Model Selection and Comparative Analysis

NAME: NANDITA R NADIG
ID: PES2UG23CS365
COURSE NAME: MACHINE- LEARNING (UE23CS352A)
SUBMISSION DATE: 30.08.2025

Introduction

The purpose of this lab assignment is to explore model selection and performance evaluation using hyperparameter tuning and classifier comparison on multiple datasets. Two main approaches are implemented:

1. **Manual Grid Search:** A custom implementation of hyperparameter tuning where parameter grids were manually iterated, models were trained and validated using cross-validation, and the best configurations were identified.
2. **Scikit-learn GridSearchCV:** The same process was automated using scikit-learn's GridSearchCV, which performs systematic hyperparameter search with integrated cross-validation.

By applying these two approaches, the assignment aimed to:

1. Understand the importance of hyperparameter tuning in optimizing model performance.
2. Compare the effectiveness and efficiency of manual implementation versus library-based implementation.
3. Evaluate different classifiers across datasets using a consistent ML pipeline.

Dataset description

Wine Quality Dataset

- Number of instances: 1599
- Number of features: 11
- Target Variable: Wine quality (binary: good vs not good)

HR Attrition Dataset

- Number of instances: 1470
- Number of features: 35
- Target Variable: Employee attrition (binary: Yes vs No)

Banknote Authentication Dataset

- Number of instances: 1375
- Number of features: 4
- Target Variable: Banknote type (binary: genuine vs forged)

QSAR Biodegradation Dataset

- Number of instances: 1055
- Number of features: 41
- Target Variable: Biodegradability (binary: readily biodegradable vs not biodegradable)

Methodology

Hyperparameter Tuning - The process of adjusting the external parameters of a machine learning model, such as the maximum depth of a decision tree or the number of estimators in a random

forest, to achieve the best performance. These parameters are not learned from the data but are set before training and tuning them ensures that the model generalizes well to unseen data.

Grid Search - A method for hyperparameter tuning. It works by exhaustively trying out all possible combinations of a predefined set of hyperparameter values. For each combination, the model is trained and evaluated, and the best-performing set of parameters is selected based on the chosen evaluation metric. While it can be computationally expensive, Grid Search ensures that no possible option within the defined grid is missed.

K-Fold Cross-Validation - A resampling technique that provides a more reliable estimate of model performance. In this method, the dataset is divided into *K* equal parts (folds). The model is trained on *K-1* folds and tested on the remaining fold, and this process is repeated *K* times so that each fold acts once as the test set. The average of the results gives the final performance score, reducing the bias that can occur from a single train-test split.

Pipeline:

The machine learning pipeline is a structured workflow that ensures data is processed and modelled in a consistent and repeatable way. This assignment uses a pipeline involving StandardScaler, VarianceThreshold, SelectKBest, and a Classifier.

1. **StandardScaler** standardizes the dataset by centering the features around zero and scaling them to unit variance. This step helps improve the performance of algorithms that are sensitive to the scale of input features.
2. **VarianceThreshold** is applied to remove low-variance features that contribute little to distinguishing between classes. By eliminating constant or nearly constant columns, the model is left with more informative features.
3. **SelectKBest** further reduces dimensionality by ranking features according to statistical tests and selecting the top *k* most relevant ones. This improves efficiency and focuses the classifier on the strongest predictors.
4. Finally, the processed data is passed into a **Classifier** (such as Decision Tree, Random Forest, or Logistic Regression), which learns from the training data and makes predictions on unseen data.

Procedure:

Before performing hyperparameter tuning, a parameter grid was explicitly defined for each classifier (Decision Tree, Random Forest, Logistic Regression). The grid specifies possible values for feature selection (k) and classifier parameters. The objective is to identify the parameter combination that yields the best performance based on AUC using cross-validation.

1) Manual Implementation

- Adjust the parameter grid so that feature selection parameter $k \leq$ total number of features (*n_features*), keeping only valid values *('all' or ≤ n_features)*.
- Generate all possible hyperparameter combinations using itertools.product.

