Learning Rate=1, Epochs= 10)	12416 (91.2203%)	1195 (8.7797%)	0.8937	0.0372	0.1386
ANN (Learning Rate=1, Epochs= 100	12499 (91.8301%)	1112 (8.1699%)	0.9011	0.0331	0.1353
ANN (Learning Rate=1, Epochs= 1000)	12484 (91.7199%)	1127 (8.2801%)	0.8998	0.0323	0.1364

According to MAE (mean absolute error) and RMSE (root mean square error) values, there is no classification which gives significant error.

3.6. Complexity reduction: extract 2-5 features with the highest merit using Information Gain and remove all other features, except class label. Build the same table as for the "5. Classification". Can you see much difference in accuracy results? Please, explain.

In this step, InfoGain attribute selection method was applied to our dataset first. According to InfoGain method attribute ranking of the dataset is Perimeter, ConvexArea, Area, EquivDiameter, MajorAxisLength, ShapeFactor2, MinorAxisLength, ShapeFactor1, ShapeFactor3, Compactness, AspectRation, Eccentricity, Roundness, ShapeFactor4, Solidity and Extent. At the first step the first two high ranked attributes were selected and RandomForest classification was applied with them. In the next step the third high ranked attribute added the first two and RandomForest classifier was used with three attributes. By following this way, one more attribute added to previous ones and RandomForest classifier was used again. Here is the result (table 4):

Table 4: Performance table of different attributes.

Classification	Classification Accuracy (%)	Kappa Statistics	MAE	RMSE	RAE	RRSE
2 attributes	80.53%	0.7642	0.0682	0.2013	28.8674%	58.5636%
3 attributes	81.96%	0.7815	0.0667	0.1929	28.2088%	56.1034%
4 attributes	81.99%	0.7819	0.0666	0.1933	28.1644%	56.2298%
5 attributes	91.13%	0.8927	0.0357	0.1369	15.0888%	39.8359%
6 attributes	91.09%	0.8922	0.0354	0.1372	14.9652%	39.9143%
7 attributes	90.91%	0.89	0.0358	0.1379	15.1650%	40.1262%
8 attributes	91.05%	0.8917	0.0356	0.138	15.0821%	40.1394%
9 attributes	90.94%	0.8904	0.0361	0.1385	15.2851%	40.3033%
10 attributes	90.74%	0.888	0.0367	0.1396	15.5214%	40.6182%
11 attributes	90.90%	0.8899	0.0368	0.1389	15.5793%	40.4133%
12 attributes	90.85%	0.8892	0.037	0.1392	15.6729%	40.4935%
13 attributes	91.43%	0.8963	0.034	0.1342	14.3997%	39.0270%
14 attributes	92.00%	0.9032	0.0323	0.1285	13.6504%	37.3705%
15 attributes	92.17%	0.9052	0.0319	0.1268	13.5150%	36.8814%
16 attributes	92.59%	0.9103	0.0315	0.1252	13.3251%	36.4189%

A figure of Classification Accuracy – Number of attributes was generated also (figure 14). By five attributes classification accuracy becomes 91.13% and there is a very low difference between this percentage and the highest percentage (92.59%). That means 5 attributes may be use in very large datasets, because classification accuracy of 16 attributes is only 1.46% higher than accuracy of 5 attributes.

