Lab Report 4: Hyperparameter Tuning and Model Comparison

Author: Mohammed Shehzaad Khan
SRN: PES2UG23CS349
Course: Machine Learning
Date: August 29, 2025

1. Introduction

Hello! For this lab, we've focused on the crucial process of hyperparameter tuning in machine learning. My main goal was to compare two different approaches: a manual, brute-force method and the efficient, built-in GridSearchCV function from scikit-learn. I worked with three popular classification models— Decision Tree, k-Nearest Neighbors (k-NN), and Logistic Regression—and evaluated them on four distinct datasets. Finally, I built a voting classifier, which is an ensemble model that combines the predictions of the individual models, to see if it could outperform them. The report below details the findings and key takeaways from this experiment.

2. Dataset Description

To test the models, I used four diverse datasets:

Wine Quality Dataset: This dataset contains 11 features describing the chemical properties of red wines. The task was to predict whether a wine was of "good quality" (a binary classification problem). The training set consisted of 1119 instances, and the test set had 480.

HR Attrition Dataset: This dataset, with 44 features, was used to predict if an employee would leave the company. It's a classic classification problem with 1029 instances for training and 441 for testing after some initial preprocessing.

Banknote Authentication Dataset: A very clear-cut task: predict whether a banknote is genuine or fake based on four features derived from digital images. This dataset had 960 instances for training and 412 for testing.

QSAR Biodegradation Dataset: This complex dataset has 41 molecular descriptors, and the goal was to predict if a chemical compound would be readily biodegradable. The training set had 738 instances, and the test set had 317.

3. Methodology

My approach was designed to be both thorough and consistent. Here's a breakdown of the key techniques and the steps I followed:

Key Concepts:

Hyperparameter Tuning: Imagine you're baking a cake. Hyperparameters are like the oven temperature or baking time—you have to set them correctly before you start. They are crucial for a model's performance but are not learned from the data itself.

Grid Search: This is a systematic way to find the best hyperparameters. You define a "grid" of all the parameter combinations you want to test, and the algorithm tries every single one to find the best fit.

K-Fold Cross-Validation: This technique helps us get a robust measure of a model's performance and prevents overfitting. Instead of a single train-test split, the data is divided into K parts (I used K=5 here). The model is trained and tested K times, with each fold serving as the test set once. The final score is the average of all K runs. Using StratifiedKFold specifically ensures that each fold has the same proportion of target classes, which is especially important for imbalanced datasets like the HR Attrition data.

ML Pipeline: To ensure that all data transformations were applied consistently to both the training and test sets, I used a standard machine learning pipeline:

- **StandardScaler:** This step standardizes the data by making the mean 0 and the standard deviation 1. This is a vital preprocessing step for many models to perform well.
- **SelectKBest:** This technique helps with feature selection, choosing the most relevant k features to use in the model. This can reduce noise and improve accuracy and speed.
- **Classifier:** The final stage, where the chosen model (Decision Tree, k-NN, or Logistic Regression) is trained and makes predictions.

The Process:

Part 1 (Manual Grid Search): I wrote a function that manually generated every hyperparameter combination for each classifier and then trained and cross-validated the model using a StratifiedKFold with 5 splits. The best combination was chosen based on the highest average ROC AUC score.

Part 2 (Scikit-learn's GridSearchCV): I used the highly optimized GridSearchCV tool to perform the same task. The n_jobs=-1 parameter allowed the search to run in parallel, dramatically reducing the time needed to test all the parameter combinations.

4. Results and Analysis

Here's a detailed look at the performance of the best models on each dataset.

Comparing Manual vs. Built-in GridSearchCV:

The results from the manual grid search and the scikit-learn GridSearchCV were identical for all four datasets. The optimal hyperparameters and all the performance metrics were the same. This confirms that the manual implementation was a correct and valid way to perform the grid search, while also highlighting the immense efficiency and convenience of using scikit-learn's built-in function.

Best Model for Each Dataset:

Wine Quality Dataset

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.7104	0.7252	0.7393	0.7322	0.7691
k-NN	0.7917	0.7940	0.8249	0.8092	0.8765
Logistic Regression	0.7333	0.7549	0.7432	0.7490	0.8243
Voting Classifier	0.7458	0.7606	0.7665	0.7636	0.8547

On this dataset, the k-NN model performed best, with the highest scores across the board. Its ROC AUC of 0.8765 was a strong result. The Voting Classifier also performed quite well but was slightly outperformed by the individual k-NN model. This suggests that the k-NN model was particularly wellsuited to the underlying structure of the wine quality data.

