ML LAB WEEK 4 - ANALYSIS REPORT

Project Title: Model Selection and Comparative Analysis

Student Name: Neema Shrivastava

Student ID: PES2UG23CS377

Course Name: Machine Learning Lab

Submission Date: 31/8/2025

1. Introduction

   The purpose of this project was to explore **model selection and performance evaluation** through the process of **hyperparameter tuning** and **comparative analysis of classifiers**. Two datasets, Wine Quality and Banknote Authentication, were used to test multiple machine learning algorithms including Decision Trees, k-Nearest Neighbors, and Logistic Regression.

   The tasks performed included:

   - ➢ Implementing manual grid search from scratch to tune hyperparameters.
   - ➢ Using Scikit-learn's GridSearchCV for automated tuning.
   - ➢ Applying Stratified K-Fold Cross-Validation to ensure fair evaluation.
   - ➢ Building ML pipelines with StandardScaler, SelectKBest, and classifiers.
   - ➢ Evaluating models on multiple metrics (Accuracy, Precision, Recall, F1, ROC AUC).
   - ➢ Comparing manual vs. built-in implementations and analyzing performance differences.
   - ➢ Combining models using a Voting Classifier to assess ensemble performance.

2. Dataset Description
   - ★ Wine Quality (Red Wine, UCI Repository)
     - Instances: 1,599 samples
     - Features: 11 chemical properties (fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol)
     - Target: Quality rating (0–10). Converted into a binary target:
       - 1 → Good quality (score > 5)
       - 0 → Not good quality (score ≤ 5)
   - ★ Banknote Authentication (UCI Repository)
     - Instances: 1,372 samples

- Features: 4 statistical properties of images of banknotes (variance, skewness, kurtosis, entropy)
- Target:
    - 1 → Genuine banknote
    - 0 → Forged banknote

3. Methodology

- Hyperparameter Tuning: Selecting the optimal set of parameters for a model to maximize predictive performance.
- Grid Search: Exhaustively evaluates all possible combinations of parameters from a predefined grid.
- K-Fold Cross-Validation: Splits training data into k folds, repeatedly training on (k−1) folds and validating on the remaining fold. We used Stratified 5-Fold to preserve class balance.

ML Pipeline

For each classifier, a Scikit-learn Pipeline was constructed with three stages:

A. StandardScaler: Standardizes features to mean 0 and variance 1.
B. SelectKBest (f_classif): Selects top k features. k is a tuned hyperparameter.
C. Classifier: One of:
    a. Decision Tree
    b. k-Nearest Neighbors (kNN)
    c. Logistic Regression

Process

In Part 1: Manual Implementation, parameter grids were created for each classifier, and all possible parameter combinations were generated. A 5-fold Stratified Cross-Validation procedure was performed, and the ROC AUC score was computed for every combination. The parameters that achieved the highest mean ROC AUC were selected, and the final pipeline was refit on the full training dataset using these best parameters.

In Part 2: Built-in Implementation (GridSearchCV), Scikit-learn's GridSearchCV was used with the same pipeline structure and parameter grids. The scoring metric chosen was ROC AUC, and cross-validation was again carried out using a Stratified 5-Fold approach. From this process, the best estimator, the best parameters, and the best cross-validation score were extracted for further evaluation.

4. Results and Analysis

Wine Quality Dataset

Manual Grid Search – Best Parameters:

- Decision Tree: {'selectkbest__k': 5, 'classifier__max_depth': 5, 'classifier__min_samples_split': 5}
- kNN: {'selectkbest__k': 5, 'classifier__n_neighbors': 7, 'classifier__weights': 'distance', 'classifier__p': 1}
- Logistic Regression: {'selectkbest__k': 11, 'classifier__C': 1, 'classifier__penalty': 'l2', 'classifier__solver': 'saga', 'classifier__max_iter': 500}

Built-in GridSearchCV – Best Parameters: Identical to manual implementation for all three classifiers.

