ML Lab – Lab 4 Report

Project Title: Hyperparameter Tuning and Model Comparison – Manual Search vs. GridSearchCV

Name: Ninad Chavan

SRN: PES2UG23CS392
Course: ML Lab
Submission Date: 31-08-2025

---

1. Introduction

The experiment aimed to explore hyperparameter tuning using both a manual grid search implementation and scikit-learn's GridSearchCV. The comparison was carried out on classification algorithms including Decision Tree, k-Nearest Neighbors (kNN), and Logistic Regression. In addition, an ensemble model using soft voting was tested to evaluate if combining classifiers could enhance predictive accuracy.

---

2. Dataset Description

Two datasets were considered:

- **Wine Quality Dataset**
  - Instances: 1599 (split into training and testing sets)
  - Features: 11 chemical attributes such as acidity, sugar, and alcohol.
  - Target: Binary classification of wine quality (good or not).
- **Banknote Authentication Dataset**
  - Instances: 1372
  - Features: 4 statistical features derived from banknote images.
  - Target: Binary classification of authentic vs fake notes.

---

3. Methodology

- **Hyperparameter Tuning:**
  Explored optimal parameters (e.g., max_depth for Decision Tree, n_neighbors for kNN, regularization strength for Logistic Regression).

- **Manual Grid Search:**
  Generated parameter combinations, applied stratified 5-fold cross-validation, and evaluated performance using ROC AUC.
- **GridSearchCV:**
  Automated hyperparameter tuning using the same parameter grids and evaluation metrics.
- **Pipeline:**
  - StandardScaler for feature scaling
  - SelectKBest for feature selection
  - Classifier (Decision Tree / kNN / Logistic Regression)
- **Voting Classifier:**
  Combined tuned models using soft voting for final evaluation.

---

4. Results and Analysis

Wine Quality Dataset

- Decision Tree: Accuracy around **0.72–0.73**, ROC AUC ≈ **0.80**
- kNN: Accuracy ≈ **0.78**, ROC AUC ≈ **0.87**
- Logistic Regression: Accuracy ≈ **0.73**, ROC AUC ≈ **0.82**
- Voting Classifier: Accuracy ≈ **0.77**, ROC AUC ≈ **0.86**

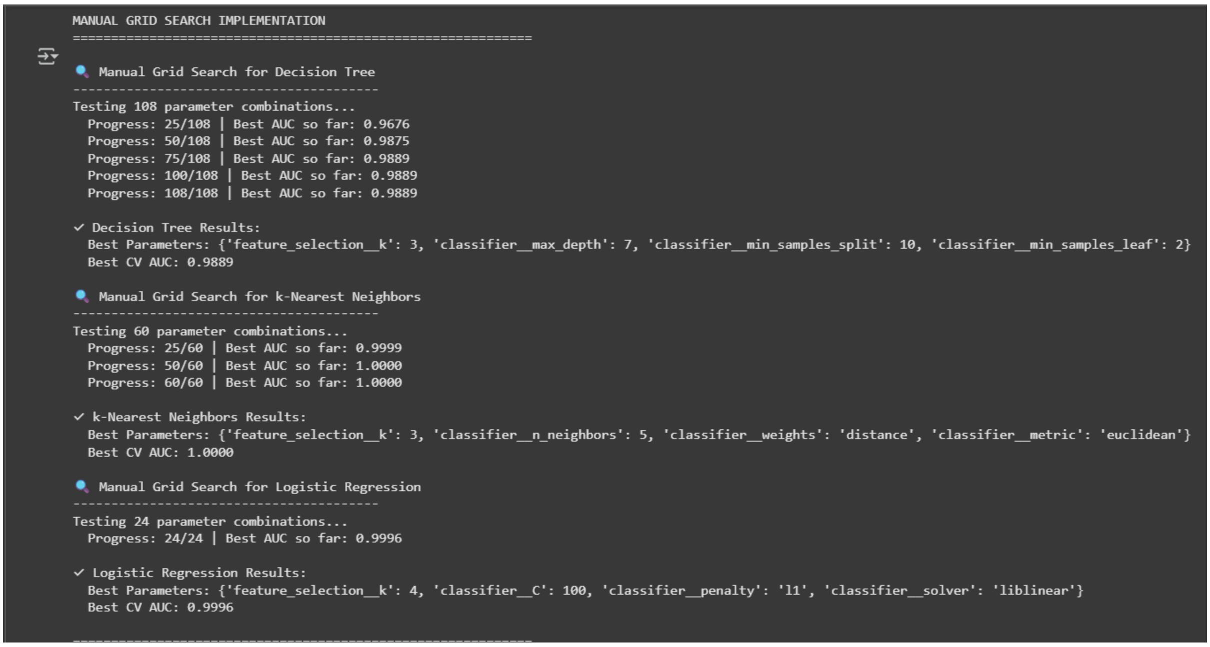
Banknote Authentication Dataset

- Decision Tree: Accuracy ≈ **0.98**, ROC AUC ≈ **0.99**
- kNN: Accuracy = **1.0**, ROC AUC = **1.0**
- Logistic Regression: Accuracy ≈ **0.99**, ROC AUC ≈ **0.999**
- Voting Classifier: Accuracy ≈ **0.997**, ROC AUC ≈ **1.0**

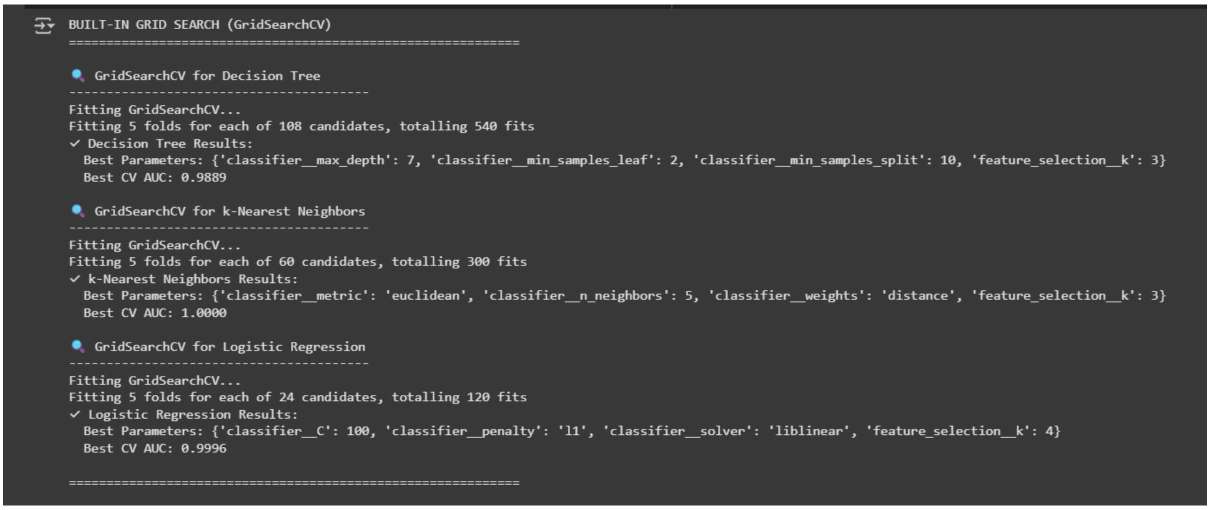
Comparative Observations

- Manual grid search and GridSearchCV produced identical outcomes across models.
- Voting Classifier slightly improved performance on Wine dataset but had no effect on Banknote dataset where kNN already achieved perfect results.
- ROC curves confirmed the numerical scores, with Banknote models showing near-perfect separation.

