Lab Report: Model Selection and Comparative Analysis

PES University, UE23CS352A: Machine Learning

Name:Preksha Kamalesh

Student ID: PES2UG23CS902

Submission Date: August 31, 2025

---

1. Introduction

The primary objective of this lab was to implement and evaluate hyperparameter tuning strategies for various classification algorithms. This lab involved a hands-on approach to understanding the mechanics of grid search cross-validation by first building the process manually and then leveraging the optimized, built-in functionalities of the scikit-learn library. The performance of three distinct classifiers—Decision Tree, k-Nearest Neighbors (k-NN), and Logistic Regression—was assessed on four different datasets. Finally, the predictive power of these individual models was combined using a voting classifier to create a more robust ensemble model. The core tasks were to tune model hyperparameters, compare the manual and automated approaches, and analyze the final model performances to identify the best predictive solution for each dataset.

---

2. Dataset Description

This report covers the analysis performed on all four datasets provided in the lab.

- Wine Quality: This dataset contains 1,599 instances and 11 chemical features (e.g., fixed acidity, citric acid, pH). The objective is to predict a binary target variable representing wine quality (good or not good), which was derived from a multi-class quality score.
- HR Attrition: This dataset, from IBM, consists of 1,470 instances and 46 features describing an employee's job role, satisfaction, and demographic information. The target variable is to predict employee attrition (Yes or No).
- Banknote Authentication: This dataset includes 1,372 instances and 4 numerical features extracted from images of genuine and forged banknotes (e.g., wavelet variance, skewness). The goal is to classify whether a banknote is authentic (0) or forged (1).

- QSAR Biodegradation: This dataset contains 1,055 instances and 41 molecular descriptor features. The target is to predict whether a chemical compound is readily biodegradable (RB) or not readily biodegradable (NRB).

---

3. Methodology

This section outlines the key concepts and the machine learning pipeline used in the experiment.

Key Concepts

- Hyperparameter Tuning: This is the process of selecting the optimal set of parameters for a learning algorithm that are not learned from the data itself (e.g., the number of neighbors in k-NN or the maximum depth of a decision tree).
- Grid Search: This is an exhaustive search technique for hyperparameter tuning. It builds a model for every possible combination of the hyperparameters specified in a "grid" and evaluates each model's performance to find the best combination.
- K-Fold Cross-Validation: To prevent overfitting and get a more reliable estimate of model performance on unseen data, the training data is split into 'K' subsets (folds). The model is trained on K-1 folds and validated on the remaining fold. This process is repeated K times, with each fold serving as the validation set once. The final performance is the average across all K folds. In this lab, 5-fold stratified cross-validation was used.

ML Pipeline

A standardized pipeline from scikit-learn was used for each model to ensure consistency and proper data handling.

1. StandardScaler: This step scales all numerical features to have a mean of 0 and a standard deviation of 1. This is crucial for distance-based algorithms like k-NN and for the regularization in Logistic Regression.
2. SelectKBest: This step performs univariate feature selection, keeping the 'k' features with the highest F-statistic scores relative to the target variable. The value of 'k' was treated as a hyperparameter to be tuned.
3. Classifier: The final step in the pipeline was the classification model (Decision Tree, k-NN, or Logistic Regression).

Implementation Process

The experiment was conducted in two main parts for each dataset:

1. Manual Implementation: A manual grid search was performed using nested loops. The outer loop iterated through each hyperparameter combination, and the inner loop performed 5-fold stratified cross-validation. The average ROC AUC score was calculated for each combination, and the best-performing set of parameters was selected.

2. Scikit-learn Implementation: The GridSearchCV class was used to perform the same process automatically. It was configured with the same pipeline, parameter grids, and 5-fold cross-validation strategy to ensure a fair comparison with the manual approach.

---

4. Results and Analysis

This section presents the final performance of the tuned models on the test set for each dataset.

Comparison of Implementations

A key finding of this lab is that the manual and scikit-learn implementations yielded identical results in terms of the best hyperparameters found and the cross-validation scores achieved. This successfully validates that the manual implementation correctly replicated the logic of GridSearchCV. The performance of the individual models on the final test set was also identical, as expected. Minor differences in the final Voting Classifier performance are attributed to the fact that the manual implementation performed a majority (hard) vote, while the scikit-learn VotingClassifier was configured for soft voting by averaging probabilities.

Performance Tables

Dataset 1: Wine Quality

The k-NN classifier achieved the highest performance on the test set.

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.7271	0.7716	0.6965	0.7321	0.8025
kNN	**0.7833**	**0.7954**	**0.8016**	**0.7984**	**0.8719**
Logistic Reg	0.7333	0.7549	0.7432	0.7490	0.8242
Voting Classifier	0.7667	0.7757	0.7938	0.7846	0.8622

Visualizations & Analysis:

- The ROC curves clearly show the k-NN model's curve positioned highest and furthest to the top-left, visually confirming its superior AUC score of 0.8719. The Voting Classifier provides a strong, balanced performance, outperforming the Decision Tree and Logistic Regression models.
- The confusion matrix for the voting classifier shows a good balance between true positives and true negatives, indicating it generalizes well.