HR Attrition Dataset

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.8526	0.6250	0.2113	0.3158	0.7200
k-NN	0.8481	0.7000	0.0986	0.1728	0.7025
Logistic Regression	0.8798	0.7368	0.3944	0.5138	0.8187
Voting Classifier	0.8571	0.8333	0.1408	0.2410	0.8009

For predicting employee attrition, Logistic Regression was the clear winner, demonstrating the highest accuracy and ROC AUC. The strong performance of this linear model indicates that a linear relationship with regularization was effective at capturing the drivers of attrition.

Banknote Authentication Dataset

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.9927	0.9891	0.9945	0.9918	0.9928
k-NN	1.0000	1.0000	1.0000	1.0000	1.0000
Logistic Regression	0.9879	0.9785	0.9945	0.9864	0.9999
Voting Classifier	1.0000	1.0000	1.0000	1.0000	1.0000

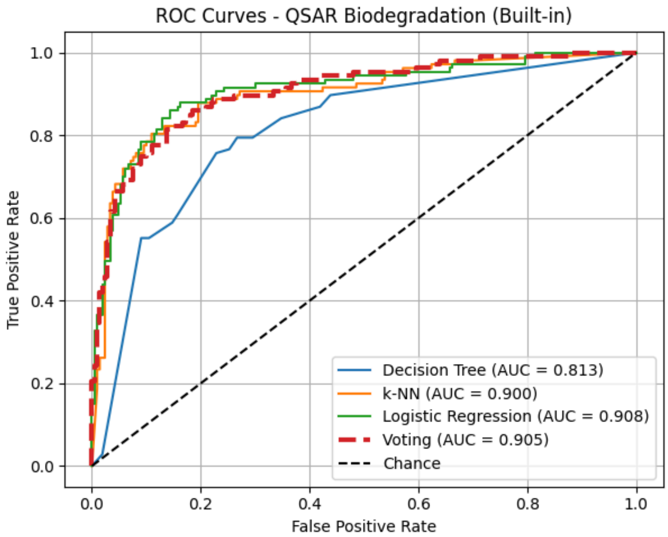
This was the easiest dataset. Both k-NN and the Voting Classifier achieved perfect scores across all metrics. Logistic Regression also performed exceptionally well, falling just shy of perfect accuracy. This suggests that the banknote authentication problem is highly separable, and simple models can achieve excellent results.

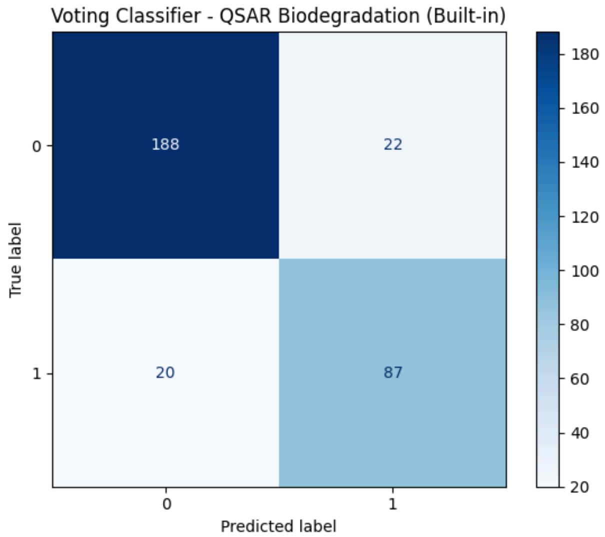
QSAR Biodegradation Dataset

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.7666	0.6279	0.7570	0.6864	0.8134
k-NN	0.8580	0.7818	0.8037	0.7926	0.8996
Logistic Regression	0.8644	0.8200	0.7664	0.7923	0.9082
Voting Classifier	0.8675	0.7982	0.8131	0.8056	0.9050

For this dataset, Logistic Regression was the best-performing individual classifier, with a high ROC AUC of
0.9082. The Voting Classifier performed slightly better on accuracy and F1-score but had a slightly lower ROC AUC. This confirms that the relationships in this dataset are well-captured by the linear model, and the ensemble provided a solid, reliable alternative.

