**Dry Bean Dataset**

Data Analysis

horizontal line

# 

# 

# 

# Introduction

This is a multi-class classification dataset with 13611 entries,16 features and 7 classes of labels representing Seker, Barbunya, Bombay, Cali, Dermosan, Horoz and Sira varieties of beans.

## Creation of dataset

Images of 13,611 grains of 7 different registered dry beans were taken with a high-resolution camera. Bean images obtained by computer vision systems 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)

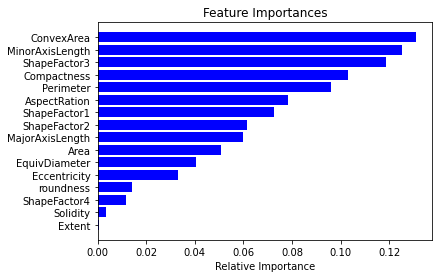
**Pandas Profile Report:**

Used the pandas-profiling to generate an overview of the dataset

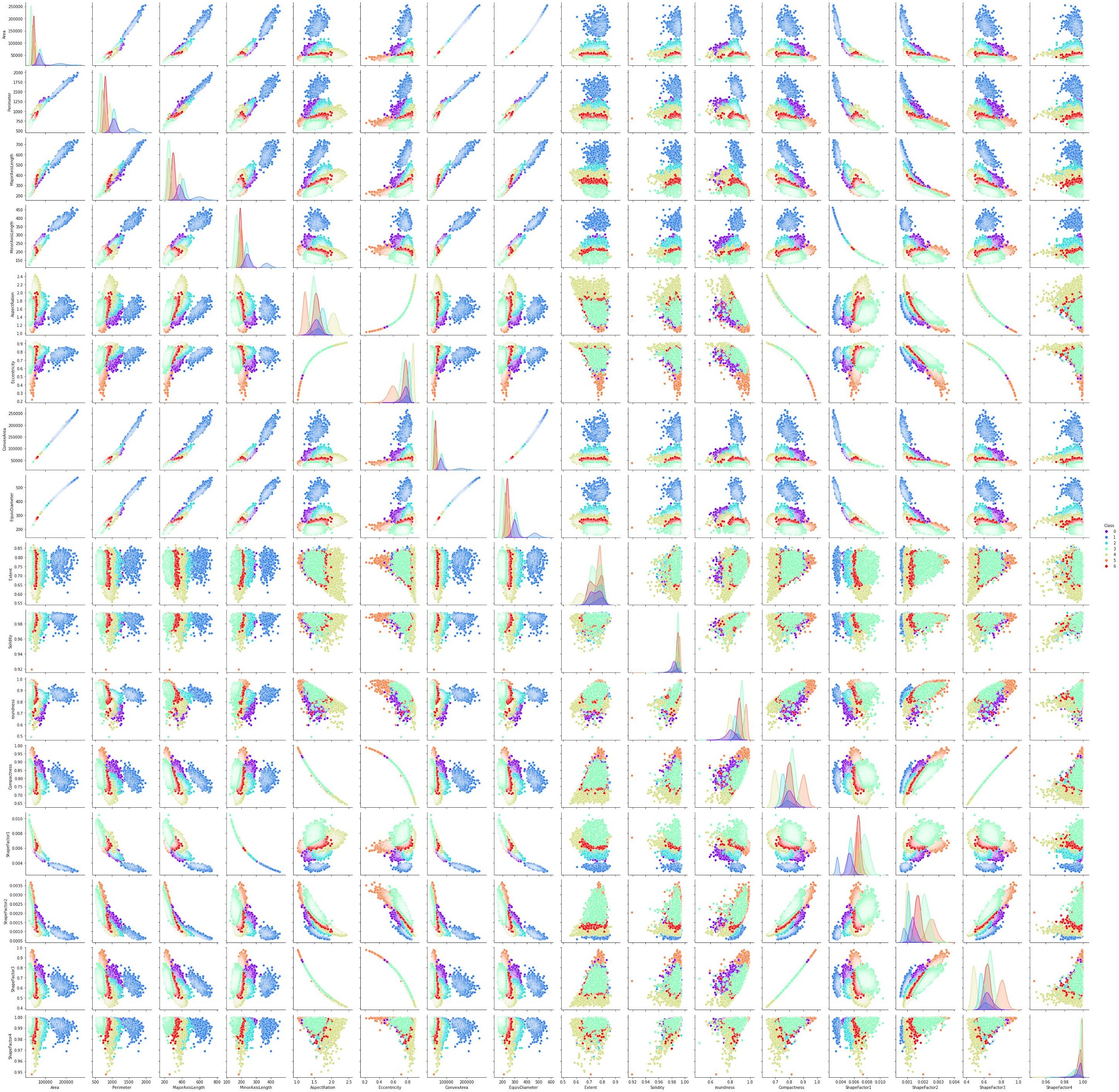
