GMM-KNN Classification Pipeline (Proposed)

---

1. Introduction

This project implements a (Hybrid) two-stage learning pipeline combining unsupervised clustering (Gaussian Mixture Models) with a supervised classifier (K-Nearest Neighbors). The goal is to classify mango images into quality categories (excluding the "Badami" class) by first identifying latent cluster structure in the feature space, then using these cluster labels as additional features for KNN classification. Model development follows a standard workflow: data ingestion, preprocessing, clustering, feature augmentation, feature selection, hyperparameter tuning, evaluation, and final inference.

---

2. Data Loading & Preprocessing

- **Datasets:**
  - **Training set:** Balanced CSV of extracted features and labels.
  - **Validation set:** Held-out CSV without outliers (Badami class removed).
- **Filtering:** Any samples belonging to unwanted classes are dropped prior to further processing.
- **Feature/Target Separation:** Input features and the target column ("target") are separated; the image-name column is also removed from the validation inputs.
- **Scaling:** A StandardScaler is fit on the training features and applied to both training and validation data to ensure zero mean and unit variance across all attributes.

---

3. Unsupervised Clustering via GMM

- **Model Specification:**
  - Number of clusters (components): 4
  - Covariance type: full
  - Random seed: 42 (for reproducibility)
- **Training:** The GMM is fit on the scaled training set.
- **Cluster Prediction:** Each sample in both training and validation is assigned a cluster label by the trained GMM.
- **Augmented Features:** The cluster label is appended as an extra feature dimension, resulting in an augmented dataset for subsequent classification.

---

4. Feature Selection with Sequential Forward Selection (SFS)

- **Base Estimator:** K-Nearest Neighbors (default settings)
- **Search Strategy:** Forward (no floating), evaluating between 1 and 19 features.

- **Validation Strategy:** 5-fold cross-validation on the training set, optimizing for accuracy.
- **Selected Features:** 14 features including edge density, color statistics in HSV and LAB spaces, shape descriptors (area, perimeter, eccentricity), and the appended GMM cluster label.
- **Selection Performance:** The SFS achieved a cross-validation accuracy of **0.9443** on the training folds.

---

5. KNN Hyperparameter Tuning

- **Parameter Grid:**
  - n_neighbors: [4, 5, 7, 10]
  - weights: [uniform, distance]
  - p (Minkowski exponent): [1, 2]
- **Grid Search:** 5-fold cross-validation on the selected 14-feature training set.
- **Optimal Parameters:**
  - n_neighbors = 4
  - p = 1 (Manhattan distance)
  - weights = distance
- **Best Training Accuracy: 0.9592** (mean cross-validation).

---

6. Model Evaluation

6.1 Validation Cross-Validation

- Performed 5-fold cross-validation on the held-out validation set using the tuned KNN.
- **Mean Accuracy:** 0.9172
- **Standard Deviation:** ± 0.0330

6.2 Hold-out Validation Metrics

- **Accuracy on Validation Set:** 0.9223
- **Confusion Matrix & ROC Curves:**
  - Visual inspection confirms high true-positive rates across most classes.
  - Per-class AUC scores (each > 0.90) indicate strong discriminative power.

6.3 Training-CV (Out-of-Fold) Metrics

- **Confusion Matrix (O-of-F Predictions):** Shows consistency between fold-level performance and hold-out results.
- **ROC Curves & AUC:** Each class achieves AUC > 0.90, validating that the model generalizes well beyond the training folds.