Performance Comparison (Test Set)

Classifier	Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual	0.7292	0.7725	0.7004	0.7347	0.8042
Decision Tree	Built-in	0.7271	0.7716	0.6965	0.7321	0.8025
kNN	Manual	0.7812	0.7836	0.8171	0.8000	0.8589
kNN	Built-in	0.7812	0.7836	0.8171	0.8000	0.8589
Logistic Regression	Manual	0.7333	0.7549	0.7432	0.7490	0.8242
Logistic Regression	Built-in	0.7333	0.7549	0.7432	0.7490	0.8243

Voting Classifier

- Manual Voting: Accuracy 0.7354, ROC AUC 0.8605
- Built-in Voting: Accuracy 0.7625, ROC AUC 0.8600

Analysis:

- ➔ The best-performing classifier for Wine Quality was kNN, achieving the highest ROC AUC (0.8589).
- ➔ Manual and built-in implementations produced almost identical results, confirming the correctness of the manual approach.
- ➔ The Voting Classifier (built-in) slightly improved overall accuracy.

Banknote Authentication Dataset

Manual Grid Search – Best Parameters:

- Decision Tree: {'selectkbest__k': 4, 'classifier__max_depth': 5, 'classifier__min_samples_split': 2}

- kNN: {'selectkbest__k': 4, 'classifier__n_neighbors': 7, 'classifier__weights': 'uniform', 'classifier__p': 1}

- Logistic Regression: {'selectkbest__k': 4, 'classifier__C': 10, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs', 'classifier__max_iter': 500}

Built-in GridSearchCV – Best Parameters: Identical to manual implementation for all three classifiers.

Performance Comparison (Test Set)

Classifier	Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual	0.9854	0.9733	0.9945	0.9838	0.9847
Decision Tree	Built-in	0.9854	0.9733	0.9945	0.9838	0.9847
kNN	Manual	1.0000	1.0000	1.0000	1.0000	1.0000
kNN	Built-in	1.0000	1.0000	1.0000	1.0000	1.0000
Logistic Regression	Manual	0.9903	0.9786	1.0000	0.9892	0.9999
Logistic Regression	Built-in	0.9903	0.9786	1.0000	0.9892	0.9999

Voting Classifier

- Manual Voting: Accuracy 1.0000, ROC AUC 1.0000
- Built-in Voting: Accuracy 1.0000, ROC AUC 1.0000

Analysis:

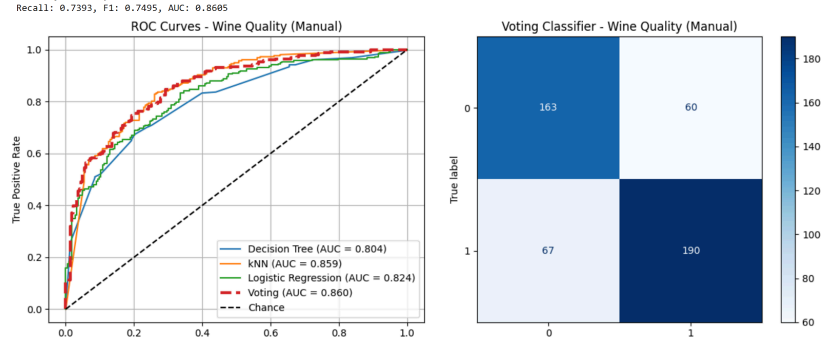
➔ All models performed exceptionally well; kNN and Logistic Regression achieved near-perfect classification.
➔ Both manual and built-in approaches gave identical results.
➔ The Voting Classifier achieved perfect accuracy and AUC.

5. Screenshots

```
################################################################################
PROCESSING DATASET: WINE QUALITY
################################################################################
Wine Quality dataset loaded and preprocessed successfully.
Training set shape: (1119, 11)
Testing set shape: (480, 11)
------------------------------

============================================================
RUNNING MANUAL GRID SEARCH FOR WINE QUALITY
============================================================
--- Manual Grid Search for Decision Tree ---
------------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'selectkbest__k': 5, 'classifier__max_depth': 5, 'classifier__min_samples_split': 5}
Best cross-validation AUC: 0.7832
--- Manual Grid Search for kNN ---
------------------------------------------------------------------------------------------
Best parameters for kNN: {'selectkbest__k': 5, 'classifier__n_neighbors': 7, 'classifier__weights': 'distance', 'classifier__p': 1}
Best cross-validation AUC: 0.8667
--- Manual Grid Search for Logistic Regression ---
------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'selectkbest__k': 11, 'classifier__C': 1, 'classifier__penalty': 'l2', 'classifier__solver': 'saga', 'classifier__max_iter': 500}
Best cross-validation AUC: 0.8052

============================================================
EVALUATING MANUAL MODELS FOR WINE QUALITY
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7292
  Precision: 0.7725
  Recall: 0.7004
  F1-Score: 0.7347
  ROC AUC: 0.8042

kNN:
  Accuracy: 0.7812
  Precision: 0.7836
  Recall: 0.8171
```

What can I help you build?