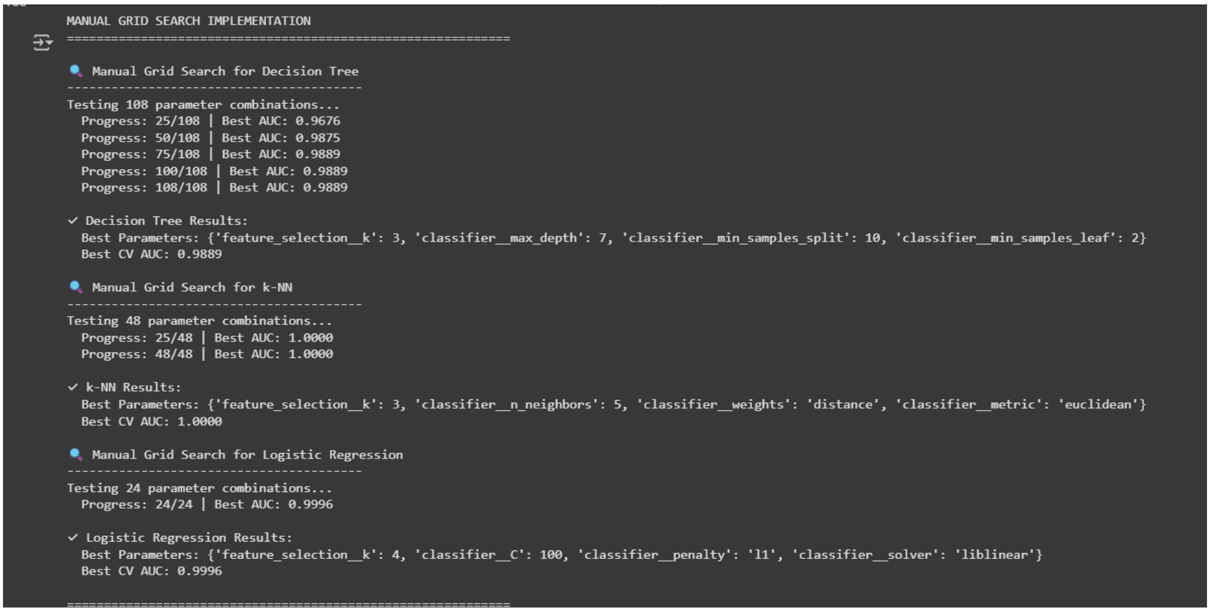
5. Screenshots



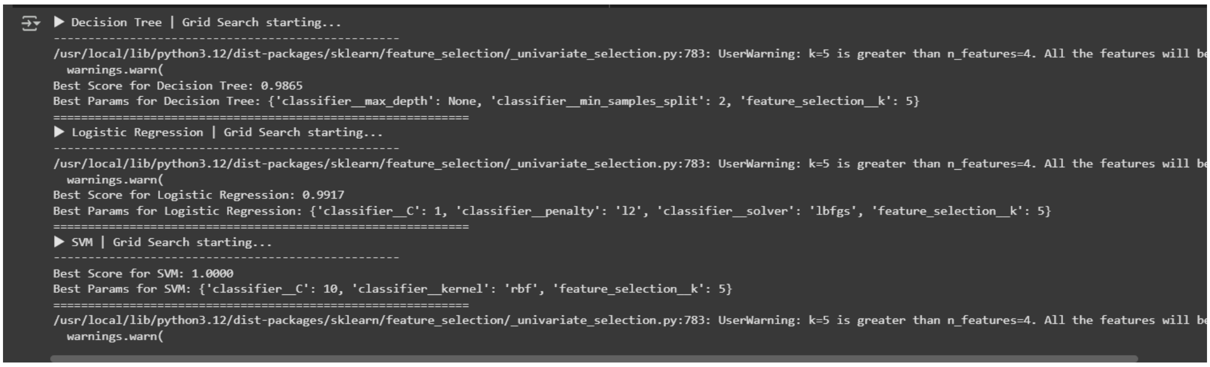
```
MANUAL GRID SEARCH IMPLEMENTATION
============================================================

🔍 Manual Grid Search for Decision Tree
----------------------------------------
Testing 108 parameter combinations...
  Progress: 25/108 | Best AUC so far: 0.9676
  Progress: 50/108 | Best AUC so far: 0.9875
  Progress: 75/108 | Best AUC so far: 0.9889
  Progress: 100/108 | Best AUC so far: 0.9889
  Progress: 108/108 | Best AUC so far: 0.9889

✓ Decision Tree Results:
  Best Parameters: {'feature_selection__k': 3, 'classifier__max_depth': 7, 'classifier__min_samples_split': 10, 'classifier__min_samples_leaf': 2}
  Best CV AUC: 0.9889

🔍 Manual Grid Search for k-Nearest Neighbors
----------------------------------------
Testing 60 parameter combinations...
  Progress: 25/60 | Best AUC so far: 0.9999
  Progress: 50/60 | Best AUC so far: 1.0000
  Progress: 60/60 | Best AUC so far: 1.0000

✓ k-Nearest Neighbors Results:
  Best Parameters: {'feature_selection__k': 3, 'classifier__n_neighbors': 5, 'classifier__weights': 'distance', 'classifier__metric': 'euclidean'}
  Best CV AUC: 1.0000

🔍 Manual Grid Search for Logistic Regression
----------------------------------------
Testing 24 parameter combinations...
  Progress: 24/24 | Best AUC so far: 0.9996

✓ Logistic Regression Results:
  Best Parameters: {'feature_selection__k': 4, 'classifier__C': 100, 'classifier__penalty': 'l1', 'classifier__solver': 'liblinear'}
  Best CV AUC: 0.9996

============================================================
```