- Perform the 5-fold StratifiedKFold cross-validation for each combination to preserve class distribution.
- In each fold:
  - Split data into training and validation subsets.
  - Build a pipeline with VarianceThreshold, StandardScaler, SelectKBest and the chosen classifier.
  - Update pipeline with the current hyperparameter set.
  - Train the pipeline on the training fold.
  - Predict probabilities on the validation fold.
  - Compute and store the AUC score.
- Calculate mean AUC across the 5 folds for each parameter combination.
- Compare with the best score so far and updated the best combination if higher.
- Report the best hyperparameter set with its corresponding AUC score.

2) Using GridSearchCV

- *feature_selection__k* values are restricted so that they did not exceed the total number of features (*n_features*).
- Build a pipeline with VarianceThreshold (constant feature removal), StandardScaler (normalization), SelectKBest (feature selection), and the chosen classifier.
- Apply 5-fold StratifiedKFold with shuffle and a fixed random seed for reproducibility.
- Initialise GridSearchCV with the pipeline, adjusted parameter grid, and AUC as the scoring metric, using parallel execution *(n_jobs=-1)*.
- Fit the grid search on the training dataset, where GridSearchCV will automatically evaluate all parameter combinations across folds.
- Retrieve the best estimator, best parameter set, and cross-validation performance metrics from the fitted GridSearchCV object.

Results and Analysis

1. Wine Quality

Manual Implementation:

Classifier	Accuracy	Precision	Recall		F1-Score	ROC AUC
Decision Tree	0.7271	0.7716	0.6965		0.7321	0.8025
KNN	0.7750	0.7854	0.7977		0.7915	0.8679
Logistic Regression	0.7333	0.7549	0.7432		0.7490	0.8242

Voting Classifier	0.7354	0.7642	0.7315		0.7475	0.8607

GridSearchCV Implementation:

Classifier	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.7271	0.7716	0.6965	0.7321	0.8025
KNN	0.7750	0.7854	0.7977	0.7915	0.8679
Logistic Regression	0.7333	0.7549	0.7432	0.7490	0.8242
Voting Classifier	0.7729	0.7868	0.7899	0.7883	0.8607

Observation: Both implementations produced identical results for individual classifiers, while the Voting Classifier performed slightly better under GridSearchCV. KNN achieved the highest overall AUC.

2. HR Attrition

Manual Implementation:

Classifier	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.8231	0.3333	0.0986	0.1522	0.7107
KNN	0.8186	0.3784	0.1972	0.2593	0.7236
Logistic Regression	0.8798	0.7368	0.3944	0.5138	0.8177
Voting Classifier	0.8299	0.4231	0.1549	0.2268	0.7960

GridSearchCV Implementation:

Classifier	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.8231	0.3333	0.0986	0.1522	0.7107
KNN	0.8186	0.3784	0.1972	0.2593	0.7236
Logistic Regression	0.8798	0.7368	0.3944	0.5138	0.8177
Voting Classifier	0.8458	0.5556	0.2113	0.3061	0.7960

Observation: Logistic Regression clearly outperformed the others in terms of AUC and balanced performance. GridSearchCV gave the Voting Classifier a slight edge in accuracy and F1.

3. Banknote Authentication

Manual Implementation:

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

GridSearchCV Implementation:

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

Observation: Both methods gave nearly perfect results. The dataset is highly separable, which explains the flawless performance from KNN, Logistic Regression, and the Voting Classifier.

4. QSAR Biodegradation

Manual Implementation:

Classifier	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.7634	0.6600	0.6168	0.6377	0.8007
KNN	0.8644	0.8077	0.7850	0.7962	0.8931
Logistic Regression	0.8644	0.8200	0.7664	0.7923	0.9082
Voting Classifier	0.8644	0.8200	0.7664	0.7923	0.9100

GridSearchCV Implementation:

Classifier	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.7634	0.6600	0.6168	0.6377	0.8007
KNN	0.8644	0.8077	0.7850	0.7962	0.8931
Logistic Regression	0.8644	0.8200	0.7664	0.7923	0.9082
Voting Classifier	0.8675	0.8218	0.7757	0.7981	0.9100

Observation: Both implementations again gave nearly identical results. Logistic Regression and KNN performed strongly, with the Voting Classifier giving a marginal boost under GridSearchCV.

