UTM

UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF

COMPUTING

SEMESTER 2/20232024

SECJ3563-06 COMPUTATIONAL INTELLIGENCE (KEPINTARAN KOMPUTER)

ASSIGNMENT 1

LECTURER: DR SHAFAATUNNUR HASAN

NAME	MATRIC NO
NG KAI ZHENG	A21EC0101
LEW CHIN HONG	A21EC0044
YEO CHUN TECK	A21EC0148
GAN HENG LAI	A21EC0176

Dataset used

[Rice (Cammeo and Osmancik) - UCI Machine Learning Repository]

Dataset Visualization and Discussion

The dataset used discusses the rice grown in Turkey. There are 2 species of rice discussed in the dataset, which are the Osmancik species and the Cammeo species. There are a total of 3810 rice grain images of those 2 species were taken and a total of 7 morphological features were obtained for data analysis.

Attribute: Species

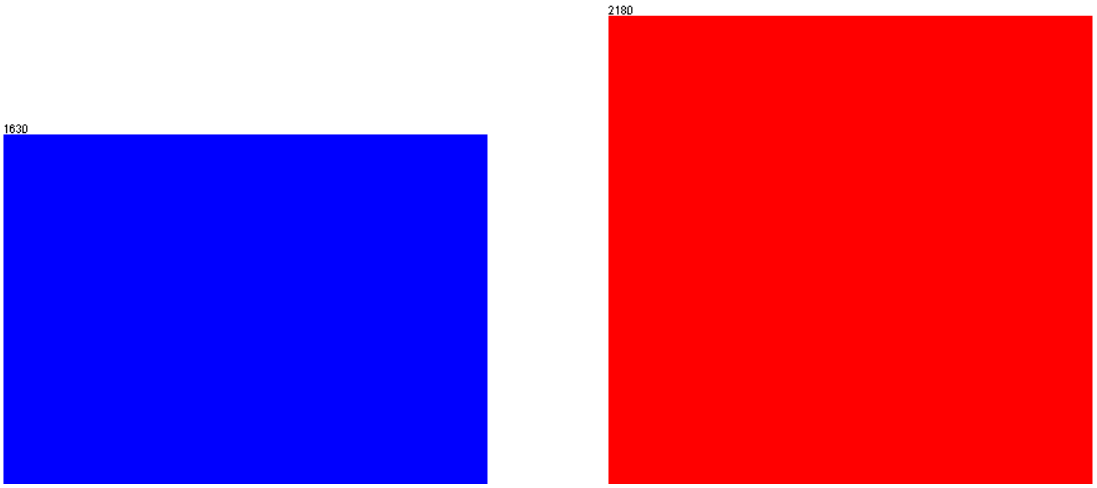
From a total of 3810 rice grain images, 1630 rice grain images are from the Cammeo species and 2180 rice grain images are from the Osmancik species.

Selected attribute

Name: Class | Type: Nominal
Missing: 0 (0%) | Distinct: 2 | Unique: 0 (0%)

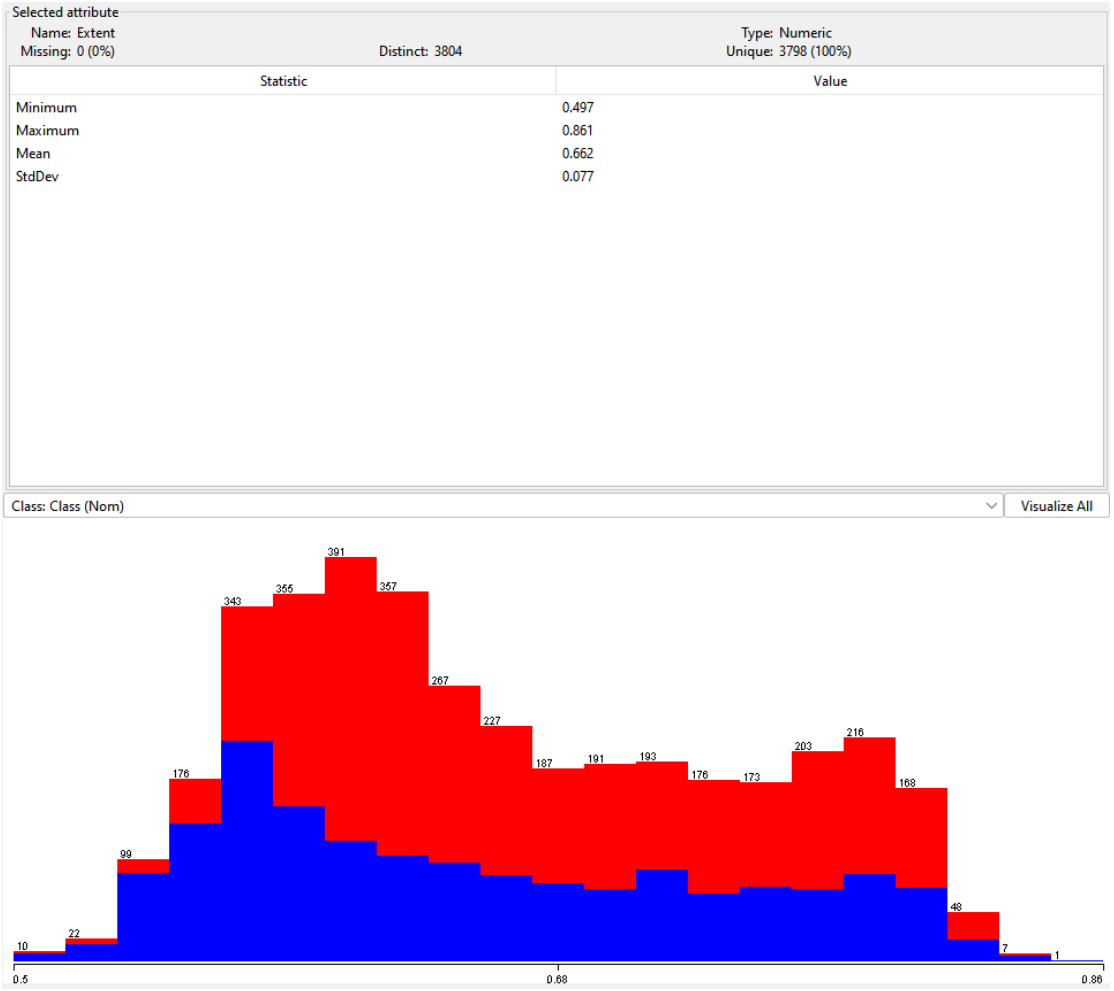
No.	Label	Count	Weight
1	Cammeo	1630	1630
2	Osmancik	2180	2180

Class: Class (Nom) Visualize All



Attribute: Extent

Extent discusses the ratio of the region formed by the rice grain to the bounding box. This is a continuous type of data. From the data, we know that the minimum range of the extent in the dataset is 0.497 (can be analyzed through visual data) the maximum is 0.861, the mean value is 0.662 and the standard deviation is 0.077.

