Statistical Learning for Engineers

Automated Classification of Dry Bean Varieties Using Machine Learning

Group 6

Krishna Barfiwala	002771997
Niraj Gandhi	002747644
Mukund Bankar	002959140
Shaurin Karnik	002775232
Utkarsh Singh	002969579

Professor:

Ramin Mohammadi

Contents

1. Abstract

Dry bean datasets consist of various attributes and features related to different varieties of dry beans. This dataset is used for classifying multi-class categories where we aim to classify the datapoint into one of the following 7 categories after training the model. Various features such as perimeter, major axis length, minor axis length, eccentricity and other relevant attributes are used as an input to the model and the model distinguishes the dataset into 7 different categories such as Seker, Barbunya, Bombay, Cali, Dermosan, Horoz, and Sira.

2. Introduction

One of the important aspects of classifying a dry bean dataset based on this is that we can try to solve the problems related to crop monitoring, disease detection and quality assessment. By providing a thorough analysis of the attributes of the dry bean dataset we can understand the pattern in diseases of dry beans, qualities of the beans and the various environmental factors which affect the quality. Using various preprocessing techniques we can understand the correlation between the bean characteristics, drop the features which do not contain any important information.

Understanding the correlation between the dry bean and its characteristics helps in identifying the features which may provide redundant information and need to be removed from the dataset. This will streamline the dataset which will eventually increase the model's efficiency. Standardization and normalization ensure that datapoints are in a range of similar scale. This is crucial for algorithms that rely on distance metrics such as support vector machines (SVM), multi-class logistic regression and Neural Network as it avoids the domination of certain features on the training model.

Insights gained after training the model and looking at its prediction can be used to iterate and improve the model exponentially. As more data is fed to the system it can adapt to the changing conditions and improve its accuracy.

The model, once developed, can become a valuable tool for research scientists and agricultural practitioners aiding them in data-driven decisions for crop management, disease prevention and quality assurance.

3. Data Description

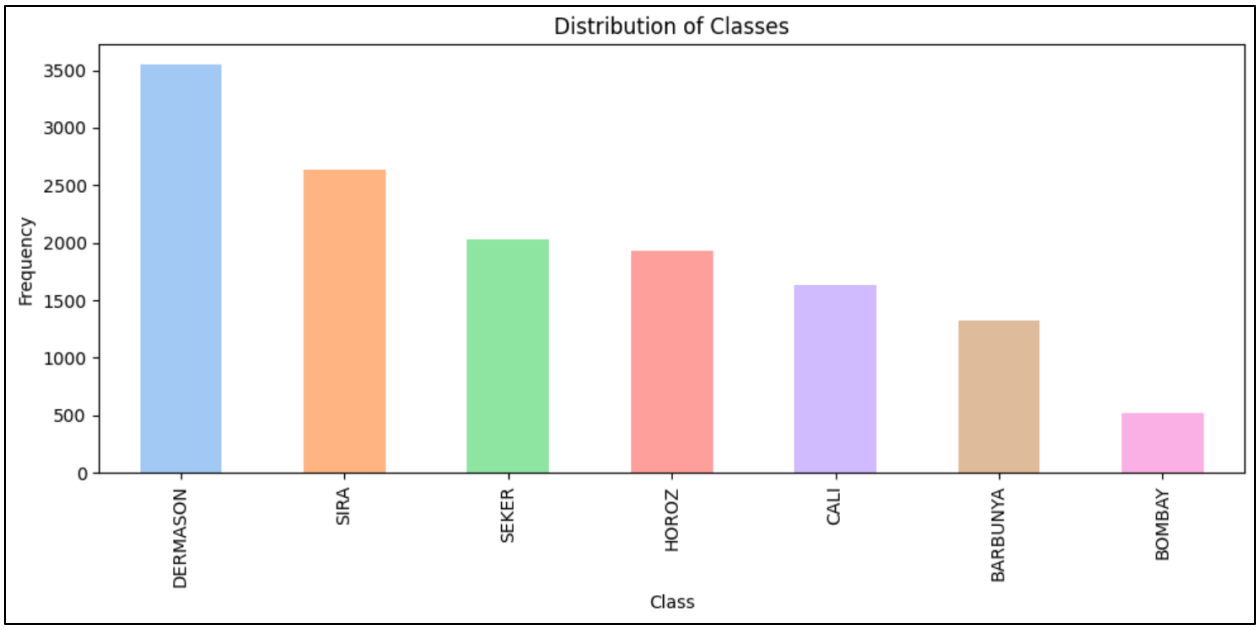
For this classification "Dry Bean Dataset" is utilized, which is available on the UCI data repository. This dataset consists of numerical values derived from a collection of high-resolution image data, encompassing a total of 13,611 grains from seven distinct registered dry bean varieties.

