ARTIFICIAL INTELLIGENCE

ASSIGNMENT REPORT

Table of Contents

Dataset

Dataset Name:

- Dry Bean Dataset

Dataset Description:

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.

Attributes:

- Area (A): The area of a bean zone and the number of pixels within its boundaries.
- Perimeter (P): Bean circumference is defined as the length of its border.
- Major axis length (L): The distance between the ends of the longest line that can be drawn from a bean.
- Minor axis length (l): The longest line that can be drawn from the bean while standing perpendicular to the main axis.
- Aspect ratio (K): Defines the relationship between L and l.
- Eccentricity (Ec): Eccentricity of the ellipse having the same moments as the region.
- Convex area (C): Number of pixels in the smallest convex polygon that can contain the area of a bean seed.
- Equivalent diameter (Ed): The diameter of a circle having the same area as a bean seed area.
- Extent (Ex): The ratio of the pixels in the bounding box to the bean area.

- Solidity (S): Also known as convexity. The ratio of the pixels in the convex shell to those found in beans.
- Roundness (R): Calculated with the following formula: (4piA)/(P^2)
- Compactness (CO): Measures the roundness of an object: Ed/L
- ShapeFactor1 (SF1)
- ShapeFactor2 (SF2)
- ShapeFactor3 (SF3)
- ShapeFactor4 (SF4)

Class/Label:

- Class: Type of bean

This is what our dataset looked like when we loaded it into a Pandas DataFrame.

	Area	Perimeter	MajorAxisLength	MinorAxisLength	AspectRation	Eccentricity	ConvexArea	EquivDiameter	Extent	Solidity	roundness	Compactness	ShapeFactor1	ShapeFactor2
0	53931	915.794	364.579544	189.297004	1.925966	0.854641	54651	262.044046	0.647400	0.986825	0.808077	0.718757	0.006760	0.00111
1	31746	669.002	242.502507	167.580320	1.447082	0.722811	32134	201.047911	0.696765	0.987926	0.891341	0.829055	0.007639	0.00222
2	63206	1035.302	416.580206	194.290238	2.144113	0.884577	64172	283.683589	0.692638	0.984947	0.741027	0.680982	0.006591	0.00087
3	38134	717.612	247.751869	196.271832	1.262289	0.610247	38522	220.349079	0.778404	0.989928	0.930557	0.889394	0.006497	0.00250
4	54471	915.945	378.793963	183.501841	2.064252	0.874826	54914	263.352675	0.806655	0.991933	0.815899	0.695240	0.006954	0.00100
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13606	39078	734.170	280.680490	178.028368	1.576605	0.773108	39474	223.059756	0.780935	0.989968	0.911064	0.794711	0.007183	0.00176
13607	60239	966.280	394.180245	195.988834	2.011238	0.867632	60865	276.945260	0.623779	0.989715	0.810740	0.702585	0.006544	0.00098
13608	50724	849.547	316.374029	205.189204	1.541865	0.761159	51283	254.133435	0.694488	0.989100	0.883179	0.803269	0.006237	0.00160
13609	54306	926.309	374.438736	191.626421	1.954004	0.859123	56858	262.953507	0.776110	0.955116	0.795327	0.702260	0.006895	0.00103
13610	33608	678.629	247.475659	173.817327	1.423769	0.711821	34022	206.859940	0.724841	0.987831	0.917039	0.835880	0.007364	0.00221

13611 rows × 17 columns

Figure 1

Data Preprocessing / Normalization

Steps:

The following steps were taken to preprocess and normalize the dataset.

1. The dataset contained a total of **68** duplicate rows which could make the model skewed so those rows were dropped from the dataset.
2. The target label/class was categorical so we encoded it by assigning each class a unique number because KNN works for numerical data. The values were saved incase we wanted to find out the true labels.
3. The dataset had 16 attributes and each attribute had a different value range, so the dataset was normalized using **Min-Max Scaling** in order for the model to perform better.
4. The dataset had a total of **13453** rows so I reduced them to **700** such that each class label had 100 rows each and the model would be optimal.

This is what the dataset looked like after performing the above steps.