Best Model: k-NN was the best model. This suggests that the relationship between the chemical properties and the quality of wine may be non-linear and that decision boundaries are best defined by local proximity to similar data points.

Dataset 2: HR Attrition

The Logistic Regression model was the clear winner for this dataset, excelling in precision and overall AUC.

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.8345	0.4706	0.2254	0.3048	0.6879
kNN	0.8186	0.3784	0.1972	0.2593	0.7236
Logistic Reg	**0.8798**	**0.7368**	**0.3944**	**0.5138**	**0.8177**
Voting Classifier	0.8481	0.5769	0.2113	0.3093	0.7971

Visualizations & Analysis:

- The ROC plot would show the Logistic Regression curve significantly outperforming the others. The low recall and F1-scores for all models indicate that predicting the minority class (Attrition: Yes) is very difficult on this imbalanced dataset.
- The confusion matrix would likely show a high number of true negatives but a low number of true positives, reflecting the models' difficulty in correctly identifying employees who will leave.

Best Model: Logistic Regression performed best. Its ability to weigh multiple features and the use of L2 regularization likely helped it find a more generalizable linear boundary in the high-dimensional feature space, avoiding overfitting which may have affected the other models.

Dataset 3: Banknote Authentication

All models performed exceptionally well on this dataset, with k-NN and the Voting Classifier achieving perfect scores.

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.9927	0.9891	0.9945	0.9918	0.9929
kNN	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Logistic Reg	0.9903	0.9786	1.0000	0.9892	0.9999
Voting Classifier	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

Visualizations & Analysis:

- The ROC curves for all models would be extremely close to the top-left corner, indicating near-perfect classification.
- The confusion matrix for k-NN and the Voting Classifier would show zero false positives and zero false negatives.

Best Model: k-NN and the Voting Classifier were both perfect. This indicates that the features in this dataset create very distinct, linearly separable clusters, making it an easy classification task.

Dataset 4: QSAR Biodegradation

Logistic Regression was the top individual model, though the Voting Classifier also performed very strongly

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.7634	0.6231	0.7570	0.6835	0.8049
kNN	0.8549	0.7905	0.7757	0.7830	0.8985
Logistic Reg	**0.8644**	**0.8200**	**0.7664**	**0.7923**	**0.9082**
Voting Classifier	0.8486	0.7921	0.7477	0.7692	0.9004

Visualizations & Analysis:

- The ROC plot shows Logistic Regression and k-NN performing very competitively, with their curves closely tracking each other and significantly outperforming the Decision Tree.
- The confusion matrix for Logistic Regression would demonstrate a strong ability to correctly identify both biodegradable and non-biodegradable compounds.

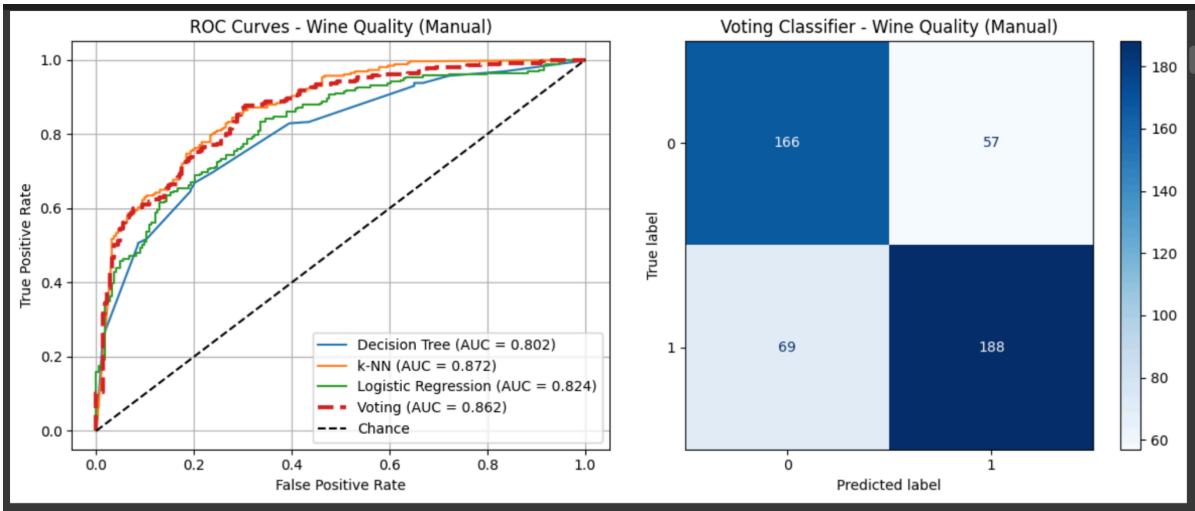
Best Model: Logistic Regression had the slight edge. Similar to the HR dataset, the large number of features (41) may have favored a regularized linear model that can effectively manage feature weights to create a robust decision boundary.