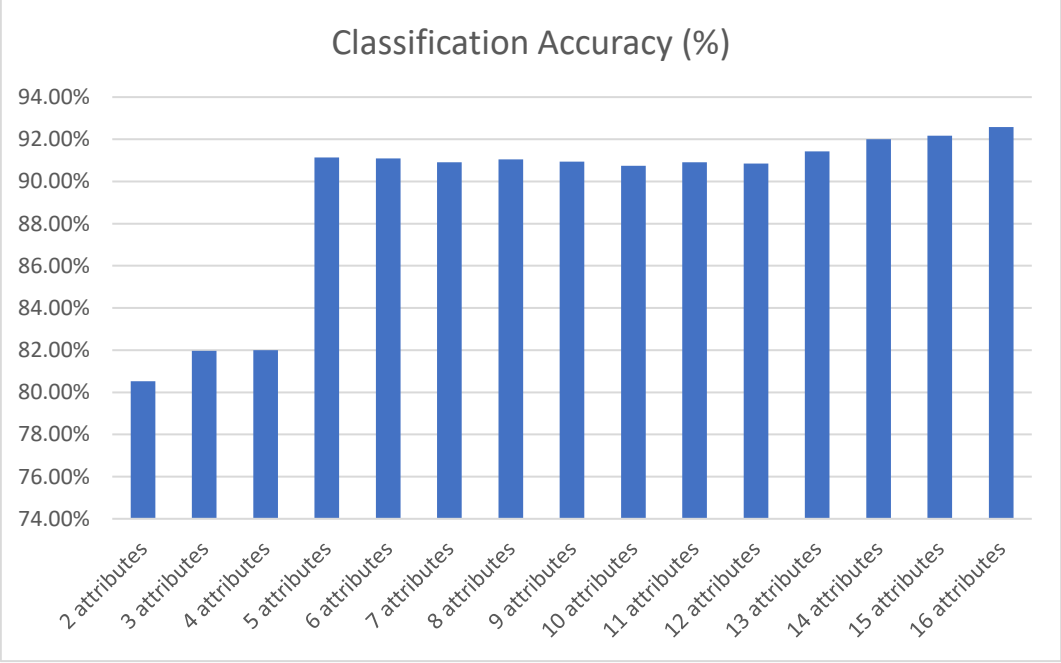


Figure 14: Classification accuracy table by InfoGain method.

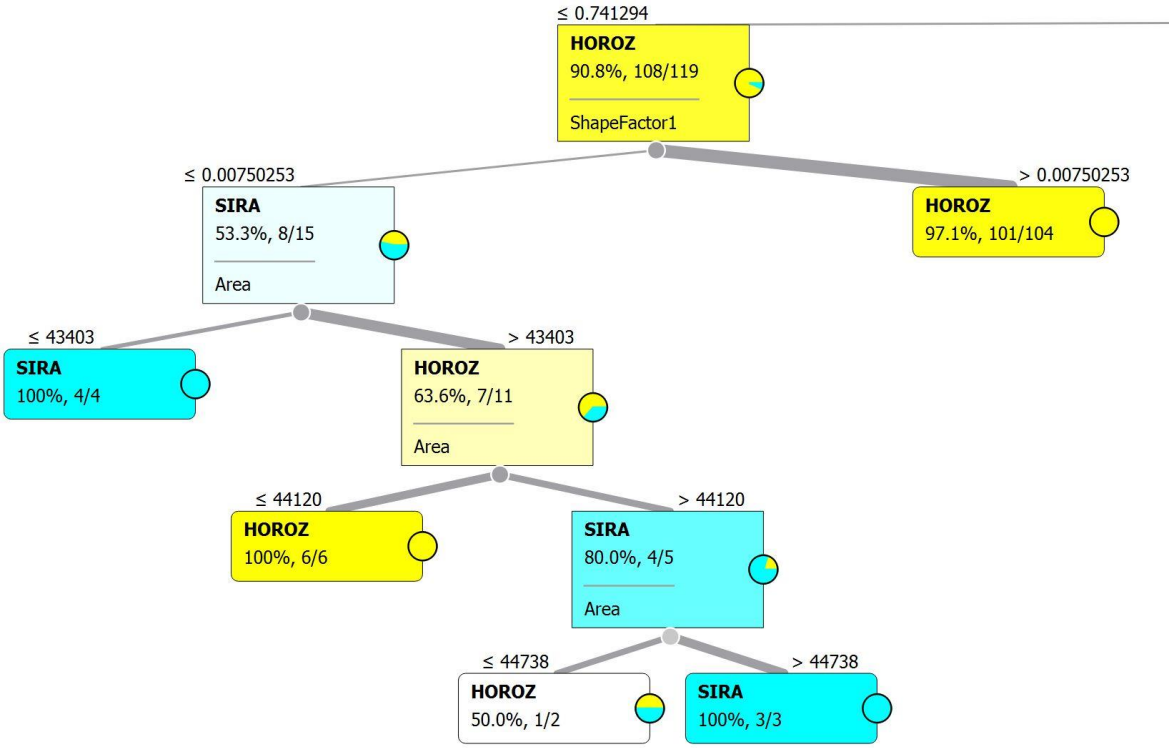
At the last step the five high ranked attributes of InfoGain method were used for different kinds of classification. The classification results are very similar to previous question. ANN method by 0.1 learning rate and 1000 epochs is gave the highest result with 91.2133%. ANN classification with 0.01 learning rate and 1 epoch gave the lowest accuracy result with 26.0525% (table 5).