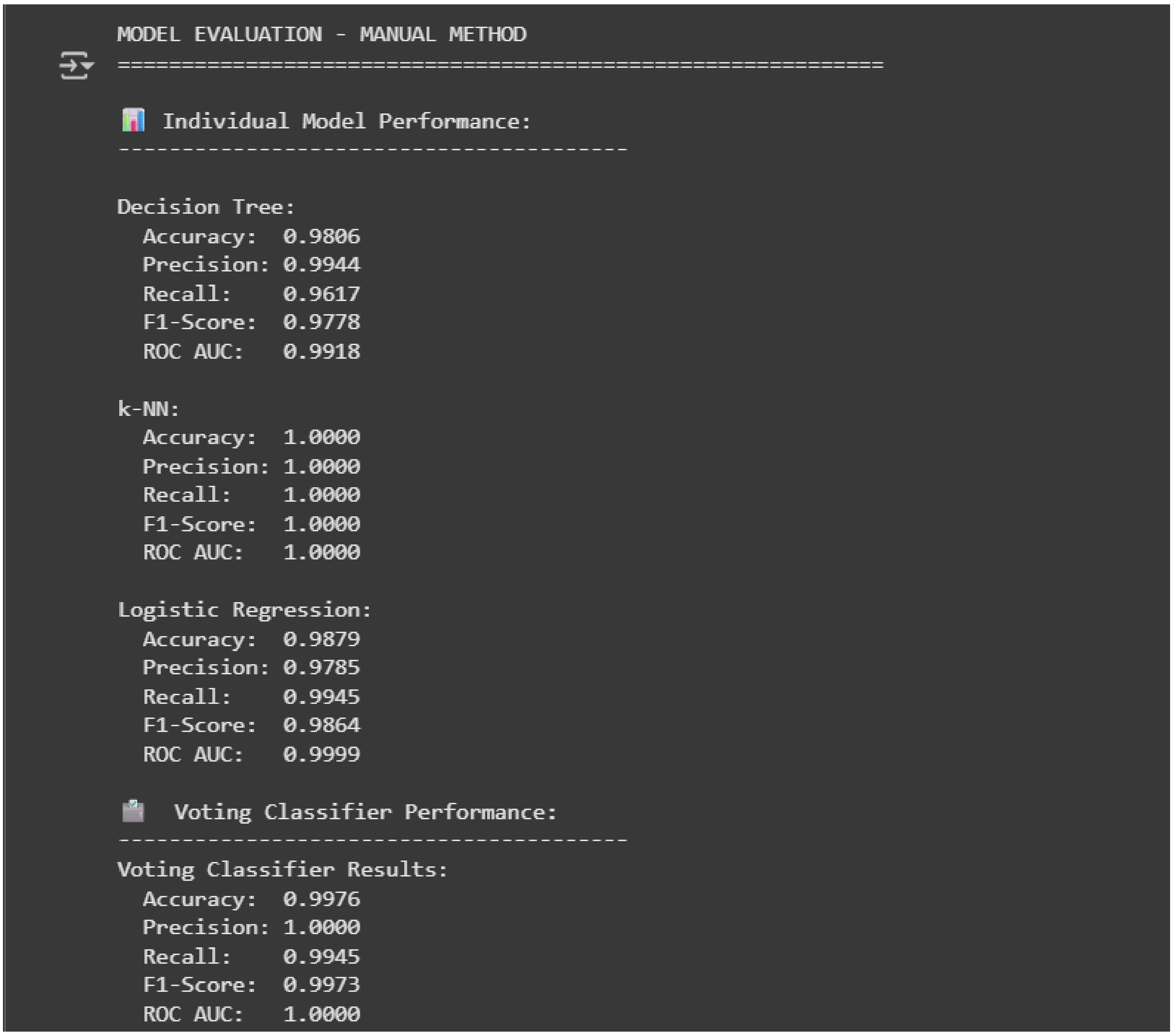
```
BUILT-IN GRID SEARCH (GridSearchCV)
============================================================

🔍 GridSearchCV for Decision Tree
----------------------------------------
Fitting GridSearchCV...
Fitting 5 folds for each of 108 candidates, totalling 540 fits
✓ Decision Tree Results:
  Best Parameters: {'classifier__max_depth': 7, 'classifier__min_samples_leaf': 2, 'classifier__min_samples_split': 10, 'feature_selection__k': 3}
  Best CV AUC: 0.9889

🔍 GridSearchCV for k-Nearest Neighbors
----------------------------------------
Fitting GridSearchCV...
Fitting 5 folds for each of 60 candidates, totalling 300 fits
✓ k-Nearest Neighbors Results:
  Best Parameters: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 5, 'classifier__weights': 'distance', 'feature_selection__k': 3}
  Best CV AUC: 1.0000

🔍 GridSearchCV for Logistic Regression
----------------------------------------
Fitting GridSearchCV...
Fitting 5 folds for each of 24 candidates, totalling 120 fits
✓ Logistic Regression Results:
  Best Parameters: {'classifier__C': 100, 'classifier__penalty': 'l1', 'classifier__solver': 'liblinear', 'feature_selection__k': 4}
  Best CV AUC: 0.9996

============================================================
```