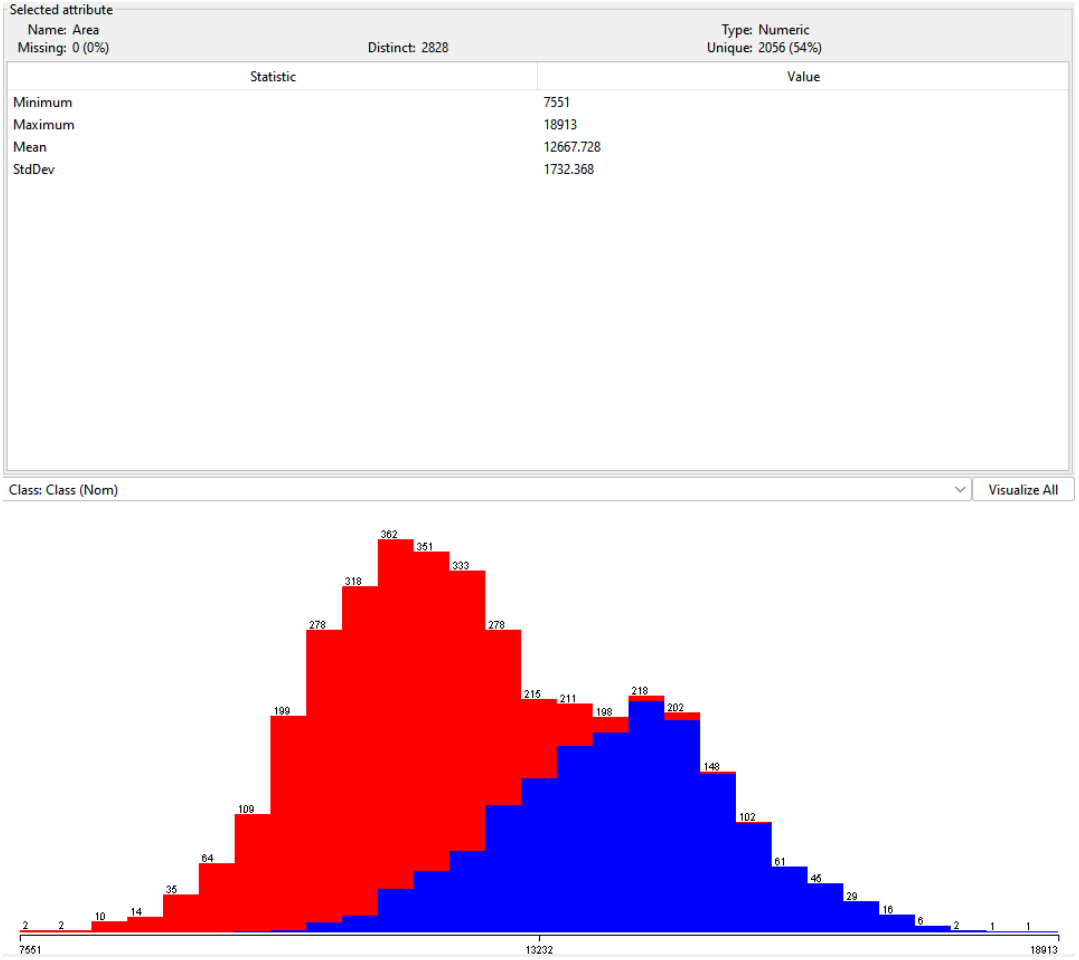


From visualizing data we can also notice that the minimum data of 0.497 was carried out by the Cammeo species as well as the maximum data of 0.861 also carried out by the Cammeo species.

Plot : Master Plot Instance: 1355 Area : 10959.0 Perimeter : 447.2330017089844 Major_Axis_Length : 197.35841369628906 Minor_Axis_Length : 71.06098175048828 Eccentricity : 0.9329289197921753 Convex_Area : 11133.0 Extent : 0.49741286039352417 Class : Cammeo	Plot : Master Plot Instance: 414 Area : 13980.0 Perimeter : 479.17999267578125 Major_Axis_Length : 206.32904052734375 Minor_Axis_Length : 87.0367202758789 Eccentricity : 0.9066728353500366 Convex_Area : 14137.0 Extent : 0.8610495328903198 Class : Cammeo

Attribute: Area

Area discusses the number of pixels within the boundaries of the rice grain. Other than the maximum value, minimum value, mean value and standard deviation, we can notice from the visual graph to obtain an initial inference which is that the Cammero species (blue) mostly have a larger area than the Osmancik species (red).



From the visual data, we can know that the minimum area of 7551 is from the Osmancik species, and the maximum area of 18913 is from the Cammeo species.

```
Plot : Master Plot
Instance: 3026
             Area : 7551.0
        Perimeter : 369.0639953613281
Major_Axis_Length : 163.40882873535156
Minor_Axis_Length : 59.532405853271484
     Eccentricity : 0.931275486946106
      Convex_Area : 7723.0
           Extent : 0.7564616203308105
            Class : Osmancik
```

```
Plot : Master Plot
Instance: 1204
             Area : 18913.0
        Perimeter : 541.7459716796875
Major_Axis_Length : 227.13548278808594
Minor_Axis_Length : 106.94835662841797
     Eccentricity : 0.8822094798088074
      Convex_Area : 19099.0
           Extent : 0.8041241765022278
            Class : Cammeo
```

Attribute: Perimeter

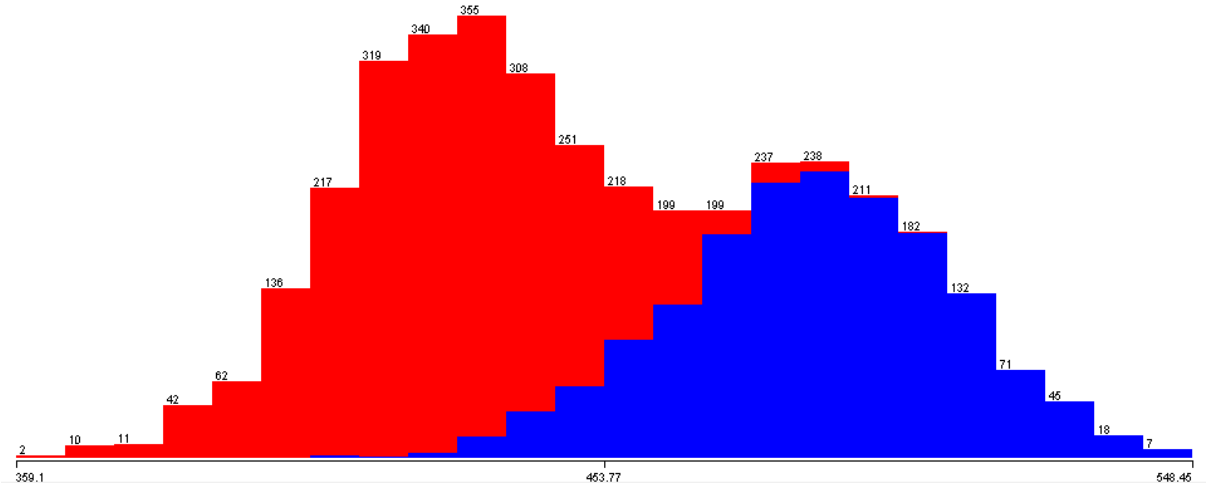
From the visual graph, we can make initial inferences that most of the Osmancik species have a smaller perimeter than the Cammeo species since most of the red colour bar lies on the left and the blue colour bar lies on the right.

