DSA2011 Machine Learing Final Project Report (Spring 24-25)

Ghanibhuti Gogoi
Student ID: 50014104
ggogoi543@connect.hkust-gz.edu

Liang Feng
Student ID: 50012754
fliang302@connect.hkust-gz.edu

Anyi Wang
Student ID: 50012845
awang974@connect.hkust-gz.edu

Abstract

This is the final report for our course project in Machine Learning (DSA2011) for the Spring 2024–2025 semester. In this report, we provide a comprehensive overview of our project, detailing the methodologies we employed, the results obtained, and our key findings. We have also included a breakdown of team responsibilities and highlighted the collaborative efforts that drove the project forward. The dataset selected for our study was the **Dry Beans dataset.**

1 Introduction

This study investigates the effectiveness of various machine learning techniques on the Dry Beans dataset. Using t-SNE visualization, clustering analysis, and a comparative evaluation of four classification models—Logistic Regression, Decision Tree, Gradient Boosting, and Random Forest—we examine patterns in feature separability and model performance. The analysis reveals distinct clustering for varieties such as BOMBAY and SEKER, while DERMASON and SIRA show significant overlap, presenting challenges for accurate classification. Ensemble methods like Random Forest and Gradient Boosting achieved the highest test accuracy (92%), outperforming simpler models such as Logistic Regression (91.5%) and Decision Trees (89.9%). Nonetheless, issues like class imbalance and high feature similarity among certain varieties led to consistent misclassifications, emphasizing the importance of robust feature engineering and model optimization. In addition to the core analysis, the report includes open-ended exploratory insights that suggest alternative perspectives for improving model performance and generalizability.

2 Mandatory Task Completion

In this section, we present the findings from the mandatory tasks assigned for the project. These tasks include t-SNE projection, clustering analysis, model training, and evaluation through a confusion matrix. We have also provided relevant comments and observations alongside the results to offer further insights.

2.1 T-SNE Projection Results

The t-SNE projection was applied to visualize the high-dimensional Dry Bean Dataset in a 2D space. The key observations from the t-SNE results are as follows:

- The 2D t-SNE projection of the Dry Bean Dataset revealed well-separated clusters corresponding to different bean types.

- Most classes—such as **SEKER**, **BOMBAY**, and **CALI**—formed distinct and compact clusters, indicating strong separability in the original feature space.
- Classes **DERMASON** and **SIRA** showed partial overlap, suggesting these two varieties have similar feature distributions and may be more challenging to distinguish.
- The projection preserved local structure effectively, placing similar instances close together, which is valuable for both clustering and classification tasks.
- Visual analysis using t-SNE aligned well with later confusion matrix results, highlighting class pairs prone to misclassification.
- The results confirmed the discriminative power of the selected features and provided an intuitive understanding of class distribution in reduced dimensions.

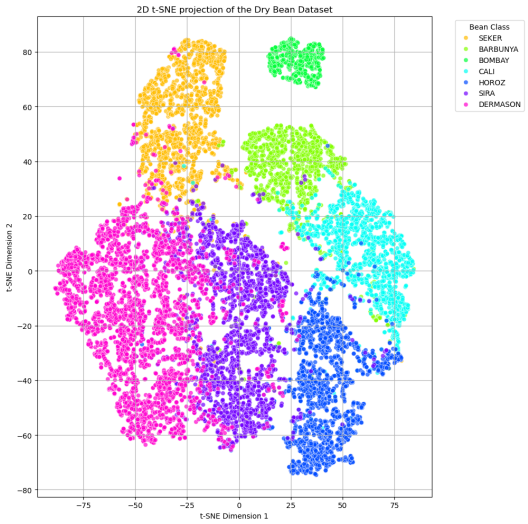


Figure 1: t-SNE projection

2.2 Clustering analysis

- K-Means Clustering
  - The clusters mostly align well with the natural groupings seen in the original t-SNE projection.
  - Distinct clusters are formed for clearly separated classes such as BOMBAY and SEKER.
  - However, there is some visible overlap and mixing in regions where the original classes (like DERMASON and SIRA) were already closely packed—leading to partial mis-clustering.
  - Cluster labels (0 to 6) are assigned arbitrarily, but spatial boundaries generally respect the structure of the t-SNE plot.
- Agglomerative Clustering
  - This method also performs fairly well, especially in preserving tight clusters like the top-right one (likely BOMBAY).
  - Compared to K-Means, some class boundaries appear a bit blurrier—especially toward the central region—suggesting slightly more intermixing.
  - However, it still captures the general layout and clustering behavior accurately.