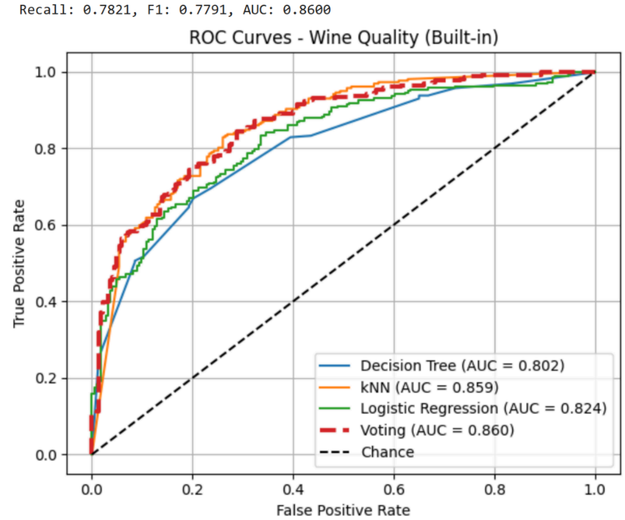
```
  F1-Score: 0.8000
  ROC AUC: 0.8589

Logistic Regression:
  Accuracy: 0.7333
  Precision: 0.7549
  Recall: 0.7432
  F1-Score: 0.7490
  ROC AUC: 0.8242

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7354, Precision: 0.7600
  Recall: 0.7393, F1: 0.7495, AUC: 0.8605
```

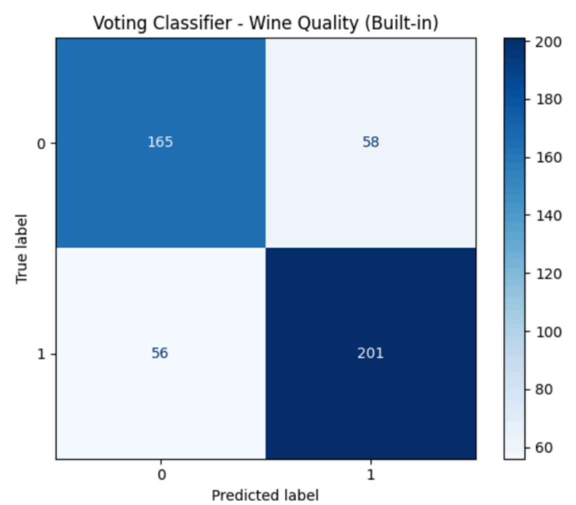


```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR WINE QUALITY
============================================================

--- GridSearchCV for Decision Tree ---
Best params for Decision Tree: {'classifier__max_depth': 5, 'classifier__min_samples_split': 5, 'selectkbest__k': 5}
Best CV score: 0.7832

--- GridSearchCV for kNN ---
Best params for kNN: {'classifier__n_neighbors': 7, 'classifier__p': 1, 'classifier__weights': 'distance', 'selectkbest__k': 5}
Best CV score: 0.8667

--- GridSearchCV for Logistic Regression ---
Best params for Logistic Regression: {'classifier__C': 1, 'classifier__max_iter': 500, 'classifier__penalty': 'l2', 'classifier__solver': 'saga', 'selectkbest__k': 11}
Best CV score: 0.8052

============================================================
EVALUATING BUILT-IN MODELS FOR WINE QUALITY
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7271
  Precision: 0.7716
  Recall: 0.6965
  F1-Score: 0.7321
  ROC AUC: 0.8025

kNN:
  Accuracy: 0.7812
  Precision: 0.7836
  Recall: 0.8171
  F1-Score: 0.8000
  ROC AUC: 0.8589

Logistic Regression:
  Accuracy: 0.7333
  Precision: 0.7549
  Recall: 0.7432
  F1-Score: 0.7490
```


```
  F1-Score: 0.7490
```


```
  ROC AUC: 0.8243

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7625, Precision: 0.7761
  Recall: 0.7821, F1: 0.7791, AUC: 0.8600
```