Selected attribute

Name: Perimeter
Missing: 0 (0%)
Distinct: 3738
Type: Numeric
Unique: 3667 (96%)

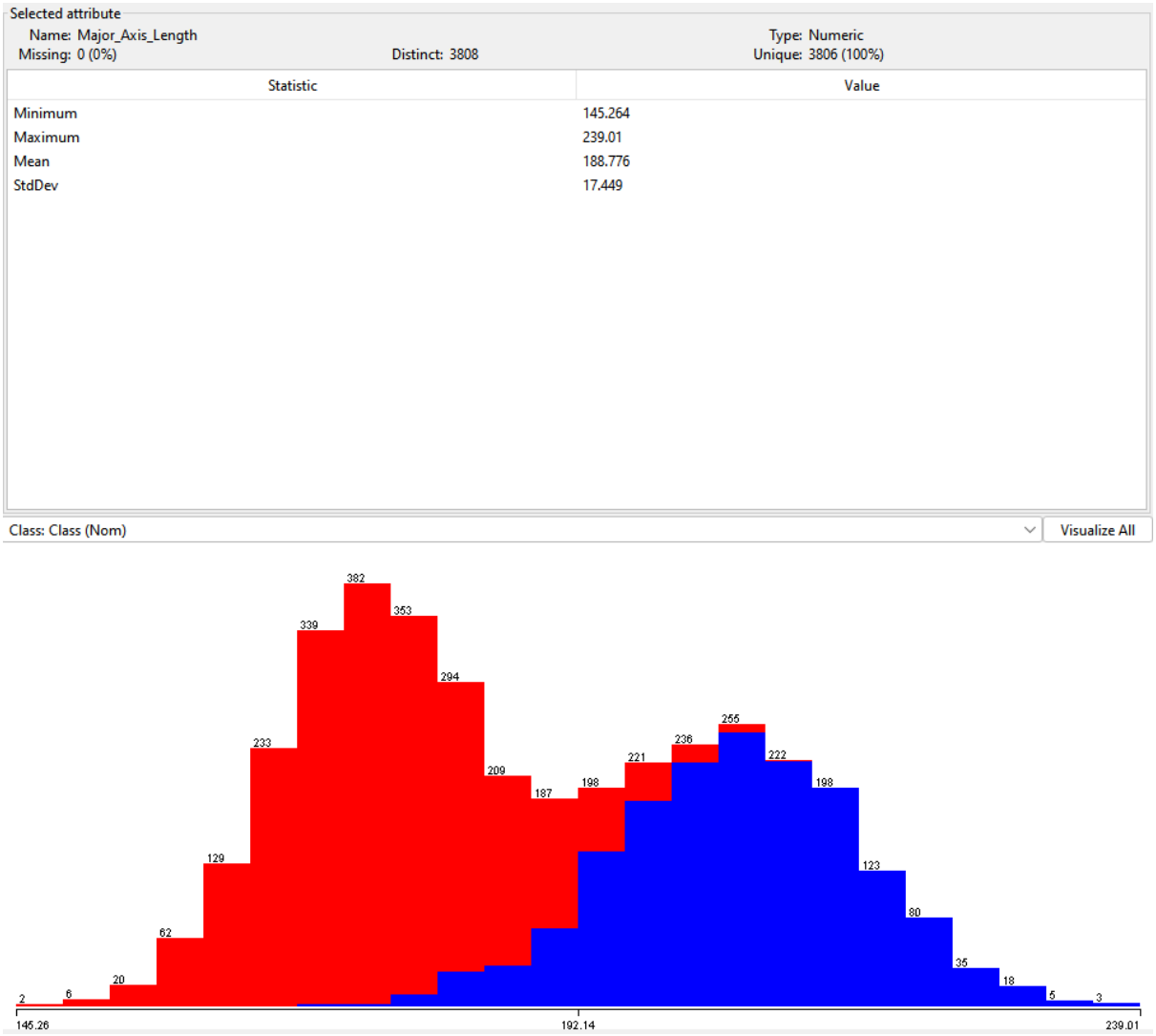
Statistic	Value
Minimum	359.1
Maximum	548.446
Mean	454.239
StdDev	35.597

Class: Class (Nom) Visualize All



Attribute: Major Axis Length

From the visualized graph, when we hover the mouse on the highest of red bar, we can notice most of the Osmancik species (382) lie in the range of 172.607 to 176.513.



Attribute: Minor Axis Length

From the visualization, the red and blue bars also closely overlap each other, so a tentative inference can be made that the two species may have similar minor axis lengths.

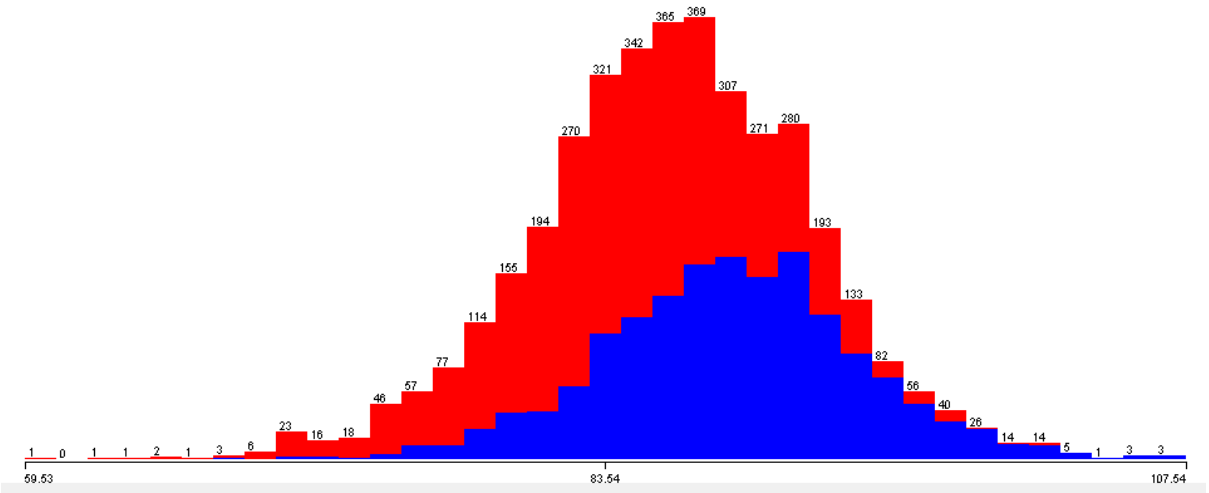
Selected attribute

Name: Minor_Axis_Length | Type: Numeric

Missing: 0 (0%) | Distinct: 3804 | Unique: 3798 (100%)