Data Set Link: https://archive.ics.uci.edu/dataset/602/dry+bean+dataset

Feature Name	**Description**	**Feature Type**
Area	The area of a bean zone and the number of pixels within its boundaries	Integer
Perimeter	Bean circumference	Continuous
MajorAxisLength	Distance between the longest line that can be drawn from a bean	Continuous
MinorAxisLength	Longest line that can be drawn from the bean which is perpendicular to the main axis	Continuous
AspectRatio	Relationship between MajorAxisLength and MinorAxisLength	Continuous
Eccentricity	Eccentricity of the ellipse having the same moments as the region	Continuous
ConvexArea	No. of pixels in the smallest convex polygon that contains the area of seed	Integer
EquivDiameter	Diameter of a circle having same area as been seed area	Continuous
Extent	Ratio of the pixels in the bounding box to the bean area	Continuous
Solidity	The ratio of the pixels in the convex shell to those found in beans.	Continuous
Roundness	Calculated with the following formula: (4piA)/(P^2)	Continuous

Compactness	Measures the roundness of an object	Continuous
ShapeFactor1	One of the shape factors, likely a dimensionless quantity derived from the geometric properties of the bean.	Continuous
ShapeFactor2	Another shape factor providing information about additional geometric information about the bean	Continuous
ShapeFactor3	Another shape factor contributing to the characterization of the bean's shape	Continuous
ShapeFactor4	Fourth shape factor giving details about the bean's shape	Continuous
Class	Target variable indicating the type or variety of dry bean. We have 7 categories of class in this dataset. They are Barbunya, Bombay, Cali, Dermason, Horoz, Sekar and Sira	Categorical

4. Exploratory Data Analysis



Above graph depicts the distribution of classes and their frequency count in the dataset. From the above graph we can observe that Dermason class has the highest number of count whereas Bombay has the lowest number of count.

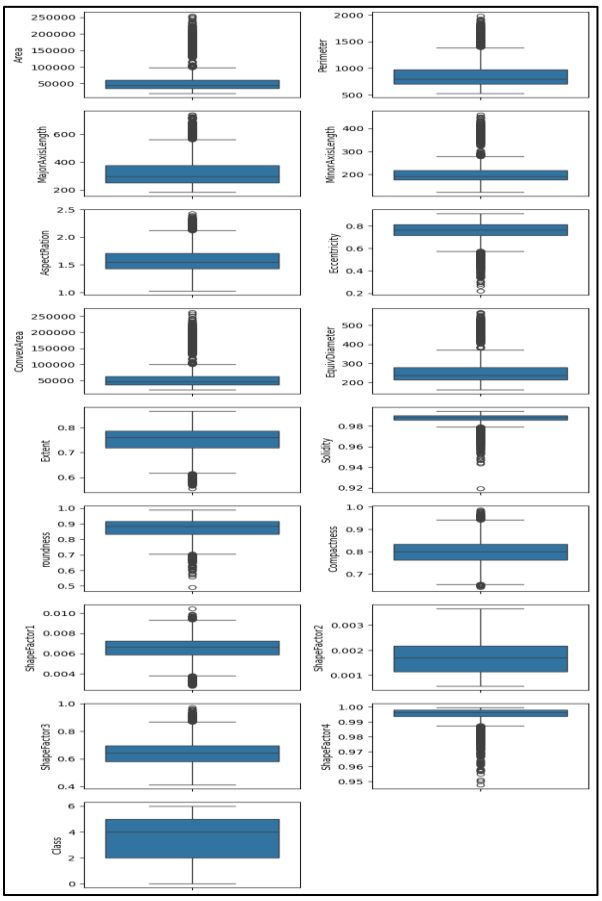
a. Feature Engineering:

In our dry bean dataset, 7 target categories namely 'SEKER', 'BARBUNYA', 'BOMBAY', 'CALI', 'DERMASON', 'HOROZ', 'SIRA'. After mapping and target names are replaced with 0 to 6 values in our original dataset as most of our models expect input in the form of numeric values.
Also after null values count check and we have found that there are no null values present in the dataset.