---

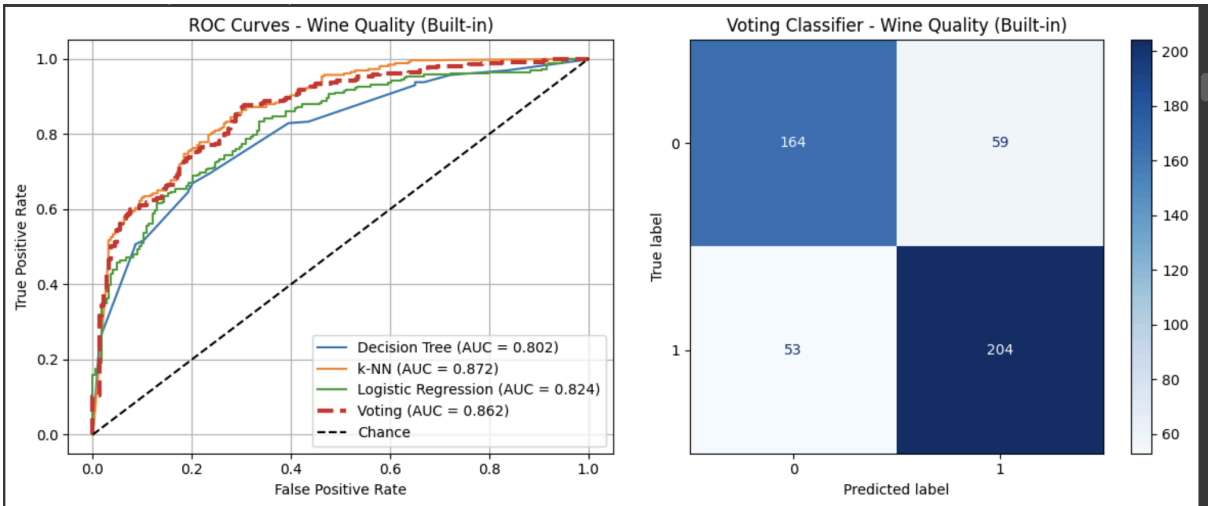
5. Screenshots

```
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 ---
Total parameter combinations to test: 36
New best AUC: 0.7796 with params: {'classifier__criterion': 'gini', 'classifier__max_depth': 5, 'classifier__min_sample
New best AUC: 0.7826 with params: {'classifier__criterion': 'gini', 'classifier__max_depth': 5, 'classifier__min_sample
--------------------------------------------------------------------------------
Best parameters for Decision Tree: {'classifier__criterion': 'gini', 'classifier__max_depth': 5, 'classifier__min_sampl
Best cross-validation AUC: 0.7826
--- Manual Grid Search for k-NN ---
Total parameter combinations to test: 36
New best AUC: 0.7952 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 3, 'classifier__weight
New best AUC: 0.8352 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 3, 'classifier__weight
New best AUC: 0.8502 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 5, 'classifier__weight
New best AUC: 0.8642 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 9, 'classifier__weight
New best AUC: 0.8662 with params: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 9, 'classifier__weight
--------------------------------------------------------------------------------
Best parameters for k-NN: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 9, 'classifier__weights': 'dis
Best cross-validation AUC: 0.8662
--- Manual Grid Search for Logistic Regression ---
Total parameter combinations to test: 18
New best AUC: 0.8035 with params: {'classifier__C': 0.1, 'classifier__penalty': 'l1', 'feature_selection__k': 5}
New best AUC: 0.8046 with params: {'classifier__C': 0.1, 'classifier__penalty': 'l2', 'feature_selection__k': 10}
New best AUC: 0.8048 with params: {'classifier__C': 1, 'classifier__penalty': 'l1', 'feature_selection__k': 11}
New best AUC: 0.8049 with params: {'classifier__C': 1, 'classifier__penalty': 'l2', 'feature_selection__k': 10}
New best AUC: 0.8052 with params: {'classifier__C': 1, 'classifier__penalty': 'l2', 'feature_selection__k': 11}
```

