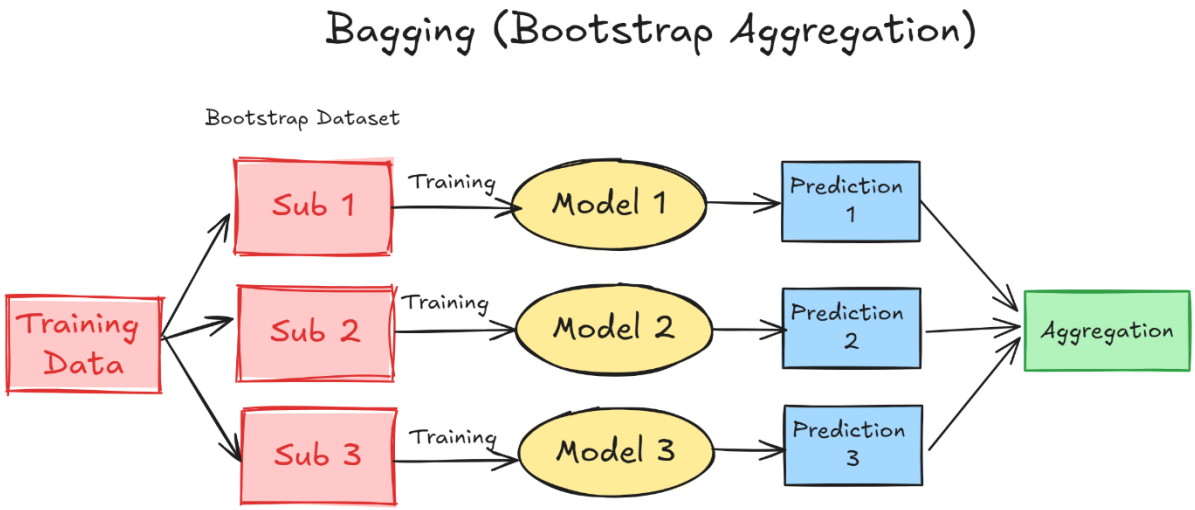
Classification Performance Report: Raisin Dataset Analysis



JUNAID

Research Scholar - Data Science

Classification Performance Report: Raisin Dataset Analysis

1. Introduction

This report provides an comparative analysis of two machine learning models – a Support Vector Classifier (SVC) and a Voting Classifier (an ensemble method) – applied to a dataset containing features of raisins. The objective is to classify raisins into "Besni" or "Kecimen" varieties, evaluating the performance of each model on a consistent test set and understanding their respective strengths and weaknesses.

2. Dataset Overview

The dataset consists of various geometric properties of raisins. A few sample rows illustrate its structure:

Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	Extent	Perimeter	Class
3086	709.33	53.109	0.9872	243.001	0.34	0.7255	Kecimen
4668	383.07	692.97	1.0340	0.7606	929.72	1Kecimen	6335
4357	362.59	471.81	0.9294	936.70	0.8466	5956	Besni
0060	0.6999	189.45	828Besni	3876	509.13	92.493	Kecimen
2712	16.827	572.08	0.8335	5569	597.06	641.94	1079
752K	ecimen	772.66	265.42	4.5594	352.03	381.29	Besni
203.3	812	92.087	793.70	487.06	280.15	1126	765Besni
6739	917.74	94.055	192.26	0.9736	480.84	9103	1044
680.6	141	751.31	765Besni				

The target variable is Class, which categorizes the raisins as either "Besni" or "Kecimen".

For both the SVC and the Voting Classifier evaluations, a **consistent test set of 40 samples** was used. The class distribution in this test set is:

- Class '0': 19 samples
- Class '1': 21 samples (These numerical labels are assumed to correspond to "Besni" and "Kecimen" varieties.)

