FINAL REPORT

DRY BEAN CLASSIFICATION MODEL

BUAN 302

Prof: Minati Rath

Varsha Gunturu

220076

TABLE OF CONTENTS

Problem Statement.....2

Methodology.....2

Results.....7

Skewness Analysis.....7

Correlation Analysis:.....8

Outlier Removal:.....9

Transformations:.....10

SMOTE and Standardization:.....11

Reduction using PCA:.....12

SVM Model:.....12

Decision Tree Model:.....14

K-Means Model:.....18

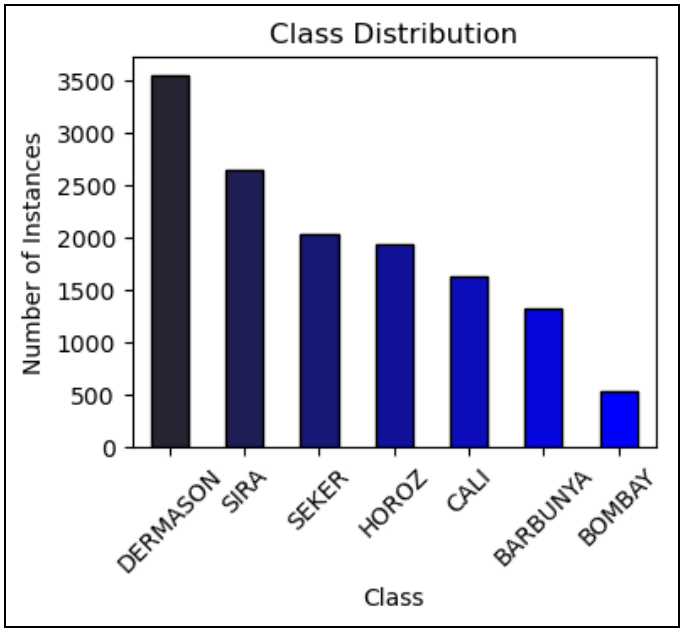
Conclusion.....19

Problem Statement

The objective of this analysis is to classify dry bean samples into one of seven distinct classes based on their geometric properties. The dataset comprises 13,611 entries with 16 features and one target variable (class). The primary challenges include class imbalance, feature redundancy, skewed distributions, and the need for optimal feature selection and modeling approaches to achieve high classification accuracy.

Methodology

There were 13,611 entries with 17 columns, 16 columns containing various features and 1 column containing the class label of the beans.The 16 columns include Area, Perimeter, MajorAxisLength, MinorAxisLength, AspectRation, Eccentricity, ConvexArea, EquivDiameter, Extent, Solidity, roundness, Compactness, ShapeFactor1, ShapeFactor2, ShapeFactor3 and ShapeFactor4. The Class Labels include Dermason, Sira, Seker, Horoz, Cali, Barbunya, and Bombay. The data did not have any null values.



The Dermason and Sira classes are predominant in the dataset forming the majority whereas the Bombay class is significantly underrepresented. This may lead to model bias, therefore imbalance necessitates techniques like oversampling are needed to ensure effective learning of models across all classes. Skewness Analysis and Correlation Analysis will be performed to see the skewness of the features and identify highly correlated features.

A bar plot and pairwise plot will be generated for better understanding of the features and their relationship. Based on the skewness analysis and correlation analysis, transformation will be applied and methods such as SMOTE , PCA will be applied and Feature engineering will be done.

After the data is processed, the data will be run through Support Vector Machines, Decision Tree and K-means model. The parameters for these models will be tuned accordingly. The results for these models will be done using plots, graphs and matrices along with the evaluation metrics.

- Class Encoding: Target variable encoded using LabelEncoder().
- Two features were generated

    - Shape_Complexity: Created by multiplying ShapeFactor1 and Compactness. Represents the complexity of the bean's shape.
    - Shape_Regularity: Created by multiplying Solidity and roundness. Captures the regularity of the bean’s shape.
- Class Balancing: SMOTE (Synthetic Minority Over-sampling Technique) applied to address class imbalance.
- Outlier Removal: IQR method used to exclude extreme values.
- Skewness Handling:
    - Log transformations applied to features with high positive skewness.
    - Yeo-Johnson transformations applied to features with high negative skewness.
- Feature Scaling: StandardScaler used to normalize features for consistent weighting.
- Dimensionality Reduction: Principal Component Analysis (PCA) applied, retaining 95% variance.
- Random Forest applied to check for important features and less important features.
    - The top 5 most important features, based on Random Forest, are:
        - ShapeFactor3: 0.106
        - AspectRatio: 0.099
        - Compactness: 0.091
        - Eccentricity: 0.084
        - Perimeter: 0.068

    The least important features are Solidity, Extent, and ShapeFactor2.
- K-fold Cross Validation will be used to train the model accordingly, so that the models are generalised to the data accordingly, which also avoids overfitting.

- The models were then tested on
- For SVM the model will be runned on ‘rbf’ and then on ‘poly’, out of both the one with higher accuracy will be chosen.
    - The SVM model with a polynomial kernel was evaluated using 5-Fold Cross-Validation and tested on a hold-out test set
- For Decision Trees, simple, post-pruned and pre-pruned trees will be generated. Out of the three the one with the highest accuracy will be chosen for evaluation.