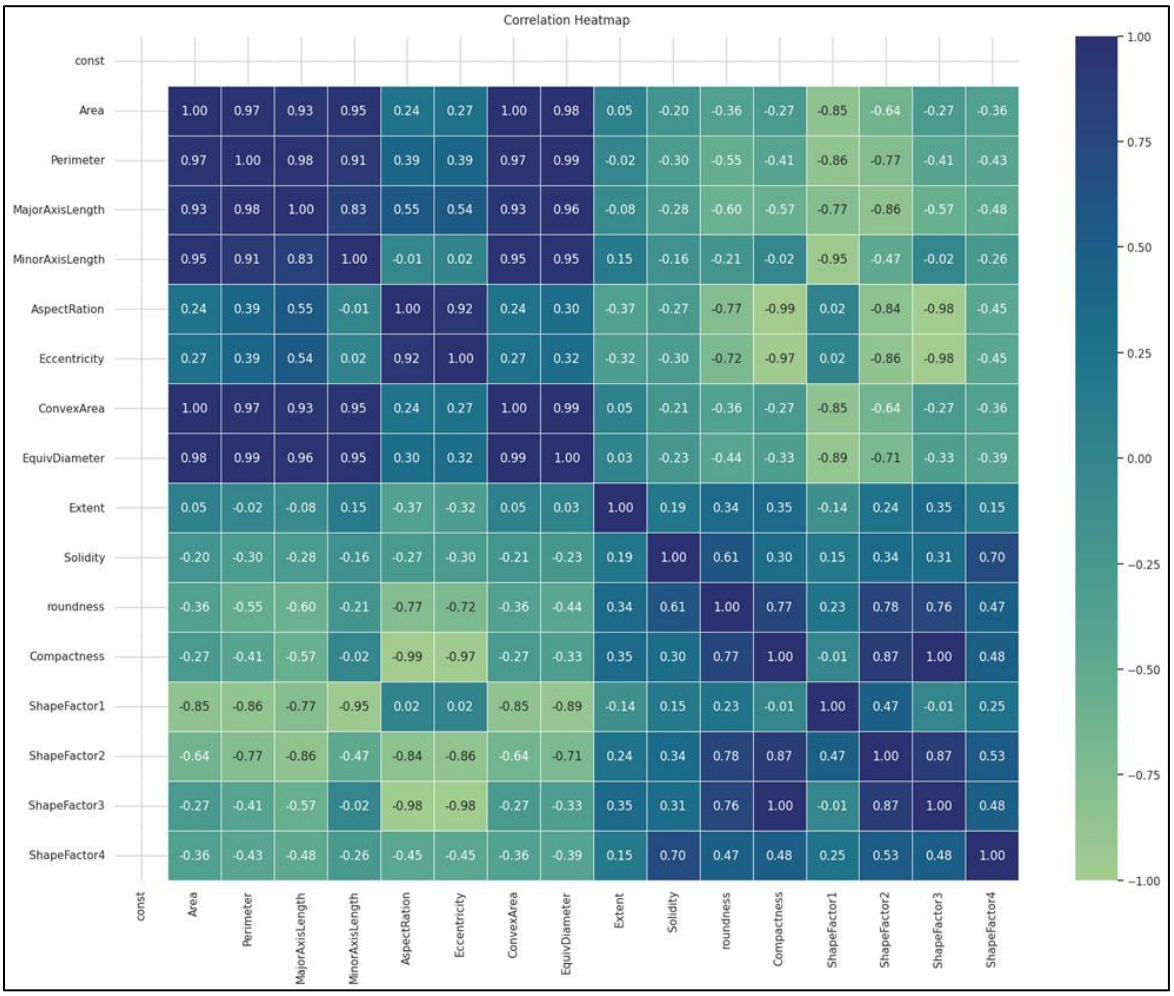
```
DBD.isnull().sum()

Area               0
Perimeter          0
MajorAxisLength    0
MinorAxisLength    0
AspectRation       0
Eccentricity       0
ConvexArea         0
EquivDiameter      0
Extent             0
Solidity           0
roundness          0
Compactness        0
ShapeFactor1       0
ShapeFactor2       0
ShapeFactor3       0
ShapeFactor4       0
Class              0
dtype: int64
```

b. Box-plot

The below graph gives an in-depth information about the data distribution of features and its outliers.



c. Correlation heatmap:



From the above correlation heatmap and after going through the documentation of the dataset and plotting the correlation pairs in a tabulated format, we came to a conclusion to drop the Area and ShapeFactor4 features from the dataset. Since the area was having a correlation mapping greater than ninety-five percent with perimeter features, we dropped it. Also from the tabulated correlation pairs, ShapeFactor4 was evidently seen to have high correlation with the other attributes and the dataset already consisted of four different shape factors, hence we decided to drop ShapeFactor4.

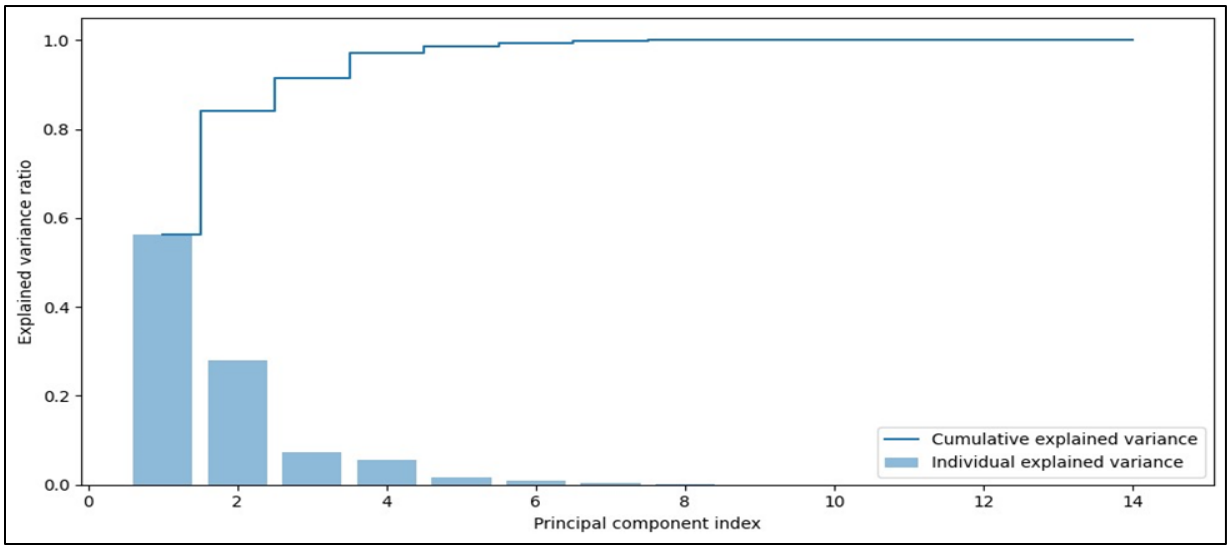
d. Standardization:

It is also called feature scaling or normalizing the data. It is a preprocessing step in which the features of a dataset are transformed in accordance with the zero mean and unit variance. This step ensures that all features have the same scale as it plays an important role in performance and faster convergence of the model resulting in making the model more robust and effective.

e. PCA:

Principal component analysis is a dimensional reduction technique which is used to transform the dataset into a new coordinate system where features are uncorrelated and are present in the order of variance they contain. The main objective of this technique is to maintain as much original information as possible while reducing the number of dimensions.

Covariance matrix is calculated to capture critical information about the variability and relationships between variables in a dataset. It forms the basis for identifying the principal components which is a key factor in capturing the most significant variance in the dataset. Using the covariance matrix, eigenvalues and eigenvectors are computed and are used to determine magnitude and direction of principal components respectively. We have chosen n=7 as PCA



components based on cumulative variance which represents the proportion of total variance explained by the first 7 principal components

The chart shows that the first few principal components explain a significant amount of variance in the data, with the first component explaining over 50% of the variance. The subsequent principal components explained have less variance subsequently, but the cumulative explained variance continues to increase as more principal components are considered. Hence, we can plan based on the above chart that the first few principal components can be used to retain the most important information and hence can be used to reduce the dimensionality of the data.

f. Splitting the data:

We are splitting the dataset into training and test as we plan to train the model on 67% of total dataset and test the model on the remaining 33% of dataset which is our test dataset will be used for evaluation of its performance on another, unseen data. This helps us in understanding how well the model generalizes to new, unseen data.

5. Methods:

a. Neural Networks

Neural Networks are advanced Machine Learning algorithms that mimic the working of a Human Brain in order to solve multiple complex problems like Human Beings are able to do in a real time scenario. They share a common building block called Neurons that help them to receive information.

Neural Networks are composed of interconnected layers – each layer comprised of processing units called neurons. The minimum layers that a Neural Network can have are either 2 (input layer and output layer) OR in some cases it Is even considered 3 (Hidden layer comes into play). The number of Layers is mostly dependent on the complexity of a task. The neurons are simple processing units that receive information/data input and apply an activation function and thus result in an output.

The Neural Network Equation:

$$Z = Bias + W_1X_1 + W_2X_2 + \ldots + W_nX_n$$

where,

- Z is the symbol for denotation of the Neural Network.
- W are the weights or the beta coefficients
- X are the inputs
- Bias or intercept = W0

b. Logistic regression

Logistic Regression is a type of statistical method used for binary and multi-class classification where we aim to categorize the output into two or more classes. The logistic model falls under classification task. It uses sigmoid, logit function and log loss cost function to transform a linear combination of input features into a probability between 0 and 1. Through iterative optimization techniques like gradient descent the model is trained to find the optimal weights that can minimize the prediction error. The assumptions of logistic regression includes the samples are independent samples, little or no multicollinearity, large sample size, dependent variables should be

$\sigma(z)$=1/(1+e^-z)

where z is linear combination of features and weights

$z= b_0+ b_1x_1+b_2x_2 \ldots.+b_nx_n$

where b_0, b_1, b_2 etc are weights and x_1, x_2, x_3 etc are input features

Gradient Descent:

$$J(\theta) = J(\theta) - \alpha \frac{\partial}{\partial \theta_J} J(\theta)$$

Where α = learning rate

J(θ) = cost function

The derivative of cost function can be modified as:

$$J(\theta) = \frac{\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^i\right)x^i}{m}$$

c. Soft Margin SVM

The Soft Margin SVM represents a refined version of the classic Support Vector Machine, specifically designed for classification tasks. This model variant permits a certain degree of error during training, thereby increasing its resilience to outliers and its effectiveness in handling datasets with classes that are not clearly linearly separated.