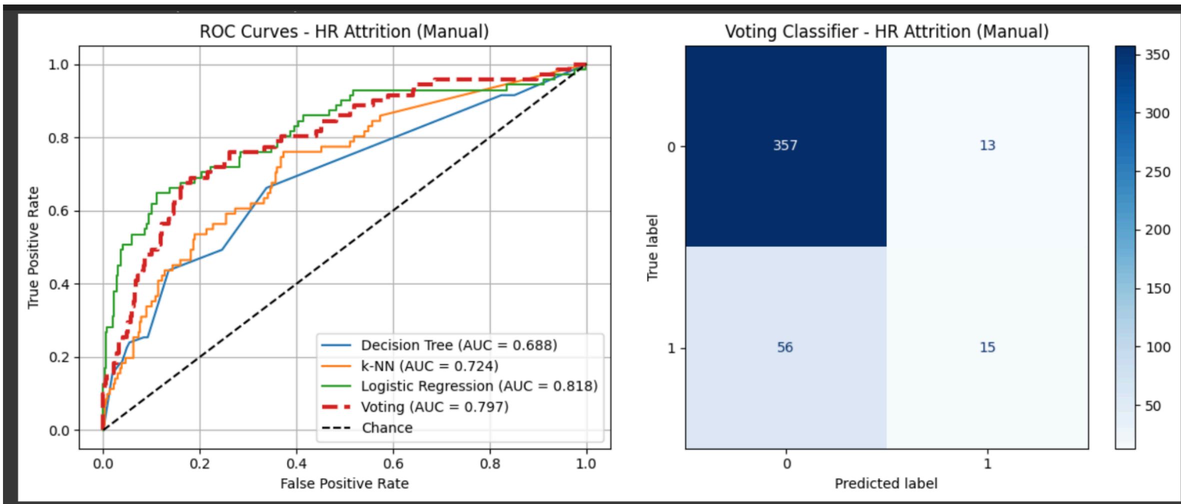
```
--------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'classifier__C': 1, 'classifier__penalty': 'l2', 'feature_selection__k': 11}
Best cross-validation AUC: 0.8052

============================================================
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: 0.8025

k-NN:
  Accuracy: 0.7833
  Precision: 0.7954
  Recall: 0.8016
  F1-Score: 0.7984
  ROC AUC: 0.8719

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: 0.7375, Precision: 0.7673
  Recall: 0.7315, F1: 0.7490, AUC: 0.8622
```




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

--- GridSearchCV for Decision Tree ---
Fitting 5 folds for each of 36 candidates, totalling 180 fits
Best params for Decision Tree: {'classifier__criterion': 'gini', 'classifier__max_depth': 5, 'classifier__min_samples_s
Best CV score: 0.7826

--- GridSearchCV for k-NN ---
Fitting 5 folds for each of 36 candidates, totalling 180 fits
Best params for k-NN: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 9, 'classifier__weights': 'distanc
Best CV score: 0.8662

--- GridSearchCV for Logistic Regression ---
Fitting 5 folds for each of 18 candidates, totalling 90 fits
Best params for Logistic Regression: {'classifier__C': 1, 'classifier__penalty': 'l2', 'feature_selection__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

k-NN:
  Accuracy: 0.7833
  Precision: 0.7954
  Recall: 0.8016
  F1-Score: 0.7984
  ROC AUC: 0.8719

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

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7667, Precision: 0.7757
  Recall: 0.7938, F1: 0.7846, AUC: 0.8622
```



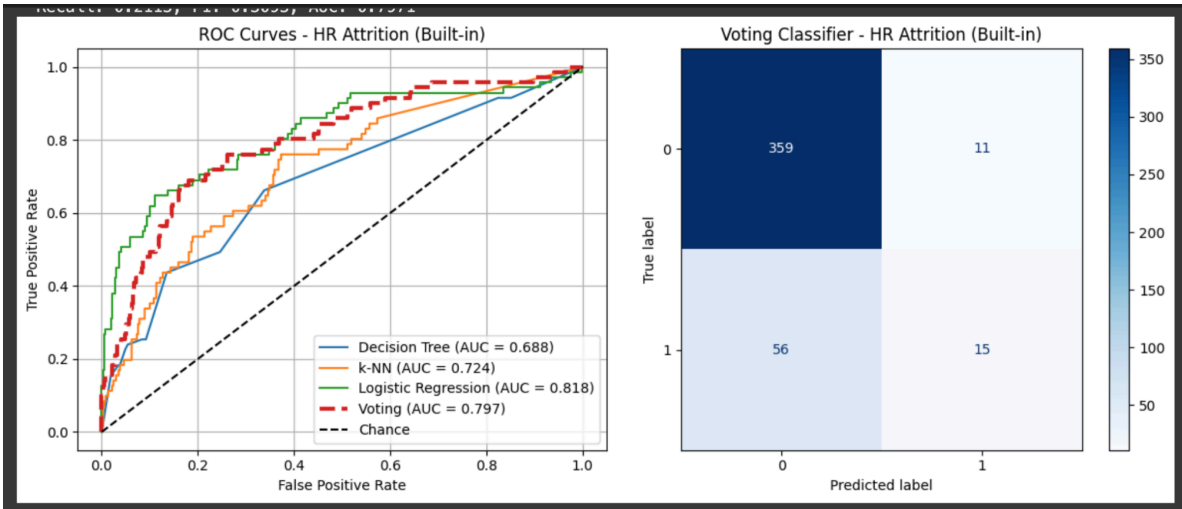