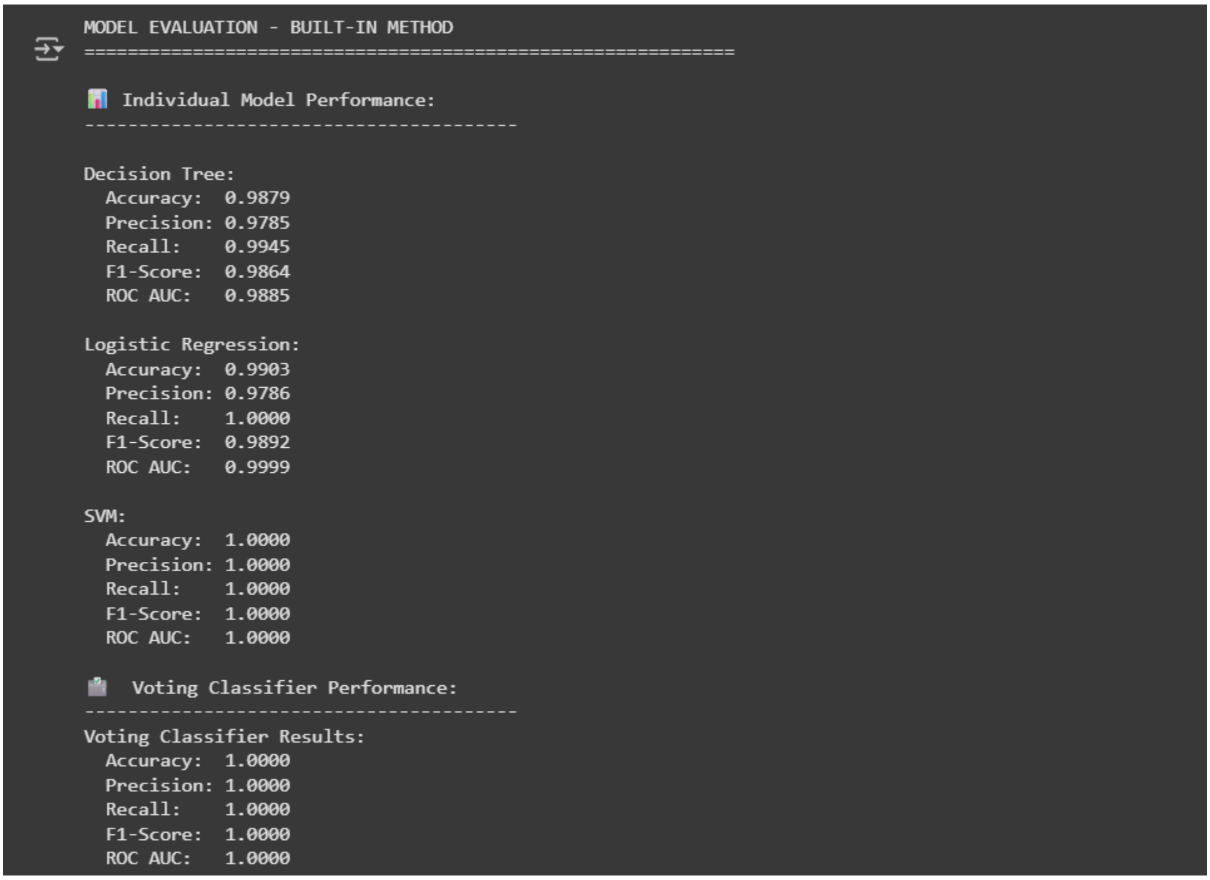
```
MANUAL GRID SEARCH IMPLEMENTATION
============================================================

🔍 Manual Grid Search for Decision Tree
----------------------------------------
Testing 108 parameter combinations...
  Progress: 25/108 | Best AUC: 0.9676
  Progress: 50/108 | Best AUC: 0.9875
  Progress: 75/108 | Best AUC: 0.9889
  Progress: 100/108 | Best AUC: 0.9889
  Progress: 108/108 | Best AUC: 0.9889

✓ Decision Tree Results:
  Best Parameters: {'feature_selection__k': 3, 'classifier__max_depth': 7, 'classifier__min_samples_split': 10, 'classifier__min_samples_leaf': 2}
  Best CV AUC: 0.9889

🔍 Manual Grid Search for k-NN
----------------------------------------
Testing 48 parameter combinations...
  Progress: 25/48 | Best AUC: 1.0000
  Progress: 48/48 | Best AUC: 1.0000

✓ k-NN Results:
  Best Parameters: {'feature_selection__k': 3, 'classifier__n_neighbors': 5, 'classifier__weights': 'distance', 'classifier__metric': 'euclidean'}
  Best CV AUC: 1.0000

🔍 Manual Grid Search for Logistic Regression
----------------------------------------
Testing 24 parameter combinations...
  Progress: 24/24 | Best AUC: 0.9996

✓ Logistic Regression Results:
  Best Parameters: {'feature_selection__k': 4, 'classifier__C': 100, 'classifier__penalty': 'l1', 'classifier__solver': 'liblinear'}
  Best CV AUC: 0.9996

============================================================
```



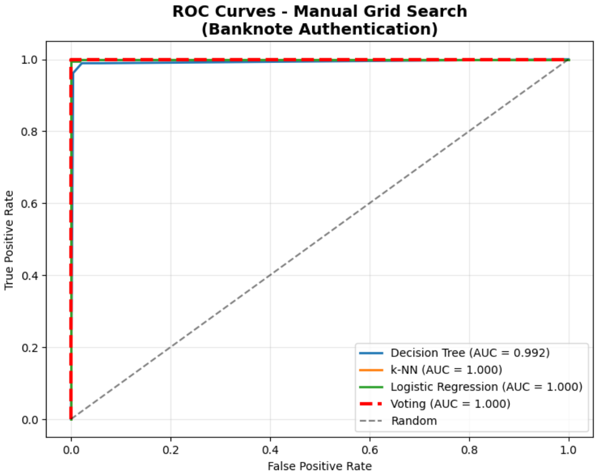
```
▶ Decision Tree | Grid Search starting...
--------------------------------------------------
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:783: UserWarning: k=5 is greater than n_features=4. All the features will be
  warnings.warn(
Best Score for Decision Tree: 0.9865
Best Params for Decision Tree: {'classifier__max_depth': None, 'classifier__min_samples_split': 2, 'feature_selection__k': 5}
============================================================
▶ Logistic Regression | Grid Search starting...
--------------------------------------------------
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:783: UserWarning: k=5 is greater than n_features=4. All the features will be
  warnings.warn(
Best Score for Logistic Regression: 0.9917
Best Params for Logistic Regression: {'classifier__C': 1, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs', 'feature_selection__k': 5}
============================================================
▶ SVM | Grid Search starting...
--------------------------------------------------
Best Score for SVM: 1.0000
Best Params for SVM: {'classifier__C': 10, 'classifier__kernel': 'rbf', 'feature_selection__k': 5}
============================================================
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:783: UserWarning: k=5 is greater than n_features=4. All the features will be
  warnings.warn(
```

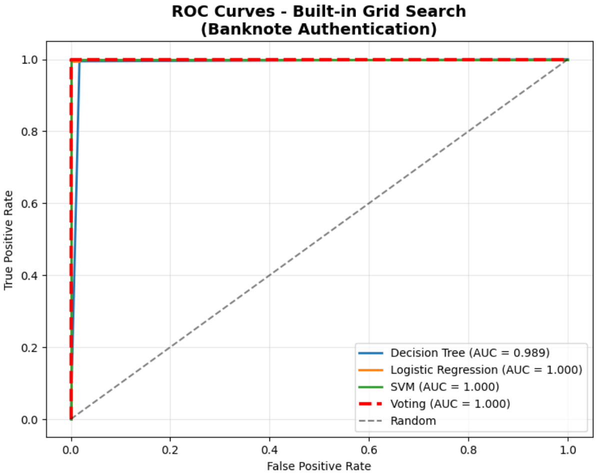


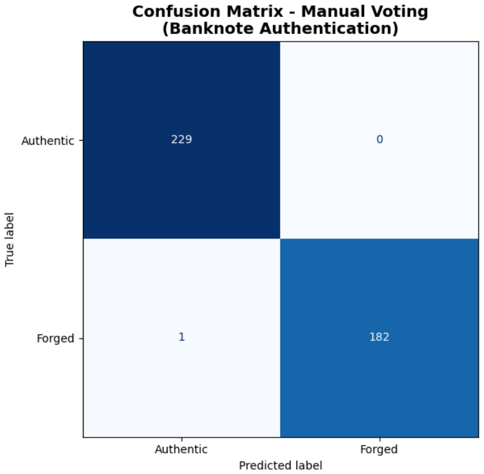
```
MODEL EVALUATION - MANUAL METHOD
============================================================

📊 Individual Model Performance:
----------------------------------------

Decision Tree:
  Accuracy:  0.9806
  Precision: 0.9944
  Recall:    0.9617
  F1-Score:  0.9778
  ROC AUC:   0.9918

k-NN:
  Accuracy:  1.0000
  Precision: 1.0000
  Recall:    1.0000
  F1-Score:  1.0000
  ROC AUC:   1.0000

Logistic Regression:
  Accuracy:  0.9879
  Precision: 0.9785
  Recall:    0.9945
  F1-Score:  0.9864
  ROC AUC:   0.9999

🗳️  Voting Classifier Performance:
----------------------------------------
Voting Classifier Results:
  Accuracy:  0.9976
  Precision: 1.0000
  Recall:    0.9945
  F1-Score:  0.9973
  ROC AUC:   1.0000
```