Visualizations:

1) Wine Quality Dataset:

All models did well here, with AUC scores between 0.802 and 0.868. The ROC curves show that **KNN (AUC = 0.868)** and the **Voting Classifier (AUC = 0.861)** are the best performers.

Looking at the confusion matrices:

- Manual Voting Classifier: Correctly predicted 165 class 0s and 188 class 1s, but it also misclassified 58 class 0s and 69 class 1s.
- Built-in Voting Classifier: Performed a little better, getting 168 class 0s and 203 class 1s right, with fewer mistakes (55 class 0s and 54 class 1s misclassified)

2) HR Attrition Dataset:

The ROC curves show that **Logistic Regression** is the best standalone model **(AUC = 0.818)**

The confusion matrices highlight a strong bias toward the majority class (no attrition):

- Manual Voting Classifier: High true negatives (355), but very poor recall for attrition (only 11 true positives).
- Built-in Voting Classifier: Slightly better with 15 true positives but still misses most attrition cases.

3) Banknote Authentication Dataset:

The ROC curves are almost perfect: **KNN, Logistic Regression, and Voting Classifier all achieve AUC = 1.000**, while Decision Tree is just a bit lower at 0.985.

The confusion matrices show perfect predictions for both manual and built-in versions:

- Manual Voting Classifier: 100% correct (229 class 0 and 183 class 1).
- Built-in Voting Classifier: Also 100% correct.

4) QSAR Biodegradation Dataset:

ROC curves show **Voting Classifier (0.910)** and **Logistic Regression (0.908)** as the strongest

The confusion matrices show only minor differences:

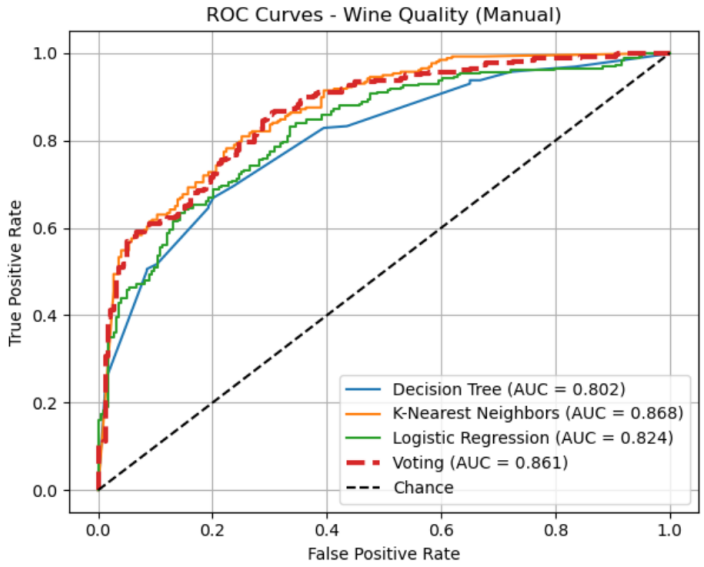
- Manual Voting Classifier: 192 true negatives, 82 true positives, 25 false negatives.
- Built-in Voting Classifier: Almost identical, but slightly better recall (83 true positives, 24 false negatives).

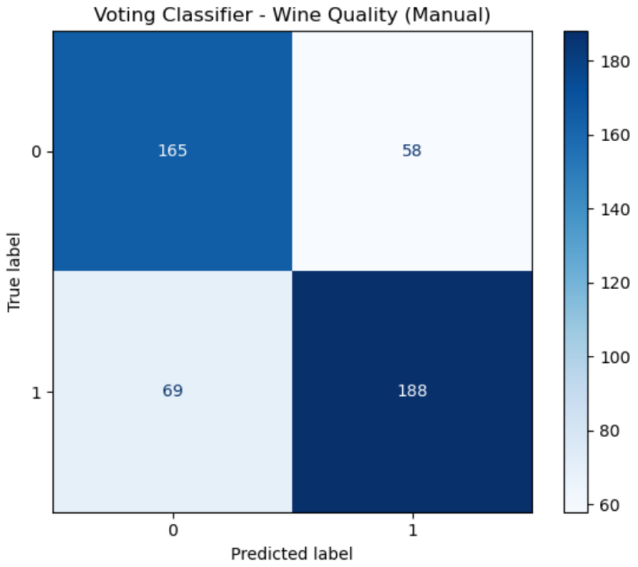
Best Models:

1. Wine Quality**:** *KNN*
2. HR Attrition: *Logistic Regression*
3. Banknote Authentication: *All models*
4. QSAR Biodegradation: *Voting Classifier / Logistic Regression*

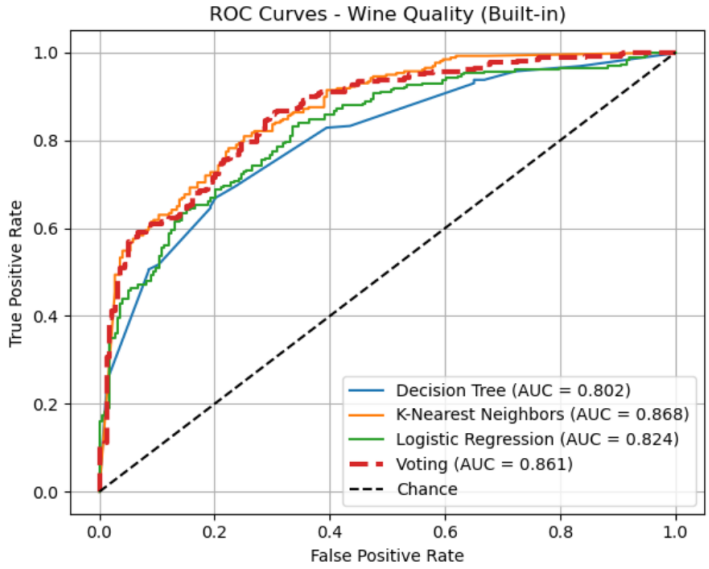
Screenshots

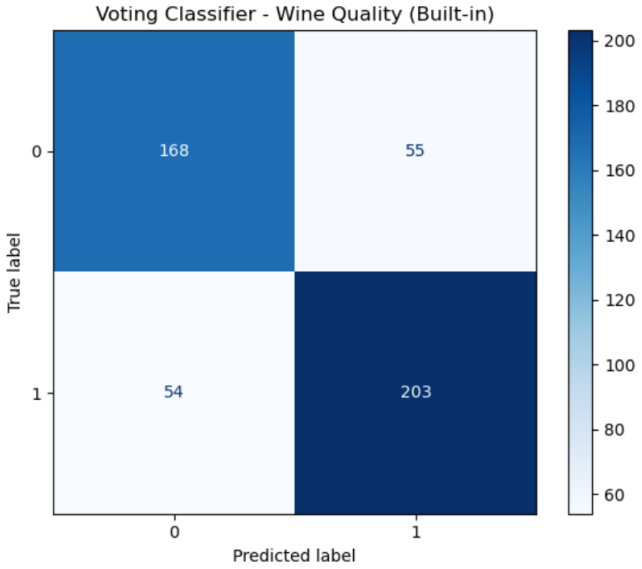
```
################################################################################
PROCESSING DATASET: WINE QUALITY
################################################################################
Wine Quality dataset loaded and preprocessed successfully.
Training set shape: (1119, 11)
Testing set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

============================================================
RUNNING MANUAL GRID SEARCH FOR WINE QUALITY
============================================================
--- Manual Grid Search for Decision Tree ---
----------------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'classifier__max_depth': 5, 'classifier__min_samples_split': 5, 'feature_selection__k': 5}
Best cross-validation AUC: 0.7832
--- Manual Grid Search for K-Nearest Neighbors ---
----------------------------------------------------------------------------------------------
Best parameters for K-Nearest Neighbors: {'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'feature_selection__k': 5}
Best cross-validation AUC: 0.8642
--- Manual Grid Search for Logistic Regression ---
----------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'classifier__C': 1.0, 'classifier__penalty': 'l2', 'feature_selection__k': 'all'}
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-Nearest Neighbors:
  Accuracy: 0.7750
  Precision: 0.7854
  Recall: 0.7977
  F1-Score: 0.7915
  ROC AUC: 0.8679

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: 0.7642
  Recall: 0.7315, F1: 0.7475, AUC: 0.8607
```