	Area	Perimeter	MajorAxisLength	MinorAxisLength	AspectRation	Eccentricity	ConvexArea	EquivDiameter	Extent	Solidity	roundness	Compactness	ShapeFactor1	ShapeFactor
0	0.158329	0.271721	0.326923	0.239338	0.548470	0.883270	0.155860	0.267885	0.591340	0.845726	0.720612	0.289757	0.465525	0.1989
1	0.143939	0.278537	0.349434	0.181036	0.733872	0.945645	0.141192	0.248164	0.122845	0.876533	0.587129	0.157543	0.547140	0.1422
2	0.118969	0.237463	0.320803	0.147074	0.765630	0.953834	0.116037	0.212424	0.941393	0.921245	0.616828	0.129364	0.614351	0.1470
3	0.180093	0.311371	0.378003	0.239821	0.646614	0.919804	0.175470	0.296645	0.273960	0.933901	0.659016	0.221739	0.457189	0.1493
4	0.144588	0.280695	0.358070	0.173940	0.772039	0.955418	0.141584	0.249066	0.151879	0.890600	0.580963	0.135181	0.556107	0.1310
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
695	0.615019	0.692742	0.680634	0.745584	0.338258	0.760336	0.599649	0.726112	0.468109	0.935712	0.769681	0.502780	0.082948	0.1175
696	0.714602	0.787131	0.819159	0.753444	0.475933	0.849361	0.697989	0.802977	0.855287	0.915453	0.702479	0.361343	0.081064	0.0507
697	0.572802	0.654753	0.618082	0.747811	0.270224	0.697947	0.558660	0.691887	0.656424	0.931881	0.789988	0.581261	0.082127	0.1590
698	0.651254	0.760342	0.827598	0.666898	0.586813	0.898652	0.639331	0.754675	0.534240	0.857115	0.644687	0.256845	0.122607	0.0277
699	0.846804	0.880590	0.898435	0.857509	0.449166	0.834944	0.836118	0.897979	0.475262	0.787693	0.695567	0.382730	0.044571	0.0399

700 rows × 17 columns

Figure 2

Train / Test Split

The dataset was split into 2 sets, training and testing. About 80% of the data was used for training and the remaining 20% was used for testing. Labels were also removed from the test data and saved for later use in evaluations.

The following was what both sets looked like after splitting.

Train Set:

	Area	Perimeter	MajorAxisLength	MinorAxisLength	AspectRation	Eccentricity	ConvexArea	EquivDiameter	Extent	Solidity	roundness	Compactness	ShapeFactor1	ShapeFactor
0	0.158329	0.271721	0.326923	0.239338	0.548470	0.883270	0.155860	0.267885	0.591340	0.845726	0.720612	0.289757	0.465525	0.19899
1	0.143939	0.278537	0.349434	0.181036	0.733872	0.945645	0.141192	0.248164	0.122845	0.876533	0.587129	0.157543	0.547140	0.14222
2	0.118969	0.237463	0.320803	0.147074	0.765630	0.953834	0.116037	0.212424	0.941393	0.921245	0.616828	0.129364	0.614351	0.14702
3	0.180093	0.311371	0.378003	0.239821	0.646614	0.919804	0.175470	0.296645	0.273960	0.933901	0.659016	0.221739	0.457189	0.14939
4	0.144588	0.280695	0.358070	0.173940	0.772039	0.955418	0.141584	0.249066	0.151879	0.890600	0.580963	0.135181	0.556107	0.13106
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
545	0.615019	0.692742	0.680634	0.745584	0.338258	0.760336	0.599649	0.726112	0.468109	0.935712	0.769681	0.502780	0.082948	0.11759
546	0.714602	0.787131	0.819159	0.753444	0.475933	0.849361	0.697989	0.802977	0.855287	0.915453	0.702479	0.361343	0.081064	0.05075
547	0.572802	0.654753	0.618082	0.747811	0.270224	0.697947	0.558660	0.691887	0.656424	0.931881	0.789988	0.581261	0.082127	0.15902
548	0.651254	0.760342	0.827598	0.666898	0.586813	0.898652	0.639331	0.754675	0.534240	0.857115	0.644687	0.256845	0.122607	0.02772
549	0.846804	0.880590	0.898435	0.857509	0.449166	0.834944	0.836118	0.897979	0.475262	0.787693	0.695567	0.382730	0.044571	0.03998

550 rows × 17 columns

Figure 3

Test Set:

	Area	Perimeter	MajorAxisLength	MinorAxisLength	AspectRation	Eccentricity	ConvexArea	EquivDiameter	Extent	Solidity	roundness	Compactness	ShapeFactor1	ShapeFactor
0	0.206916	0.347448	0.436982	0.248446	0.740090	0.947294	0.202146	0.330525	0.824330	0.910647	0.644091	0.156317	0.444436	0.10490
1	0.132675	0.234975	0.295306	0.197753	0.577267	0.894967	0.129060	0.232294	0.478336	0.943322	0.737068	0.277175	0.517651	0.21354
2	0.178778	0.339848	0.426278	0.198032	0.849838	0.973000	0.174765	0.294941	0.076557	0.904737	0.521026	0.084998	0.517323	0.08862
3	0.139879	0.264671	0.338357	0.180466	0.711553	0.939528	0.136287	0.242488	0.371152	0.931192	0.628712	0.172757	0.548326	0.15261
4	0.181408	0.328360	0.394386	0.229036	0.704074	0.937407	0.177548	0.298345	0.218041	0.894224	0.586821	0.179951	0.472025	0.12896
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
145	0.611189	0.684719	0.704483	0.714995	0.394442	0.801507	0.594096	0.723050	0.785557	0.970810	0.786960	0.442360	0.095940	0.09584
146	0.621646	0.684360	0.694098	0.743424	0.354582	0.773126	0.606071	0.731390	0.705105	0.936490	0.814605	0.482818	0.084657	0.10866
147	0.691476	0.759846	0.765133	0.778095	0.394491	0.801540	0.675637	0.785573	0.874033	0.911328	0.734545	0.436577	0.072785	0.07914
148	0.750619	0.836885	0.898924	0.732943	0.583647	0.897440	0.737444	0.829590	0.910173	0.847429	0.634960	0.263935	0.091432	0.01689
149	0.843981	0.870061	0.949588	0.806009	0.552290	0.884874	0.823301	0.896024	0.804363	0.930951	0.719200	0.290384	0.062777	0.01383

150 rows × 16 columns

Figure 4

K-Nearest Neighbors

Implementation:

The following was the way the algorithm was implemented.

1. For each row in the test dataset, the Euclidean distance of the test data row was calculated from all the rows in the training set.
2. After that, K rows from the train data which had the least Euclidean distance from the test data were selected.
3. Finally, the label which was the most common from the K neighbors was assigned to the test row.
4. Similarly, this process was carried out for each test row and varying values of K.

Evaluation / Visualization:

The true positives, true negatives, false positives and false negatives were calculated from the confusion matrix that was created. Using those, the following were calculated.

- F1-score, Precision, Recall and Accuracy were calculated for each class label and each value of K.
- Micro and Macro scores were calculated for each value of K.

Macro-Scores:

Macro scores were plotting using bar plots. Here is what they looked like. The different evaluation metrics such as F1 score, Accuracy score, Recall score and Precision are on the X-axis and their value on Y-axis.