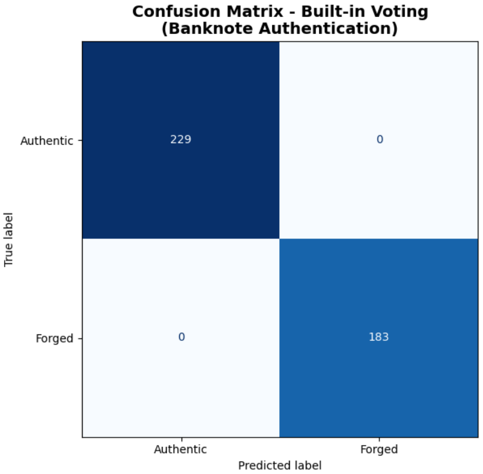


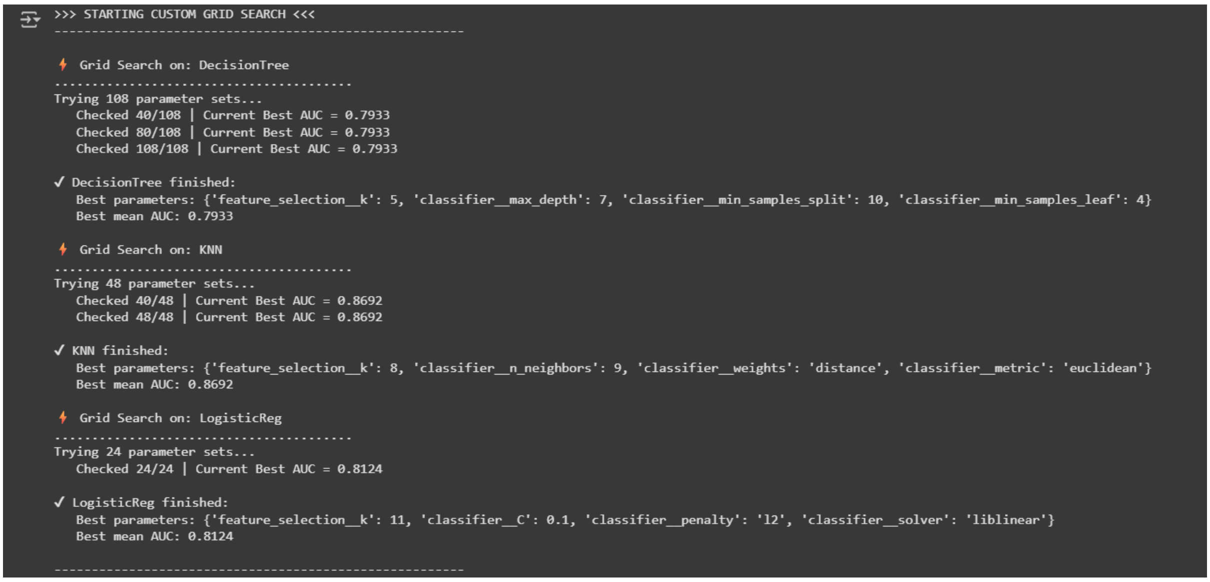
```
MODEL EVALUATION - BUILT-IN METHOD
============================================================

📊 Individual Model Performance:
----------------------------------------

Decision Tree:
  Accuracy:  0.9879
  Precision: 0.9785
  Recall:    0.9945
  F1-Score:  0.9864
  ROC AUC:   0.9885

Logistic Regression:
  Accuracy:  0.9903
  Precision: 0.9786
  Recall:    1.0000
  F1-Score:  0.9892
  ROC AUC:   0.9999

SVM:
  Accuracy:  1.0000
  Precision: 1.0000
  Recall:    1.0000
  F1-Score:  1.0000
  ROC AUC:   1.0000

📋 Voting Classifier Performance:
----------------------------------------
Voting Classifier Results:
  Accuracy:  1.0000
  Precision: 1.0000
  Recall:    1.0000
  F1-Score:  1.0000
  ROC AUC:   1.0000
```