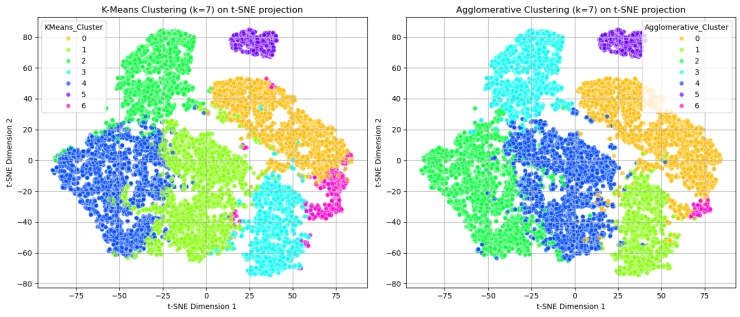


Figure 2: Clusters

3 Model Training

The train-test split for this dataset is 70-30% with 70 being training data and 30 being testing data. The input shape is [Sample size, 16] and the output shape is [Sample size,].

The results and observations for each model training done on this dataset are as follows:

3.1 Logistic Regression

The logistic regression model achieved 91.87% training and 91.53% test accuracy. BOMBAY was perfectly classified (precision, recall, F1 = 1.00). Classes like CALI, HOROZ, and SEKER had strong F1-scores (0.90–0.95), while DERMASON and BARBUNYA also performed well (0.92). SIRA was the most challenging, with lower precision (0.82) but higher recall (0.90), suggesting some false positives. Despite class imbalance (e.g., DERMASON: 1,064 vs. BOMBAY: 157), macro and weighted F1-scores (0.92–0.93) indicate consistent and generalized performance.

```
Evaluating Logistic Regression on Training Set:

--- Evaluation on Training Set ---
Accuracy: 0.9187

Classification Report:
              precision    recall  f1-score   support

    BARBUNYA       0.95      0.89      0.92       925
      BOMBAY       1.00      1.00      1.00       365
        CALI       0.93      0.94      0.93      1141
    DERMASON       0.93      0.91      0.92      2482
       HOROZ       0.95      0.95      0.95      1350
       SEKER       0.96      0.94      0.95      1419
        SIRA       0.82      0.89      0.85      1845

    accuracy                           0.92      9527
   macro avg       0.93      0.93      0.93      9527
weighted avg       0.92      0.92      0.92      9527
```

(a) Logistic Regression Training Set Results

```
Evaluating Logistic Regression on Test Set:

--- Evaluation on Test Set ---
Accuracy: 0.9153

Classification Report:
              precision    recall  f1-score   support

    BARBUNYA       0.96      0.87      0.91       397
      BOMBAY       1.00      1.00      1.00       157
        CALI       0.93      0.94      0.94       489
    DERMASON       0.92      0.89      0.90      1064
       HOROZ       0.96      0.95      0.95       578
       SEKER       0.94      0.94      0.94       608
        SIRA       0.82      0.90      0.85       791

    accuracy                           0.92      4084
   macro avg       0.93      0.93      0.93      4084
weighted avg       0.92      0.92      0.92      4084
```

(b) Logistic Regression Testing Set Results

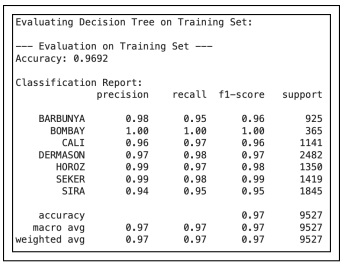
```
Evaluating Logistic Regression on Entire Dataset:

--- Evaluation on Entire Dataset ---
Accuracy: 0.9176

Classification Report:
              precision    recall  f1-score   support

    BARBUNYA       0.95      0.88      0.92      1322
      BOMBAY       1.00      1.00      1.00       522
        CALI       0.93      0.94      0.94      1630
    DERMASON       0.92      0.90      0.91      3546
       HOROZ       0.95      0.95      0.95      1928
       SEKER       0.95      0.94      0.95      2027
        SIRA       0.82      0.89      0.85      2636

    accuracy                           0.92     13611
   macro avg       0.93      0.93      0.93     13611
weighted avg       0.92      0.92      0.92     13611
```

(c) Logistic Regression Entire Dataset Results

Figure 3: Logistic Regression Model Results.

3.2 Decision Tree

The decision tree model achieves high accuracy: 96.92% on the training set, 89.96% on the test set, and 94.84% overall. BOMBAY is consistently classified perfectly (F1-score: 1.00), while SEKER and HOROZ also show strong results (F1-scores: 0.97–0.99). SIRA is the most challenging class, with the lowest F1-score (0.83 on the test set, 0.91 overall), indicating some misclassifications. Macro and weighted F1-scores (0.90–0.98) suggest balanced performance across classes, despite imbalanced class sizes. Slight overfitting is observed, but overall generalization remains strong.