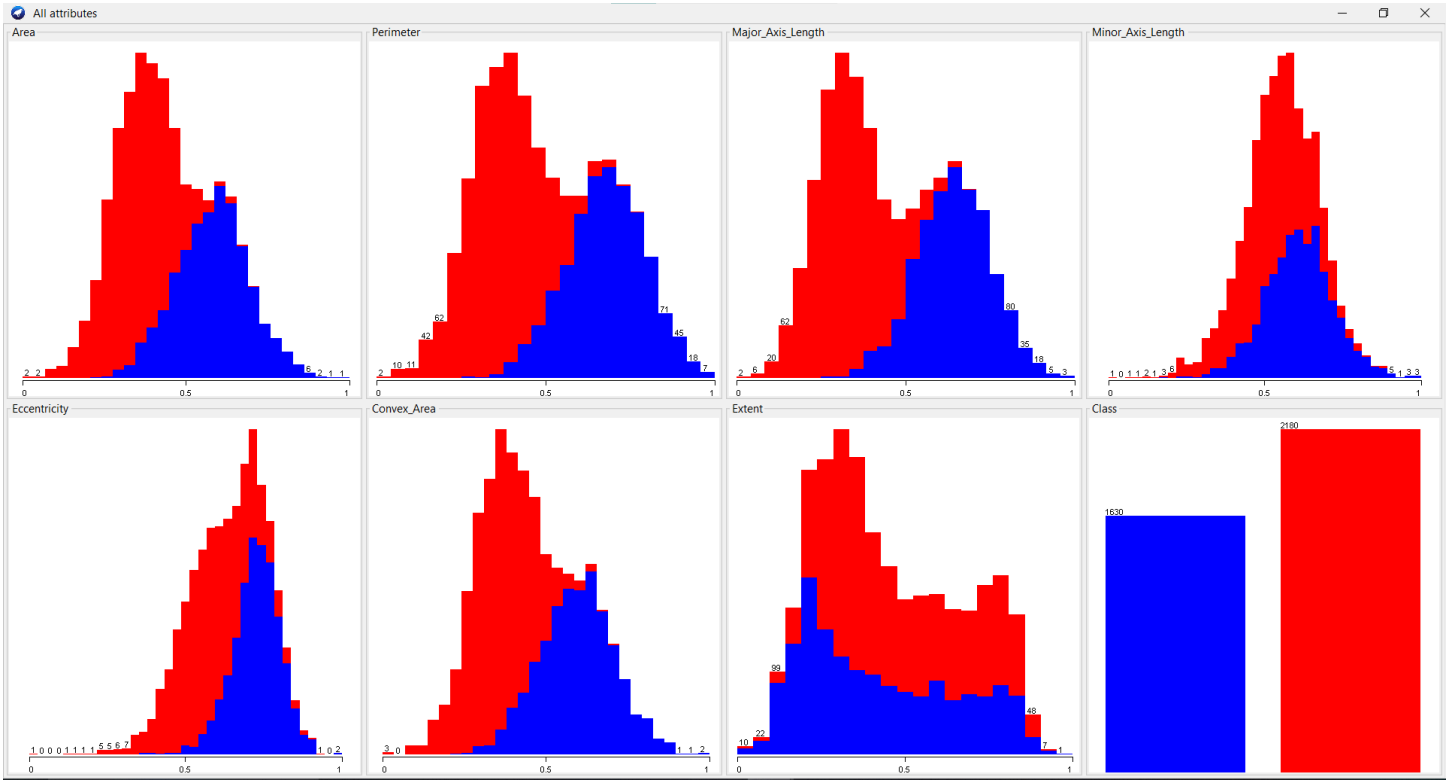
Statistic	Value
Minimum	59.532
Maximum	107.542
Mean	86.314
StdDev	5.73

Class: Class (Nom) | Visualize All



Visualization of all attribute

There are also other attributes that can be easily inferences and analysis when we look into it. However, it is preferred to perform the pre-processing of the data when we look into it.