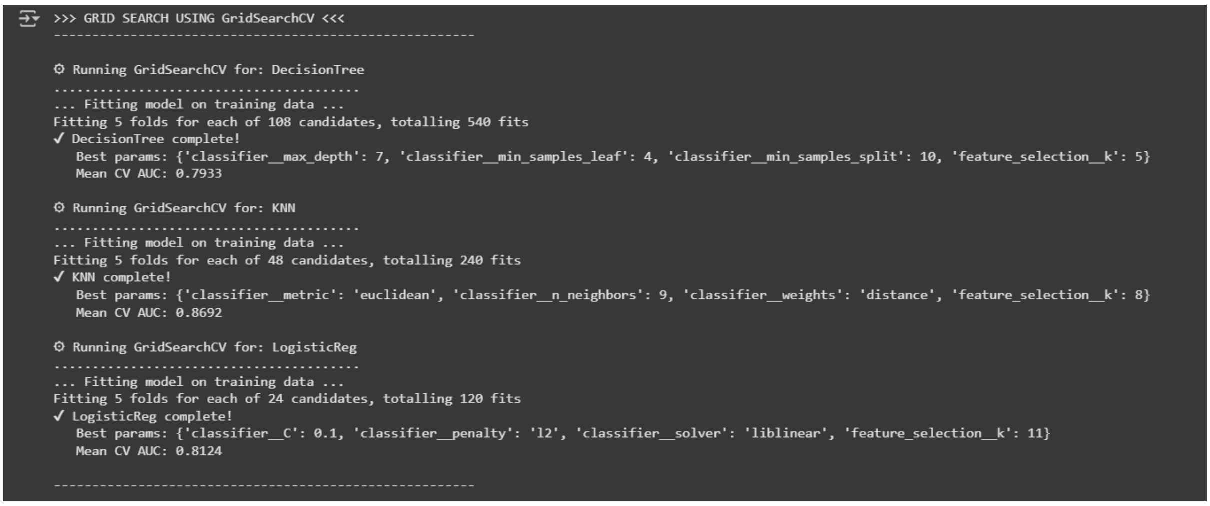




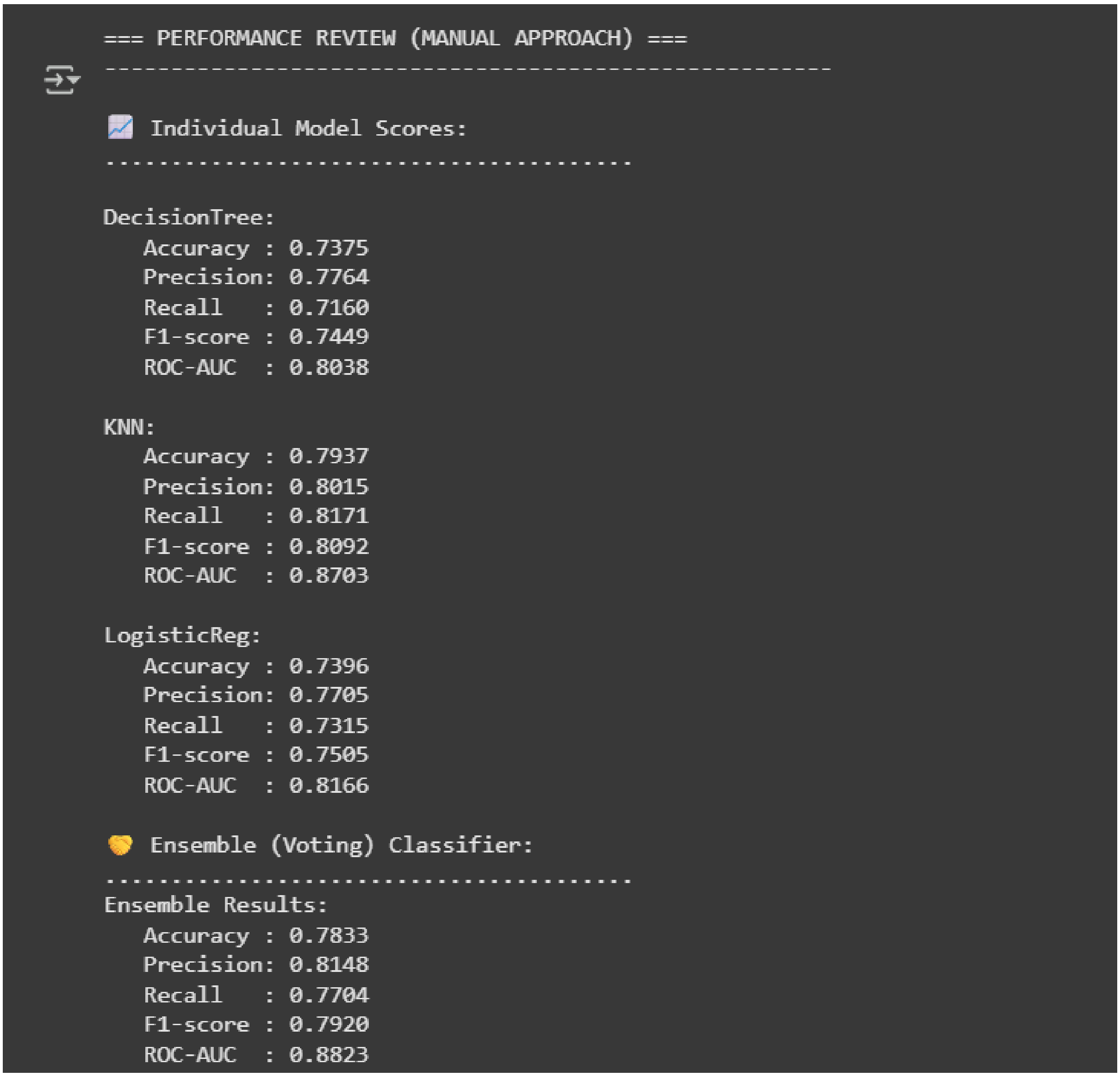




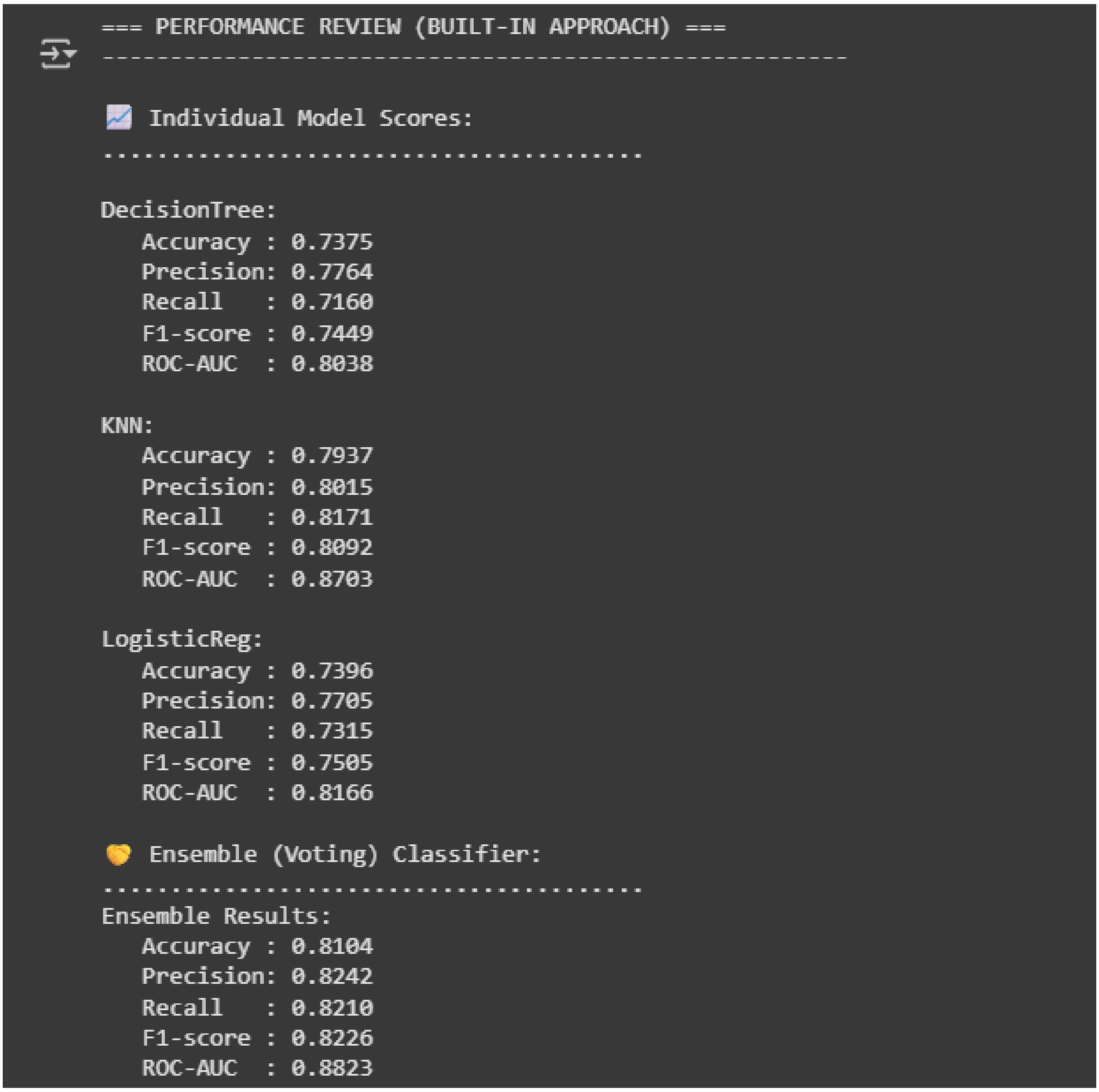
```
>>> STARTING CUSTOM GRID SEARCH <<<
------------------------------------------------------

⚡ Grid Search on: DecisionTree
........................................
Trying 108 parameter sets...
   Checked 40/108 | Current Best AUC = 0.7933
   Checked 80/108 | Current Best AUC = 0.7933
   Checked 108/108 | Current Best AUC = 0.7933

✓ DecisionTree finished:
   Best parameters: {'feature_selection__k': 5, 'classifier__max_depth': 7, 'classifier__min_samples_split': 10, 'classifier__min_samples_leaf': 4}
   Best mean AUC: 0.7933

⚡ Grid Search on: KNN
........................................
Trying 48 parameter sets...
   Checked 40/48 | Current Best AUC = 0.8692
   Checked 48/48 | Current Best AUC = 0.8692

✓ KNN finished:
   Best parameters: {'feature_selection__k': 8, 'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'classifier__metric': 'euclidean'}
   Best mean AUC: 0.8692

⚡ Grid Search on: LogisticReg
........................................
Trying 24 parameter sets...
   Checked 24/24 | Current Best AUC = 0.8124

✓ LogisticReg finished:
   Best parameters: {'feature_selection__k': 11, 'classifier__C': 0.1, 'classifier__penalty': 'l2', 'classifier__solver': 'liblinear'}
   Best mean AUC: 0.8124

------------------------------------------------------
```