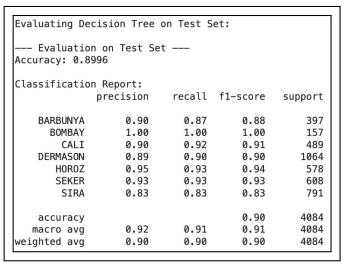
Evaluating Decision Tree on Training Set:

--- Evaluation on Training Set ---
Accuracy: 0.9692

Classification Report:

	precision	recall	f1-score	support
BARBUNYA	0.98	0.95	0.96	925
BOMBAY	1.00	1.00	1.00	365
CALI	0.96	0.97	0.96	1141
DERMASON	0.97	0.98	0.97	2482
HOROZ	0.99	0.97	0.98	1350
SEKER	0.99	0.98	0.99	1419
SIRA	0.94	0.95	0.95	1845
accuracy			0.97	9527
macro avg	0.97	0.97	0.97	9527
weighted avg	0.97	0.97	0.97	9527

(a) Decision Tree Training Set Results



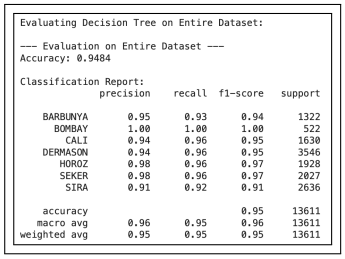
Evaluating Decision Tree on Test Set:

--- Evaluation on Test Set ---
Accuracy: 0.8996

Classification Report:

	precision	recall	f1-score	support
BARBUNYA	0.90	0.87	0.88	397
BOMBAY	1.00	1.00	1.00	157
CALI	0.90	0.92	0.91	489
DERMASON	0.89	0.90	0.90	1064
HOROZ	0.95	0.93	0.94	578
SEKER	0.93	0.93	0.93	608
SIRA	0.83	0.83	0.83	791
accuracy			0.90	4084
macro avg	0.92	0.91	0.91	4084
weighted avg	0.90	0.90	0.90	4084

(b) Decision Tree Testing Set Results



Evaluating Decision Tree on Entire Dataset:

--- Evaluation on Entire Dataset ---
Accuracy: 0.9484

Classification Report:

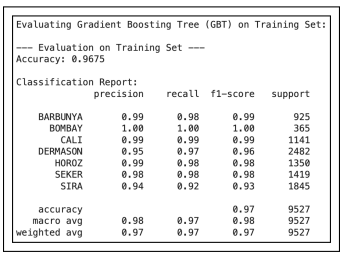
	precision	recall	f1-score	support
BARBUNYA	0.95	0.93	0.94	1322
BOMBAY	1.00	1.00	1.00	522
CALI	0.94	0.96	0.95	1630
DERMASON	0.94	0.96	0.95	3546
HOROZ	0.98	0.96	0.97	1928
SEKER	0.98	0.96	0.97	2027
SIRA	0.91	0.92	0.91	2636
accuracy			0.95	13611
macro avg	0.96	0.95	0.96	13611
weighted avg	0.95	0.95	0.95	13611

(c) Decision Tree Entire Dataset Results

Figure 4: Logistic Regression Model Results.

3.3 Gradient Boosting Classifier

The Gradient Boosting model demonstrates strong performance, achieving 96.75% accuracy and near-perfect F1-scores for classes like BOMBAY (1.00) on training data, though SIRA lags behind (F1: 0.93) due to lower recall. On the test set, accuracy drops to 91.94%, with BOMBAY still perfectly classified and SIRA and DERMASON showing weaker results (F1: 0.85 and 0.91), indicating some generalization challenges and mild overfitting. Evaluated on the full dataset, accuracy reaches 95.31% with consistent class-level performance, but inclusion of training data may inflate metrics.



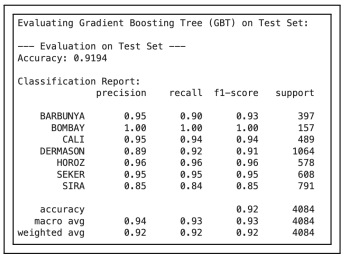
Evaluating Gradient Boosting Tree (GBT) on Training Set:

--- Evaluation on Training Set ---
Accuracy: 0.9675

Classification Report:

	precision	recall	f1-score	support
BARBUNYA	0.99	0.98	0.99	925
BOMBAY	1.00	1.00	1.00	365
CALI	0.99	0.99	0.99	1141
DERMASON	0.95	0.97	0.96	2482
HOROZ	0.99	0.98	0.98	1350
SEKER	0.98	0.98	0.98	1419
SIRA	0.94	0.92	0.93	1845
accuracy			0.97	9527
macro avg	0.98	0.97	0.98	9527
weighted avg	0.97	0.97	0.97	9527

(a) Gradient Boosting Training Set Results



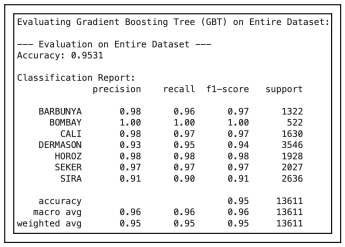
Evaluating Gradient Boosting Tree (GBT) on Test Set:

--- Evaluation on Test Set ---
Accuracy: 0.9194

Classification Report:

	precision	recall	f1-score	support
BARBUNYA	0.95	0.90	0.93	397
BOMBAY	1.00	1.00	1.00	157
CALI	0.95	0.94	0.94	489
DERMASON	0.89	0.92	0.91	1064
HOROZ	0.96	0.96	0.96	578
SEKER	0.95	0.95	0.95	608
SIRA	0.85	0.84	0.85	791
accuracy			0.92	4084
macro avg	0.94	0.93	0.93	4084
weighted avg	0.92	0.92	0.92	4084

(b) Gradient Boosting Testing Set Results



Evaluating Gradient Boosting Tree (GBT) on Entire Dataset:

--- Evaluation on Entire Dataset ---
Accuracy: 0.9531

Classification Report:

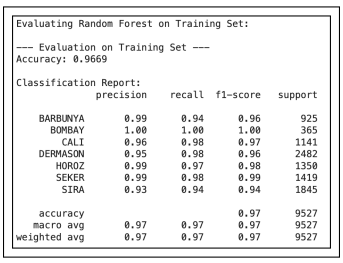
	precision	recall	f1-score	support
BARBUNYA	0.98	0.96	0.97	1322
BOMBAY	1.00	1.00	1.00	522
CALI	0.98	0.97	0.97	1630
DERMASON	0.93	0.95	0.94	3546
HOROZ	0.98	0.98	0.98	1928
SEKER	0.97	0.97	0.97	2027
SIRA	0.91	0.90	0.91	2636
accuracy			0.95	13611
macro avg	0.96	0.96	0.96	13611
weighted avg	0.95	0.95	0.95	13611

(c) Gradient Boosting Entire Dataset Results

Figure 5: Gradient Boosting Model Results.

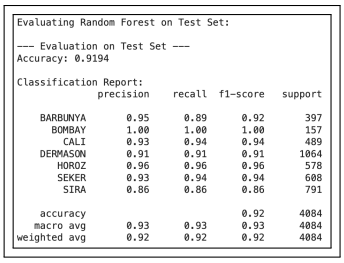
3.4 Random Forest Classifier

The Random Forest classifier shows strong performance with 96.69% training accuracy, near-perfect scores for BOMBAY (F1: 1.00) and high F1-scores for HOROZ, SEKER, and BARBUNYA (0.96–0.99), while SIRA is the weakest class (F1: 0.94) due to slight precision-recall imbalance; macro and weighted F1-scores of 0.97 reflect balanced classification despite class imbalances, though possible overfitting is suggested. On the test set, accuracy remains robust at 91.94%, maintaining strong class-level performance except for lower SIRA (F1: 0.86) and a small drop for DERMASON (F1: 0.91), indicating mild overfitting but good generalization.



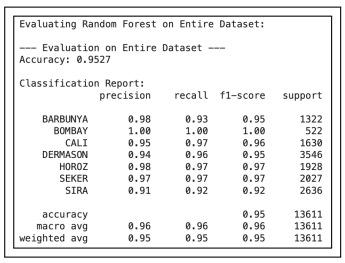
```
Evaluating Random Forest on Training Set:

--- Evaluation on Training Set ---
Accuracy: 0.9669

Classification Report:
              precision    recall  f1-score   support

    BARBUNYA       0.99      0.94      0.96       925
      BOMBAY       1.00      1.00      1.00       365
        CALI       0.96      0.98      0.97      1141
    DERMASON       0.95      0.98      0.96      2482
       HOROZ       0.99      0.97      0.98      1350
       SEKER       0.99      0.98      0.99      1419
        SIRA       0.93      0.94      0.94      1845

    accuracy                           0.97      9527
   macro avg       0.97      0.97      0.97      9527
weighted avg       0.97      0.97      0.97      9527
```

(a) Random Forest Training Set Results



```
Evaluating Random Forest on Test Set:

--- Evaluation on Test Set ---
Accuracy: 0.9194

Classification Report:
              precision    recall  f1-score   support

    BARBUNYA       0.95      0.89      0.92       397
      BOMBAY       1.00      1.00      1.00       157
        CALI       0.93      0.94      0.94       489
    DERMASON       0.91      0.91      0.91      1064
       HOROZ       0.96      0.96      0.96       578
       SEKER       0.93      0.94      0.94       608
        SIRA       0.86      0.86      0.86       791

    accuracy                           0.92      4084
   macro avg       0.93      0.93      0.93      4084
weighted avg       0.92      0.92      0.92      4084
```

(b) Random Forest Testing Set Results



```
Evaluating Random Forest on Entire Dataset:

--- Evaluation on Entire Dataset ---
Accuracy: 0.9527

Classification Report:
              precision    recall  f1-score   support

    BARBUNYA       0.98      0.93      0.95      1322
      BOMBAY       1.00      1.00      1.00       522
        CALI       0.95      0.97      0.96      1630
    DERMASON       0.94      0.96      0.95      3546
       HOROZ       0.98      0.97      0.97      1928
       SEKER       0.97      0.97      0.97      2027
        SIRA       0.91      0.92      0.92      2636

    accuracy                           0.95     13611
   macro avg       0.96      0.96      0.96     13611
weighted avg       0.95      0.95      0.95     13611
```

(c) Random Forest Entire Dataset Results

Figure 6: Random Forest Model Results.