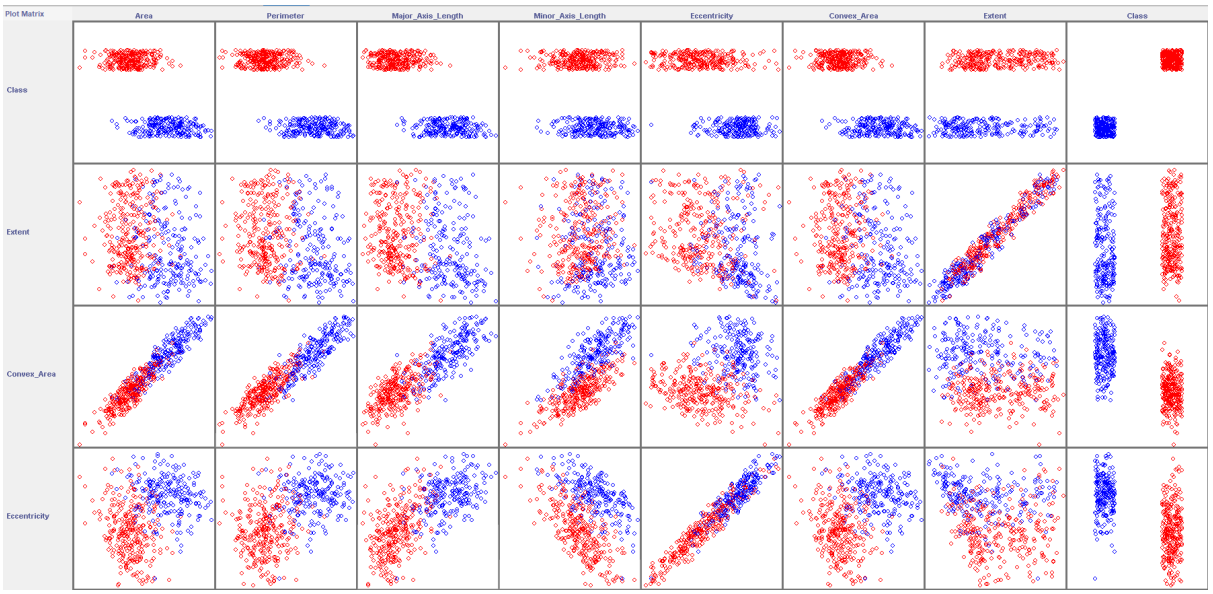


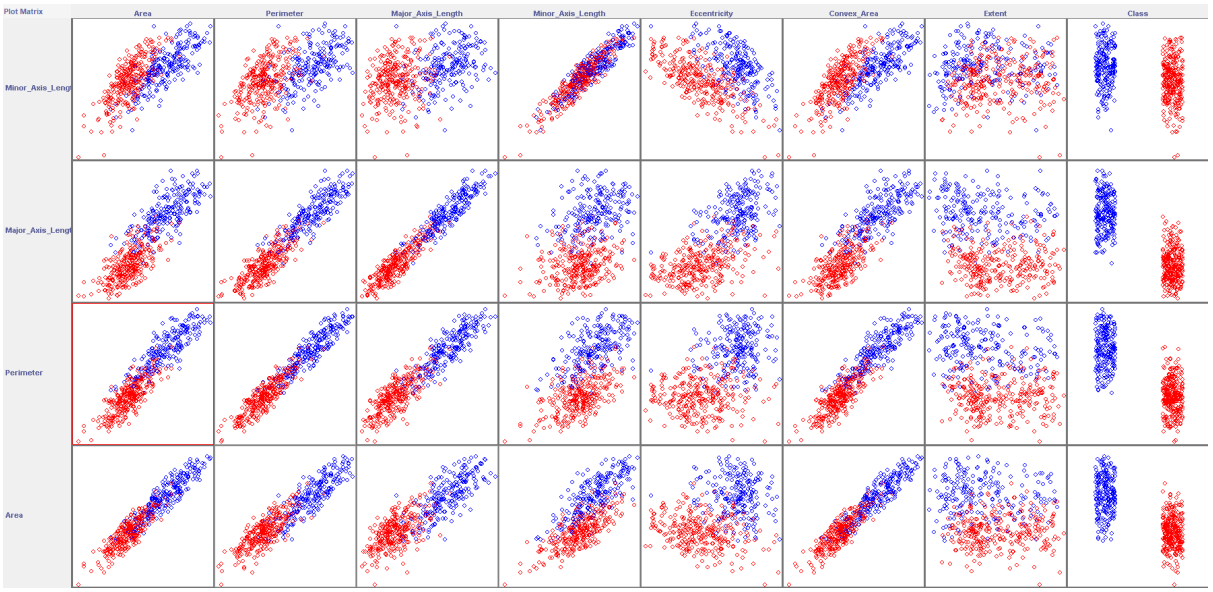
Data Visualization

By using WEKA, it allows users to visualize a 2-D plot of the dataset. We can adjust the PlotSize, PointSize and Jitter to have a better visualization. In this case, all of the attributes are visualized. From the visualization, we can notice that most of the relationships have a positive correlation, which means as one of the variables increases, the other variable increases as well.

For example:

1) As the area of a species increases, the perimeter also increases.
2) As the major axis length increases, the area also increases.
3) As the convex area increases, the perimeter also increases.





Pre-processing of data

It is preferred to perform the pre-processing of the data when we look into it so that it can ensure that all features and variables have the same scale (patterns and relationships can be found in the data regardless of scale differences), so that it becomes easier to interpret the importance of each feature in the model.

Dataset before pre-processing

No.	1: Area Numeric	2: Perimeter Numeric	3: Major_Axis_Length Numeric	4: Minor_Axis_Length Numeric	5: Eccentricity Numeric	6: Convex_Area Numeric	7: Extent Numeric	8: **Class** Nominal
1	15231.0	525.5789794921875	229.7498779296875	85.09378814697266	0.9288820028305054	15617.0	0.5728955268859863	Cammeo
2	14656.0	494.3110046386719	206.0200653076172	91.73097229003906	0.8954049944877625	15072.0	0.615436315536499	Cammeo
3	14634.0	501.12200927734375	214.10678100585938	87.7682876586914	0.9121180772781372	14954.0	0.6932588219642639	Cammeo
4	13176.0	458.3429870605469	193.33738708496094	87.44839477539062	0.8918609023094177	13368.0	0.640669047832489	Cammeo
5	14688.0	507.1669921875	211.74337768554688	89.31245422363281	0.9066908955574036	15262.0	0.6460239291191101	Cammeo
6	13479.0	477.0159912109375	200.0530548095703	86.6502914428711	0.9013283252716064	13786.0	0.6578972935676575	Cammeo
7	15757.0	509.281005859375	207.2966766357422	98.33613586425781	0.8803234696388245	16150.0	0.5897080898284912	Cammeo
8	16405.0	526.5700073242188	221.61251831054688	95.43670654296875	0.9025205969810486	16837.0	0.6588882803916931	Cammeo
9	14534.0	483.6409912109375	196.65081787109375	95.0506820678711	0.8754285573959351	14932.0	0.6496513485908508	Cammeo
10	13485.0	471.57000732421875	198.27264404296875	87.72728729248047	0.8967893123626709	13734.0	0.5723198652267456	Cammeo
11	14930.0	499.92498779296875	212.24581909179688	90.0174789428711	0.9056064486503601	15248.0	0.6243727207183838	Cammeo
12	14626.0	496.58599853515625	204.5341339111328	92.97486877441406	0.8907115459442139	15070.0	0.5702144503593445	Cammeo
13	15926.0	522.739990234375	225.73605346679688	91.05709075927734	0.9150331616401672	16240.0	0.7797688841819763	Cammeo
14	14076.0	479.677001953125	199.48915100097656	90.7099838256836	0.8906388878822327	14434.0	0.7812187671661377	Cammeo
15	13500.0	476.9150085449219	202.5466766357422	85.40547180175781	0.9067548513412476	13800.0	0.7177033424377441	Cammeo
16	14349.0	496.9460144042969	213.5440216064453	86.16077423095703	0.9149883985519409	14678.0	0.6668370962142944	Cammeo
17	15209.0	496.56500244140625	214.05007934570312	91.02632141113281	0.9050725698471069	15395.0	0.5693695545196533	Cammeo
18	15238.0	496.8710021972656	208.5317840576172	93.82839965820312	0.8930549025535583	15487.0	0.7323144674301147	Cammeo
19	13509.0	480.46600341796875	207.1371612548828	83.9401626586914	0.9142105579376221	13732.0	0.595634937286377	Cammeo
20	13699.0	480.72900390625	197.97091674804688	89.9490737915039	0.8908207416534424	14178.0	0.6720467209815979	Cammeo
21	13921.0	481.9930114746094	203.4448699951172	88.12581634521484	0.9013131856918335	14210.0	0.5613306164741516	Cammeo
22	14405.0	488.23699951171875	204.4301300048828	90.62645721435547	0.896367073059082	14742.0	0.7637857794761658	Cammeo
23	13377.0	472.96099853515625	199.94078063964844	86.68286895751953	0.9011330008506775	13685.0	0.7948306798934937	Cammeo
24	15043.0	498.29901123046875	206.01170349121094	94.76668548583984	0.8879156708717346	15414.0	0.5868836045265198	Cammeo