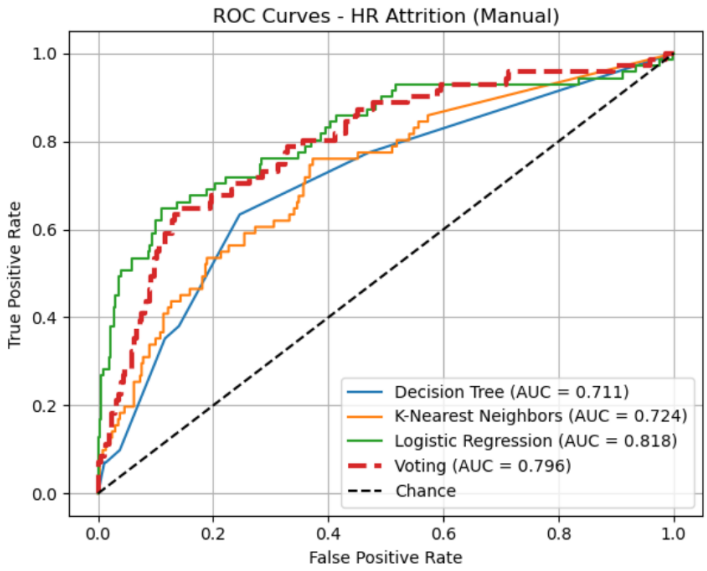


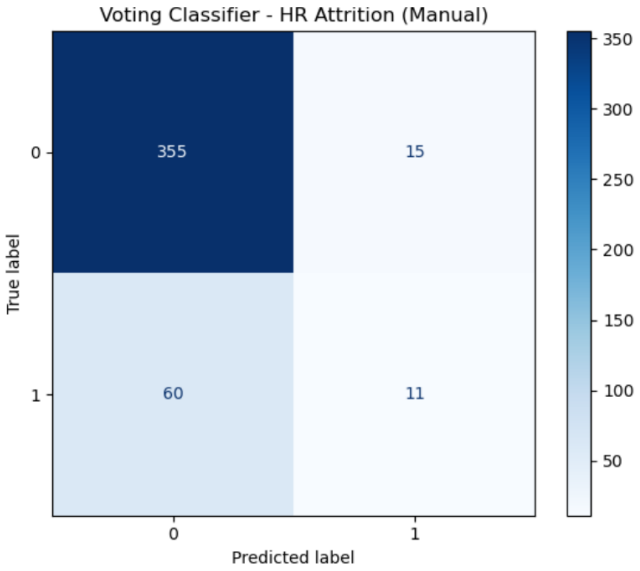

```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR WINE QUALITY
============================================================

--- GridSearchCV for Decision Tree ---
Fitting 5 folds for each of 36 candidates, totalling 180 fits
Best params for Decision Tree: {'classifier__max_depth': 5, 'classifier__min_samples_split': 5, 'feature_selection__k': 5}
Best CV score: 0.7832

--- GridSearchCV for K-Nearest Neighbors ---
Fitting 5 folds for each of 24 candidates, totalling 120 fits
Best params for K-Nearest Neighbors: {'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'feature_selection__k': 5}
Best CV score: 0.8642

--- GridSearchCV for Logistic Regression ---
Fitting 5 folds for each of 24 candidates, totalling 120 fits
Best params for Logistic Regression: {'classifier__C': 1.0, 'classifier__penalty': 'l2', 'feature_selection__k': 'all'}
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-Nearest Neighbors:
  Accuracy: 0.7750
  Precision: 0.7854
  Recall: 0.7977
  F1-Score: 0.7915
  ROC AUC: 0.8679

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: 0.7729, Precision: 0.7868
  Recall: 0.7899, F1: 0.7883, AUC: 0.8607
```



