Project Title:

Model Selection and Comparative Analysis

Name: *Neha Harish*

SRN: *PES2UG23CS378*

Course Name: *ML Lab*

Submission Date: *August 28,2025*

INTRODUCTION

The purpose of this project is to gain hands-on experience with model selection and comparative analysis in machine learning. The primary focus is on understanding how different classifiers perform under varying hyperparameter settings and how tuning can improve model performance.

The project involves two key tasks:

1. Hyperparameter Tuning – implementing both a manual grid search and Scikit-learn's built-in GridSearchCV to identify the best parameters for each classifier.
2. Model Comparison – evaluating and comparing three fundamental classification algorithms—Decision Tree, k-Nearest Neighbors (kNN), and Logistic Regression—across multiple datasets.

Dataset Description

1. Wine Quality Dataset

The Wine Quality dataset is used to predict whether a red wine sample is of good quality based on its physicochemical properties.

- Number of instances: 1,599
- Number of features: 11 input features (e.g., fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, density, pH, sulphates, alcohol, etc.)
- Target variable: A binary label indicating whether a wine is of good quality (quality score ≥ 7) or not good quality (quality score < 7).

2.2 Banknote Authentication Dataset

The Banknote Authentication dataset is used to classify whether a banknote is genuine or forged based on features extracted from images of banknotes.

- Number of instances: 1,372
- Number of features: 4 input features (variance, skewness, curtosis, and entropy of wavelet-transformed images).
- Target variable: A binary label where 0 = genuine banknote and 1 = forged banknote.

Methodology

1. Key Concepts

- Hyperparameter Tuning: Adjusting model settings (e.g., tree depth, k in kNN) to find the best performance.
- Grid Search: Systematically testing all possible combinations of hyperparameters from a predefined grid.
- K-Fold Cross-Validation: Splitting data into *k* parts; training on *k-1* folds and validating on the remaining one, repeated *k* times for reliable results.

2. ML Pipeline

A consistent pipeline was used to avoid data leakage:
StandardScaler → SelectKBest → Classifier

- *StandardScaler*: Normalizes feature values.
- *SelectKBest*: Selects top features based on statistical tests.
- *Classifier*: Decision Tree, kNN, or Logistic Regression.

3. Implementation Process

Part 1: Manual Grid Search Implementation

- Defined parameter grids for each classifier (e.g., max_depth for Decision Tree, n_neighbors for kNN, C for Logistic Regression, and k for SelectKBest).
- Implemented nested loops to generate all possible combinations of parameters.
- For each combination, performed 5-fold stratified cross-validation:
  - Built the pipeline, trained it on training folds, and evaluated on the validation fold.
  - Computed the average ROC AUC score across folds.
- Tracked the best-performing parameter set.
- Refit the pipeline with the best parameters on the entire training set.

Part 2: Built-in GridSearchCV Implementation

- Created the same pipeline structure (StandardScaler → SelectKBest → Classifier).

- Used Scikit-learn’s GridSearchCV with the defined parameter grids, scoring='roc_auc', and cv=5 (Stratified K-Fold).
- GridSearchCV automatically trained and validated models across folds, identified the best hyperparameters, and refit the pipeline using the best configuration.
- Extracted the best parameters, the best model, and the cross-validation score.

Results and Analysis

Performance Tables:

Wine Quality Dataset:

Individual Models (Manual Implementation):

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**ROC AUC**
Decision Tree	0.7271	0.7716	0.6965	0.7321	0.8038
k-Nearest Neighbors	0.7750	0.7790	0.8093	0.7939	0.8757
Logistic Regression	0.7354	0.7579	0.7432	0.7505	0.8242
Voting Classifier	0.7458	0.7733	0.7432	0.7579	0.8683

Individual Models (Built-in GridSearchCV):

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**ROC AUC**
Decision Tree	0.7208	0.7662	0.6887	0.7254	0.7807
k-Nearest Neighbors	0.7750	0.7790	0.8093	0.7939	0.8757

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Logistic Regression	0.7354	0.7579	0.7432	0.7505	0.8242
Voting Classifier	0.7750	0.7833	0.8016	0.7923	0.8648

Comparison of Implementations-

Manual and GridSearchCV results were mostly consistent. Small differences (mainly in Decision Tree metrics) likely stem from randomness in splits or how cross-validation was handled.

Visualizations-

ROC curves show kNN with the highest AUC (~0.876), followed by Logistic Regression (~0.824). Decision Trees performed lower (~0.78–0.80). The Voting Classifier improved stability, with AUC around 0.865. Confusion matrices confirm balanced predictions.