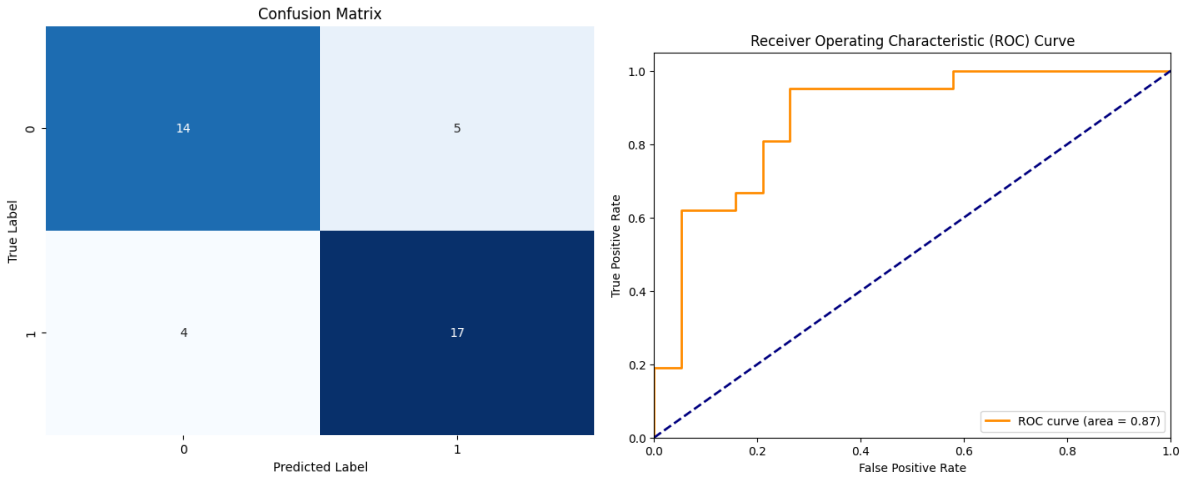
3. Model Architectures

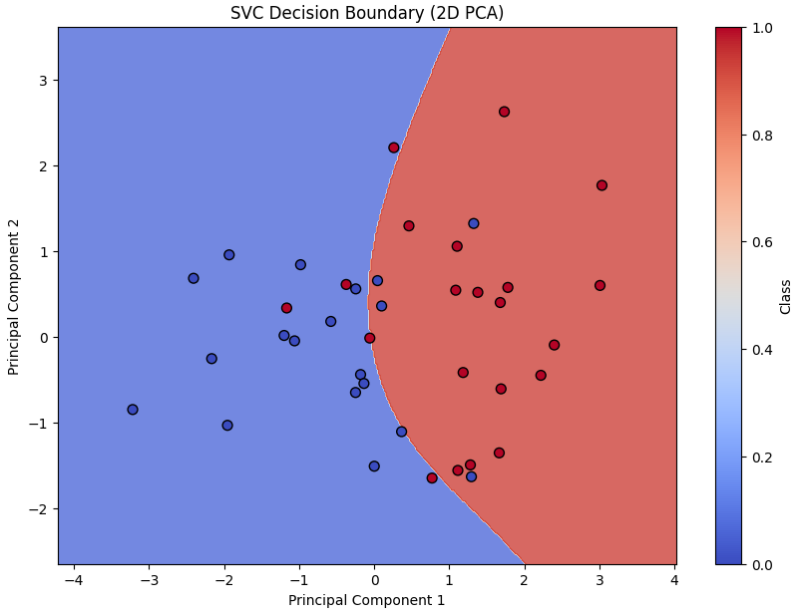
- **Support Vector Classifier (SVC):** A standalone SVC model was used, likely with a Radial Basis Function (RBF) kernel.
- **Voting Classifier (Ensemble Model):** This model combines the predictions of three individual base estimators:
    - **Logistic Regression (log_model):** A linear model often used as a baseline for classification.
    - **Support Vector Classifier (svc_model):** Another SVC model, contributing its non-linear classification capabilities.
    - **Decision Tree Classifier (dt_model):** A tree-based model known for its ability to capture complex decision rules. The VotingClassifier was configured with voting='hard', meaning it makes predictions based on the majority vote of its constituent classifiers.

4. Performance Evaluation

The performance of both models was evaluated using standard classification metrics: Precision, Recall, F1-score, and Accuracy, all on the same 40-sample test set.