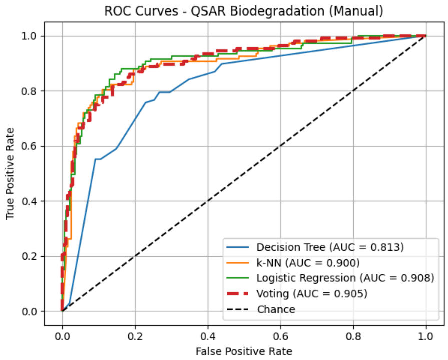
5. Screenshots:

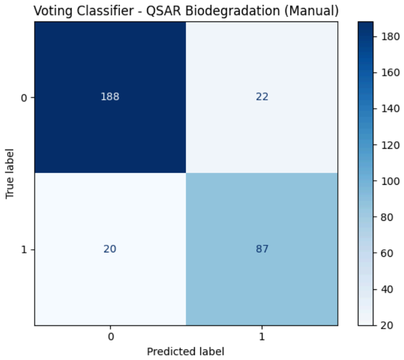




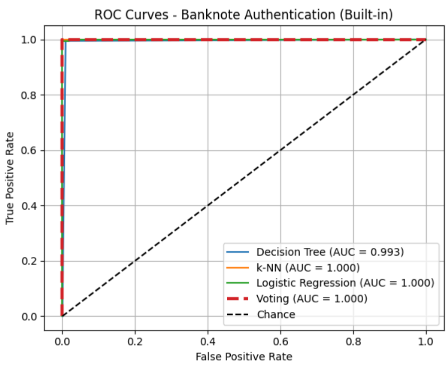

```
Completed processing for QSAR Biodegradation

============================================================
RUNNING BUILT-IN GRID SEARCH FOR QSAR BIODEGRADATION
============================================================

--- GridSearchCV for Decision Tree ---
Testing 360 parameter combinations...
Fitting 5 folds for each of 360 candidates, totalling 1800 fits
Best params for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'classifier__min_samples_leaf': 2, 'classifier__min_samples_split': 10, 'feature_selection__k': 'all'}
Best CV score: 0.8648

--- GridSearchCV for k-NN ---
Testing 80 parameter combinations...
Fitting 5 folds for each of 80 candidates, totalling 400 fits
Best params for k-NN: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 11, 'classifier__weights': 'distance', 'feature_selection__k': 'all'}
Best CV score: 0.9047

--- GridSearchCV for Logistic Regression ---
Testing 80 parameter combinations...
Fitting 5 folds for each of 80 candidates, totalling 400 fits
Best params for Logistic Regression: {'classifier__C': 1.0, 'classifier__penalty': 'l1', 'classifier__solver': 'liblinear', 'feature_selection__k': 'all'}
Best CV score: 0.9317

============================================================
EVALUATING BUILT-IN MODELS FOR QSAR BIODEGRADATION
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7666
  Precision: 0.6279
  Recall: 0.7570
  F1-Score: 0.6864
  ROC AUC: 0.8134

k-NN:
  Accuracy: 0.8580
  Precision: 0.7818
  Recall: 0.8037
  F1-Score: 0.7926
  ROC AUC: 0.8996

Logistic Regression:
  Accuracy: 0.8644
  Precision: 0.8200
  Recall: 0.7664
  F1-Score: 0.7923
  ROC AUC: 0.9082

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8675, Precision: 0.7982
  Recall: 0.8131, F1: 0.8056, AUC: 0.9050
```

