1 Data Description

1.1 Data Source

The data used for this assignment is the 'Dry Bean Data Set' retrieved from the UCI Machine Learning Repository.[1] The data contains features for 7 different varieties of dry beans. The features were obtained from high resolution images of the beans using a computer vision system for segmentation and feature extraction. In total, the data set is comprised of 16 features for 13611 beans, each belonging to one of 7 classes.

1.2 Variable Details

The data set is made up of 16 features, split between 12 dimension measurements and 4 shape forms. Additionally, each observation was labelled according to the type of bean it represented. All features are numeric values. There were no missing values for any of the observations. The features used in the analysis are described further here:

- **Area:** Number of pixels within bean boundary
- **Perimeter:** Length of bean border
- **Major Axis Length:** Longest distance between two points on bean
- **Minor Axis Length:** Longest distance perpendicular to major axis
- **Aspect Ratio:** Ratio between major and minor axis lengths
- **Eccentricity:** Eccentricity of the ellipse having the same moments as the region
- **Convex Area:** Number of pixels in the smallest convex area in the bean
- **Equivalent Diameter:** Diameter of a circle having bean's area
- **Extent:** Ratio of pixels in the background to the bean area
- **Solidity:** Ratio of pixels in the convex shell to the bean area
- **Roundness:** Ratio between area and perimeter
- **Compactness:** Ratio between equivalent diameter and major axis length
- **Shape Factor 1-4:** Numerical description of shape
- **Class:** Class of dry bean

1.3 Descriptive Analysis

The 7 classes of dry beans are: Barbunya, Bombay, Cali, Dermason, Horoz, Seker, and Sira. Each class is numbered alphabetically for future reference. The class distribution within the data set and the class numbers are detailed in Table 1.

Table 1: Unbalanced Bean Class Distribution

No.	Class	Count	Percentage
1	Barbunya	1322	9.71 %
2	Bombay	522	3.84 %
3	Cali	1630	11.98 %
4	Dermason	3546	26.05 %
5	Horoz	1928	14.17 %
6	Seker	2027	14.89 %
7	Sira	2636	19.37 %

Since all the features are numerical, a scaled parallel coordinates plot is shown in Fig. 1, displaying each feature for every instance of the data set. The value of the y-axis is scaled so that each feature only spans values between 0 and 1.

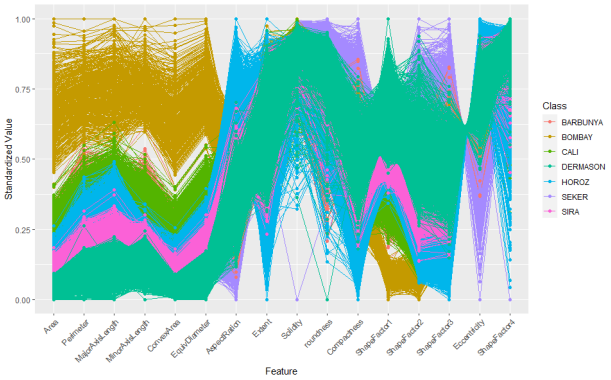

Figure 1: Parallel coordinate plot of features

From the parallel coordinates plot, we can see better separation between the classes for the first 6 features, as well as the first 2 shape factors. We can observe this more closely in the first two plots of Fig. 2. Otherwise, for the remaining features do not separate the classes as well, as seen in the last plot of Fig. 2.

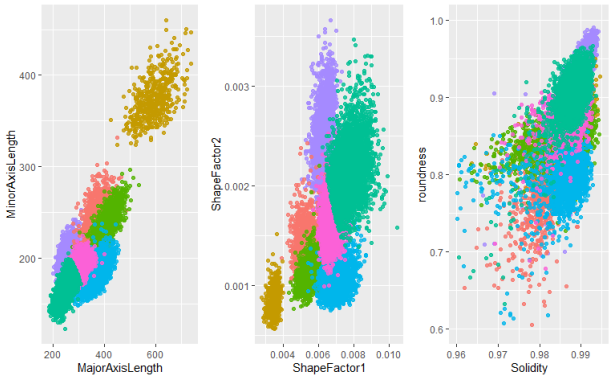

Figure 2: Selected pairs plots showing difference in group separation between features

[1] UCI Machine Learning Repository. *Dry Bean Data Set*. 2015. URL: https://archive.ics.uci.edu/ml/datasets/Dry+Bean+Dataset.

2 Problem Description

The aim of this analysis is to explore the data set using mixture discriminant analysis for the purposes of classification. This data set exemplifies features extracted from a computer vision system, which could be similarly used across many fields and industries. For this analysis, we will further investigate the effectiveness of mixture discriminant analysis for classification when compared against another classification technique: random forests. The analysis will show if the features gathered by the previous system are well suited to separate the beans by class, and how the different analysis techniques will compare.