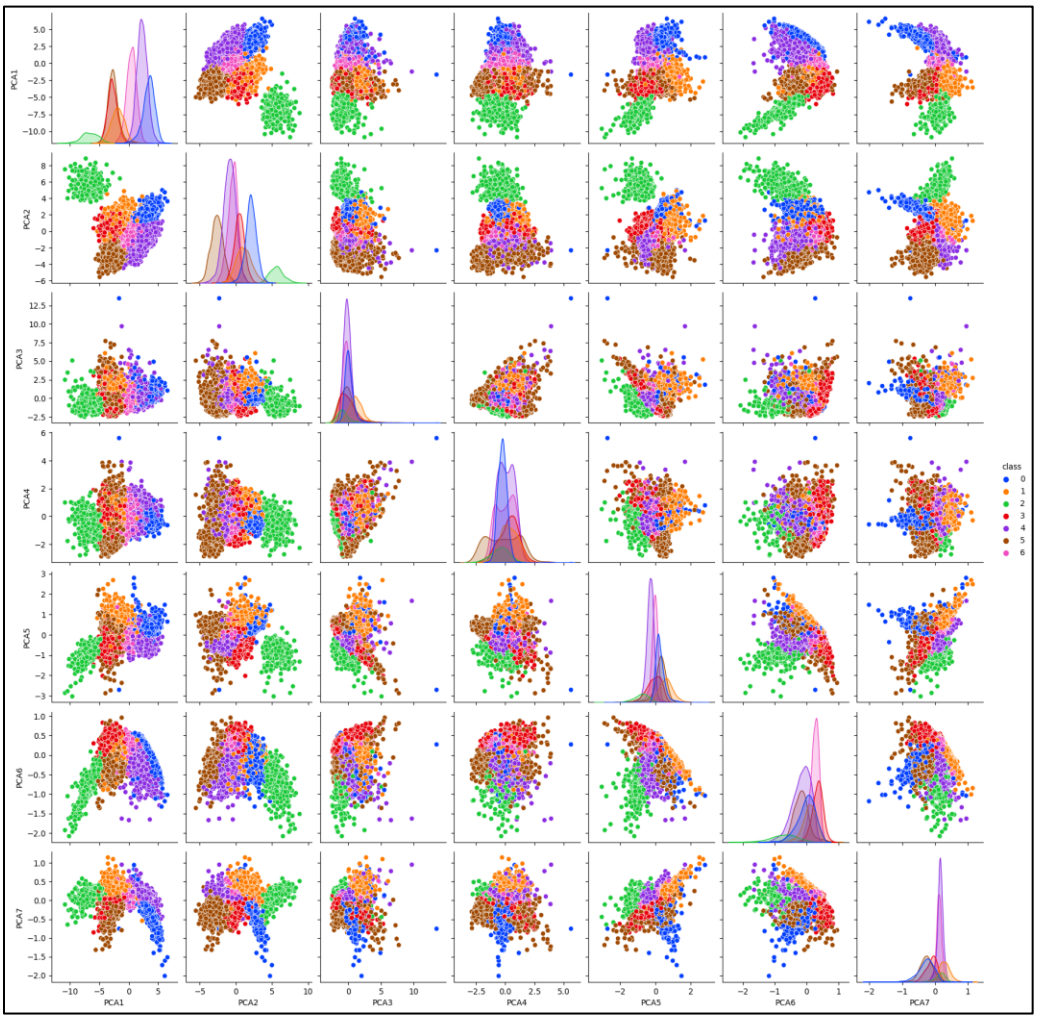
$$\begin{aligned} min \quad & \frac{1}{2}w^T w + C\sum_{i=1}^{m}\xi_i \\ s.t. \quad & y_i(x_i^T w + b) \geq 1 - \xi_i \qquad \xi_i \geq 0 \end{aligned}$$

Key Advantages of Soft Margin SVM Include:

1. Enhanced Tolerance to Data Anomalies: By incorporating slack variables, the Soft Margin SVM tolerates data points that fall on the incorrect side of the margin, thus providing a buffer against outlier influence, a significant advancement over the Hard Margin SVM.
2. Adaptability to Complex Datasets: Given the complexity of real-world datasets, which rarely present clear-cut separations, the Soft Margin SVM is adept at determining an ideal hyperplane that effectively reduces misclassification.
3. Regulated Margin-Misclassification Equilibrium: The model achieves a delicate balance between enlarging the margin and curbing classification errors, a process modulated by the regularization parameter.

Initial data exploration revealed substantial class overlap within the dataset, suggesting a lack of inherent linear separability. This observation was further confirmed by examining pairwise

relationships between features. However, specific principal components, particularly PCA1 and PCA2, exhibited a greater capacity to differentiate classes. This finding, likely attributable to their higher variance capture, suggests potential benefits for classification accuracy.



The MultiClass SVM leverages a one-vs-rest strategy, where individual binary SVM classifiers are trained for each class. The final prediction for each instance is determined by the classifier with the highest decision function score.

Components of Multiclass SVM:

Hingeloss:

The hinge loss is a crucial part of the SVM's cost function and is defined as:

$hingeloss$=[0,1−$yf(x)$]

Regularization Component:

The regularization term, often represented by the parameter C, plays a critical role in balancing margin maximization and minimizing classification errors. In effect, C determines the trade-off between achieving the widest possible margin and accommodating for potential misclassifications.