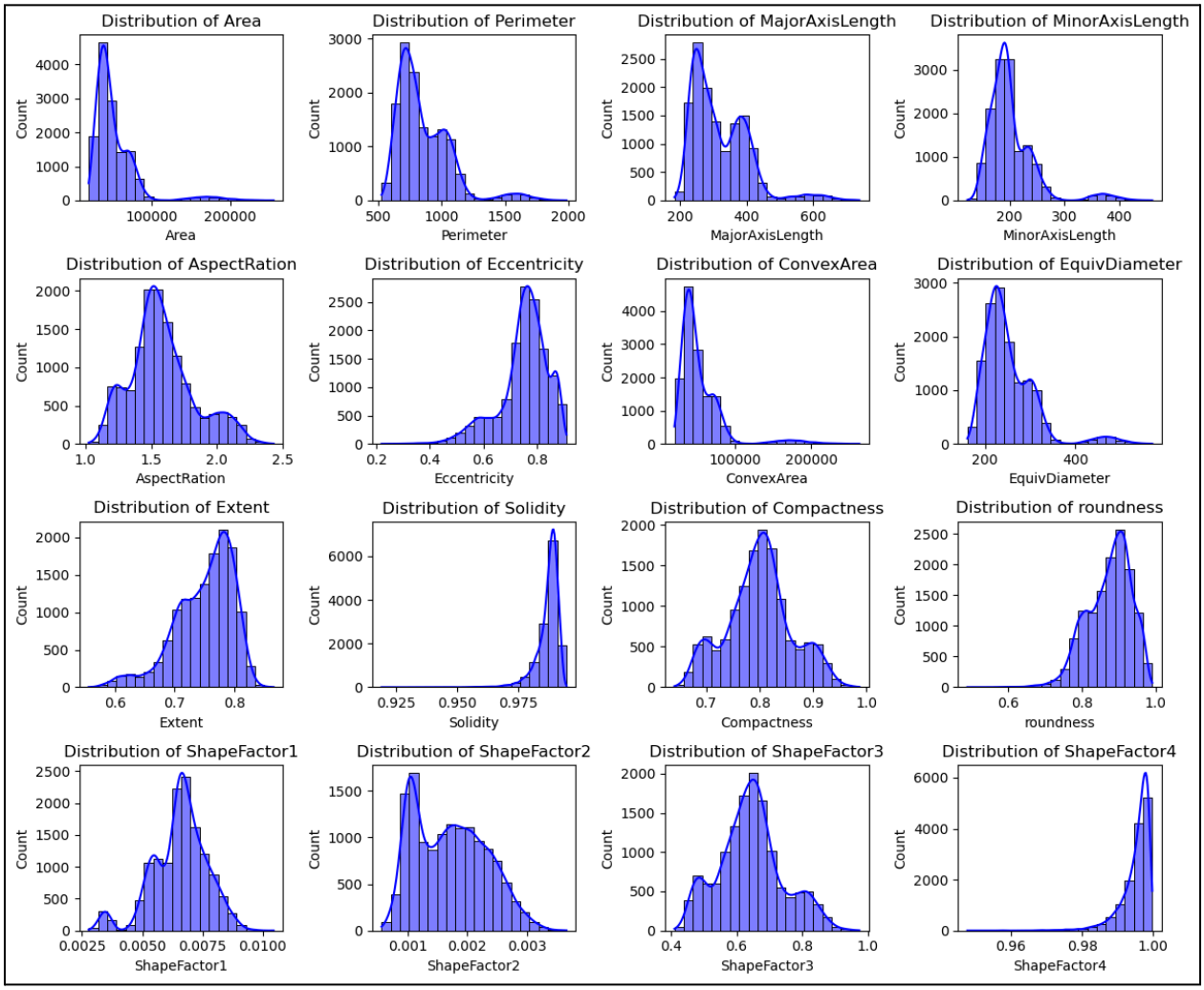
  trained and evaluated a Decision Tree Classifier on the dataset using three variations:
    - Simple Decision Tree: A basic decision tree with default hyperparameters.
    - Pre-pruned Decision Tree: A decision tree with a maximum depth of 3 to prevent overfitting.
    - Post-pruned Decision Tree: A decision tree with cost complexity pruning to optimize the model and reduce overfitting.
    - The following evaluation metrics were calculated for each model:
        - Accuracy: The proportion of correct predictions.
        - Precision: The proportion of true positive predictions among all positive predictions.
        - Recall: The proportion of true positive predictions among all actual positives.
        - F1 Score: The harmonic mean of precision and recall.
        - Sensitivity: The ability of the model to identify positive class instances (calculated from the confusion matrix).

        - Specificity: The ability of the model to identify negative class instances (calculated from the confusion matrix).
        - ROC AUC: The area under the Receiver Operating Characteristic curve, which indicates how well the model distinguishes between classes

- For k-means, first the elbow method will be used to check the optimal number of clusters required after which the model will be trained on the optimal number.
    - The following metrics are used to assess the model:
        - Silhouette Score: Measures how similar each point is to its own cluster compared to other clusters (higher values are better).
        - Davies-Bouldin Index: Measures cluster separation (lower values are better).
        - Adjusted Rand Index (ARI): Measures how well the predicted clusters match the actual clusters (higher values indicate better clustering).