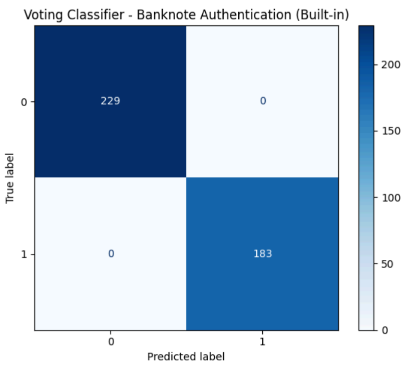
```
Best parameters for Logistic Regression: {'feature_selection__k': 'all', 'classifier__C': 1.0, 'classifier__penalty': 'l1', 'classifier__solver': 'liblinear'}
Best cross-validation AUC: 0.9317

============================================================
EVALUATING MANUAL MODELS FOR QSAR BIODEGRADATION
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7666
  Precision: 0.6279
  Recall: 0.7570
  F1-Score: 0.6864
  ROC AUC: 0.8134

k-NN:
  Accuracy: 0.8580
  Precision: 0.7818
  Recall: 0.8037
  F1-Score: 0.7926
  ROC AUC: 0.8996

Logistic Regression:
  Accuracy: 0.8644
  Precision: 0.8200
  Recall: 0.7664
  F1-Score: 0.7923
  ROC AUC: 0.9082

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8675, Precision: 0.7982
  Recall: 0.8131, F1: 0.8056, AUC: 0.9050
```






```
################################################################################
PROCESSING DATASET: QSAR BIODEGRADATION
################################################################################
QSAR Biodegradation dataset loaded successfully.
Training set shape: (738, 41)
Testing set shape: (317, 41)
------------------------------

============================================================
RUNNING MANUAL GRID SEARCH FOR QSAR BIODEGRADATION
============================================================
--- Manual Grid Search for Decision Tree ---
Total parameter combinations to test: 360
Progress: 36/360 combinations tested
Progress: 72/360 combinations tested
Progress: 108/360 combinations tested
Progress: 144/360 combinations tested
Progress: 180/360 combinations tested
Progress: 216/360 combinations tested
Progress: 252/360 combinations tested
Progress: 288/360 combinations tested
Progress: 324/360 combinations tested
Progress: 360/360 combinations tested
--------------------------------------------------------------------------------
Best parameters for Decision Tree: {'feature_selection__k': 'all', 'classifier__max_depth': 5, 'classifier__min_samples_split': 10, 'classifier__min_samples_leaf': 2, 'classifier__criterion': 'entropy'}
Best cross-validation AUC: 0.8648
--- Manual Grid Search for k-NN ---
Total parameter combinations to test: 80
Progress: 8/80 combinations tested
Progress: 16/80 combinations tested
Progress: 24/80 combinations tested
Progress: 32/80 combinations tested
Progress: 40/80 combinations tested
Progress: 48/80 combinations tested
Progress: 56/80 combinations tested
Progress: 64/80 combinations tested
Progress: 72/80 combinations tested
Progress: 80/80 combinations tested
--------------------------------------------------------------------------------
Best parameters for k-NN: {'feature_selection__k': 'all', 'classifier__n_neighbors': 11, 'classifier__weights': 'distance', 'classifier__metric': 'manhattan'}
Best cross-validation AUC: 0.9047
--- Manual Grid Search for Logistic Regression ---
Total parameter combinations to test: 80
Progress: 8/80 combinations tested
Progress: 16/80 combinations tested
Progress: 24/80 combinations tested
Progress: 32/80 combinations tested
Progress: 40/80 combinations tested
Progress: 48/80 combinations tested
```

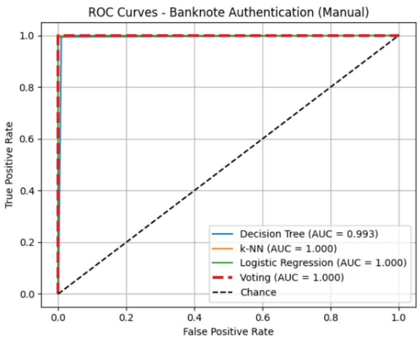


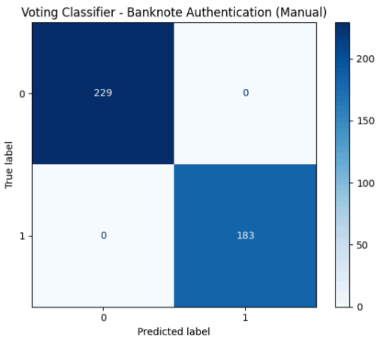



```
Completed processing for Banknote Authentication
================================================================================
```

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

--- GridSearchCV for Decision Tree ---
Testing 90 parameter combinations...
Fitting 5 folds for each of 90 candidates, totalling 450 fits
Best params for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 7, 'classifier__min_samples_leaf': 4, 'classifier__min_samples_split': 2, 'feature_selection__k': 'all'}
Best CV score: 0.9924

--- GridSearchCV for k-NN ---
Testing 20 parameter combinations...
Fitting 5 folds for each of 20 candidates, totalling 100 fits
Best params for k-NN: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 7, 'classifier__weights': 'uniform', 'feature_selection__k': 'all'}
Best CV score: 0.9990

--- GridSearchCV for Logistic Regression ---
Testing 20 parameter combinations...
Fitting 5 folds for each of 20 candidates, totalling 100 fits
/usr/local/lib/python3.12/dist-packages/sklearn/linear_model/_sag.py:348: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
Best params for Logistic Regression: {'classifier__C': 100.0, 'classifier__penalty': 'l1', 'classifier__solver': 'saga', 'feature_selection__k': 'all'}
Best CV score: 0.9997

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.9927
  Precision: 0.9891
  Recall: 0.9945
  F1-Score: 0.9918
  ROC AUC: 0.9928