---

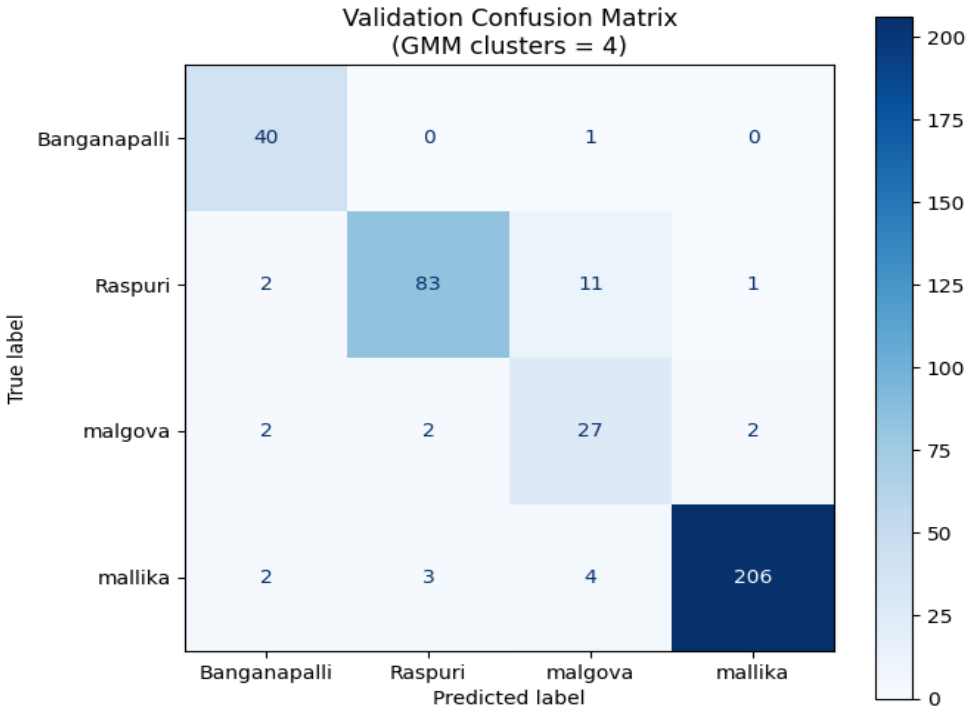
7. Model Persistence & Inference

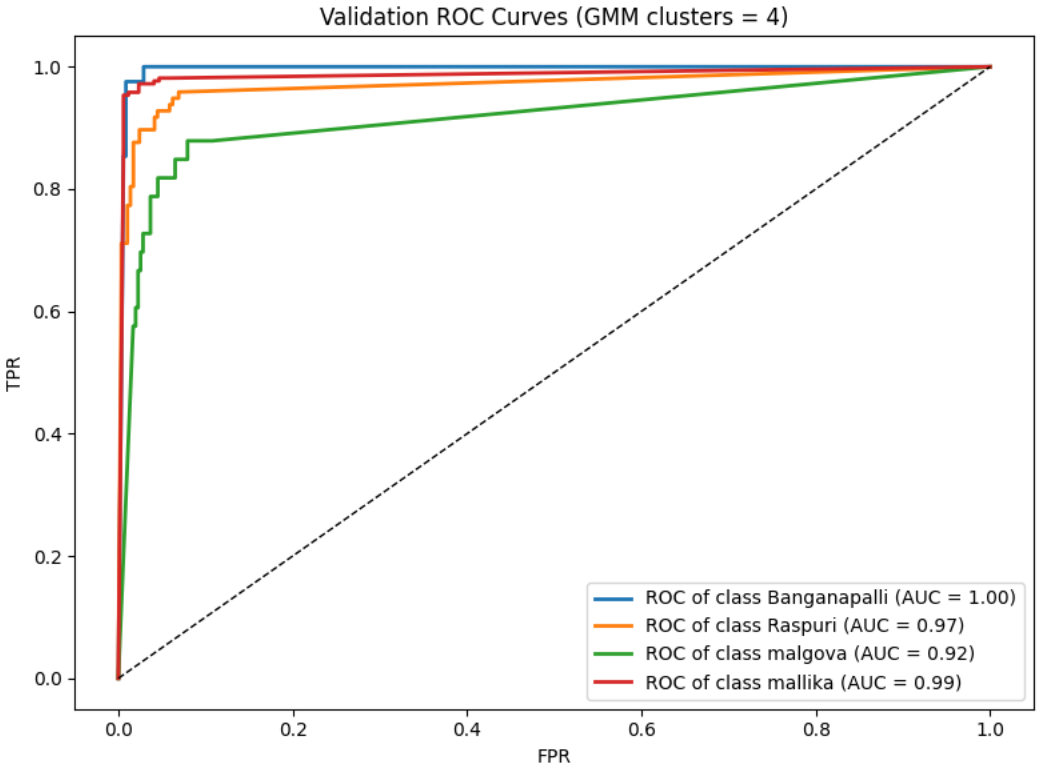
- **Saved Components:**
  - `StandardScaler`
  - Trained GMM
  - Tuned KNN classifier
- **Inference Function:**
  1. Load saved components and metadata.
  2. Scale raw inputs, predict GMM clusters, augment features.
  3. Subset to selected features and predict with KNN.
  4. Optionally compute accuracy and display confusion matrix if true labels are provided.
- **Inference Accuracy on Validation Data: 0.9223**, matching the hold-out evaluation.