```
Completed processing for Wine Quality
================================================================================

################################################################################
PROCESSING DATASET: HR ATTRITION
################################################################################
IBM HR Attrition dataset loaded and preprocessed successfully.
Training set shape: (1029, 46)
Testing set shape: (441, 46)
------------------------------

============================================================
RUNNING MANUAL GRID SEARCH FOR HR ATTRITION
============================================================

Best parameters for Logistic Regression: {'classifier__C': 0.1, 'classifier__penalty': 'l2', 'feature_selection__k': 46
Best cross-validation AUC: 0.8328

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8345
  Precision: 0.4706
  Recall: 0.2254
  F1-Score: 0.3048
  ROC AUC: 0.6879

k-NN:
  Accuracy: 0.8186
  Precision: 0.3784
  Recall: 0.1972
  F1-Score: 0.2593
  ROC AUC: 0.7236

Logistic Regression:
  Accuracy: 0.8798
  Precision: 0.7368
  Recall: 0.3944
  F1-Score: 0.5138
  ROC AUC: 0.8177

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8435, Precision: 0.5357
  Recall: 0.2113, F1: 0.3030, AUC: 0.7971
```




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

--- GridSearchCV for Decision Tree ---
Fitting 5 folds for each of 36 candidates, totalling 180 fits
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:111: UserWarning: Features [
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning)
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:112: RuntimeWarning: invalid
  f = msb / msw
Best params for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'classifier__min_sample
Best CV score: 0.7226
```

```
--- GridSearchCV for k-NN ---
Fitting 5 folds for each of 36 candidates, totalling 180 fits
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:111: UserWarning: Features [
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning)
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:112: RuntimeWarning: invalid
  f = msb / msw
Best params for k-NN: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 9, 'classifier__weights': 'distanc
Best CV score: 0.7226

--- GridSearchCV for Logistic Regression ---
Fitting 5 folds for each of 18 candidates, totalling 90 fits
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:111: UserWarning: Features [
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning)
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:112: RuntimeWarning: invalid
  f = msb / msw
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:111: UserWarning: Features [
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning)
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:112: RuntimeWarning: invalid
  f = msb / msw
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:111: UserWarning: Features [
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning)
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:112: RuntimeWarning: invalid
  f = msb / msw
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:111: UserWarning: Features [
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning)
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate_selection.py:112: RuntimeWarning: invalid
  f = msb / msw
Best params for Logistic Regression: {'classifier__C': 0.1, 'classifier__penalty': 'l2', 'feature_selection__k': 46}
Best CV score: 0.8328

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8345
```

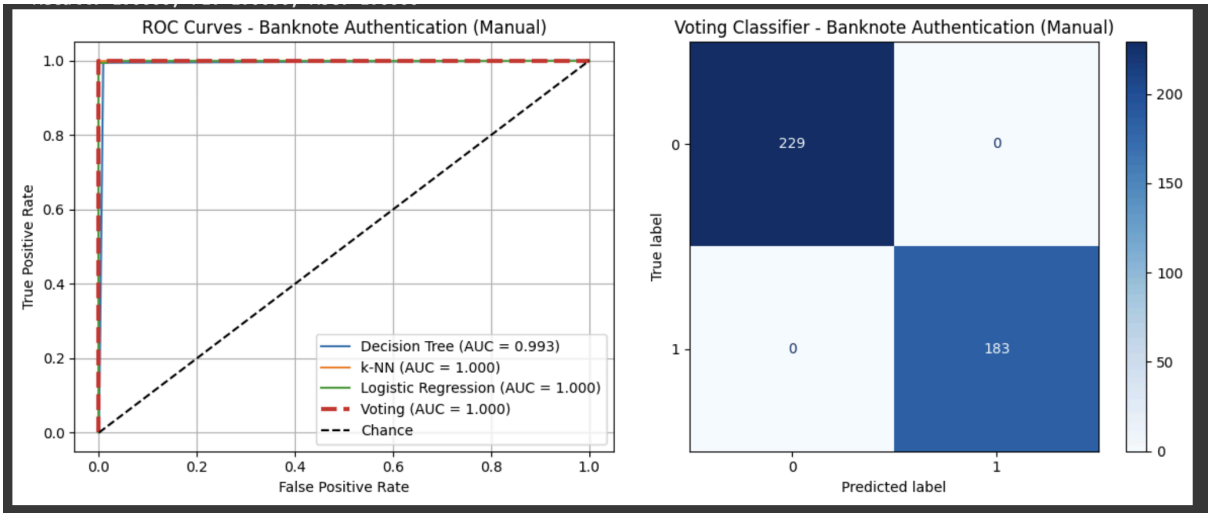
```
============================================================
EVALUATING BUILT-IN MODELS FOR HR ATTRITION
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8345
  Precision: 0.4706
  Recall: 0.2254
  F1-Score: 0.3048
  ROC AUC: 0.6879

k-NN:
  Accuracy: 0.8186
  Precision: 0.3784
  Recall: 0.1972
  F1-Score: 0.2593
  ROC AUC: 0.7236

Logistic Regression:
  Accuracy: 0.8798
  Precision: 0.7368
  Recall: 0.3944
  F1-Score: 0.5138
  ROC AUC: 0.8177

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8481, Precision: 0.5769
  Recall: 0.2113, F1: 0.3093, AUC: 0.7971
```