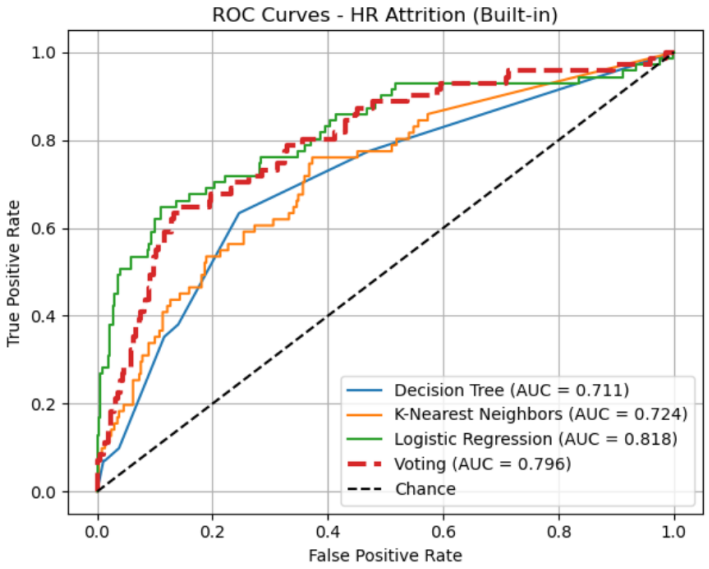


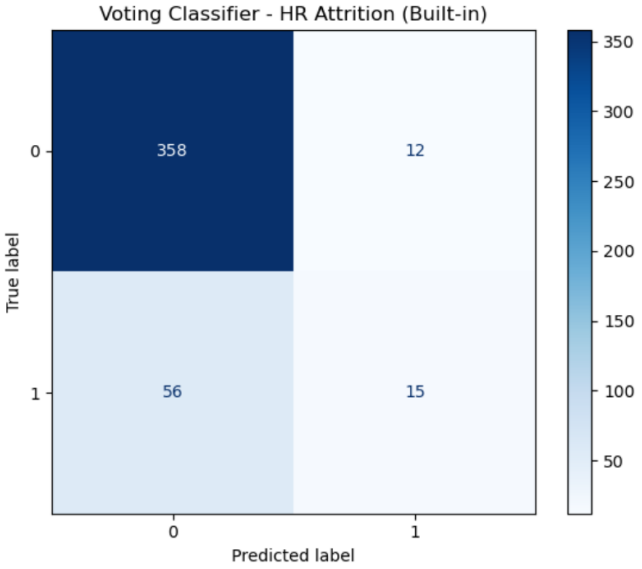

```
################################################################################
PROCESSING DATASET: HR ATTRITION
################################################################################
IBM HR Attrition dataset loaded and preprocessed successfully.
Training set shape: (1029, 46)
Testing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

============================================================
RUNNING MANUAL GRID SEARCH FOR HR ATTRITION
============================================================
--- Manual Grid Search for Decision Tree ---
----------------------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'classifier__max_depth': 3, 'classifier__min_samples_split': 2, 'feature_selection__k': 5}
Best cross-validation AUC: 0.7152
--- Manual Grid Search for K-Nearest Neighbors ---
----------------------------------------------------------------------------------------------------
Best parameters for K-Nearest Neighbors: {'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'feature_selection__k': 10}
Best cross-validation AUC: 0.7226
--- Manual Grid Search for Logistic Regression ---
----------------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'classifier__C': 0.1, 'classifier__penalty': 'l2', 'feature_selection__k': 'all'}
Best cross-validation AUC: 0.8328

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8231
  Precision: 0.3333
  Recall: 0.0986
  F1-Score: 0.1522
  ROC AUC: 0.7107

K-Nearest Neighbors:
  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: 0.8299, Precision: 0.4231
  Recall: 0.1549, F1: 0.2268, AUC: 0.7960
```