Updating Weights and Biases:

If hingeloss condition is true then :

$$w \leftarrow w - lr \times 2 \times C \times w$$

Else:

$$w \leftarrow w - lr \times (2 \times C \times w - yi \times xi)$$

$$b \leftarrow b - lr \times (-yi)$$

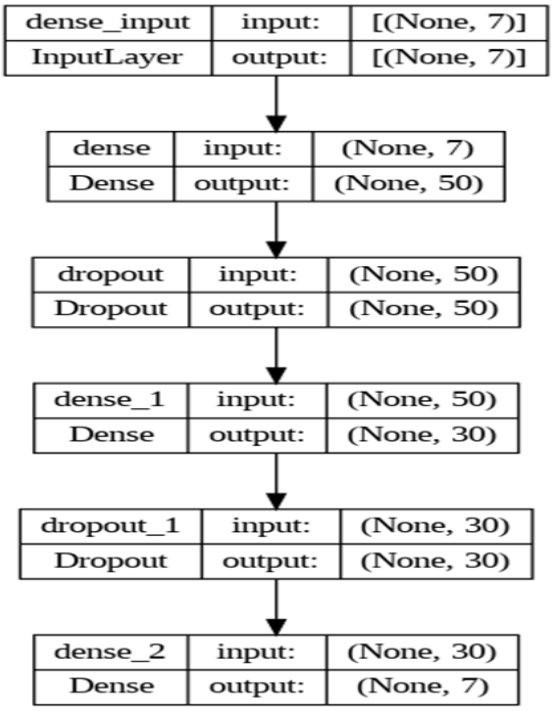
Lr is learning Rate

6. Result:

	F1- Score	Accuracy	Precision	Recall
Logistic Regression	91.37%	89.95%	91.86%	90.96%
Neural Nets	91.83%	91.81%	91.13%	91.81%
Soft Margin SVM	47.40%	52.11%	45.62%	52.11%

a. Neural Networks

Our Neural Network in total consists of 4 standard Layers and 2 Dropout Layers (Dropout layers in general are not considered Standard layers as they do not perform any transformations on the data). The 2 Hidden layers contain 50 and 30 Neurons respectively and use the Rectified Linear Unit (ReLU) activation function. The dropout layers after each hidden layer randomly drop 20% of the Neurons in order to prevent overfitting of the data. Finally, the output layer has neurons equal to the number of unique classes in our target variable i.e. 7 and uses SoftMax activation function for multi-class classification.



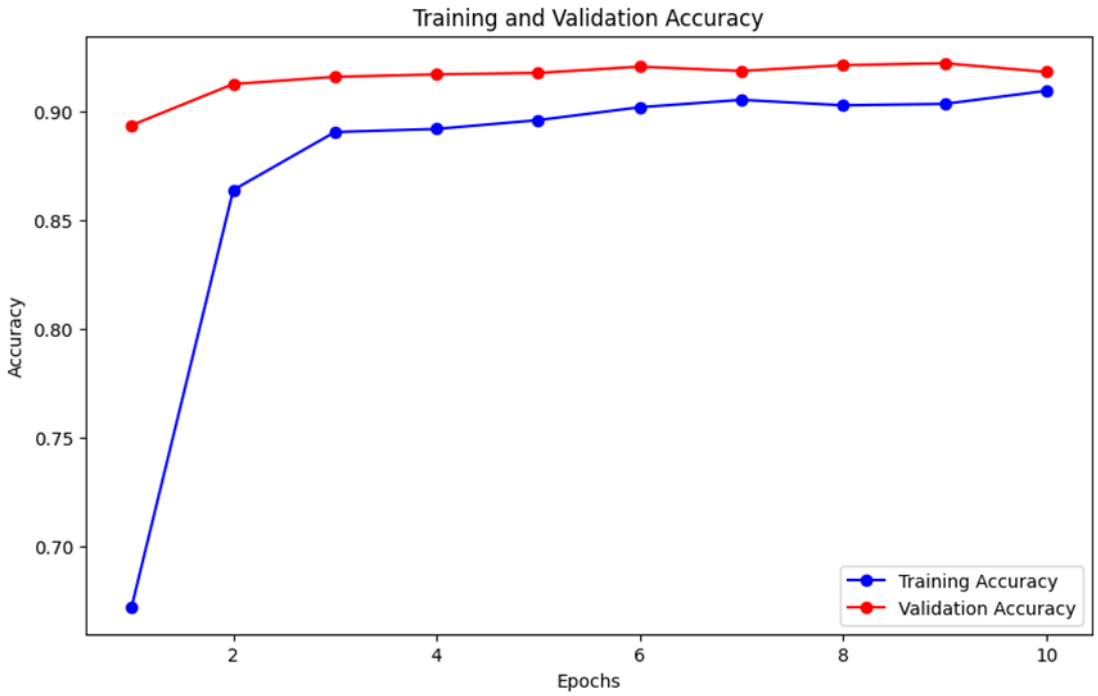

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 50)                400

 dropout (Dropout)           (None, 50)                0

 dense_1 (Dense)             (None, 30)                1530

 dropout_1 (Dropout)         (None, 30)                0

 dense_2 (Dense)             (None, 7)                 217

=================================================================
Total params: 2147 (8.39 KB)
Trainable params: 2147 (8.39 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Various Performance Metrics for Each Class