Check the Dry\_Beans.html file

**Feature Wise Importance in the model**

Trained a random forest classifier for 7-class classification and used the model’s feature importances argument for comparing and plotting the relative importance of features.



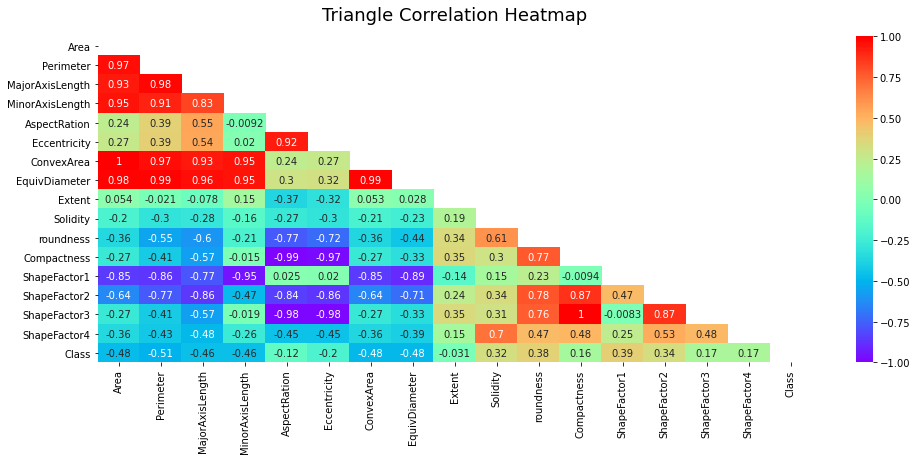
**Complete Pairplot of all features:**



This plot does show the relationships between each possible pair of features but is hard to gather significant insights from it. Hence, I tried to find the most correlated feature pairs and then plot their pairplots.

**Correlation Grid**

Heatmap correlating every pair of features hence denoting pairwise relationships.