Pre-processing technique - Normalization

No.	1: Area Numeric	2: Perimeter Numeric	3: Major_Axis_Length Numeric	4: Minor_Axis_Length Numeric	5: Eccentricity Numeric	6: Convex_Area Numeric	7: Extent Numeric	8: **Class** Nominal
1	0.6759373349762365	0.8792316337413122	0.9012158717654067	0.5324173883610126	0.8880105461624966	0.6939170182841069	0.20757715654525274	Cammeo
2	0.6253300475268438	0.714094907735351	0.6480871586362704	0.6706631298048674	0.6919798765705287	0.6460091420534458	0.32456422596930146	Cammeo
3	0.6233937687026931	0.7500661214807537	0.7343491098256111	0.5881244713677477	0.7898463518780029	0.6356364275668073	0.5385759368713439	Cammeo
4	0.49507129026579827	0.5241356701575263	0.5127995325323011	0.5814614305538616	0.6712268046202634	0.49622011251758086	0.3939541808457925	Cammeo
5	0.6281464530892449	0.7819917159978848	0.7091384075497986	0.6202878778789477	0.758066508953558	0.6627109704641351	0.40868009187630955	Cammeo
6	0.5217391304347826	0.6227541059384585	0.5844363531008799	0.5648377563307747	0.7266650099577051	0.5329641350210971	0.44133181637654406	Cammeo
7	0.7222320014082028	0.7931565355341914	0.661704922408826	0.8082419156262204	0.6036672835115342	0.7407700421940928	0.2538116653671127	Cammeo
8	0.7792642140468228	0.884465586126432	0.8144136958920788	0.7478497752779081	0.7336465723226757	0.8011603375527426	0.4440570278279396	Cammeo
9	0.61459250132019	0.6577429664231816	0.5481442899601872	0.739809281203158	0.5750042406630372	0.6337025316455697	0.41865548695083327	Cammeo
10	0.5222672064777328	0.5939920273853334	0.5654445012321485	0.5872704757720068	0.7000859999483442	0.5283931082981715	0.20599408832694538	Cammeo
11	0.6494455201549023	0.7437442471088271	0.7144980093558647	0.6349728200089865	0.7517163341963577	0.6614803094233473	0.34913931934623116	Cammeo
12	0.622689667312093	0.7261099181576436	0.63223655297913	0.6965722183663878	0.6644965408056492	0.6458333333333334	0.20020420235932568	Cammeo
13	0.737106143284633	0.8642379735059568	0.8583999322888013	0.6566268683636665	0.8069161549367251	0.7486814345991561	0.7764784059037397	Cammeo
14	0.574282696708326	0.6368078015772128	0.578421124591861	0.6493969867812638	0.6640710783988653	0.5899261603375527	0.7804655807236119	Cammeo
15	0.5235873965851082	0.6222207824044816	0.6110361115541999	0.5389094393607905	0.7584410135987418	0.5341947960618847	0.6057983110770027	Cammeo
16	0.5983101566625594	0.7280112834433263	0.7283460888644376	0.5546416146460086	0.8066540365876729	0.6113748241912799	0.46591625277360665	Cammeo
17	0.6740010561520859	0.7259990307018032	0.7337442665216952	0.6559859743856941	0.7485901104247713	0.6744022503516175	0.19788074077364468	Cammeo
18	0.6765534236930117	0.727615118666501	0.674879959112853	0.7143503916596381	0.6782184887324012	0.6824894514767933	0.6459788706780132	Cammeo
19	0.5243795106495336	0.6409747845471239	0.6600033530064748	0.5083885521048155	0.8020992503624632	0.5282172995780591	0.27011048203263466	Cammeo
20	0.5411019186762894	0.6423637788704275	0.5622259405508697	0.6335480108330629	0.6651359560060005	0.5674226441631505	0.4802427087152839	Cammeo
21	0.5606407322654462	0.6490394293551008	0.6206172461725268	0.595571427379585	0.7265763574135515	0.5702355836849508	0.175773679925505	Cammeo
22	0.6032388663967612	0.6820160370983009	0.6311271310309355	0.6476572130964531	0.6976135014334488	0.6170007032348804	0.732524905295378	Cammeo
23	0.5127618377046295	0.6013383213347626	0.5832387112735891	0.5655163125633416	0.7255212525278191	0.5240857946554149	0.8178983089297472	Cammeo
24	0.6593909522971307	0.7351569162263	0.6479979621533464	0.7338939235448286	0.6481247892756947	0.6760724331926864	0.24604433738399697	Cammeo

If data has outliers, use standardization, otherwise use normalization. Applying standardization makes all attributes have the same standard deviation of 1. However, in this case, we apply normalization. (Folds cross-validation and Percentage-split)

Experimental results

Epoch	Random Seed	MLP Architecture (No. of Hidden layers)	Learning rates	momentum rates	10-fold cv (accuracy, precision , recall and etc)	Hold-out (accuracy, precision , recall and etc)
1	1	8, 4	0.3	0.2	**Class Cammeo** Accuracy: 0.918 Precision: 0.895 Recall: 0.917 F-measure: 0.906 **Class Osmancik** Accuracy: 0.918 Precision: 0.937 Recall: 0.919 F-measure: 0.928 **Average** Accuracy: 0.918 Precision: 0.919 Recall: 0.918 F-measure: 0.918	**Class Cammeo** Accuracy: 0.623 Precision: 0.530 Recall: 0.998 F-measure: 0.692 **Class Osmancik** Accuracy: 0.623 Precision: 0.996 Recall: 0.346 F-measure: 0.513 **Average** Accuracy: 0.623 Precision: 0.798 Recall: 0.623 F-measure: 0.589
2	2	4,2	0.4	0.6	**Class Cammeo** Accuracy: 0.920 Precision: 0.905 Recall: 0.909 F-measure: 0.907 **Class Osmancik** Accuracy: 0.920 Precision: 0.932 Recall: 0.929 F-measure: 0.930 **Average** Accuracy: 0.920	**Class Cammeo** Accuracy: 0.914 Precision: 0.922 Recall: 0.872 F-measure: 0.896 **Class Osmancik** Accuracy: 0.914 Precision: 0.909 Recall: 0.945 F-measure: 0.927 **Average** Accuracy: 0.914