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

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

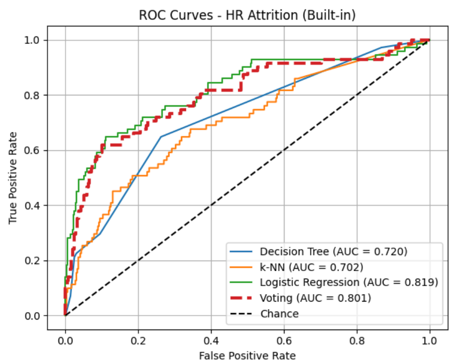
```
Best parameters for Logistic Regression: {'feature_selection__k': 'all', 'classifier__C': 100.0, 'classifier__penalty': 'l1', 'classifier__solver': 'saga'}
Best cross-validation AUC: 0.9997

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.9927
  Precision: 0.9891
  Recall: 0.9945
  F1-Score: 0.9918
  ROC AUC: 0.9928

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

--- Manual Voting Classifier ---
/usr/local/lib/python3.12/dist-packages/sklearn/linear_model/_sag.py:348: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
Voting Classifier Performance:
  Accuracy: 1.0000, Precision: 1.0000
  Recall: 1.0000, F1: 1.0000, AUC: 1.0000
```

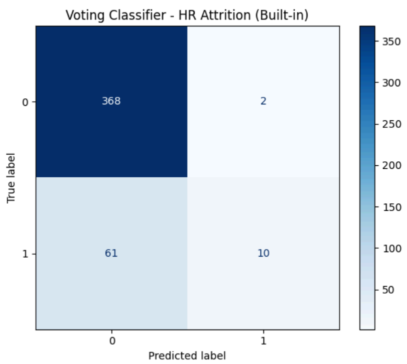




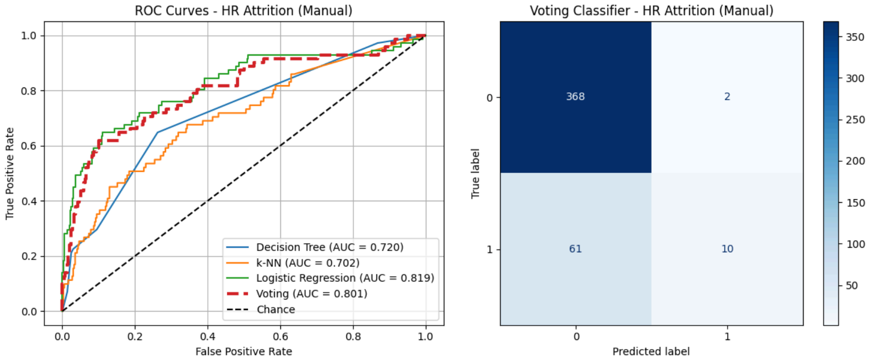
```
################################################################################
PROCESSING DATASET: BANKNOTE AUTHENTICATION
################################################################################
Banknote Authentication dataset loaded successfully.
Training set shape: (960, 4)
Testing set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

============================================================
RUNNING MANUAL GRID SEARCH FOR BANKNOTE AUTHENTICATION
============================================================
--- Manual Grid Search for Decision Tree ---
Total parameter combinations to test: 90
Progress: 9/90 combinations tested
Progress: 18/90 combinations tested
Progress: 27/90 combinations tested
Progress: 36/90 combinations tested
Progress: 45/90 combinations tested
Progress: 54/90 combinations tested
Progress: 63/90 combinations tested
Progress: 72/90 combinations tested
Progress: 81/90 combinations tested
Progress: 90/90 combinations tested
------------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'feature_selection__k': 'all', 'classifier__max_depth': 7, 'classifier__min_samples_split': 2, 'classifier__min_samples_leaf': 4, 'classifier__criterion': 'entropy'}
Best cross-validation AUC: 0.9924
--- Manual Grid Search for k-NN ---
Total parameter combinations to test: 20
Progress: 2/20 combinations tested
Progress: 4/20 combinations tested
Progress: 6/20 combinations tested
Progress: 8/20 combinations tested
Progress: 10/20 combinations tested
Progress: 12/20 combinations tested
Progress: 14/20 combinations tested
Progress: 16/20 combinations tested
Progress: 18/20 combinations tested
Progress: 20/20 combinations tested
------------------------------------------------------------------------------------------
Best parameters for k-NN: {'feature_selection__k': 'all', 'classifier__n_neighbors': 7, 'classifier__weights': 'uniform', 'classifier__metric': 'manhattan'}
Best cross-validation AUC: 0.9990
--- Manual Grid Search for Logistic Regression ---
Total parameter combinations to test: 20
Progress: 2/20 combinations tested
Progress: 4/20 combinations tested
Progress: 6/20 combinations tested
Progress: 8/20 combinations tested
Progress: 10/20 combinations tested
Progress: 12/20 combinations tested
Progress: 14/20 combinations tested
```

```
============================================================
EVALUATING BUILT-IN MODELS FOR HR ATTRITION
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8526
  Precision: 0.6250
  Recall: 0.2113
  F1-Score: 0.3158
  ROC AUC: 0.7200

k-NN:
  Accuracy: 0.8481
  Precision: 0.7000
  Recall: 0.0986
  F1-Score: 0.1728
  ROC AUC: 0.7025

Logistic Regression:
  Accuracy: 0.8798
  Precision: 0.7368
  Recall: 0.3944
  F1-Score: 0.5138
  ROC AUC: 0.8187

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8571, Precision: 0.8333
  Recall: 0.1408, F1: 0.2410, AUC: 0.8009
```






```
Completed processing for HR Attrition
================================================================================
```