Results

Skewness Analysis

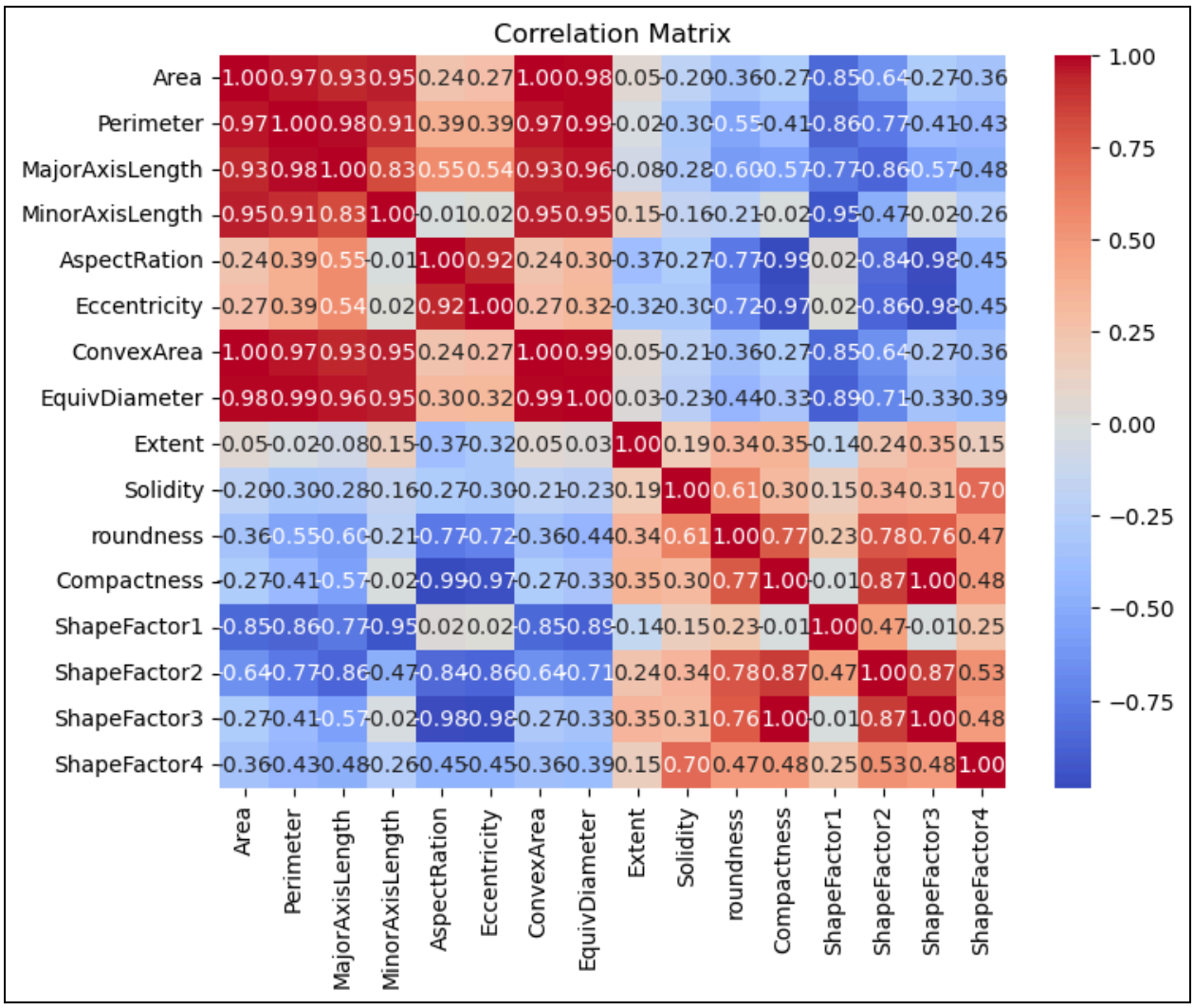


- Highly Skewed (|Skewness| > 1):
    - Positive Skew: Area, Perimeter, MajorAxisLength, MinorAxisLength, ConvexArea, EquivDiameter
    - Negative Skew: Eccentricity, Solidity, ShapeFactor4
- Moderately Skewed (0.5 < |Skewness| < 1):
    - Positive Skew: AspectRation

- Negative Skew: Extent, Roundness

- Low Skewness (|Skewness| < 0.5):
  - Compactness, ShapeFactor1, ShapeFactor2, ShapeFactor3

Correlation Analysis:

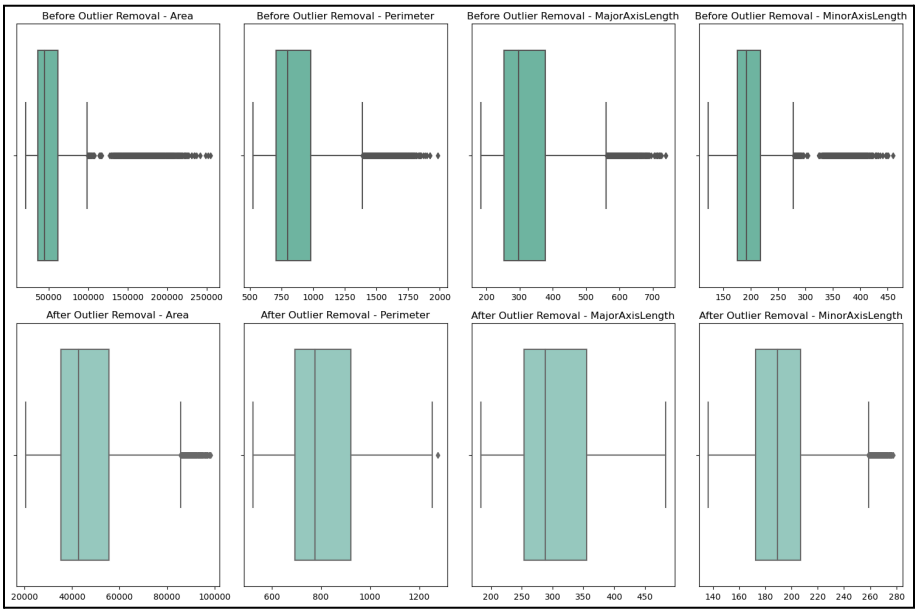


- Strongly Correlated Features (High Positive Correlation):

- Area, Perimeter, ConvexArea, and EquivDiameter. These features show a very high correlation (close to 1). This is expected since these measurements are related geometrically. Including all these features might introduce redundancy.
- ShapeFactor1, ShapeFactor2, and ShapeFactor3: These features also have high mutual correlation (e.g., ShapeFactor2 and ShapeFactor3 > 0.8). Similar to the above, redundancy might exist.
- Since these features are highly redundant, indicating that one of them could represent the group effectively without losing much information.

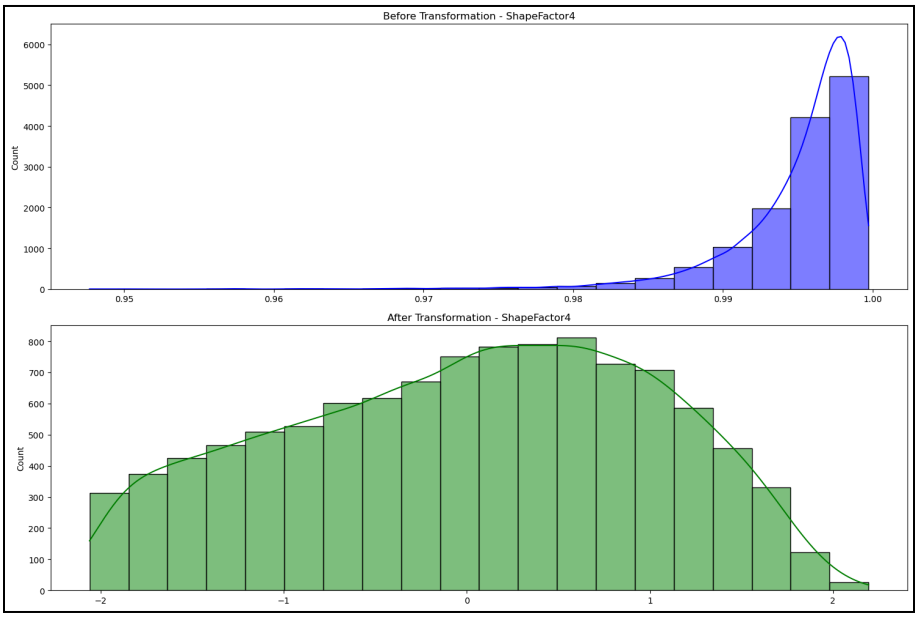
- Weakly or Negatively Correlated Features:
  - Features like Extent, Solidity, and Roundness have weak or moderate correlation with most others. These features might capture unique aspects of the data that are less related to others, making them potentially valuable for distinguishing classes.

Outlier Removal:



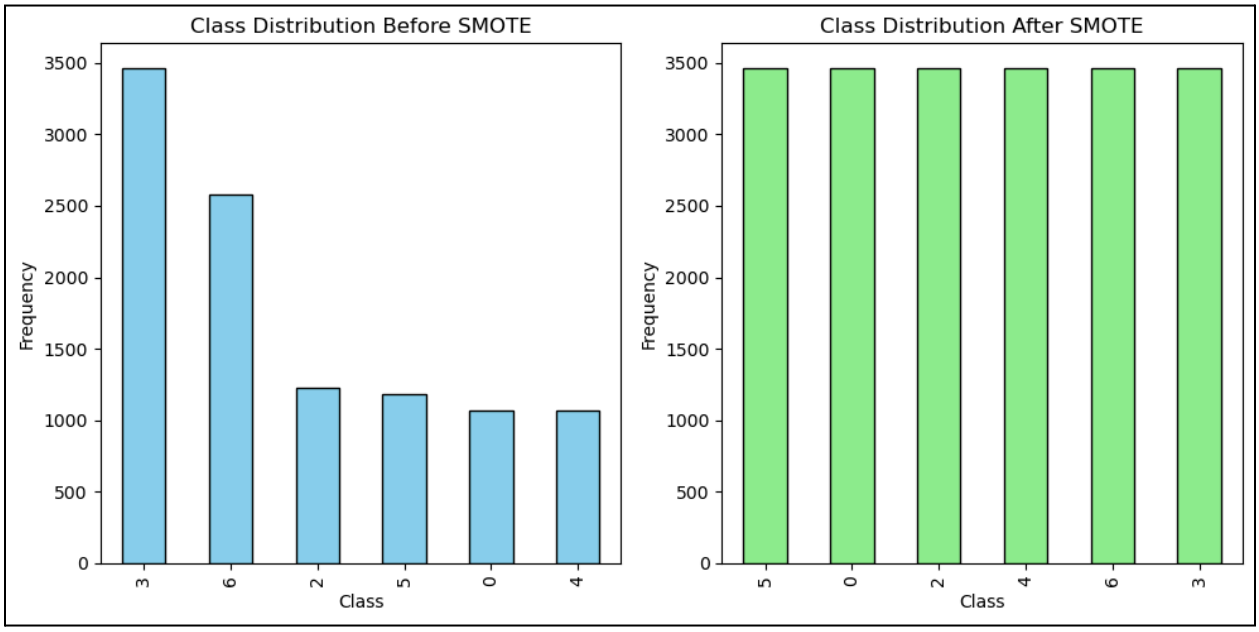
Outliers are removed using the IQR (Interquartile Range) method. Features with values beyond 1.5 times the IQR above or below the 25th and 75th percentiles are considered outliers and excluded.

Transformations:



- Log and Yeo-Johnson transformations help in normalizing the data, ensuring that the model handles non-normal distributions efficiently
    - Log transformation is applied to features with positive skewness (skew > 1) to normalize the distribution.
    - Yeo-Johnson transformation is applied to features with negative skewness (skew < -1) to make the data more Gaussian.
- This ensures the features are in a suitable form for model training, improving model accuracy and performance.

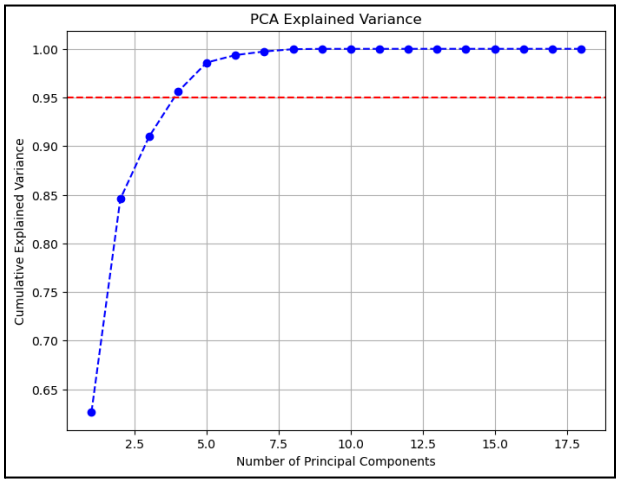
SMOTE and Standardization:

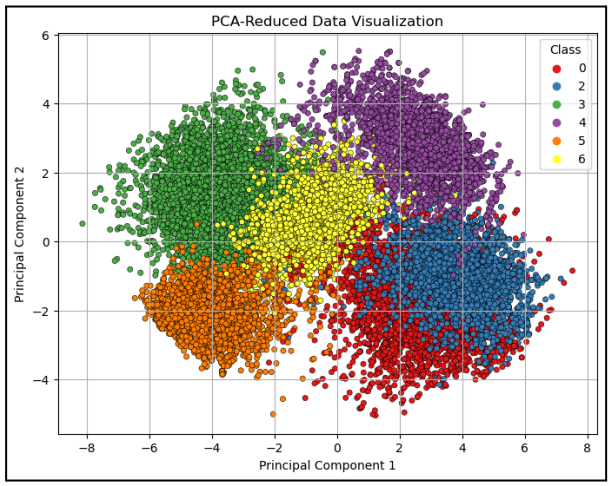


SMOTE (Synthetic Minority Over-sampling Technique) is applied to balance the classes, addressing any class imbalance and ensuring the model doesn't favor the majority class. Feature Scaling:

Standardization using StandardScaler() is applied to scale the features to have a mean of 0 and standard deviation of 1, ensuring equal weighting in distance-based models like KNN or SVM.