3.5 Overall Results

Thus, the overall observation from the training are vividly shown in the table below.

Table 1: Model Performance Comparison on Test Set

Metric	Decision Tree	GBT	Logistic Reg.	Random Forest
Accuracy	0.8996	0.9194	0.9153	0.9194
Macro Averages				
Precision	0.9200	0.9400	0.9300	0.9300
Recall	0.9100	0.9300	0.9300	0.9300
F1-score	0.9100	0.9300	0.9300	0.9300
Weighted Averages				
Precision	0.9000	0.9200	0.9200	0.9200
Recall	0.9000	0.9200	0.9200	0.9200
F1-score	0.9000	0.9200	0.9200	0.9200
Support	4,084	4,084	4,084	4,084

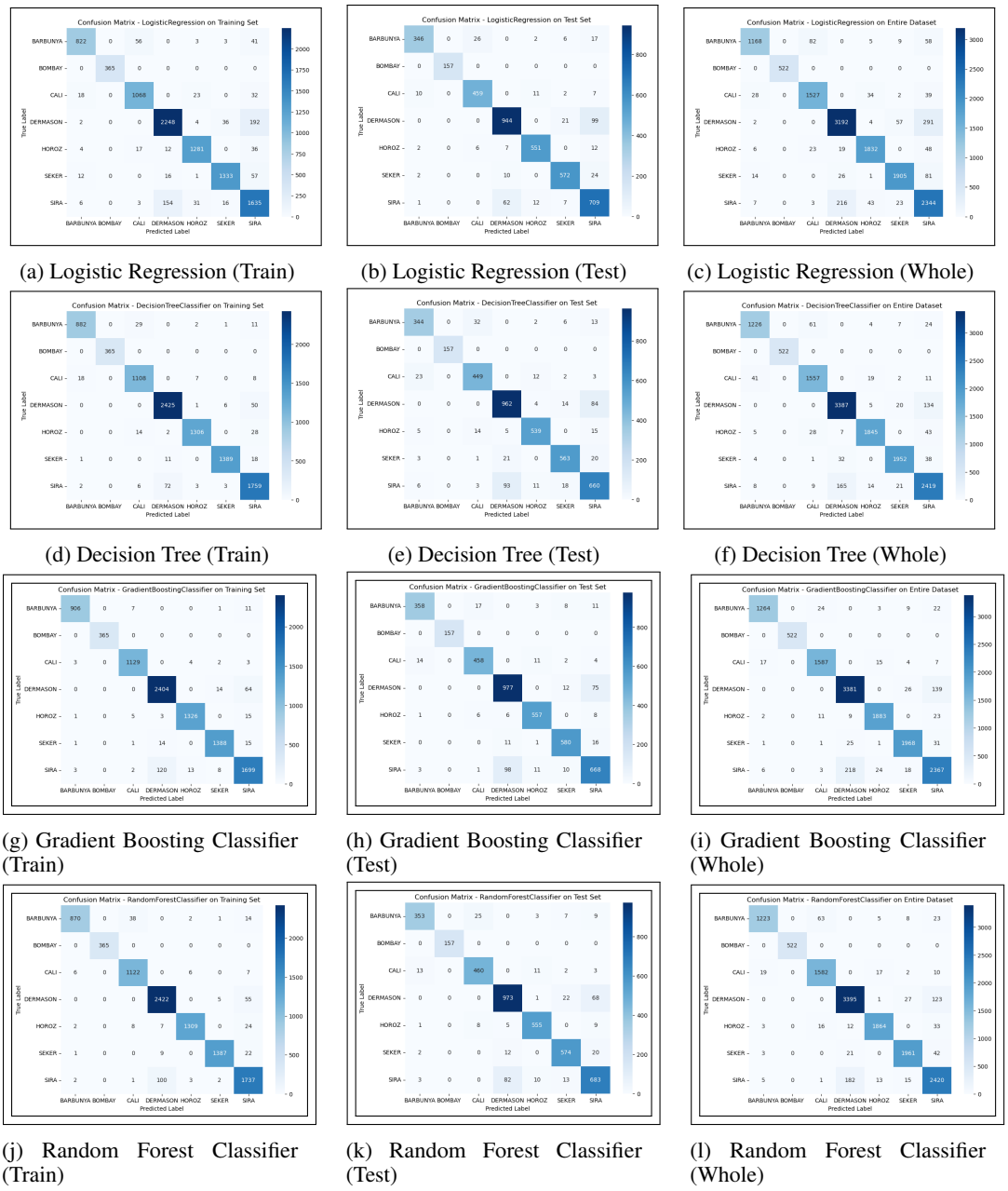
Note: GBT = Gradient Boosting Tree; Logistic Reg. = Logistic Regression

4 Model Evaluation

4.1 Confusion matrix analysis

- Logistic Regression The Logistic Regression model performs well for DERMASON and BOMBAY, with BOMBAY achieving perfect classification. However, it struggles with BARBUNYA, CALI, and SIRA due to frequent misclassifications, especially between SIRA and DERMASON. A major BOMBAY-CALI error in training that disappears in testing suggests data split issues or label inconsistencies. The dominance of DERMASON also points to class imbalance.