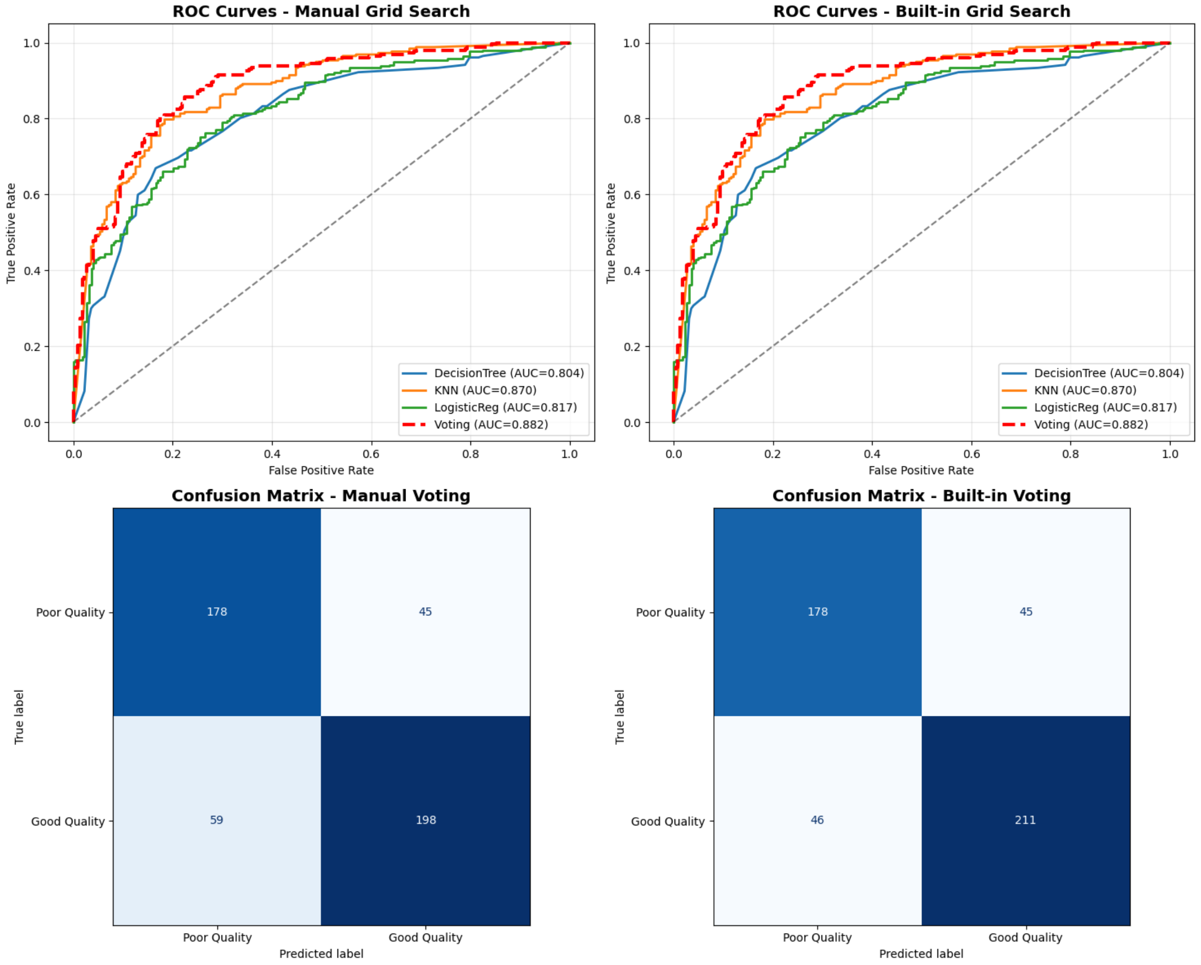
```
>>> GRID SEARCH USING GridSearchCV <<<
--------------------------------------------------------

⚙ Running GridSearchCV for: DecisionTree
.........................................
... Fitting model on training data ...
Fitting 5 folds for each of 108 candidates, totalling 540 fits
✓ DecisionTree complete!
   Best params: {'classifier__max_depth': 7, 'classifier__min_samples_leaf': 4, 'classifier__min_samples_split': 10, 'feature_selection__k': 5}
   Mean CV AUC: 0.7933

⚙ Running GridSearchCV for: KNN
.........................................
... Fitting model on training data ...
Fitting 5 folds for each of 48 candidates, totalling 240 fits
✓ KNN complete!
   Best params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'feature_selection__k': 8}
   Mean CV AUC: 0.8692

⚙ Running GridSearchCV for: LogisticReg
.........................................
... Fitting model on training data ...
Fitting 5 folds for each of 24 candidates, totalling 120 fits
✓ LogisticReg complete!
   Best params: {'classifier__C': 0.1, 'classifier__penalty': 'l2', 'classifier__solver': 'liblinear', 'feature_selection__k': 11}
   Mean CV AUC: 0.8124

--------------------------------------------------------
```