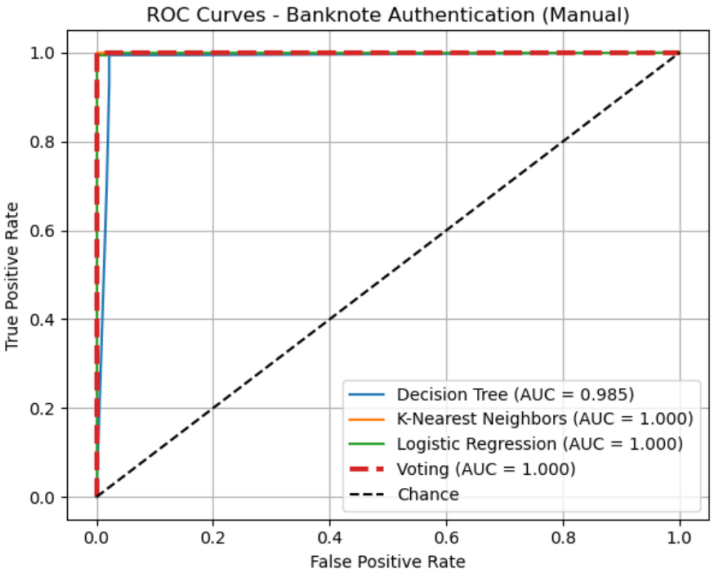


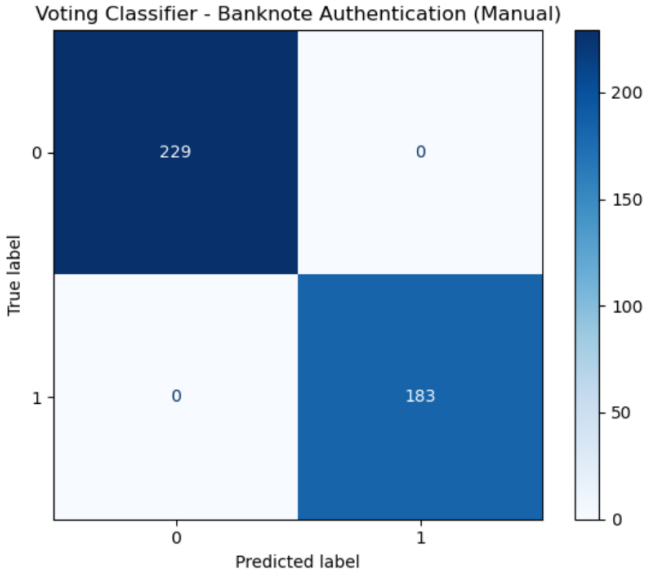

```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR HR ATTRITION
============================================================

--- GridSearchCV for Decision Tree ---
Fitting 5 folds for each of 36 candidates, totalling 180 fits
Best params for Decision Tree: {'classifier__max_depth': 3, 'classifier__min_samples_split': 2, 'feature_selection__k': 5}
Best CV score: 0.7152

--- GridSearchCV for K-Nearest Neighbors ---
Fitting 5 folds for each of 24 candidates, totalling 120 fits
Best params for K-Nearest Neighbors: {'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'feature_selection__k': 10}
Best CV score: 0.7226

--- GridSearchCV for Logistic Regression ---
Fitting 5 folds for each of 24 candidates, totalling 120 fits
Best params for Logistic Regression: {'classifier__C': 0.1, 'classifier__penalty': 'l2', 'feature_selection__k': 'all'}
Best CV score: 0.8328

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8231
  Precision: 0.3333
  Recall: 0.0986
  F1-Score: 0.1522
  ROC AUC: 0.7107

K-Nearest Neighbors:
  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: 0.8458, Precision: 0.5556
  Recall: 0.2113, F1: 0.3061, AUC: 0.7960
```