Table 5: Classification accuracy results by five high ranked attributes (Perimeter, ConvexArea, Area, EquivDiameter, MajorAxisLength)

Classification	Correctly Classified Instances	Incorrectly Classified Instances	Kappa Statistics	MAE	RMSE
J48	12292 (90.3093%)	1319 (9.6907%)	0.8827	0.0373	0.1523
BayesNet	9283 (68.2022%)	4328 (31.7978%)	0.6164	0.0951	0.2681
Random Forest	12404 (91.1322%)	1207 (8.8678%)	0.8927	0.0357	0.1369
Support Vector Machine	11999 (88.1566%)	1612 (11.8434%)	0.8564	0.2062	0.3046
KNN (k = 1)	12086 (88.7958%)	1525 (11.2042%)	0.8645	0.0321	0.1789
KNN (k = 3)	12316 (90.4856%)	1295 (9.5144%)	0.8849	0.0331	0.1484
KNN (k = 5)	12395 (91.066%)	1216 (8.934%)	0.8919	0.0332	0.1407
ANN (Learning Rate=0.01, Epochs= 1)	3546 (26.0525%)	10065 (73.9475%)	0	0.236	0.3429
ANN (Learning Rate=0.01, Epochs= 10)	6626 (48.6812%)	6985 (51.3188%)	0.3569	0.2069	0.3069
ANN (Learning Rate=0.01, Epochs= 100)	11979 (88.0097%)	1632 (11.9903%)	0.8548	0.0819	0.1735
ANN (Learning Rate=0.01, Epochs= 1000)	12413 (91.1983%)	1198 (8.8017%)	0.8935	0.0426	0.1379
ANN (Learning Rate=0.1, Epochs= 1)	6577 (48.3212%)	7034 (51.6788%)	0.3523	0.2065	0.31
ANN (Learning Rate=0.1, Epochs= 10)	11812 (86.7827%)	1799 (13.2173%)	0.8399	0.0846	0.1784
ANN (Learning Rate=0.1, Epochs= 100)	12341 (90.6693%)	1270 (9.3307%)	0.8871	0.0415	0.1398
ANN (Learning Rate=0.1, Epochs= 1000)	12415 (91.213%)	1196 (8.787%)	0.8937	0.0381	0.1372
ANN (Learning Rate=1, Epochs= 1)	10538 (77.4227%)	3073 (22.5773%)	0.7256	0.1086	0.215
ANN (Learning Rate=1, Epochs= 10)	12049 (88.524%)	1562 (11.476%)	0.8612	0.0482	0.1579
ANN (Learning Rate=1, Epochs= 100	12239 (89.9199%)	1372 (10.0801%)	0.878	0.0405	0.151
ANN (Learning Rate=1, Epochs= 1000)	12309 (90.4342%)	1302 (9.5658%)	0.8841	0.0373	0.149

Tree Example:

Due to relatively high number of classes (7) in the dataset, all software produces hundreds of trees and it is very hard to visualize them in Weka. In that case Orange3 software is used to visualize a branch of classification tree only for an example (figure 15). Basically, it shows how algorithm works for classification of instances by using rules.


≤ 0.741294
HOROZ
90.8%, 108/119
ShapeFactor1
≤ 0.00750253
SIRA
53.3%, 8/15
Area
> 0.00750253
HOROZ
97.1%, 101/104
≤ 43403
SIRA
100%, 4/4
> 43403
HOROZ
63.6%, 7/11
Area
≤ 44120
HOROZ
100%, 6/6
> 44120
SIRA
80.0%, 4/5
Area
≤ 44738
HOROZ
50.0%, 1/2
> 44738
SIRA
100%, 3/3

Bibliography

Muhammed Kursad Ucar, M. N. (2020). The Effect of Training and Testing Process on Machine Learning in Biomedical Datasets. *The Effect of Training and Testing Process on Machine Learning in Biomedical Datasets*.

GeeksforGeeks. (2020, Oct 28). Retrieved from www.geeksforgeeks.org: https://www.geeksforgeeks.org/attributes-and-its-types-in-data-analytics/

statistics.com . (2021). Retrieved from www.statistics.com: https://www.statistics.com/glossary/attribute/

Delua, J. (2021, March 12). *Supervised vs. Unsupervised Learning: What's the Diffference?* Retrieved from www.ibm.com: https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning

javatpoint. (2021). Retrieved from www.javatpoint.com: https://www.javatpoint.com/association-rule-learning

tutorialspoint. (2021). Retrieved from www.tutorialspoint.com: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_quick_guide.htm

DeepAI. (2021). Retrieved from www.deepai.org: https://deepai.org/machine-learning-glossary-and-terms/k-means

Garbade, D. M. (2018, Sep 12). *towards data science*. Retrieved from www.towardsdatascience.com: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

Tyagi, N. (2021, Mar 22). *analyticssteps*. Retrieved from www.analyticssteps.com: https://www.analyticssteps.com/blogs/what-gini-index-and-information-gain-decision-trees