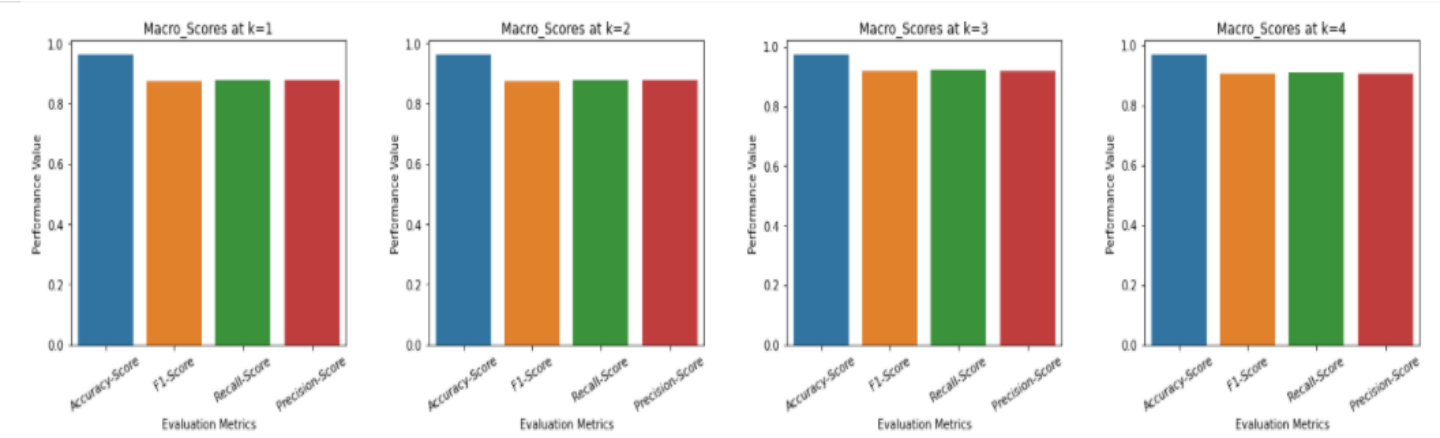

Figure 5

Micro-Scores:

Micro scores were plotting using bar plots. Here is what they looked like. The different evaluation metrics such as F1 score, Accuracy score, Recall score and Precision are on the X-axis and their value on Y-axis.

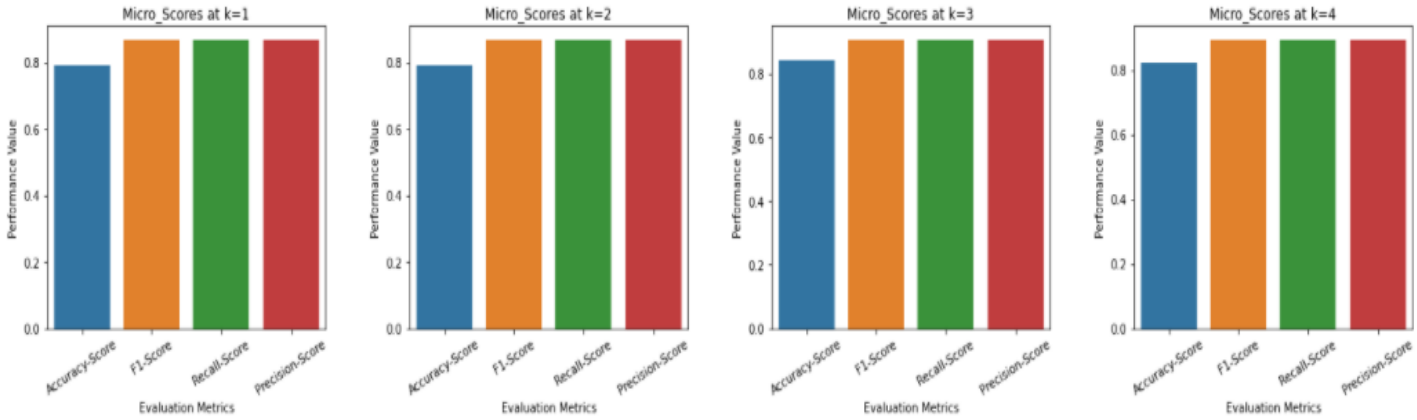

Figure 6

Best Performing K Value:

The best k value was calculated on the basis of the weighted average of the f1 scores. Here are how the varying values of K performed. We can see that it performs the best on values **3** and **5**.

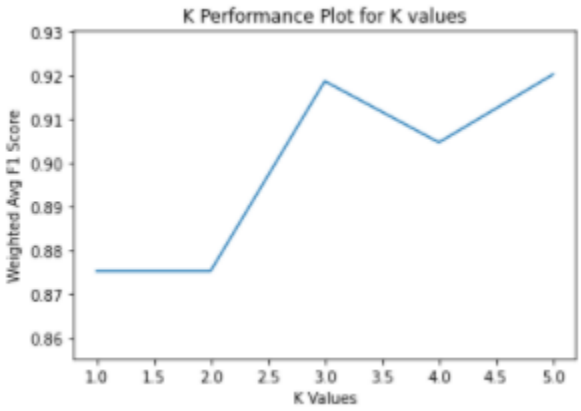

Figure 7

Confusion Matrix Of Best K Value:

A confusion matrix was created highlighted all the labels with their predicted values. Here is what it looked like. The values on the X and Y axis are the class labels which indicate the type of beans.