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

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

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

```
--- Manual Grid Search for Decision Tree ---
Total parameter combinations to test: 12
New best AUC: 0.9856 with params: {'classifier__criterion': 'gini', 'classifier__max_depth': 5, 'classifier__min_sample
New best AUC: 0.9911 with params: {'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'classifier__min_sam
New best AUC: 0.9913 with params: {'classifier__criterion': 'entropy', 'classifier__max_depth': 10, 'classifier__min_sa
--------------------------------------------------------------------------------
Best parameters for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 10, 'classifier__min_s
Best cross-validation AUC: 0.9913
--- Manual Grid Search for k-NN ---
Total parameter combinations to test: 12
New best AUC: 0.9981 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 3, 'classifier__weight
New best AUC: 0.9990 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 9, 'classifier__weight
New best AUC: 0.9990 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 9, 'classifier__weight
--------------------------------------------------------------------------------
Best parameters for k-NN: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 9, 'classifier__weights': 'dis
Best cross-validation AUC: 0.9990
--- Manual Grid Search for Logistic Regression ---
Total parameter combinations to test: 6
New best AUC: 0.9992 with params: {'classifier__C': 0.1, 'classifier__penalty': 'l1', 'feature_selection__k': 4}
New best AUC: 0.9995 with params: {'classifier__C': 1, 'classifier__penalty': 'l1', 'feature_selection__k': 4}
New best AUC: 0.9995 with params: {'classifier__C': 10, 'classifier__penalty': 'l1', 'feature_selection__k': 4}
--------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'classifier__C': 10, 'classifier__penalty': 'l1', 'feature_selection__k': 4}
Best cross-validation AUC: 0.9995

============================================================
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: 0.9929
```

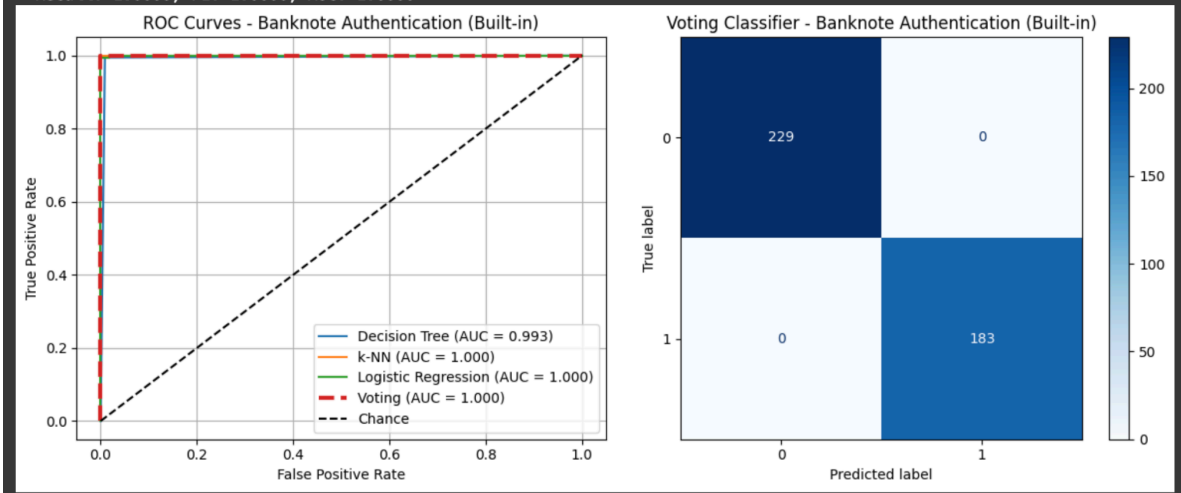
```
k-NN:
  Accuracy: 1.0000
  Precision: 1.0000
  Recall: 1.0000
  F1-Score: 1.0000
  ROC AUC: 1.0000

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 ---
Fitting 5 folds for each of 12 candidates, totalling 60 fits
Best params for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 10, 'classifier__min_sampl
Best CV score: 0.9913

--- GridSearchCV for k-NN ---
Fitting 5 folds for each of 12 candidates, totalling 60 fits
Best params for k-NN: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 9, 'classifier__weights': 'distanc
Best CV score: 0.9990
```