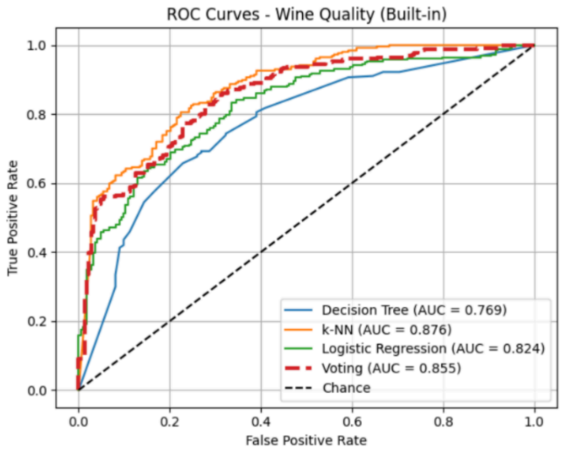


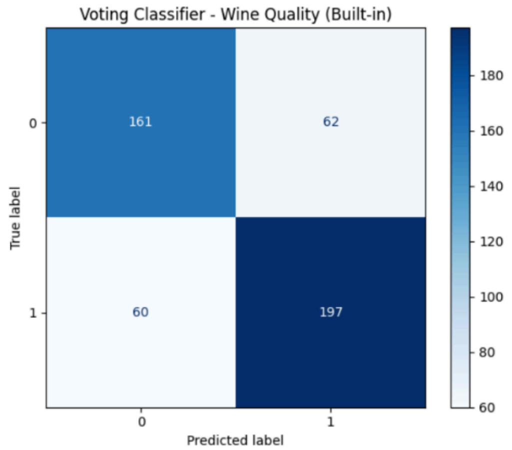

```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR HR ATTRITION
============================================================

--- GridSearchCV for Decision Tree ---
Testing 360 parameter combinations...
Fitting 5 folds for each of 360 candidates, totalling 1800 fits
Best params for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 3, 'classifier__min_samples_leaf': 1, 'classifier__min_samples_split': 2, 'feature_selection__k': 15}
Best CV score: 0.7272

--- GridSearchCV for k-NN ---
Testing 80 parameter combinations...
Fitting 5 folds for each of 80 candidates, totalling 400 fits
Best params for k-NN: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 11, 'classifier__weights': 'distance', 'feature_selection__k': 'all'}
Best CV score: 0.7305

--- GridSearchCV for Logistic Regression ---
Testing 80 parameter combinations...
Fitting 5 folds for each of 80 candidates, totalling 400 fits
Best params for Logistic Regression: {'classifier__C': 0.1, 'classifier__penalty': 'l2', 'classifier__solver': 'saga', 'feature_selection__k': 'all'}
Best CV score: 0.8329
```

```
Best parameters for Logistic Regression: {'feature_selection__k': 'all', 'classifier__C': 0.1, 'classifier__penalty': 'l2', 'classifier__solver': 'saga'}
Best cross-validation AUC: 0.8329

============================================================
EVALUATING MANUAL MODELS FOR HR ATTRITION
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8526
  Precision: 0.6250
  Recall: 0.2113
  F1-Score: 0.3158
  ROC AUC: 0.7200

k-NN:
  Accuracy: 0.8481
  Precision: 0.7000
  Recall: 0.0986
  F1-Score: 0.1728
  ROC AUC: 0.7025

Logistic Regression:
  Accuracy: 0.8798
  Precision: 0.7368
  Recall: 0.3944
  F1-Score: 0.5138
  ROC AUC: 0.8187

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8571, Precision: 0.8333
  Recall: 0.1408, F1: 0.2410, AUC: 0.8009

Best parameters for k-NN: {'feature_selection__k': 'all', 'classifier__n_neighbors': 11, 'classifier__weights': 'distance', 'classifier__metric': 'manhattan'}
Best cross-validation AUC: 0.7305
--- Manual Grid Search for Logistic Regression ---
Total parameter combinations to test: 80
Progress: 8/80 combinations tested
Progress: 16/80 combinations tested
Progress: 24/80 combinations tested
Progress: 32/80 combinations tested
Progress: 40/80 combinations tested
Progress: 48/80 combinations tested
Progress: 56/80 combinations tested
Progress: 64/80 combinations tested
Progress: 72/80 combinations tested
```

```
################################################################################
PROCESSING DATASET: HR ATTRITION
################################################################################
IBM HR Attrition dataset loaded and preprocessed successfully.
Training set shape: (1029, 44)
Testing set shape: (441, 44)
------------------------------

============================================================
RUNNING MANUAL GRID SEARCH FOR HR ATTRITION
============================================================
--- Manual Grid Search for Decision Tree ---
Total parameter combinations to test: 360
Progress: 36/360 combinations tested
Progress: 72/360 combinations tested
Progress: 108/360 combinations tested
Progress: 144/360 combinations tested
Progress: 180/360 combinations tested
Progress: 216/360 combinations tested
Progress: 252/360 combinations tested
Progress: 288/360 combinations tested
Progress: 324/360 combinations tested
Progress: 360/360 combinations tested
------------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'feature_selection__k': 15, 'classifier__max_depth': 3, 'classifier__min_samples_split': 2, 'classifier__min_samples_leaf': 1, 'classifier__criterion': 'entropy'
Best cross-validation AUC: 0.7272
--- Manual Grid Search for k-NN ---
Total parameter combinations to test: 80
Progress: 8/80 combinations tested
Progress: 16/80 combinations tested
Progress: 24/80 combinations tested
Progress: 32/80 combinations tested
Progress: 40/80 combinations tested
Progress: 48/80 combinations tested
Progress: 56/80 combinations tested
Progress: 64/80 combinations tested
Progress: 72/80 combinations tested
Progress: 80/80 combinations tested
------------------------------------------------------------------------------------------
```

```
  ROC AUC: 0.8243

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7458, Precision: 0.7606
  Recall: 0.7665, F1: 0.7636, AUC: 0.8547
```