					Precision: 0.921 Recall: 0.920 F-measure: 0.920	Precision: 0.915 Recall: 0.914 F-measure: 0.914
3	3	8	0.4	0.1	**Class Cammeo** Accuracy: 0.922 Precision: 0.904 Recall: 0.915 F-measure: 0.909 **Class Osmancik** Accuracy: 0.922 Precision: 0.936 Recall: 0.927 F-measure: 0.931 **Average** Accuracy: 0.922 Precision: 0.922 Recall: 0.922 F-measure: 0.922	**Class Cammeo** Accuracy: 0.921 Precision: 0.894 Recall: 0.924 F-measure: 0.909 **Class Osmancik** Accuracy: 0.921 Precision: 0.942 Recall: 0.919 F-measure: 0.931 **Average** Accuracy: 0.921 Precision: 0.922 Recall: 0.921 F-measure: 0.921
4	4	4	0.3	0.3	**Class Cammeo** Accuracy: 0.927 Precision: 0.909 Recall: 0.921 F-measure: 0.915 **Class Osmancik** Accuracy: 0.927 Precision: 0.941 Recall: 0.931 F-measure: 0.936 **Average** Accuracy: 0.927 Precision: 0.927 Recall: 0.927 F-measure: 0.927	**Class Cammeo** Accuracy: 0.923 Precision: 0.936 Recall: 0.879 F-measure: 0.907 **Class Osmancik** Accuracy: 0.923 Precision: 0.914 Recall: 0.956 F-measure: 0.935 **Average** Accuracy: 0.923 Precision: 0.924 Recall: 0.923 F-measure: 0.923

5	5	5	0.1	0.2	**Class Cammeo** Accuracy: 0.928 Precision: 0.918 Recall: 0.913 F-measure: 0.916 **Class Osmancik** Accuracy: 0.928 Precision: 0.936 Recall: 0.939 F-measure: 0.937 **Average** Accuracy: 0.928 Precision: 0.928 Recall: 0.928 F-measure: 0.928	**Class Cammeo** Accuracy: 0.923 Precision: 0.916 Recall: 0.901 F-measure: 0.909 **Class Osmancik** Accuracy: 0.923 Precision: 0.928 Recall: 0.939 F-measure: 0.933 **Average** Accuracy: 0.923 Precision: 0.923 Recall: 0.923 F-measure: 0.923
6	6	3	0.1	0.4	**Class Cammeo** Accuracy: 0.928 Precision: 0.920 Recall: 0.912 F-measure: 0.916 **Class Osmancik** Accuracy: 0.928 Precision: 0.935 Recall: 0.940 F-measure: 0.938 **Average** Accuracy: 0.928 Precision: 0.928 Recall: 0.928 F-measure: 0.928	**Class Cammeo** Accuracy: 0.918 Precision: 0.894 Recall: 0.916 F-measure: 0.904 **Class Osmancik** Accuracy: 0.918 Precision: 0.936 Recall: 0.919 F-measure: 0.928 **Average** Accuracy: 0.918 Precision: 0.918 Recall: 0.918 F-measure: 0.918
7	7	5,2	0.2	0.3	**Class Cammeo** Accuracy: 0.926 Precision: 0.921	**Class Cammeo** Accuracy: 0.923 Precision: 0.934

					Recall: 0.904 F-measure: 0.912 **Class Osmancik** Accuracy: 0.926 Precision: 0.929 Recall: 0.942 F-measure: 0.936 **Average** Accuracy: 0.926 Precision: 0.926 Recall: 0.926 F-measure: 0.926	Recall: 0.881 F-measure: 0.907 **Class Osmancik** Accuracy: 0.923 Precision: 0.915 Recall: 0.954 F-measure: 0.934 **Average** Accuracy: 0.923 Precision: 0.923 Recall: 0.923 F-measure: 0.923
8	8	6,3	0.1	0.3	**Class Cammeo** Accuracy: 0.927 Precision: 0.924 Recall: 0.903 F-measure: 0.913 **Class Osmancik** Accuracy: 0.927 Precision: 0.929 Recall: 0.944 F-measure: 0.937 **Average** Accuracy: 0.927 Precision: 0.927 Recall: 0.927 F-measure: 0.927	**Class Cammeo** Accuracy: 0.922 Precision: 0.963 Recall: 0.850 F-measure: 0.903 **Class Osmancik** Accuracy: 0.922 Precision: 0.898 Recall: 0.976 F-measure: 0.935 **Average** Accuracy: 0.922 Precision: 0.925 Recall: 0.922 F-measure: 0.921
9	9	5	0.1	0.1	**Class Cammeo** Accuracy: 0.927 Precision: 0.916 Recall: 0.914 F-measure: 0.915 **Class Osmancik**	**Class Cammeo** Accuracy: 0.924 Precision: 0.911 Recall: 0.909 F-measure: 0.910 **Class Osmancik**

					Accuracy: 0.927 Precision: 0.936 Recall: 0.937 F-measure: 0.937 **Average** Accuracy: 0.927 Precision: 0.927 Recall: 0.927 F-measure: 0.927	Accuracy: 0.924 Precision: 0.933 Recall: 0.935 F-measure: 0.934 **Average** Accuracy: 0.924 Precision: 0.924 Recall: 0.924 F-measure: 0.924
10	10	6,4	0.2	0.4	**Class Cammeo** Accuracy: 0.923 Precision: 0.913 Recall: 0.907 F-measure: 0.910 **Class Osmancik** Accuracy: 0.923 Precision: 0.931 Recall: 0.935 F-measure: 0.933 **Average** Accuracy: 0.923 Precision: 0.923 Recall: 0.923 F-measure: 0.923	**Class Cammeo** Accuracy: 0.923 Precision: 0.936 Recall: 0.879 F-measure: 0.907 **Class Osmancik** Accuracy: 0.923 Precision: 0.914 Recall: 0.956 F-measure: 0.935 **Average** Accuracy: 0.923 Precision: 0.924 Recall: 0.923 F-measure: 0.923
	Average				**Class Cammeo** Accuracy: 0.924 Precision: 0.913 Recall: 0.912 F-measure: 0.912 **Class Osmancik** Accuracy: 0.924 Precision: 0.934 Recall: 0.934	**Class Cammeo** Accuracy: 0.891 Precision: 0.884 Recall: 0.901 F-measure: 0.884 **Class Osmancik** Accuracy: 0.891 Precision: 0.929 Recall: 0.885

					F-measure: 0.934 **Average** Accuracy: 0.924 Precision: 0.924 Recall: 0.924 F-measure: 0.924	F-measure: 0.891 **Average** Accuracy: 0.891 Precision: 0.910 Recall: 0.891 F-measure: 0.888

Discussion

From the table, we run the experiment using the MLP-BP algorithm for 1 to 10 epochs, as well as 1 to 10 seeds. Several of the parameters have been changed to compare the performances of the 10-fold cross-validation and hold-out (training-70%/testing-30%).

The parameters being adjusted:

1) MLP architecture being applied
2) Learning rate
3) Momentum rates

We applied 2 different MLP architectures into it which are one hidden layer and two hidden layers. For comparison, we analyze the accuracy, precision, recall and F-measure for each class and their average. Lastly, we found out that for this rice (Cammeo and Osmancik) dataset, using the 10-fold cross-validation would be more accurate than using the hold-out (training-70%/testing-30%). From the aspect of accuracy, precision, recall, and F-measure, all of the results taken from the 10-fold cross-validation are higher than using the Hold-out algorithm.