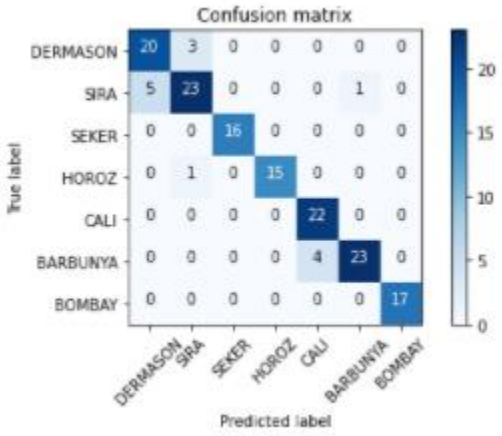

Figure 8

K-Means

Implementation:

The following was the way the algorithm was implemented.

1. Based on the value of K, K initial centroids were chosen from the data set to produce K clusters.
2. The Euclidean Distance of each row in the dataset was computed from each of the K clusters.
3. The closest cluster based on the least Euclidean distance was assigned to the data point.
4. This way, all rows in the data set were assigned a cluster.
5. The clusters were then recomputed and recentered based on the average sample in their respective clusters.
6. This process was repeated till the values of cluster did not change for 2 consecutive iterations.
7. This was carried out for varying values of K as well.

Evaluation / Visualization:

Davies Bouldin Index:

The Davies Bouldin index was used to evaluate the clusters.

Measurement with Davies- Bouldin Index is maximizing inter-cluster distance and on same time try for minimize distance between point in a cluster. Here is the plot showing the DBI of the different values of K.

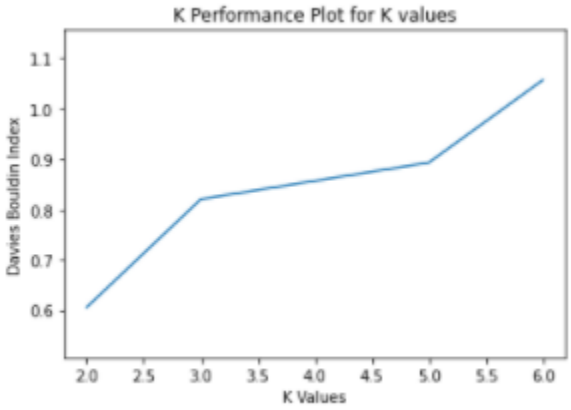

Figure 9

We can see that the DBI of K=2 is the least so it performs the best.

Here are how the different clusters looked like for varying values of K.

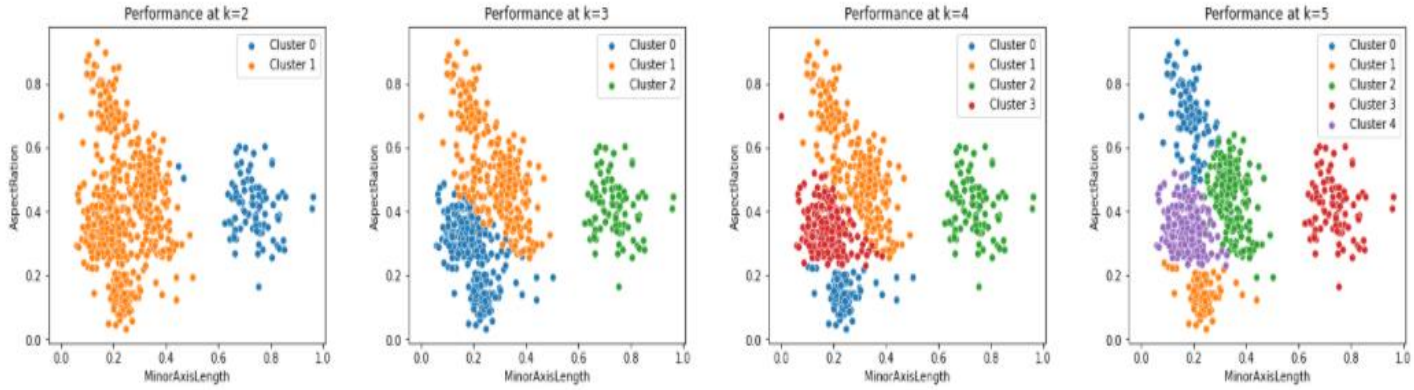

Figure 10