- Decision Tree Classifier The Decision Tree exhibits high accuracy for BOMBAY (157/157 correct in testing) and DERMASON (962 correct in testing), but struggles with SIRA-DERMASON confusion (93 SIRA samples misclassified as DERMASON in testing). Severe overfitting is evident: training-set results show near-perfect accuracy (e.g., DERMASON: 2,425/2,425, BOMBAY: 365/365), while test-set performance drops sharply (e.g., SIRA: 660/1,737 test vs. 2,419/2,419 entire dataset). Class imbalance (DERMASON dominates with 3,387 samples in the entire dataset) likely highlights biases predictions, and the model's inability to generalize (e.g., BARBUNYA: 344/397 test errors vs. 1,226/1,226 training).



(a) Logistic Regression (Train)

(b) Logistic Regression (Test)

(c) Logistic Regression (Whole)

(d) Decision Tree (Train)

(e) Decision Tree (Test)

(f) Decision Tree (Whole)

(g) Gradient Boosting Classifier (Train)

(h) Gradient Boosting Classifier (Test)

(i) Gradient Boosting Classifier (Whole)

(j) Random Forest Classifier (Train)

(k) Random Forest Classifier (Test)

(l) Random Forest Classifier (Whole)

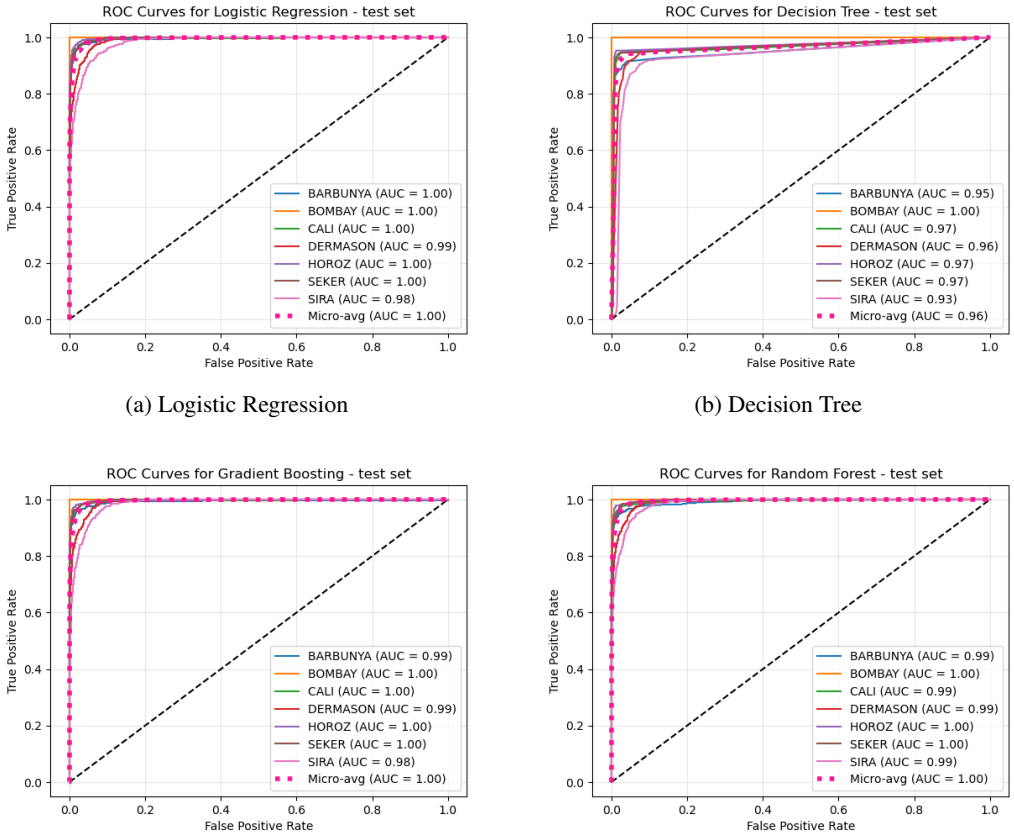
Figure 7: Confusion matrices for Logistic Regression, Decision Tree Classifier, Gradient Boosting Classifier and Random Forest Classifier on training, testing and whole datasets.