Class	Precision	Recall	F1-Score	Support
0	0.96	0.92	0.94	664
1	0.92	0.88	0.9	450
2	1	1	1	178
3	0.89	0.95	0.92	528
4	0.91	0.93	0.92	1155
5	0.98	0.93	0.95	638
6	0.86	0.89	0.87	879

Total Metrics for the Neural Network

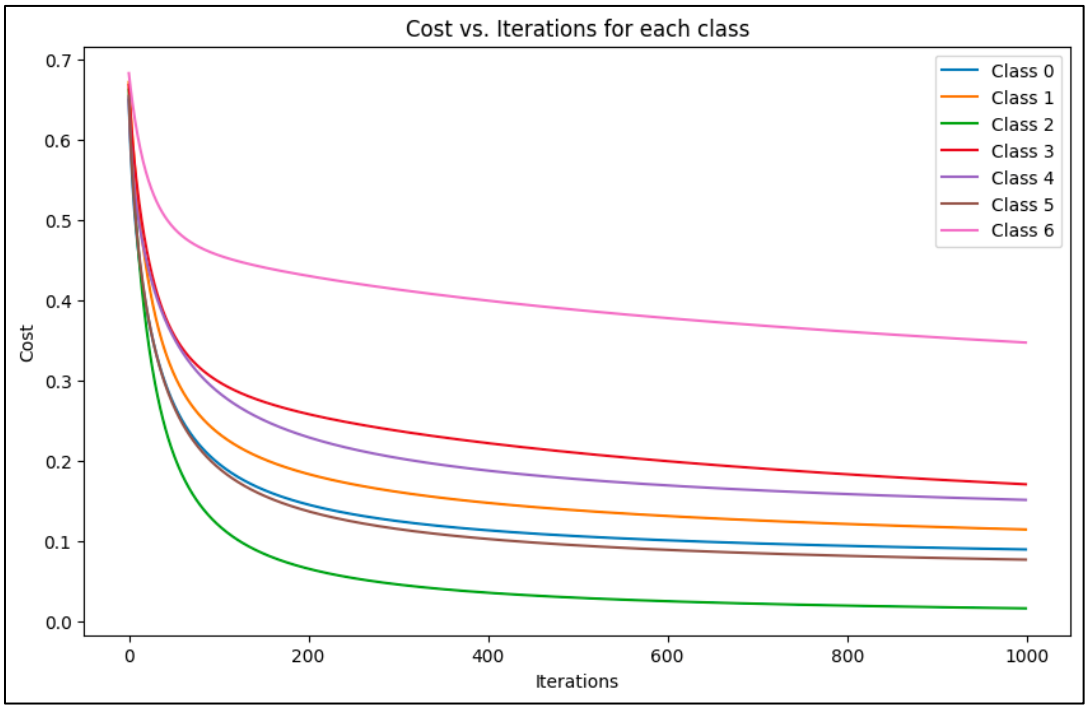
Metric	Value
Test Loss	0.2182
Test Accuracy	0.9181
Precision (Weighted)	0.9193
Recall (Weighted)	0.9181
F1 Score (Weighted)	0.9183



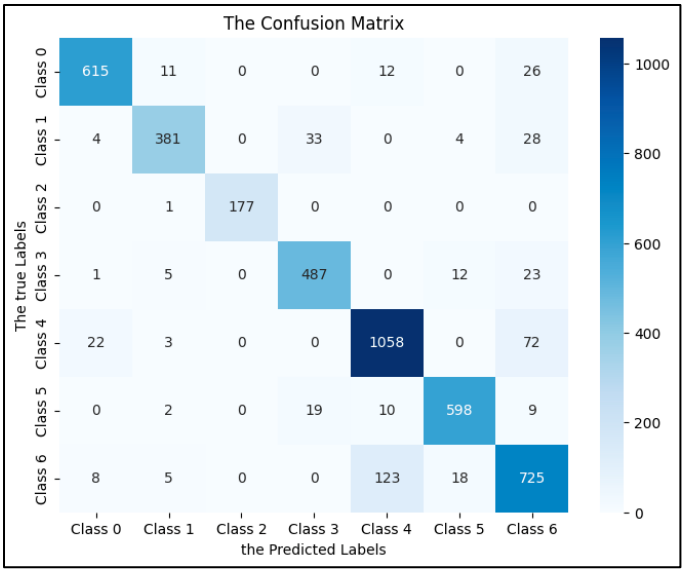
In the above graph it is clearly visible that initially there is rapid improvement in the training and validation accuracy after the 1st epoch. After 3rd epoch the accuracies begin to plateau which is an indication towards the convergence of the model to an optimal state. The small gap between training and validation accuracy implies that the model is not overfitting significantly to training data.

b. Logistic Regression

We get the best training and testing accuracy of 89.72% and 89.96% respectively. The precision, recall, F1 score for each class is high, greater than 84%. The overall model accuracy is 89.95%, recall being 90.96%, F1 score as 91.37%, precision as 91.86%. The model uses average = 'Macro' in finding the recall, precision, F1 score since the dataset is not imbalanced.