Reduction using PCA:

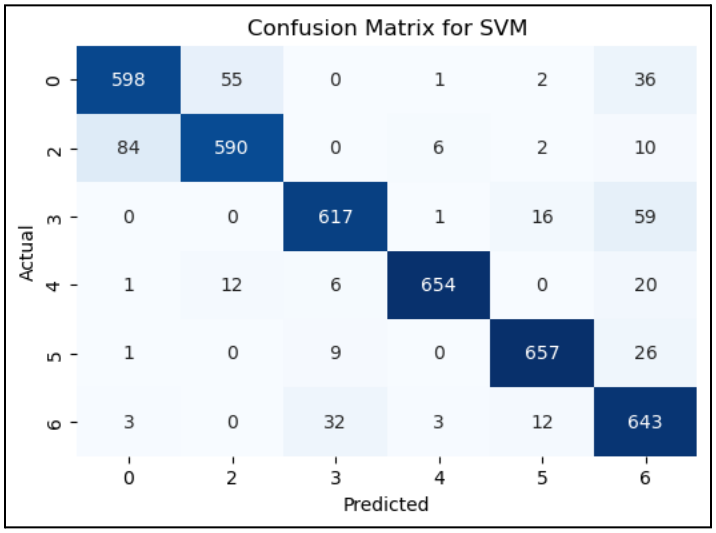




Dimensionality Reduction:

PCA (Principal Component Analysis) is applied to reduce dimensionality while retaining 95% of the variance in the dataset. This reduces the feature space and helps prevent overfitting.

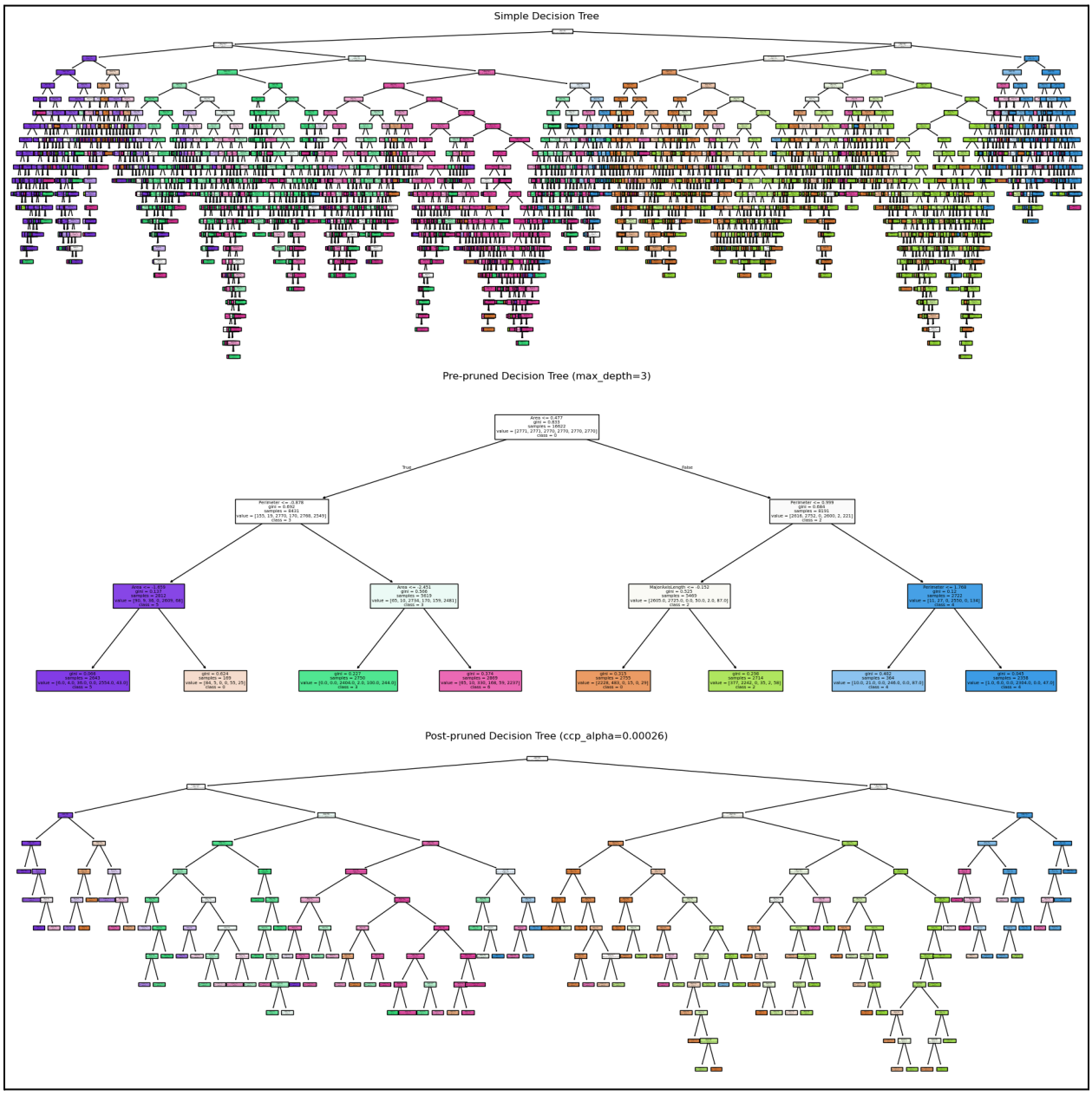
SVM Model:



- The SVM model was trained on PCA transformed and standardised Data, using K-fold cross validation.