3 Technique Description

Classification is used to predict the category to which a new observation belongs based on previously observed data, thus making it a supervised learning concept. Classification is done by splitting the data into a training, containing labelled observations and a test set, containing unlabelled observations.[2] We suppose that each of the labelled observations belong to one of G classes. Similarly, each unlabelled observation, is assumed to come from one of the G classes, which is the goal of the classification prediction. Mixture discriminant analysis (MDA) is a supervised approach in which we use the labelled observations in order to predict the class of the unlabelled observations. Mixture discriminant analysis clusters the labelled observations into each known class, and uses the resulting rule to predict the class of the unlabelled observations.[3] Using the Gaussian model-based discriminant analysis is detailed as follows. For each class in G, the Bayesian information criterion (BIC) is used to perform model-based clustering for the labelled observations of that class. From this, the class is clustered according to a $\mathcal{G}$-component model, with a $\mathcal{G}$ being the number of Gaussian components in the mixture. We classify observations based on the following equation:

$$\hat{z}_g = \hat{\pi}_g \phi(x|\hat{\mu}_g, \hat{\Sigma}_g) / \sum_{h=1}^{\mathcal{G}} \hat{\pi}_h \phi(x_j|\hat{\mu}_h, \hat{\Sigma}_h)$$

The observation is classified according to the maximum $\hat{z}_g$ value, and it is predicted to belong to the class g, for which the value is maximized. The Gaussian parsimonious clustering models (GPCMs) are made by parameterizing the covariance structure to give a variety of models.[4]. An eigenvalue decomposition of the covariance matrices results in:

$$\sum_g = \lambda_g \Gamma_g \Delta_g \Gamma_g'$$

λ_g is a constant,
Γ_g is a matrix of eigenvectors of $\sum_g$,
Δ_g is a unitary diagonal matrix with entries proportional to the eigenvalues of $\sum_g$

This decomposition is used to represent the family of GPCMs. The best model and number of Gaussians are fitted using the BIC. An example of a model in the GPCM would be 'VVV' in which the volume, shape, and orientation are all variable.

As a comparison to the MDA classification we will be using random forests (RF) , an extension of the decision tree method. A decision tree is composed of nodes and branches where decision are made to classify data into a predicted class based on the observed features. The splitting rule for a feature at each node is based on the Gini index, given by:

$$\sum_{g=1}^{G} \hat{p}_{mg}(1 - \hat{p}_{mg}),$$

where $\hat{p}_{mg}$ is the proportion of observations from class g in node m.

Alone, decision trees can have high variability between train/test splits, and can overfit to the training data. In ensemble methods such as bagging, bootstrapping is used to re-sample the entire training set with repetition, and fit a decision tree to each ensemble. Aggregating the predictions from each tree gives a classification estimate with less variance than using a single decision tree. Random forests are an ensemble method based on bagging, however they do not consider every feature at each split in the tree. The use of random forests diversifies the trees being used in the aggregation, leading to a resistance against overfitting, which other ensemble methods can be more susceptible to. The random forest method can be adjusted by tuning the number of trees and the number of predictors at each split. A value for $\mathcal{M}$ often used is: $\mathcal{M} = \sqrt{p}$, where p is the number of predictors.

Measures of classification used are the misclassification rate (MCR) and the Rand index (RI). The misclassification rate is given by the ratio of incorrectly classified observations to all the observations in the test set. The Rand index is given by:

$$RI = \frac{TP+TN}{TP+TN+FP+FN}$$

where TP is the amount of true positives, TN is true negatives, FP is false positives, and FN is false negatives.

The adjusted Rand index (ARI) corrects the Rand index by reducing agreement by chance, and is given by:

$$ARI = \frac{RI - \text{Expected}\, RI}{\text{Max}\, RI - \text{Expected}\, RI}$$

The ARI usually ranges from a value of 0 to 1, for random classification to perfect classification, respectively.

[2] Prof. McNicholas. *CSE780: Introduction to Classification*. McMaster University, School of Computational Science & Engineering. 2021.
[3] Prof. McNicholas. *CSE780: Classification Using Mixtures*. McMaster University, School of Computational Science & Engineering. 2021.
[4] Prof. McNicholas. *CSE780: Model Based Clustering I*. McMaster University, School of Computational Science & Engineering. 2021.