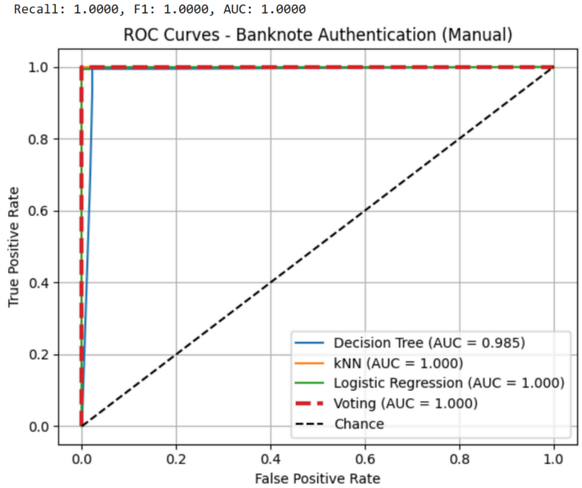


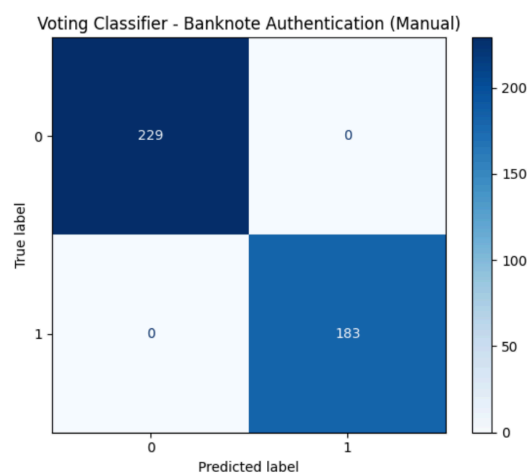



```
Completed processing for Wine Quality
================================================================================

================================================================================
ALL DATASETS PROCESSED!
================================================================================
```

```
################################################################################
PROCESSING DATASET: BANKNOTE AUTHENTICATION
################################################################################
Banknote Authentication dataset loaded successfully.
Training set shape: (960, 4)
Testing set shape: (412, 4)
------------------------------

============================================================
RUNNING MANUAL GRID SEARCH FOR BANKNOTE AUTHENTICATION
============================================================
--- Manual Grid Search for Decision Tree ---
------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'selectkbest__k': 4, 'classifier__max_depth': 5, 'classifier__min_samples_split': 10}
Best cross-validation AUC: 0.9862
--- Manual Grid Search for kNN ---
------------------------------------------------------------------------------------
Best parameters for kNN: {'selectkbest__k': 4, 'classifier__n_neighbors': 7, 'classifier__weights': 'uniform', 'classifier__p': 1}
Best cross-validation AUC: 0.9990
--- Manual Grid Search for Logistic Regression ---
------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'selectkbest__k': 4, 'classifier__C': 10, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs', 'classifier__max_iter': 500}
Best cross-validation AUC: 0.9995

============================================================
EVALUATING MANUAL MODELS FOR BANKNOTE AUTHENTICATION
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.9854
  Precision: 0.9733
  Recall: 0.9945
  F1-Score: 0.9838
  ROC AUC: 0.9847

kNN:
  Accuracy: 1.0000
  Precision: 1.0000
  Recall: 1.0000
  F1-Score: 1.0000
```