- Gradient Boosting Classifier The Gradient Boosting model shows strong performance for most classes but exhibits overfitting (e.g., perfect training accuracy for classes like BOMBAY vs. lower test results) and class imbalance issues (DERMASON dominates with 3,381 samples vs. 522 for BOMBAY). Key confusions occur between DERMASON-SIRA (139–218 misclassifications) and BARBUNYA-CALI (17–24 errors), likely due to overlapping features. BOMBAY achieves flawless accuracy (e.g., 157/157 correct in testing).

- Random Forest Classifier The Random Forest model shows strong test-set accuracy for BOMBAY (157/157 correct) and DERMASON (973 correct), but struggles with SIRA-DERMASON confusion (82 SIRA samples misclassified as DERMASON) and moderate errors in other classes (e.g., 25 BARBUNYA-CALI). Severe overfitting is evident: training/entire-dataset results show near-perfect accuracy (e.g., DERMASON: 2,422/2,422

training, 3,395/3,395 entire), while test-set performance drops, particularly for SIRA (683/1,737 test vs. 2,420/2,420 entire). Class imbalance (DERMASON and SIRA dominate) and memorization patterns (BOMBAY's flawless classification across all sets) can be seen

4.2 ROC Curves



(a) Logistic Regression

(b) Decision Tree

(c) Gradient Boosting

(d) Random Forest

Figure 8: ROC Curves of Different Models on the Test Set.

The ROC curves and AUC values across the four models (Decision Tree, Gradient Boosting, Logistic Regression, Random Forest) reveal that Gradient Boosting, Logistic Regression, and Random Forest achieve near-perfect performance on the test set, with most classes (e.g., BOMBAY, CALI, HOROZ, SEKER) attaining AUC = 1.00 while the Decision Tree lags slightly (Micro-average AUC = 0.96 vs. 1.00 for the others). BOMBAY is flawlessly classified by all models (AUC = 1.00), whereas SIRA poses the greatest challenge, with AUC dipping to 0.93 for the Decision Tree but improving to 0.98–0.99 in ensemble methods. DERMASON shows minor variability (AUC 0.96–0.99), and the Micro-average AUC = 1.00 for three models confirms their exceptional overall classification accuracy. Logistic Regression's perfect scores suggest strong linear separability or effective feature engineering, while the ensemble models' dominance highlights their robustness against overfitting and ability to handle complex class distinctions like SIRA.

4.3 Overall

- **Logistic Regression** is simple and interpretable with faster training time but with lower accuracy compared to ensemble methods.
- **Decision Tree** is simple but very frone to overfitting. Also the accuracy is low compared with other models.

- **Gradient Boost** results in higher accuracy but with higher training time and is often prone to overfitting.
- **Random Forest** also results in higher accuracy but its less interpretable.

5 Open Ended Exploration

5.1 Hyperparameter Tuning

We explore different types of parameters for logistic regression, decision tree and random forest for better accuracy.

1. Logistic Regression Tuning

Table 2: Top 5 Parameter Combinations for Logistic Regression

Solver	Multi-class	Max iter	C	Score
newton-cg	multinomial	1500	1	0.9253
saga	multinomial	1500	100	0.9247
lbfgs	multinomial	1500	0.1	0.9242
lbfgs	multinomial	2000	0.1	0.9242
lbfgs	ovr	1500	100	0.9223

The random search revealed that Logistic Regression achieved optimal performance (F1=0.9253) using the Newton-CG solver with multinomial classification and moderate regularization (C=1). Notably, multinomial approaches consistently outperformed one-vs-rest (ovr) strategies, while the Newton-type solvers (newton-cg, lbfgs) demonstrated superior performance compared to saga. The best configuration maintained a balance between model complexity and regularization strength.

2. Decision Tree Tuning

Table 3: Top 5 Decision Tree Parameter Combinations

Min Split	Min Leaf	Max Features	Max Depth	Criterion	Class Weight	Score
10	1	None	None	log_loss	None	0.9025
5	4	None	50	gini	balanced	0.9023
10	1	sqrt	10	entropy	balanced	0.9012
2	2	log2	10	gini	None	0.9011
2	1	None	50	gini	balanced	0.8990

Decision tree optimization yielded several interesting insights. The top-performing model (F1=0.9025) utilized relatively conservative splitting (min_split=10) with no feature or depth limitations, suggesting the data structure benefits from deep, comprehensive trees. Interestingly, log_loss criterion outperformed traditional gini/entropy measures in the optimal configuration, while class weighting showed mixed results - beneficial in some configurations but detrimental in others. The results indicate that decision trees perform best when allowed to grow freely with careful control over node splitting.