4 Result Description

For all analyses, a labelled/unlabelled split of 75/25 was used. The analyses were carried out 10 times, reshuffling the data to obtain unique training and testing sets. The MDA classification was carried out using the 'mclust' package in R. The results on the training set shown in Table 2, demonstrate that each class was fit by a 'VVV' model with a mixture of 5 Gaussian's, as determined by the best BIC. The testing data was then predicted for one iteration, resulting in the confusion matrix seen in Table 3.

Table 2: MDA Model Specifications

Class	n	%	Model	G
Barbunya	991	9.71	VVV	5
Bombay	392	3.84	VVV	5
Cali	1222	11.97	VVV	5
Dermason	2660	26.06	VVV	5
Horoz	1446	14.16	VVV	5
Seker	1521	14.90	VVV	5
Sira	1977	19.37	VVV	5

Table 3: Confusion Matrix for MDA

Class	1	2	3	4	5	6	7
1	310	0	21	0	2	2	4
2	0	131	0	0	0	0	0
3	28	0	355	0	3	0	3
4	0	0	0	805	2	14	69
5	2	0	11	5	446	0	12
6	2	0	0	12	0	472	32
7	7	0	0	62	8	6	576

The classification with random forests was done using the 'randomForest' package in R. A resulting confusion matrix from this method can be seen in Table 4, taken from the same iteration as the confusion matrix in Table 3. The RF parameters were tuned giving optimal results at 500 trees and 6 features at each split.

Table 4: Confusion Matrix for RF

Class	1	2	3	4	5	6	7
1	312	0	18	0	3	3	3
2	0	131	0	0	0	0	0
3	14	0	362	0	11	0	2
4	0	0	0	818	2	16	54
5	1	0	11	3	449	0	12
6	0	0	0	13	0	484	21
7	1	0	2	68	6	9	573

From the random forest analysis, we can see a plot for the feature importance measured by the mean decrease in Gini index, in Fig. 3. The most important feature according to this method is perimeter of the bean, with Fig. 4, showing the degree of class separation achieved when looking only at the perimeter.

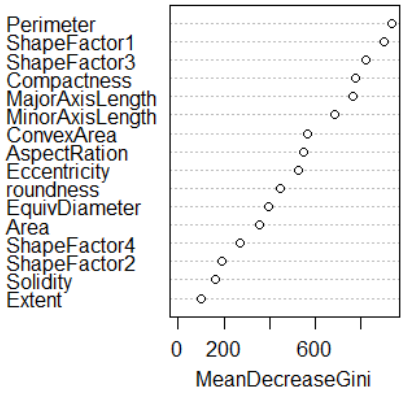

Figure 3: Feature Importance by Decrease in Gini

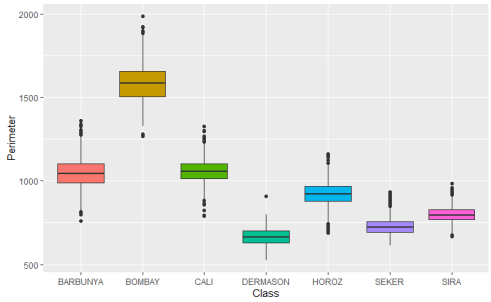

Figure 4: Perimeter Importance in Class Seperation

To compare the results, Table 5 shows the MCR and ARI for both techniques at each of the 10 iterations using a uniquely generated training/testing split. The mean and standard deviation across all 10 iterations are reported for each metric. The random forest can be observed to have a lower MCR, higher ARI and less variation for both metrics.

5 Conclusions

The classification analysis showed similar results between the MDA and RF methods, with the RF having a overall better performance. When looking at the confusion matrices between the two methods, similar patterns arise showing results that seem invariant to classification method. The Bombay class of bean is well separable from the other classes as can be seen in the confusion matrices and the parallel coordinates plot. Both models showed a tendency to misclassify unlabelled beans as Sira beans. Overall, the models can accurately predict the class to which an unlabelled bean belongs. This could be useful in industrial agricultural applications for sorting dried products or doing quality analysis to ensure that there are no outliers in a food processing plant.

Table 5: MDA vs. RF Model Results Comparison

Iteration		1	2	3	4	5	6	7	8	9	10	Mean	Std.
MDA	MCR	0.088	0.088	0.094	0.098	0.087	0.085	0.094	0.089	0.086	0.105	0.092	0.006
	ARI	0.793	0.789	0.777	0.771	0.797	0.802	0.778	0.800	0.801	0.761	0.787	0.014
RF	MCR	0.080	0.074	0.075	0.076	0.081	0.067	0.077	0.069	0.074	0.079	0.075	0.005
	ARI	0.806	0.820	0.813	0.814	0.807	0.837	0.812	0.835	0.821	0.810	0.817	0.011