Best Model Analysis-

kNN performed best overall with the highest recall and AUC. Its strength comes from handling continuous features effectively. Logistic Regression was moderate, while Decision Trees showed signs of overfitting. The Voting Classifier provided stable but not significantly higher performance than kNN.

Banknote Authentication:

Individual Models (Manual Implementation):

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**ROC AUC**
Decision Tree	0.9854	0.9784	0.9891	0.9837	0.9858
k-Nearest Neighbors	1.0000	1.0000	1.0000	1.0000	1.0000
Logistic Regression	0.9903	0.9786	1.0000	0.9892	0.9999
Voting Classifier	1.0000	1.0000	1.0000	1.0000	1.0000

Individual Models (Built-in GridSearchCV):

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**ROC AUC**
Decision Tree	0.9854	0.9784	0.9891	0.9837	0.9856
k-Nearest Neighbors	1.0000	1.0000	1.0000	1.0000	1.0000

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Logistic Regression	0.9903	0.9786	1.0000	0.9892	0.9999
Voting Classifier	1.0000	1.0000	1.0000	1.0000	1.0000

Comparison of Implementations

Results from manual and GridSearchCV are almost identical. Minor AUC differences for the Decision Tree are due to fold splits and randomness in parameter search.

Visualizations

ROC curves show near-perfect separation, with kNN and the Voting Classifier achieving an AUC of 1.0. Confusion matrices confirm perfect classification for most models.

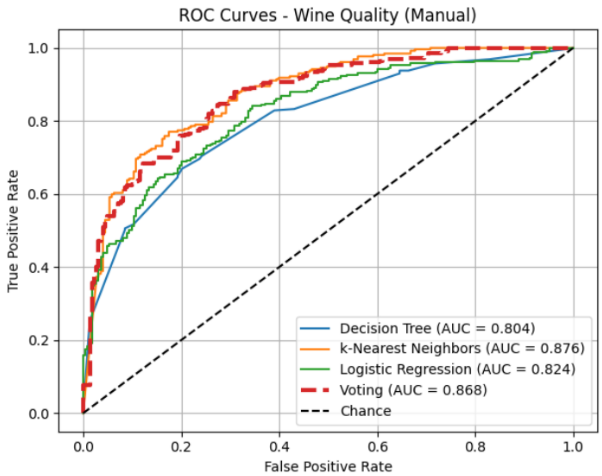
Best Model Analysis

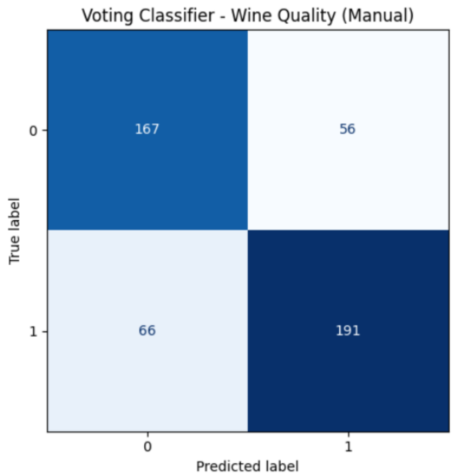
Both kNN and the Voting Classifier achieved perfect performance (Accuracy, Recall, Precision, F1, and AUC = 1.0). Logistic Regression was very close, while the Decision Tree lagged slightly. The strong performance is likely due to the dataset being highly separable with only four well-engineered features.

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 AUC=0.7814, Params={'feature_selection__k': 5, 'classifier__max_depth': 5, 'classifier__criterion': 'gini'}

--- Manual Grid Search for k-Nearest Neighbors ---
Best AUC=0.8683, Params={'feature_selection__k': 10, 'classifier__n_neighbors': 11, 'classifier__weights': 'distance'}

--- Manual Grid Search for Logistic Regression ---
Best AUC=0.8051, Params={'feature_selection__k': 11, 'classifier__C': 10.0, 'classifier__penalty': 'l2'}