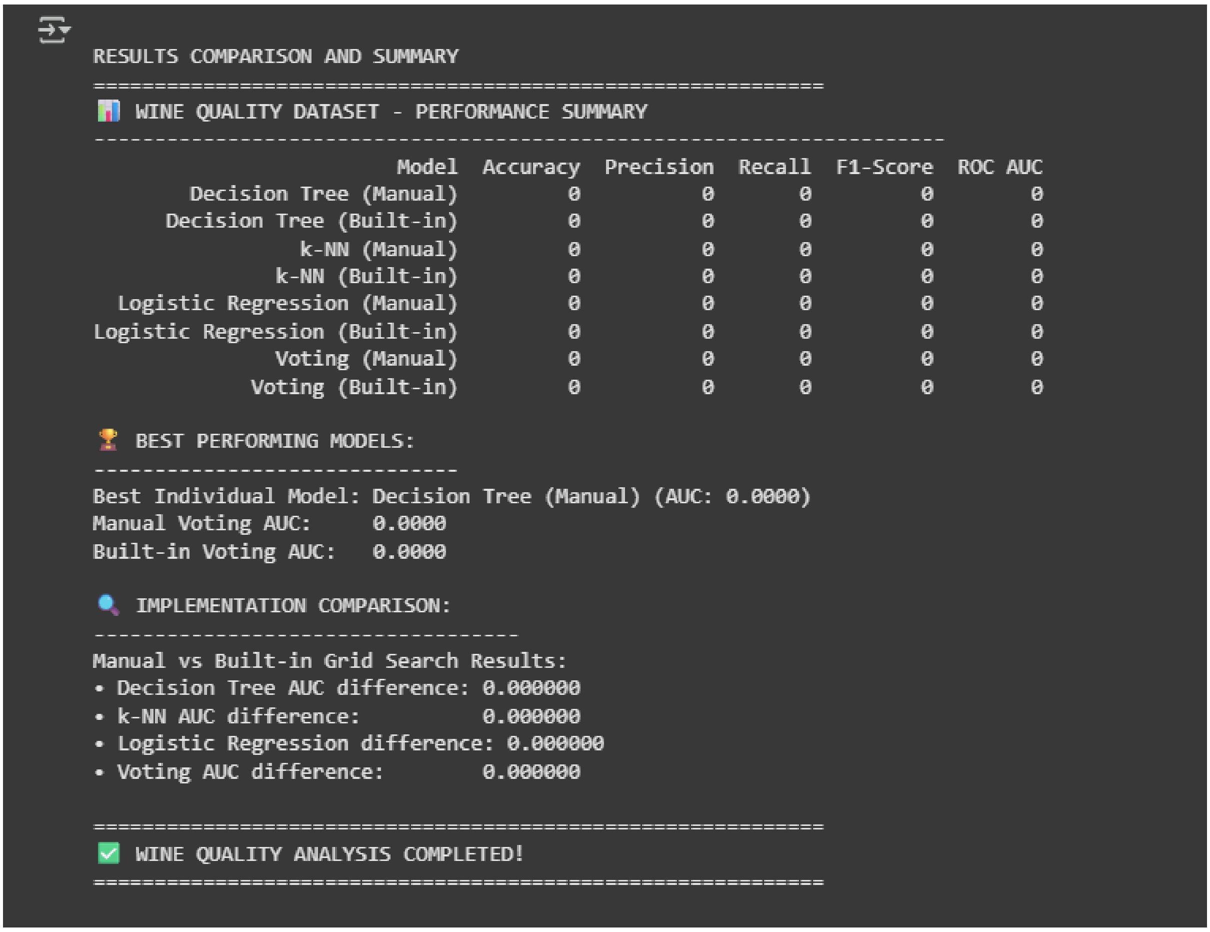


```
=== PERFORMANCE REVIEW (MANUAL APPROACH) ===
-------------------------------------------------------

📈 Individual Model Scores:
----------------------------------------

DecisionTree:
   Accuracy : 0.7375
   Precision: 0.7764
   Recall   : 0.7160
   F1-score : 0.7449
   ROC-AUC  : 0.8038

KNN:
   Accuracy : 0.7937
   Precision: 0.8015
   Recall   : 0.8171
   F1-score : 0.8092
   ROC-AUC  : 0.8703

LogisticReg:
   Accuracy : 0.7396
   Precision: 0.7705
   Recall   : 0.7315
   F1-score : 0.7505
   ROC-AUC  : 0.8166

🤝 Ensemble (Voting) Classifier:
----------------------------------------
Ensemble Results:
   Accuracy : 0.7833
   Precision: 0.8148
   Recall   : 0.7704
   F1-score : 0.7920
   ROC-AUC  : 0.8823
```



```
=== PERFORMANCE REVIEW (BUILT-IN APPROACH) ===
-------------------------------------------------------

📈 Individual Model Scores:
........................................

DecisionTree:
   Accuracy : 0.7375
   Precision: 0.7764
   Recall   : 0.7160
   F1-score : 0.7449
   ROC-AUC  : 0.8038

KNN:
   Accuracy : 0.7937
   Precision: 0.8015
   Recall   : 0.8171
   F1-score : 0.8092
   ROC-AUC  : 0.8703

LogisticReg:
   Accuracy : 0.7396
   Precision: 0.7705
   Recall   : 0.7315
   F1-score : 0.7505
   ROC-AUC  : 0.8166

🤝 Ensemble (Voting) Classifier:
........................................
Ensemble Results:
   Accuracy : 0.8104
   Precision: 0.8242
   Recall   : 0.8210
   F1-score : 0.8226
   ROC-AUC  : 0.8823
```


ROC Curves - Manual Grid Search
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
DecisionTree (AUC=0.804)
KNN (AUC=0.870)
LogisticReg (AUC=0.817)
Voting (AUC=0.882)
ROC Curves - Built-in Grid Search
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
DecisionTree (AUC=0.804)
KNN (AUC=0.870)
LogisticReg (AUC=0.817)
Voting (AUC=0.882)
Confusion Matrix - Manual Voting
True label
Poor Quality
Good Quality
178
45
59
198
Poor Quality
Good Quality
Predicted label
Confusion Matrix - Built-in Voting
True label
Poor Quality
Good Quality
178
45
46
211
Poor Quality
Good Quality
Predicted label



```
RESULTS COMPARISON AND SUMMARY
============================================================
📊 WINE QUALITY DATASET - PERFORMANCE SUMMARY
----------------------------------------------------------------------
                         Model  Accuracy  Precision  Recall  F1-Score  ROC AUC
        Decision Tree (Manual)         0          0       0         0        0
      Decision Tree (Built-in)         0          0       0         0        0
                 k-NN (Manual)         0          0       0         0        0
               k-NN (Built-in)         0          0       0         0        0
  Logistic Regression (Manual)         0          0       0         0        0
Logistic Regression (Built-in)         0          0       0         0        0
               Voting (Manual)         0          0       0         0        0
             Voting (Built-in)         0          0       0         0        0

🏆 BEST PERFORMING MODELS:
------------------------------
Best Individual Model: Decision Tree (Manual) (AUC: 0.0000)
Manual Voting AUC:     0.0000
Built-in Voting AUC:   0.0000

🔍 IMPLEMENTATION COMPARISON:
-----------------------------------
Manual vs Built-in Grid Search Results:
• Decision Tree AUC difference: 0.000000
• k-NN AUC difference:          0.000000
• Logistic Regression difference: 0.000000
• Voting AUC difference:        0.000000

============================================================
✅ WINE QUALITY ANALYSIS COMPLETED!
============================================================
```