4.1. SVC Model results





The SVC model's performance on the 40-sample test set:

Class	Precision	Recall	F1-score	Support
0	0.78	0.74	0.76	19
1	0.77	0.81	0.79	21
Accuracy	**0.78**			**40**
Macro Avg	0.78	0.77	0.77	40
Weighted Avg	0.78	0.78	0.77	40

Analysis of SVC Results:

- **Overall Accuracy:** The SVC model achieved an accuracy of **78%**.
- **Class 0 Performance:** For Class 0, the model has a precision of 0.78 and a recall of 0.74, resulting in an F1-score of 0.76.
- **Class 1 Performance:** For Class 1, the model shows a precision of 0.77 and a higher recall of 0.81, with an F1-score of 0.79.
- **Balance:** The model's performance is relatively balanced across both classes, with a slight edge in recalling Class 1 instances.

4.2. Voting Classifier Results

The Voting Classifier's performance on the 40-sample test set:

Class	Precision	Recall	F1-score	Support
0	0.88	0.79	0.83	19
1	0.83	0.90	0.86	21
Accuracy	**0.85**			**40**
Macro Avg	0.85	0.85	0.85	40
Weighted Avg	0.85	0.85	0.85	40

Analysis of Voting Classifier Results:

- **Overall Accuracy:** The Voting Classifier achieved a higher accuracy of **85%**.
- **Class 0 Performance:** For Class 0, it achieved a precision of 0.88 and a recall of 0.79, leading to an F1-score of 0.83.
- **Class 1 Performance:** For Class 1, it demonstrated a precision of 0.83 and a strong recall of 0.90, with an F1-score of 0.86.
- **Improvement:** Compared to the standalone SVC, the Voting Classifier shows clear improvements across all metrics for both classes.

5. Comparative Analysis

Metric / Model	Standalone SVC (40 samples)	Voting Classifier (40 samples)
Overall Accuracy	78%	**85%**
Class 0 (Besni)	Precision: 0.78, Recall: 0.74, F1: 0.76	Precision: **0.88**, Recall: **0.79**, F1: **0.83**
Class 1 (Kecimen)	Precision: 0.77, Recall: 0.81, F1: 0.79	Precision: **0.83**, Recall: **0.90**, F1: **0.86**
Test Set Size	Consistent (40 samples)	Consistent (40 samples)

Conclusion from Comparison:

This evaluation, performed on a consistent test set, clearly demonstrates that the **Voting Classifier significantly outperforms the standalone SVC model.** The ensemble approach, by combining the predictions from Logistic Regression, SVC, and Decision Tree, has effectively boosted the classification accuracy by **7 percentage points** (from 78% to 85%). This improvement is also reflected in the higher precision, recall, and F1-scores for both individual classes, indicating a more robust and effective model.

The Voting Classifier's enhanced performance suggests that the ensemble successfully leverages the diverse strengths of its base learners, leading to a more generalized and accurate predictive model for this dataset.

6. Conclusion and Recommendations

- The **Voting Classifier is the superior model** for this raisin classification task based on the current evaluation, achieving a robust 85% accuracy on the test set. Its ensemble nature effectively leverages the diverse perspectives of its base learners to improve overall performance compared to the standalone SVC.
- The results underscore the value of ensemble methods in machine learning, particularly in situations where individual models may have different strengths or weaknesses.

Recommendations for further investigation:

1. **Soft Voting:** While hard voting proved effective, consider experimenting with voting='soft' for the VotingClassifier. This often yields even better results by weighting predictions based on the confidence (probabilities) of individual base models. (Note: Ensure base estimators like SVC have probability=True enabled for soft voting).
2. **Hyperparameter Tuning:** Systematically tune the hyperparameters of all individual models (log_model, svc_model, dt_model) within the Voting Classifier to further optimize their individual contributions and the ensemble's overall performance.
3. **Cross-Validation:** Implement k-fold cross-validation on the entire dataset to obtain a more reliable and less volatile estimate of the models' performance, especially given the relatively small absolute size of the current test set.
4. **Feature Engineering/Importance:** Explore feature engineering techniques or analyze feature importance (if applicable to the chosen models) to gain deeper insights into which raisin properties are most crucial for accurate classification.
5. **Error Analysis:** Perform detailed error analysis on misclassified samples to identify patterns and potential areas for model improvement.