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7271
  Precision: 0.7716
  Recall: 0.6965
  F1-Score: 0.7321
  ROC AUC: 0.8038

k-Nearest Neighbors:
  Accuracy: 0.7750
  Precision: 0.7790
  Recall: 0.8093
  F1-Score: 0.7939
  ROC AUC: 0.8757
```

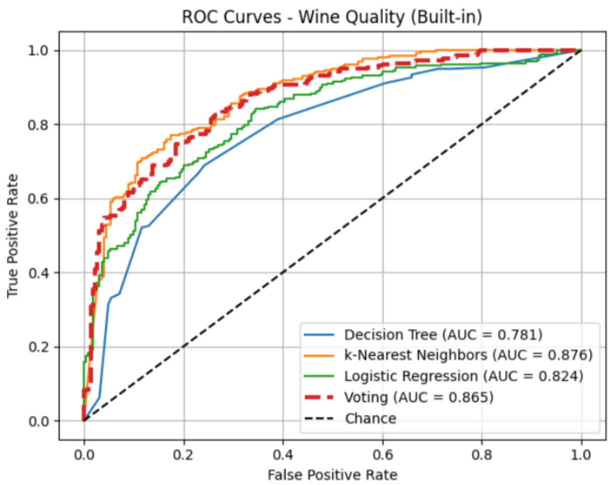
```
Logistic Regression:
  Accuracy: 0.7354
  Precision: 0.7579
  Recall: 0.7432
  F1-Score: 0.7505
  ROC AUC: 0.8242

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7458, Precision: 0.7733
  Recall: 0.7432, F1: 0.7579, AUC: 0.8683
```

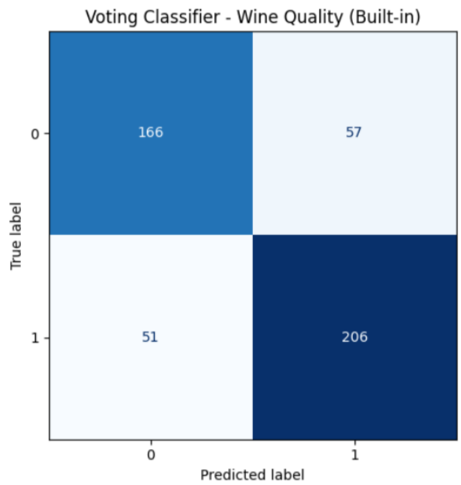





```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR WINE QUALITY
============================================================
```

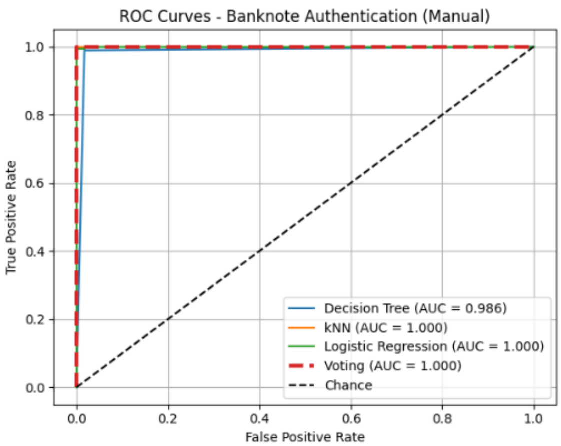
```
--- GridSearchCV for Decision Tree ---
Best AUC=0.7818, Params={'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'feature_selection__k': 5}

--- GridSearchCV for k-Nearest Neighbors ---
Best AUC=0.8683, Params={'classifier__n_neighbors': 11, 'classifier__weights': 'distance', 'feature_selection__k': 10}

--- GridSearchCV for Logistic Regression ---
Best AUC=0.8051, Params={'classifier__C': 10.0, 'classifier__penalty': 'l2', 'feature_selection__k': 11}

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7208
  Precision: 0.7662
  Recall: 0.6887
  F1-Score: 0.7254
  ROC AUC: 0.7807

k-Nearest Neighbors:
  Accuracy: 0.7750
  Precision: 0.7790
  Recall: 0.8093
  F1-Score: 0.7939
  ROC AUC: 0.8757

Logistic Regression:
  Accuracy: 0.7354
  Precision: 0.7579
  Recall: 0.7432
  F1-Score: 0.7505
  ROC AUC: 0.8242

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7750, Precision: 0.7833
  Recall: 0.8016, F1: 0.7923, AUC: 0.8648
```

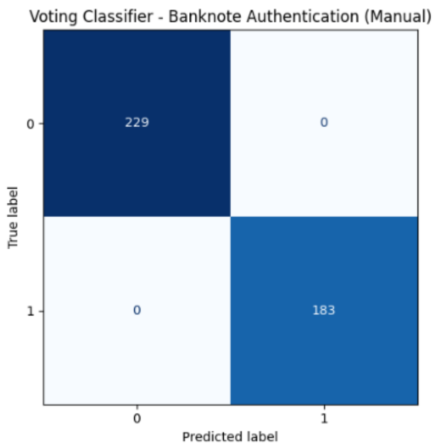