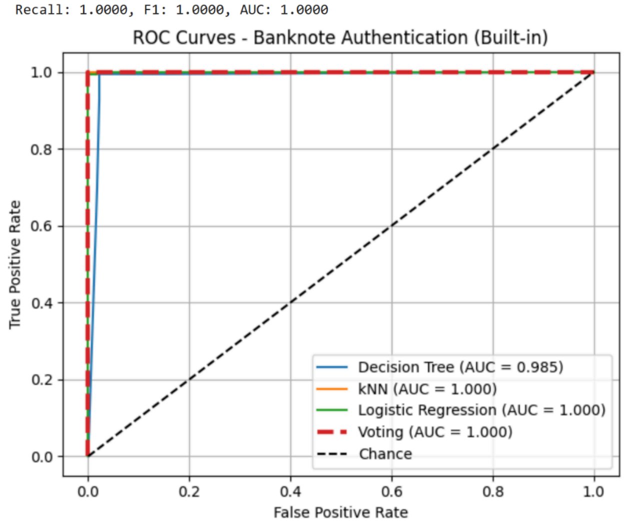
```
Logistic Regression:
  Accuracy: 0.9903
  Precision: 0.9786
  Recall: 1.0000
  F1-Score: 0.9892
  ROC AUC: 0.9999

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 1.0000, Precision: 1.0000
  Recall: 1.0000, F1: 1.0000, AUC: 1.0000
```

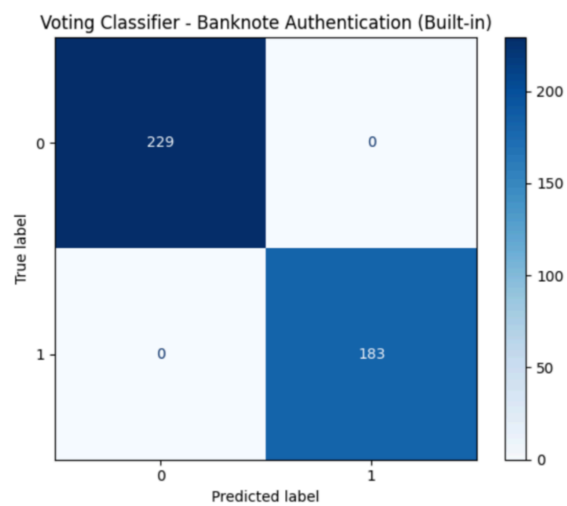




```
RUNNING BUILT-IN GRID SEARCH FOR BANKNOTE AUTHENTICATION
============================================================

--- GridSearchCV for Decision Tree ---
Best params for Decision Tree: {'classifier__max_depth': 5, 'classifier__min_samples_split': 2, 'selectkbest__k': 4}
Best CV score: 0.9856

--- GridSearchCV for kNN ---
Best params for kNN: {'classifier__n_neighbors': 7, 'classifier__p': 1, 'classifier__weights': 'uniform', 'selectkbest__k': 4}
Best CV score: 0.9990

--- GridSearchCV for Logistic Regression ---
Best params for Logistic Regression: {'classifier__C': 10, 'classifier__max_iter': 500, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs', 'selectkbest__k': 4}
Best CV score: 0.9995

============================================================
EVALUATING BUILT-IN MODELS FOR BANKNOTE AUTHENTICATION
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.9854
  Precision: 0.9733
  Recall: 0.9945
  F1-Score: 0.9838
  ROC AUC: 0.9847

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

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

```
  ROC AUC: 0.9999

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 1.0000, Precision: 1.0000
  Recall: 1.0000, F1: 1.0000, AUC: 1.0000
```






```
Completed processing for Banknote Authentication
================================================================================

================================================================================
ALL DATASETS PROCESSED!
================================================================================
```

6. Conclusion

Key Findings:

- Both manual and built-in Grid Search found the same optimal hyperparameters across datasets.
- Performance metrics were nearly identical between manual and built-in implementations.
- kNN consistently performed best on Wine Quality, while all models excelled on Banknote Authentication.
- Voting Classifier often provided small performance boosts and, in the case of Banknote Authentication, perfect results.

Main Takeaways:

- Manual implementation deepens understanding of hyperparameter tuning and model evaluation.
- Scikit-learn's GridSearchCV is more efficient, less error-prone, and scales better for larger parameter spaces.
- Model performance depends heavily on dataset characteristics:
    - Complex, noisy datasets (Wine Quality) benefit from non-linear models like kNN.
    - Cleaner, linearly separable datasets (Banknote Authentication) allow multiple models to achieve near-perfect accuracy.