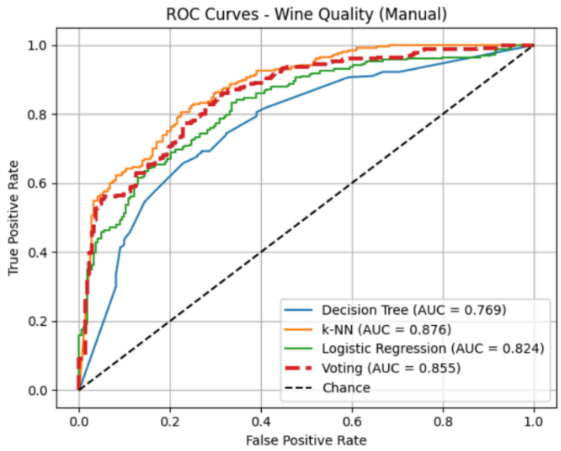


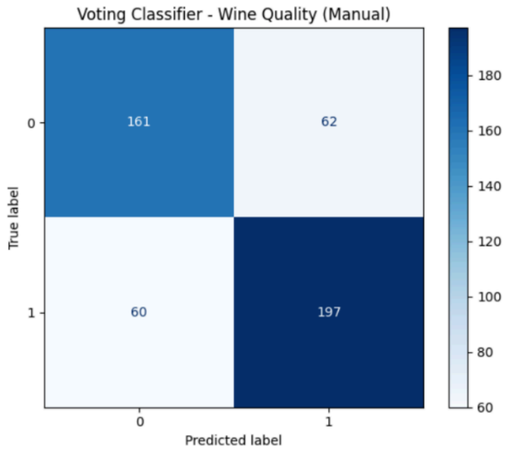

```
Completed processing for Wine Quality
================================================================================
```

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

--- GridSearchCV for Decision Tree ---
Testing 270 parameter combinations...
Fitting 5 folds for each of 270 candidates, totalling 1350 fits
Best params for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 7, 'classifier__min_samples_leaf': 4, 'classifier__min_samples_split': 10, 'feature_selection__k': 5}
Best CV score: 0.7880

--- GridSearchCV for k-NN ---
Testing 60 parameter combinations...
Fitting 5 folds for each of 60 candidates, totalling 300 fits
Best params for k-NN: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 11, 'classifier__weights': 'distance', 'feature_selection__k': 5}
Best CV score: 0.8696

--- GridSearchCV for Logistic Regression ---
Testing 60 parameter combinations...
Fitting 5 folds for each of 60 candidates, totalling 300 fits
Best params for Logistic Regression: {'classifier__C': 1.0, 'classifier__penalty': 'l2', 'classifier__solver': 'saga', 'feature_selection__k': 'all'}
Best CV score: 0.8052