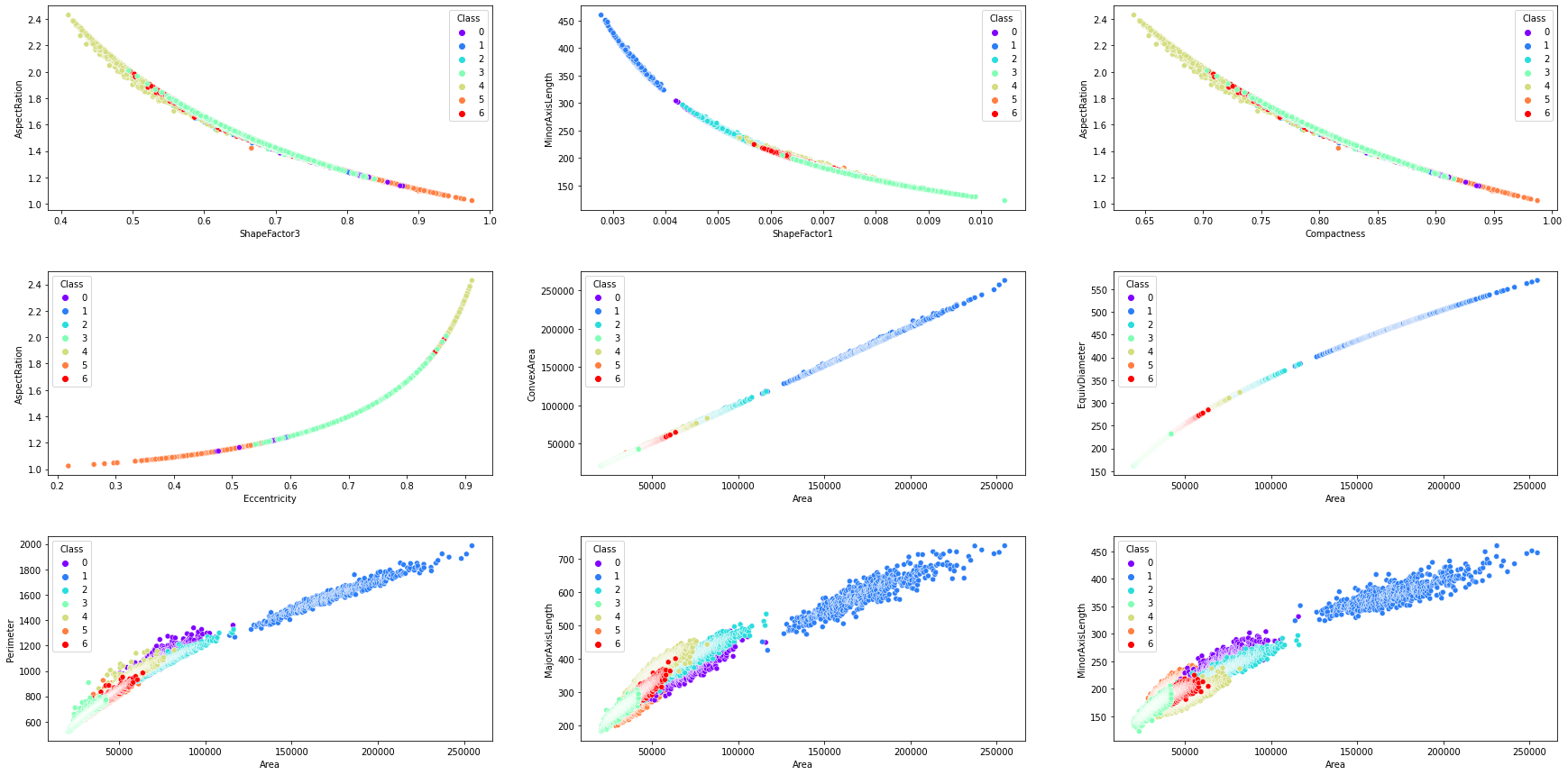
From the above heat map we can observe the following.

**Most correlated pairs:**

* ShapeFactor3 -> AspectRation
* ShapeFactor1 -> MinorAxisLength
* Compactness -> AspectRation
* Eccentricity -> AspectRation
* Area -> ConvexArea
* Area -> EquivDiameter
* Area -> Perimeter
* Area -> MajorAxisLength
* Area -> MinorAxisLength

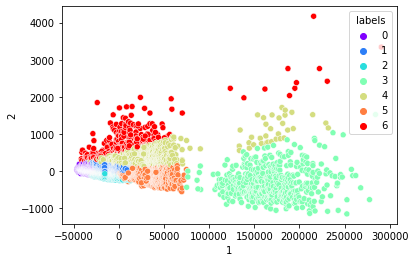
**Pairplots of most correlated pairs**

Following are the pairplots for highly correlated features where we observe the pattern of their linear relations.



**Visualizations:**

Reduced the dimensions of data by selecting the important features as found from the first plot using random forest to plot 2D, 3D as well as TSNE 3D visualizations. These indicate the spatial distribution of different class points.



Refer to the notebook for further interactive plots.

**Techniques Used :**

* Standard Scalar on input data for standardization.
* Converted the 7 output classes (which was in categorical form) using label encoding to 0-6 labels.
* Applied Grid Search on all possible combinations of parameters for different classifiers to choose the best set of parameters for the classification task.
* Reducing the dimensions to 13 using PCA improved accuracy of the model really less or negligible by dealing with correlated features in the dataset by loading them on the same eigenvector. Preferred this instead of dropping the highly correlated features to prevent data loss.

**Models Used:**

* **RandomForest Classifier -** It works by performing bagging on ensemble models of decision trees and after grid search for hyper parameter tuning, provided a test accuracy of more than 92%.
* **Support Vector Machine -** Is one of the most efficient classification machine learning algorithms where no specific data distribution is needed and the kernel can also be applied and after hyper parameter tuning provided a test accuracy of about 94%. It outperformed ensemble models and is the best fit model for this data in machine learning.
* **XGBoosting -** Is a really robust ensembling approach using boosting. Provided similar results as SVM but took too long to train and even much longer to find the best set of hyper-parameters.
* **Deep Neural Network -** Utilised early stopping, tuned hyperparameters such as number of layers, neurons in each layer and so on and made sure that the model is not overfitting on the training set. Again similar results were achieved.