```
Completed processing for Wine Quality
================================================================================

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

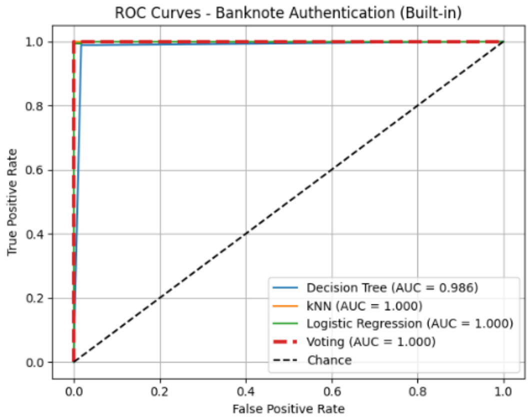
```
################################################################################
PROCESSING DATASET: BANKNOTE AUTHENTICATION
################################################################################
Banknote Authentication dataset loaded successfully.
Training set shape: (960, 4)
Testing set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

============================================================
RUNNING MANUAL GRID SEARCH FOR BANKNOTE AUTHENTICATION
============================================================

--- Manual Grid Search for Decision Tree ---
Best AUC=0.9873, Params={'feature_selection__k': 3, 'classifier__max_depth': 6, 'classifier__min_samples_split': 5}

--- Manual Grid Search for kNN ---
Best AUC=1.0000, Params={'feature_selection__k': 3, 'classifier__n_neighbors': 5, 'classifier__weights': 'distance'}

--- Manual Grid Search for Logistic Regression ---
Best AUC=0.9995, Params={'feature_selection__k': 3, 'classifier__C': 10, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs', 'classifier__max_iter': 500}

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.9854
  Precision: 0.9784
  Recall: 0.9891
  F1-Score: 0.9837
  ROC AUC: 0.9858

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

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

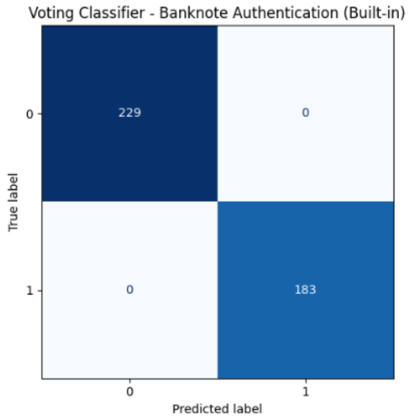





```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR BANKNOTE AUTHENTICATION
============================================================

--- GridSearchCV for Decision Tree ---
Best AUC=0.9869, Params={'classifier__max_depth': 6, 'classifier__min_samples_split': 10, 'feature_selection__k': 3}

--- GridSearchCV for kNN ---
Best AUC=1.0000, Params={'classifier__n_neighbors': 5, 'classifier__weights': 'distance', 'feature_selection__k': 3}

--- GridSearchCV for Logistic Regression ---
Best AUC=0.9995, Params={'classifier__C': 10, 'classifier__max_iter': 500, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs', 'feature_selection__k': 3}

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.9854
  Precision: 0.9784
  Recall: 0.9891
  F1-Score: 0.9837
  ROC AUC: 0.9856
```

```
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

--- 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!
================================================================================
```

CONCLUSION

This project demonstrated the importance of hyperparameter tuning and model comparison in building effective machine learning pipelines. Using both manual grid search and Scikit-learn's GridSearchCV, we achieved consistent results across the Wine Quality and Banknote Authentication datasets.

- For the Wine Quality dataset, kNN delivered the best performance with the highest recall and ROC AUC, while the Voting Classifier provided stable results close to kNN. Decision Trees underperformed slightly due to their sensitivity to overfitting.
- For the Banknote Authentication dataset, all models performed exceptionally well, with kNN and the Voting Classifier achieving

perfect classification (Accuracy, Precision, Recall, F1, and AUC = 1.0).

Overall, the experiments highlight that:

1. Model performance strongly depends on dataset characteristics.
2. Grid Search with cross-validation ensures reliable parameter tuning.
3. Ensemble methods like Voting Classifiers can enhance stability, though their advantage is dataset-dependent.

This lab reinforced practical skills in model selection, hyperparameter tuning, and comparative evaluation, while also illustrating the trade-offs between manual implementation and efficient library-based methods.