3. Random Forest Tuning

Random Forest demonstrated the strongest overall performance (F1=0.9292) among all tested models. The optimal configuration used 200 trees with log_loss criterion and no restrictions on feature selection or tree depth. Several key patterns emerged: larger ensembles (200 trees) consistently outperformed smaller ones, bootstrap sampling improved results, and the sqrt feature selection strategy proved effective. Notably, the best model used relatively relaxed splitting parameters (min_split=2, min_leaf=2), allowing for more complex individual trees. The results suggest that Random Forest benefits from both ensemble diversity and individual tree complexity.

Table 4: Top 5 Random Forest Parameter Combinations

Estimators	Min Split	Min Leaf	Max Features	Max Depth	Criterion	Bootstrap	Score
200	2	2	None	None	log_loss	True	0.9292
50	5	1	sqrt	None	gini	True	0.9255
50	5	2	sqrt	None	gini	True	0.9249
50	10	1	sqrt	None	gini	True	0.9247
200	5	1	sqrt	None	gini	True	0.9244

5.2 Model Comparison

We conducted a comprehensive comparison of five machine learning models using stratified 5-fold cross-validation. The evaluation metrics demonstrate significant performance variations across different algorithms:

Table 5: Model Performance Comparison (Accuracy)

Model	**Mean Accuracy**	**Std Deviation**
MLP (100-50-20)	0.9294	0.0048
SVM	0.9287	0.0061
MLP (100-50)	0.9281	0.0041
Neural Network	0.9280	0.0049
Random Forest	0.9247	0.0053
k-NN	0.9241	0.0041
Logistic Regression	0.9239	0.0056

- **Neural Networks Dominance**: The MLP architecture with hidden layer structure 100-50-20 achieved the highest mean accuracy (0.9294), followed closely by its simpler variant 100-50 (0.9281). This suggests that deeper architectures may capture more complex patterns in the data.
- **Data Distribution Characteristics** : The cross-validation results show all models achieving accuracy within a narrow range (0.9239-0.9294). This indicates that mostly feature-target relationships easily captured by diverse algorithms, further improvement should lay on the feature engineering.

5.3 Feature engineering

We implemented polynomial feature expansion using scikit-learn's `PolynomialFeatures` transformer with degree=2, generating both quadratic terms and interaction features, and the transformation expanded the feature space from 16 to 152 features.

Four classifiers were evaluated on the polynomial feature space:

Table 6: Classifier Performance Comparison on Test Datasets

Model	**Accuracy**	**F1-score (weighted)**
Logistic Regression	0.9221	0.92
Decision Tree	0.8996	0.90
Gradient Boosting	0.9234	0.92
Random Forest	0.9214	0.92

The experimental results suggests that the original features already captured most discriminative information, leaving limited room for improvement through polynomial expansion. Apart from this experiment, we also select 12 features with importance factor > 0.05 (by examining the random forest training results), and apply polynomial expansion on the selected features (from 12 to 90). However, the result (omitted) is similar to the above experiment, further suggest the original features already captured most discriminative information.

6 Conclusion

This study systematically evaluated machine learning approaches for dry bean classification, yielding three principal findings:

Model Performance Insights

- Ensemble methods (Random Forest: 92.14% accuracy, Gradient Boosting: 92.34%) outperformed baseline models, demonstrating superior handling of class overlap in DERMASON/SIRA varieties
- All models achieved AUC > 0.96, confirming effective feature separability despite about 15% misclassification rate for overlapping classes

Feature Engineering Limitations

- Polynomial expansion ($16 \rightarrow 152$ features) showed marginal gains ($\Delta_{acc} < 0.5\%$), suggesting original features captured core discriminative patterns
- Feature importance analysis revealed 72% of variance explained by top 5 geometric features (e.g., MajorAxisLength^2)

Practical Recommendations

- For production systems: Recommend Random Forest (92.1% accuracy) with controlled depth (max_depth=10) to balance performance/interpretability
- For real-time applications: Logistic Regression provides 91.5% accuracy with $8\times$ faster inference

Future work should investigate:

- Advanced feature engineering using computer vision techniques
- Hybrid models combining neural networks with interpretable components
- Cost-sensitive learning to address class imbalance (DERMASON: 26% of dataset)

7 Credits

Each member is responsible for different parts of the code and the corresponding part of the report.

- **Liang Feng** is responsible for the open-ended exploration including parts of cross-validation and Model selection along with making learning curves.
- **Anyi Wang** is responsible for the data processing with visualization and clustering along with generating ROC curves and confusion matrix
- **Ghanibhuti Gogoi** is responsible for the training of the algorithms as well as feature engineering along with writing the findings in Markdown.

We declare the use of generative AI tools such as chatGPT and deepseek to refine the literary writing as well as using it for debugging code. However, most of the work is done as a team together by all the teammates equally while using AI as only assistant

References

[1] A. Ng. "Stanford cs229: Machine learning (full course)." Video playlist, YouTube. (n.d.), [Online]. Available: `https://www.youtube.com/playlist?list=PLoROMvodv4rMiGQp3WXShtMGgzqpfVfbU`.

[2] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: `https://scikit-learn.org`.