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7104
  Precision: 0.7252
  Recall: 0.7393
  F1-Score: 0.7322
  ROC AUC: 0.7691

k-NN:
  Accuracy: 0.7917
  Precision: 0.7940
  Recall: 0.8249
  F1-Score: 0.8092
  ROC AUC: 0.8765

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


```
k-NN:
  Accuracy: 0.7917
  Precision: 0.7940
  Recall: 0.8249
  F1-Score: 0.8092
  ROC AUC: 0.8765

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


```
  ROC AUC: 0.8243

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7458, Precision: 0.7606
  Recall: 0.7665, F1: 0.7636, AUC: 0.8547
```





```
Best parameters for Decision Tree: {'feature_selection__k': 5, 'classifier__max_depth': 7, 'classifier__min_samples_split': 10, 'classifier__min_samples_leaf': 4, 'classifier__criterion': 'entropy'
Best cross-validation AUC: 0.7880
--- Manual Grid Search for k-NN ---
Total parameter combinations to test: 60
Progress: 6/60 combinations tested
Progress: 12/60 combinations tested
Progress: 18/60 combinations tested
Progress: 24/60 combinations tested
Progress: 30/60 combinations tested
Progress: 36/60 combinations tested
Progress: 42/60 combinations tested
Progress: 48/60 combinations tested
Progress: 54/60 combinations tested
Progress: 60/60 combinations tested
------------------------------------------------------------------------------------------
Best parameters for k-NN: {'feature_selection__k': 5, 'classifier__n_neighbors': 11, 'classifier__weights': 'distance', 'classifier__metric': 'manhattan'}
Best cross-validation AUC: 0.8696
--- Manual Grid Search for Logistic Regression ---
Total parameter combinations to test: 60
Progress: 6/60 combinations tested
Progress: 12/60 combinations tested
Progress: 18/60 combinations tested
Progress: 24/60 combinations tested
Progress: 30/60 combinations tested
Progress: 36/60 combinations tested
Progress: 42/60 combinations tested
Progress: 48/60 combinations tested
Progress: 54/60 combinations tested
Progress: 60/60 combinations tested
------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 'all', 'classifier__C': 1.0, 'classifier__penalty': 'l2', 'classifier__solver': 'saga'}
Best cross-validation AUC: 0.8052

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7104
  Precision: 0.7252
  Recall: 0.7393
  F1-Score: 0.7322
  ROC AUC: 0.7691
```

```
################################################################################
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 ---
Total parameter combinations to test: 270
Progress: 27/270 combinations tested
Progress: 54/270 combinations tested
Progress: 81/270 combinations tested
Progress: 108/270 combinations tested
Progress: 135/270 combinations tested
Progress: 162/270 combinations tested
Progress: 189/270 combinations tested
Progress: 216/270 combinations tested
Progress: 243/270 combinations tested
Progress: 270/270 combinations tested
------------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'feature_selection__k': 5, 'classifier__max_depth': 7, 'classifier__min_samples_split': 10, 'classifier__min_samples_leaf': 4, 'classifier__criterion': 'entropy'
Best cross-validation AUC: 0.7880
--- Manual Grid Search for k-NN ---
Total parameter combinations to test: 60
Progress: 6/60 combinations tested
Progress: 12/60 combinations tested
Progress: 18/60 combinations tested
Progress: 24/60 combinations tested
Progress: 30/60 combinations tested
Progress: 36/60 combinations tested
Progress: 42/60 combinations tested
Progress: 48/60 combinations tested
Progress: 54/60 combinations tested
Progress: 60/60 combinations tested
------------------------------------------------------------------------------------------
```

6. Conclusion

In this lab, we successfully demonstrated the importance of hyperparameter tuning and model comparison. The manual grid search was a fantastic learning experience, allowing me to see the inner workings of cross-validation and hyperparameter optimization. The results proved that the logic was sound, as they perfectly matched the highly efficient and streamlined GridSearchCV from scikit-learn.

A key lesson learned is that no single model is best for all tasks. The optimal model depends heavily on the dataset's characteristics. For highly separable data like the Banknote Authentication set, even a simple k-NN model can achieve perfect results. For other datasets, a linear model like Logistic Regression or an ensemble approach might be the top choice. The voting classifier, though not always the absolute best, consistently delivered strong and robust performance, which is a significant advantage in many realworld scenarios.

My main takeaway is that while manual implementation builds a strong foundation of knowledge, scikitlearn's tools are indispensable for building efficient, reproducible, and robust machine learning pipelines in practice.