---

8. Conclusions

- **Effectiveness of GMM Augmentation:** Incorporating unsupervised cluster labels into the feature set improved classification performance, as evidenced by high accuracy and AUC.
- **Robust Feature Selection:** Sequential forward selection identified a compact set of 14 highly informative features, balancing dimensionality reduction with predictive power.
- **Strong Generalization:** Consistent metrics across training-CV, validation-CV, and hold-out sets demonstrate the pipeline's reliability.





Results:
Train accuracy : 95.92%

Validation accuracy: 92.23%

KMeans-KNN Classification with Feature Selection

The primary goal of this implementation is to classify mango varieties using a hybrid machine learning pipeline that combines **unsupervised clustering (KMeans)** and **supervised classification (K-Nearest Neighbors)**. The pipeline includes **feature standardization**, **feature selection**, **model training**, and **evaluation** on both training and test datasets.

---

2. Data Preparation

- **Input Data**: Two CSV files representing the **training** and **test** datasets.
- **Preprocessing**:
  - Dropping unused columns (image_name).
  - Removing samples belonging to the class **'malgova'**, likely due to insufficient samples or irrelevance.
  - Separating features (X_train, X_test) and target labels (y_train, y_test).

---

3. Pipeline Architecture

The model uses a **scikit-learn Pipeline** comprising:

1. **StandardScaler**: Normalizes feature values.
2. **KMeans (n_clusters=4)**: Performs unsupervised clustering; likely acts as an embedding or cluster-based transformation layer.
3. **KNeighborsClassifier (n_neighbors=5)**: Performs the final classification based on clustered and scaled feature space.

---

4. Feature Selection

- The **Sequential Forward Selection (SFS)** method from mlxtend is applied to the pipeline.
- SFS iteratively selects the best feature subset (up to 19 features) based on **cross-validated accuracy** (cv=5).
- Output1 indicates that **8 features** were selected:
  bash
  CopyEdit
  ['edge_density', 'HSV_S_mean', 'HSV_S_std', 'HSV_V_std',
  'LAB_L_mean', 'LAB_A_mean', 'LAB_B_std', 'area']
  These features span texture (edge_density), color statistics in HSV and LAB spaces, and geometric properties (area), indicating a well-rounded feature representation for visual classification.

---

5. Model Evaluation

- **Training Accuracy**: 85.71%
- **Test Accuracy**: 82.49%
  This minor performance gap suggests that the model generalizes well and is **not significantly overfitting**. The pipeline effectively retains classification power on unseen validation data.

---

6. Visualization and Metrics

- **Confusion Matrix** is generated for multi-class evaluation using ConfusionMatrixDisplay.
- **ROC AUC Curve** generation is **conditional** on binary classification. Since more than two classes are present, it is **skipped**, which is appropriate.

---

7. Model Persistence

- The trained pipeline (pipe) and the selected feature list are saved using joblib, ensuring **reproducibility** and **reusability** for deployment or future inference.

---

8. Highlights and Strengths

- **Pipeline Design**: Combining clustering and classification is a unique and potentially powerful approach, particularly when class boundaries are not linearly separable.
- **Feature Selection**: Use of wrapper-based SFS allows the pipeline to identify features that enhance model performance in combination, rather than individually.
- **Cross-Validation**: Ensures robustness of selected features and model performance.
- **Reproducibility**: The use of joblib for saving models and features aligns with best practices.
- **Visualization**: Clear confusion matrix support enhances model interpretability.

---

9. Recommendations

- **Multiclass ROC AUC**: Consider using **macro or micro-averaged ROC AUC** with `OneVsRestClassifier` for future extensions.
- **Model Variants**: Experiment with alternative classifiers (e.g., SVM, RandomForest) after KMeans clustering to assess comparative performance.
- **Hyperparameter Tuning**: Consider automated tuning for `n_clusters` and `k_neighbors`.

---

9. Conclusion

This implementation demonstrates a **thoughtful blend of unsupervised and supervised learning**, leveraging robust preprocessing, feature selection, and evaluation techniques. With a **test accuracy of 82.49%**, the pipeline shows requirements of better clustering methods and hyper-parameter tuning techniques.