A good model has a smooth decreasing curve in the cost vs iterations graph for each class. This shows the model is improving to predict the dry bean Classes.



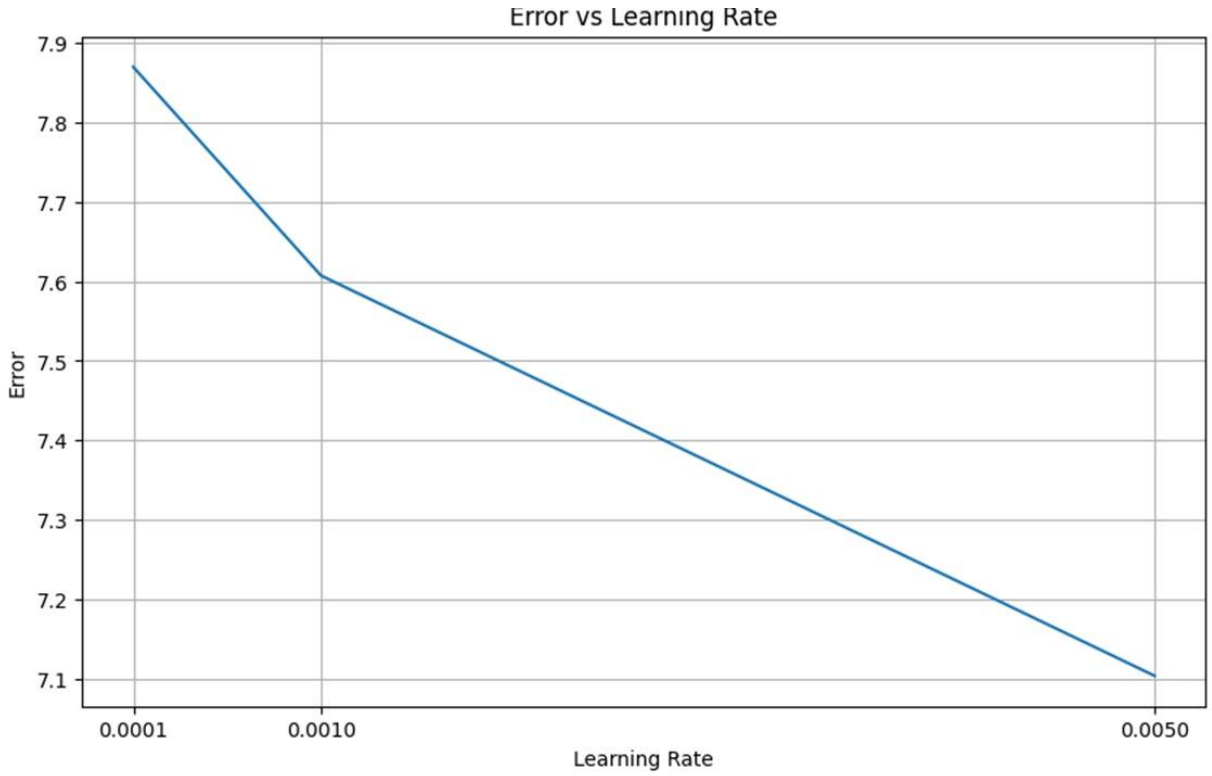
The confusion matrix shows the correct classification of the classes and the incorrect/ misclassified classifications by our model. Dark blue indicates higher number of correct classifications of the classes. Vertical line indicates the true labels while the horizontal line indicates predicted labels.

c. Soft Margin SVM

The MultiClassSVM's performance metrics paint a mixed picture. While the model achieves an accuracy of 52%, suggesting it can correctly predict outcomes slightly better than chance, the precision of 46% raises concerns about false positives. This high rate indicates that the model frequently misclassifies non-relevant instances as relevant.

Despite a moderate F1 score of 47%, signifying a balance between precision and recall, the overall performance suggests room for improvement. This is likely due to the significant class overlap observed in the data, which presents a challenge for the model to achieve clear distinctions.

To enhance the MultiClassSVM's discriminative capabilities, further exploration is warranted. This may involve parameter tuning, experimenting with alternative kernel functions, or exploring different classification algorithms that are better equipped to handle complex data landscapes with overlapping classes.



SVM has a hyper parameter α, for getting an optimal model we implement Bias Variance Tradeoff. While implementing different learning rates we found the ideal value to be 0.005.

7. Conclusion / Discussion:

The examination of the data indicates that Neural Networks stands out as the most effective and reliable model for these classification tasks. It excels in important metrics such as F1-Score, and Recall, showcasing its outstanding capability in balancing accuracy and thoroughness in predictions. Logistic Regression also displays commendable performance, although they slightly trail behind Neural Networks. Their intricate architecture might be beneficial for identifying nuanced patterns in complex situations. In contrast, the performance of Soft Margin SVM is noticeably weaker, suggesting it may not be as suitable for this particular application without extensive adjustments. Therefore, in upcoming projects where precise and balanced classification is essential, our preference would be towards using Neural Networks, with Logistic Regression as a potential option for more complex scenarios.