- K-Fold Cross-Validation Scores: [0.90158807, 0.90976901, 0.90279115, 0.90300842, 0.90060168]
- Mean CV Accuracy: 0.904
- Standard Deviation: 0.003
- These results indicate consistent performance across different folds, with minimal variance.
- The cross-validation results demonstrate minimal variability, suggesting that the model generalizes well to unseen data.
- Classes 4 and 5 exhibit the highest precision and recall, indicating the model's strong ability to identify these categories accurately.
- Class 6 shows lower precision compared to other classes, suggesting potential overlap or misclassification with other classes.
- A mean accuracy of 90.4% across folds and a similar test accuracy confirm that the model performs well with the selected kernel and hyperparameters.
- The classification report indicates a good balance between precision, recall, and F1-score, with no significant class imbalances impacting performance.

Decision Tree Model:



Decision Tree Results:

	precision	recall	f1-score	support
0	0.93	0.74	0.83	692
2	0.79	0.93	0.85	692
3	0.85	0.91	0.88	693
4	0.97	0.94	0.95	693
5	0.96	0.92	0.94	693
6	0.83	0.85	0.84	693
accuracy			0.88	4156
macro avg	0.89	0.88	0.88	4156
weighted avg	0.89	0.88	0.88	4156

Simple Decision Tree Metrics:

Accuracy: 0.897

Precision: 0.897

Recall: 0.897

F1 Score: 0.897

Sensitivity: nan

Specificity: nan

ROC AUC: 0.938

Pre-pruned Decision Tree Metrics:

Accuracy: 0.860

Precision: 0.862

Recall: 0.860

F1 Score: 0.861

Sensitivity: nan

Specificity: nan

ROC AUC: 0.963

Post-pruned Decision Tree Metrics:

Accuracy: 0.918

Precision: 0.918

Recall: 0.918

F1 Score: 0.918

Sensitivity: nan

Specificity: nan

ROC AUC: 0.987

Simple Tree Accuracy: 0.897

Pre-pruned Tree Accuracy: 0.860

Best Post-pruned Tree Accuracy: 0.918

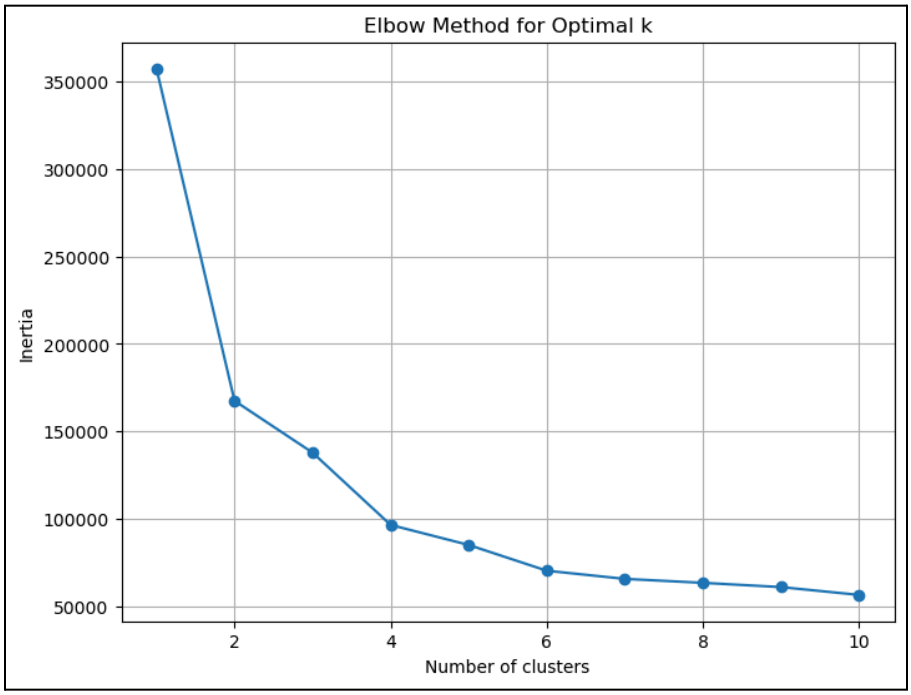
- The Decision Tree Model was run on PCA resampled and standardized data. Here holdout method was used, 80% for training and 20% for testing.
- Simple Decision Tree:
    - Accuracy: 0.897
    - Precision, Recall, and F1-Score: Moderate performance across all classes, with particularly good performance on class 4 and class 5.
- Pre-pruned Decision Tree (max depth=3):
    - Accuracy: 0.860