Comparison	10-Fold Cross Validation	Hold-Out (Training-70%/Testing-30%)
Accuracy	0.924	0.891
Precision	0.924	0.910
Recall	0.924	0.891
F-measure	0.924	0.888

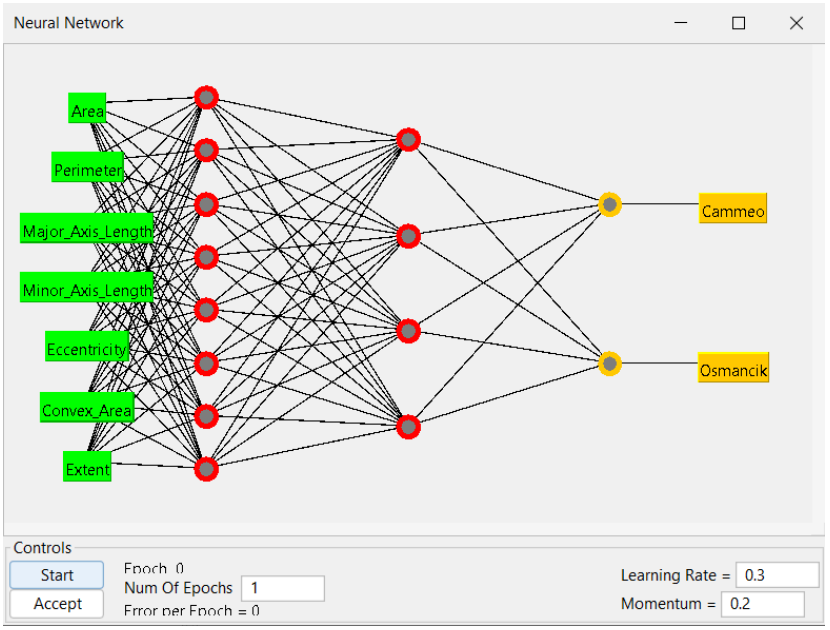


Figure: Neural Network Diagram with first hidden layer has 8 nodes, second hidden layer has 4 nodes, epoch = 1, Learning rate = 0.3 and Momentum rate = 0.2

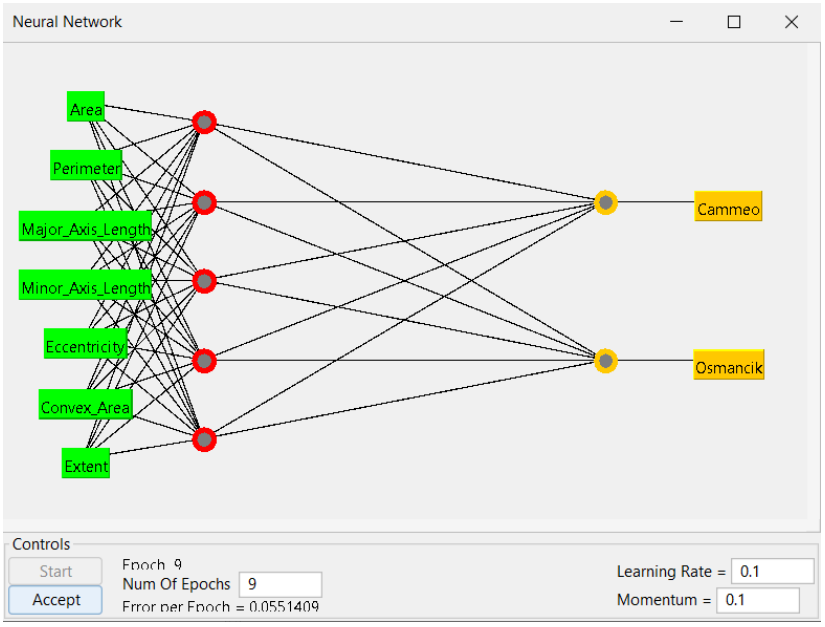


Figure: Neural Network Diagram with first hidden layer has 5 nodes, Epoch = 9, Learning rate = 0.1 and Momentum rate = 0.1

Below is a figure taken from a 10-fold cross-validation algorithm. By setting the x-axis and y-axis to the predicted class. We can visualize the classes that are predicted correctly and incorrectly.

Figure: Visualize Classifier Errors

For example, when the red rectangle at the left-bottom corner is clicked, the instance info will be shown. It shows the details of the instance, including the class of it actually is Osmancik, while the predicted class is Cammeo, so it is an incorrectly classified instance.

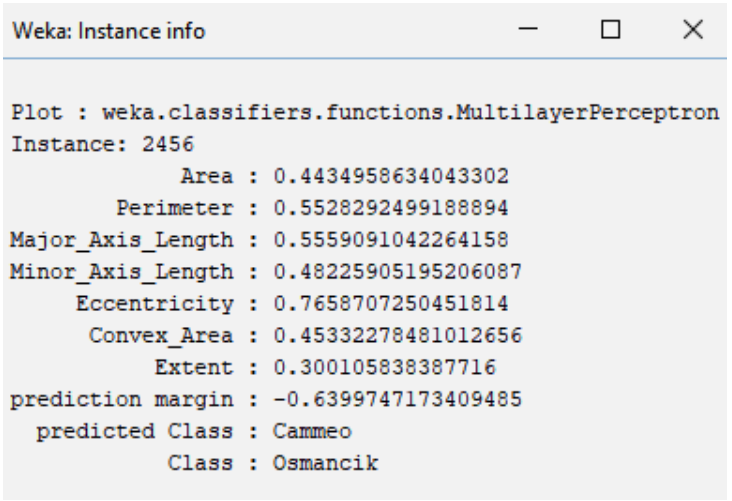


Figure: Incorrectly Classified Instance Info

Conclusion

In this assignment, we selected a dataset from the UCI Machine Learning Repository for a classification problem. After loading the data into Weka, we visually inspected its attributes and target variables. Pre-processing techniques like normalization or standardization were applied to prepare the dataset for modelling, and both the original and pre-processed datasets were saved for comparison. The MLP-BP algorithm was utilized for classification in our study. We conducted experiments over 10 epochs, employing different random seeds ranging from 1 to 10. To compare the effectiveness of two data division methods, namely 10-fold cross-validation and a Hold-out(70% training / 30% testing) split, we adjusted parameters such as learning rates, momentum rates, and hidden layers. The effects of these adjustments were observed by extracting and comparing performance metrics like accuracy, precision, and recall for both data division approaches. Furthermore, we thoroughly discussed which approaches have better accuracy on the tested dataset, and supported our findings with visualizations of misclassification results.

Through this assignment, the utilization of machine learning techniques provided us with valuable insights into the step-by-step process of solving a classification problem. We have acquired knowledge of dataset selection, pre-processing, and experimentation using the MLP-BP algorithm. By comparing various data division methods, we have gained a comprehensive understanding of their influence on model evaluation. Through the analysis of results and the visualization of misclassification outcomes, we have deepened our understanding of the model's behaviour and identified specific areas that can be improved. In short, this assignment has provided us with comprehensive practical experience in solving classification problems in real-world scenarios, emphasizing the iterative process of developing and evaluating models.