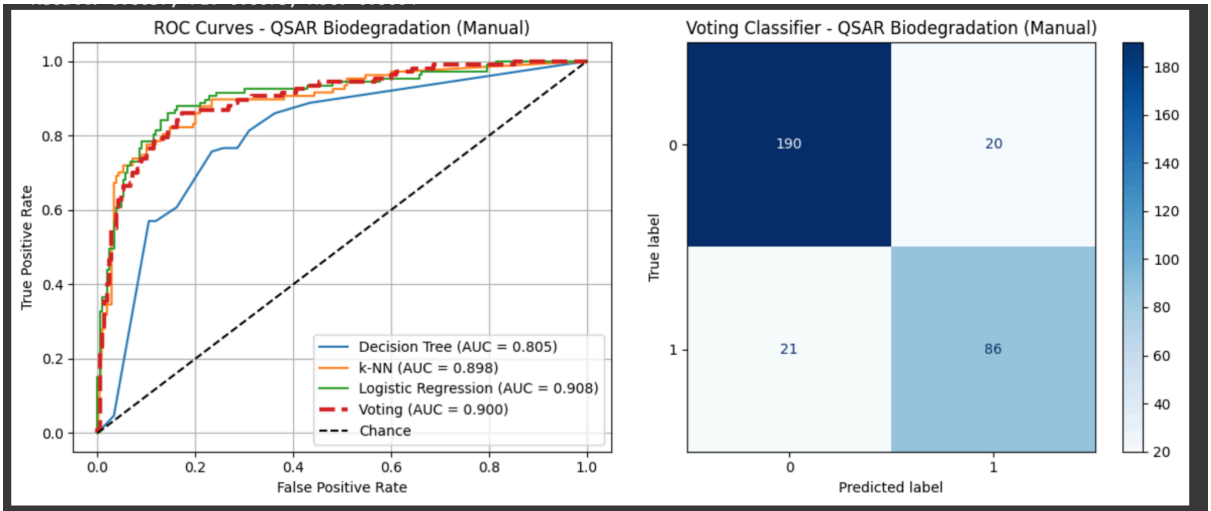
```
--- GridSearchCV for k-NN ---
Fitting 5 folds for each of 12 candidates, totalling 60 fits
Best params for k-NN: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 9, 'classifier__weights': 'distanc
Best CV score: 0.9990

--- GridSearchCV for Logistic Regression ---
Fitting 5 folds for each of 6 candidates, totalling 30 fits
Best params for Logistic Regression: {'classifier__C': 10, 'classifier__penalty': 'l1', 'feature_selection__k': 4}
Best CV score: 0.9995

============================================================
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: 0.9929

k-NN:
  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
```