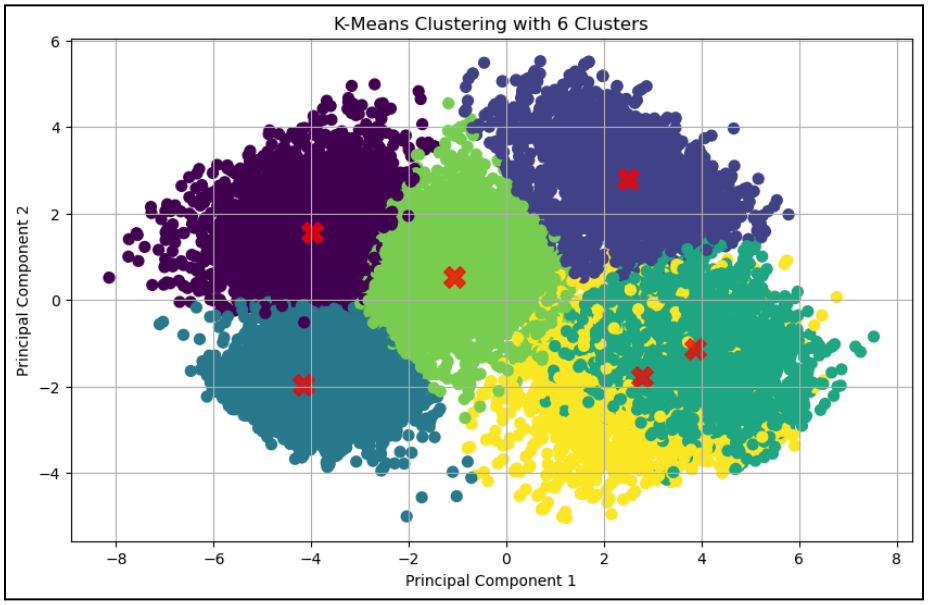
    - The model performs slightly worse compared to the simple tree, suggesting that the pruning step helped prevent overfitting but may have made the model too simple.

- Post-pruned Decision Tree (Cost Complexity Pruning):
    - Accuracy: 0.918
    - This model outperforms the simple and pre-pruned trees, indicating that the post-pruning process helped optimize the model and improve its performance by reducing overfitting while retaining sufficient complexity.
    - Best performance in both precision and recall, particularly in class 4 and 5, where the model has the highest accuracy and F1 score.
- Confusion matrices were used to calculate sensitivity (recall for the positive class) and specificity (recall for the negative class). These values were derived from the confusion matrix for each model, and the results showed that the post-pruned decision tree achieved the best performance in both sensitivity and specificity, reflecting its ability to identify both positive and negative class instances effectively.
- For a multi-class classification, a confusion matrix will not return just four values, but rather a square matrix with dimensions corresponding to the number of classes. This is why when the code tries to unpack the confusion matrix into just tn, fp, fn, tp, it fails, resulting in NaN values.
- For each model, the ROC AUC score was calculated, providing an overall measure of the model's ability to distinguish between classes:
    - Simple Tree: Reasonable performance with a decent ROC AUC score.
    - Pre-pruned Tree: Slightly lower performance due to oversimplification.

- Post-pruned Tree: Highest ROC AUC score, confirming that the post-pruned model effectively differentiates between the classes.

K-Means Model:





- The k-means model is trained on PCA reduced and standardised data, using k-fold cross validation.
- The Elbow Method helped determine the optimal number of clusters (k=6).
- K-Fold Cross-Validation was performed to ensure robust evaluation and generalization to unseen data.
- The model achieved decent clustering performance, though improvements in clustering separation could be achieved through further fine-tuning.
- K-Fold Cross-Validation Results
    - Mean Silhouette Score: 0.343 (indicates the quality of clustering).
    - Mean Davies-Bouldin Index: 1.118 (indicates the separation of clusters).
    - Mean ARI: 0.743 (measures how well the predicted clusters match the true labels).
- The model achieved decent clustering performance, though improvements in clustering separation could be achieved through further fine-tuning.

Conclusion

In this analysis of multiple machine learning models for classifying the Dry Bean dataset, the Post-Pruned Decision Tree stands out as the most effective model. By applying cost-complexity pruning, this model achieves a high test accuracy of 91.8%, outperforming both the simple and pre-pruned decision tree models. The pruning process helps avoid overfitting by limiting the model's complexity, while still maintaining its ability to capture key patterns in the data. This balance between accuracy and generalization makes the post-pruned decision tree highly

effective for unseen data. Additionally, the interpretability of this model, due to its clear decision rules, makes it especially valuable for understanding the classification process.

The Support Vector Machine (SVM) model also shows strong performance with a test accuracy of 90.4%. The SVM avoids overfitting, as there is no significant gap between training and test accuracies. However, recall for some classes (e.g., Class 2 and Class 6) is slightly lower, indicating that the model may not be as effective at detecting minority class patterns.

The K-Means clustering model achieved an Adjusted Rand Index (ARI) of 0.743, which shows a decent alignment between its predicted clusters and the true labels. However, the moderate silhouette score (0.343) and Davies-Bouldin Index (1.118) suggest some underfitting, as the model struggles with cluster separation and compactness. These metrics indicate that there is room for improvement, possibly through feature transformations or fine-tuning of the clusters. While K-Means can be useful for exploratory analysis, it is not the best choice for classification tasks, given the superior performance of the supervised models.

Overall, the Post-Pruned Decision Tree achieves the highest accuracy (91.8%) and strikes a good balance between model complexity and generalization. By effectively reducing overfitting, it performs better than both the simple decision tree (which slightly overfits) and the pre-pruned decision tree (which underfits). The SVM model, despite being slightly less accurate, is a strong alternative for classification tasks that require high stability and generalization. Finally, K-Means, while useful for exploratory analysis, does not perform as well as the supervised models in classification tasks. This evaluation highlights the importance of selecting models that align well with the dataset's characteristics and the specific goals of the classification task, while managing the risks of overfitting and underfitting effectively.