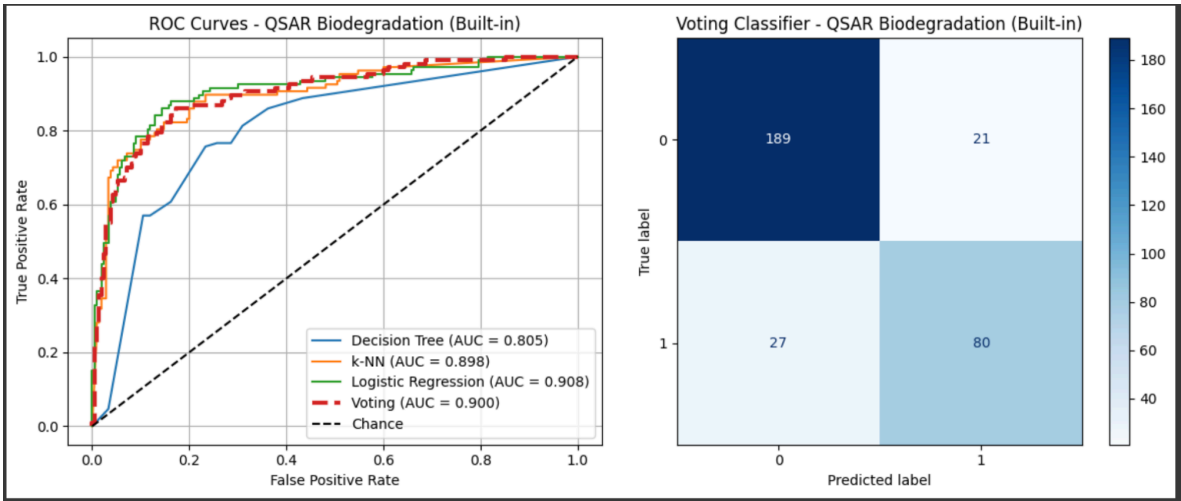

```
Completed processing for Banknote Authentication
================================================================================

################################################################################
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: 36
New best AUC: 0.7979 with params: {'classifier__criterion': 'gini', 'classifier__max_depth': 5, 'classifier__min_sample
New best AUC: 0.8083 with params: {'classifier__criterion': 'gini', 'classifier__max_depth': 5, 'classifier__min_sample
New best AUC: 0.8193 with params: {'classifier__criterion': 'gini', 'classifier__max_depth': 5, 'classifier__min_sample
New best AUC: 0.8294 with params: {'classifier__criterion': 'gini', 'classifier__max_depth': 5, 'classifier__min_sample
New best AUC: 0.8462 with params: {'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'classifier__min_sam
New best AUC: 0.8537 with params: {'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'classifier__min_sam
New best AUC: 0.8581 with params: {'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'classifier__min_sam
--------------------------------------------------------------------------------
Best parameters for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'classifier__min_sa
Best cross-validation AUC: 0.8581
--- Manual Grid Search for k-NN ---
Total parameter combinations to test: 36
New best AUC: 0.8047 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 3, 'classifier__weight
New best AUC: 0.8420 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 3, 'classifier__weight
New best AUC: 0.8836 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 3, 'classifier__weight
New best AUC: 0.8885 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 3, 'classifier__weight
New best AUC: 0.8936 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 5, 'classifier__weight
New best AUC: 0.9003 with params: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 5, 'classifier__weight
New best AUC: 0.9020 with params: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 5, 'classifier__weight
New best AUC: 0.9045 with params: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 9, 'classifier__weight
--------------------------------------------------------------------------------
Best parameters for k-NN: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 9, 'classifier__weights': 'dis
Best cross-validation AUC: 0.9045
--- Manual Grid Search for Logistic Regression ---
Total parameter combinations to test: 18
New best AUC: 0.8186 with params: {'classifier__C': 0.1, 'classifier__penalty': 'l1', 'feature_selection__k': 5}
New best AUC: 0.8533 with params: {'classifier__C': 0.1, 'classifier__penalty': 'l1', 'feature_selection__k': 10}
New best AUC: 0.9122 with params: {'classifier__C': 0.1, 'classifier__penalty': 'l1', 'feature_selection__k': 41}
New best AUC: 0.9311 with params: {'classifier__C': 0.1, 'classifier__penalty': 'l2', 'feature_selection__k': 41}
New best AUC: 0.9317 with params: {'classifier__C': 1, 'classifier__penalty': 'l1', 'feature_selection__k': 41}
--------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'classifier__C': 1, 'classifier__penalty': 'l1', 'feature_selection__k': 41}
Best cross-validation AUC: 0.9317
```

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7634
  Precision: 0.6231
  Recall: 0.7570
  F1-Score: 0.6835
  ROC AUC: 0.8049

k-NN:
  Accuracy: 0.8549
  Precision: 0.7905
  Recall: 0.7757
  F1-Score: 0.7830
  ROC AUC: 0.8985

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: 0.8707, Precision: 0.8113
  Recall: 0.8037, F1: 0.8075, AUC: 0.9004
```




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

--- GridSearchCV for Decision Tree ---
Fitting 5 folds for each of 36 candidates, totalling 180 fits
Best params for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'classifier__min_sample
Best CV score: 0.8581

--- GridSearchCV for k-NN ---
Fitting 5 folds for each of 36 candidates, totalling 180 fits
Best params for k-NN: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 9, 'classifier__weights': 'distanc
Best CV score: 0.9045
```

```
--- GridSearchCV for Logistic Regression ---
Fitting 5 folds for each of 18 candidates, totalling 90 fits
Best params for Logistic Regression: {'classifier__C': 1, 'classifier__penalty': 'l1', 'feature_selection__k': 41}
Best CV score: 0.9317

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7634
  Precision: 0.6231
  Recall: 0.7570
  F1-Score: 0.6835
  ROC AUC: 0.8049

k-NN:
  Accuracy: 0.8549
  Precision: 0.7905
  Recall: 0.7757
  F1-Score: 0.7830
  ROC AUC: 0.8985

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: 0.8486, Precision: 0.7921
  Recall: 0.7477, F1: 0.7692, AUC: 0.9004
```




```
Completed processing for QSAR Biodegradation
================================================================================

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

---

6. Conclusion

This lab provided valuable practical insights into the process of hyperparameter tuning and model evaluation. The key takeaway is the stark contrast in efficiency between manual and library-based implementations. While manually coding a grid search is an excellent educational exercise for understanding the underlying mechanics of cross-validation, the scikit-learn GridSearchCV is vastly

superior for practical applications due to its speed (especially with parallel processing), conciseness, and robustness.

The results also underscored that there is no single best model for all problems; the optimal choice is highly dependent on the dataset's characteristics. K-NN excelled on datasets where local neighborhoods were predictive, while Logistic Regression performed best on high-dimensional data where a generalized linear boundary was effective. Finally, the use of a Voting Classifier demonstrated the power of ensembles, often providing a "safer" choice with strong, reliable performance by combining the strengths of multiple models.