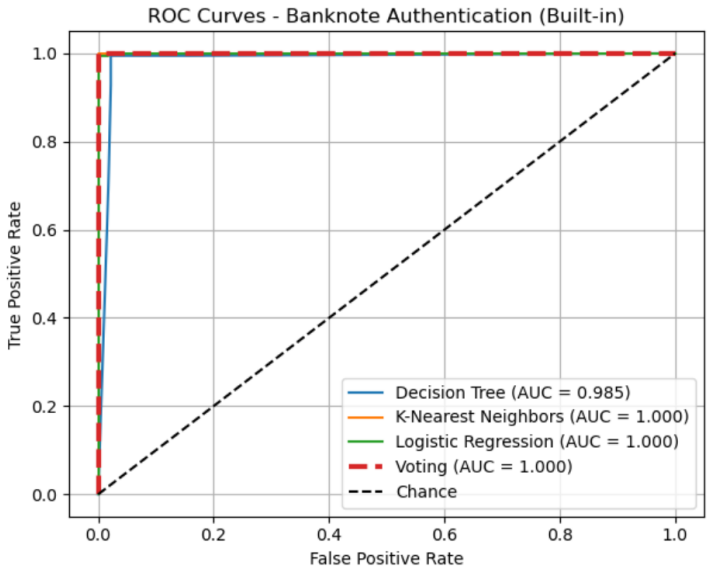


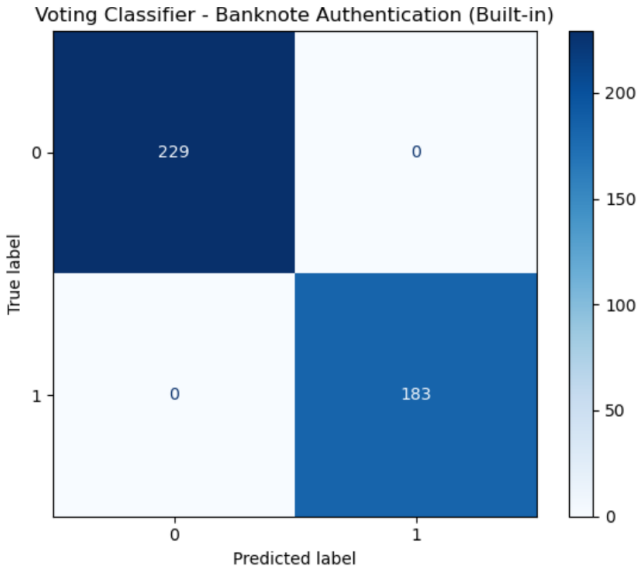

```
Completed processing for HR Attrition
================================================================================

################################################################################
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: {'classifier__max_depth': 5, 'classifier__min_samples_split': 2, 'feature_selection__k': 'all'}
Best cross-validation AUC: 0.9856
--- Manual Grid Search for K-Nearest Neighbors ---
----------------------------------------------------------------------------------------------------
Best parameters for K-Nearest Neighbors: {'classifier__n_neighbors': 7, 'classifier__weights': 'distance', 'feature_selection__k': 'all'}
Best cross-validation AUC: 0.9990
--- Manual Grid Search for Logistic Regression ---
----------------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'classifier__C': 10.0, 'classifier__penalty': 'l1', 'feature_selection__k': 'all'}
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

K-Nearest Neighbors:
  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
```



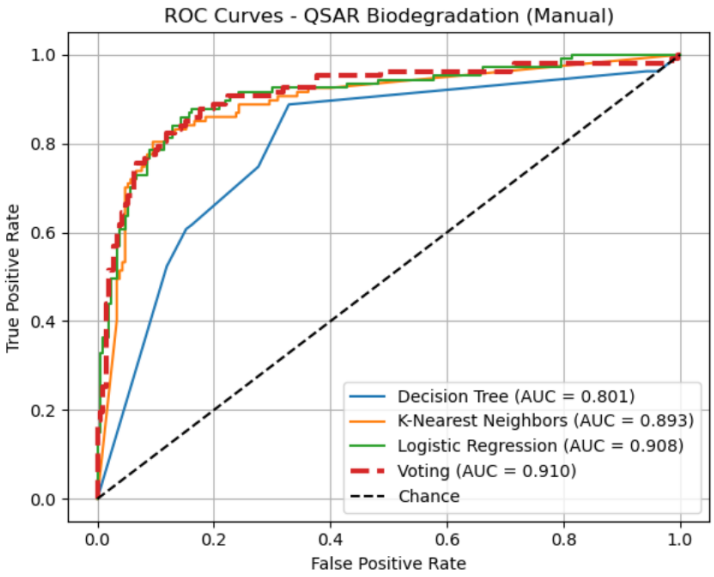


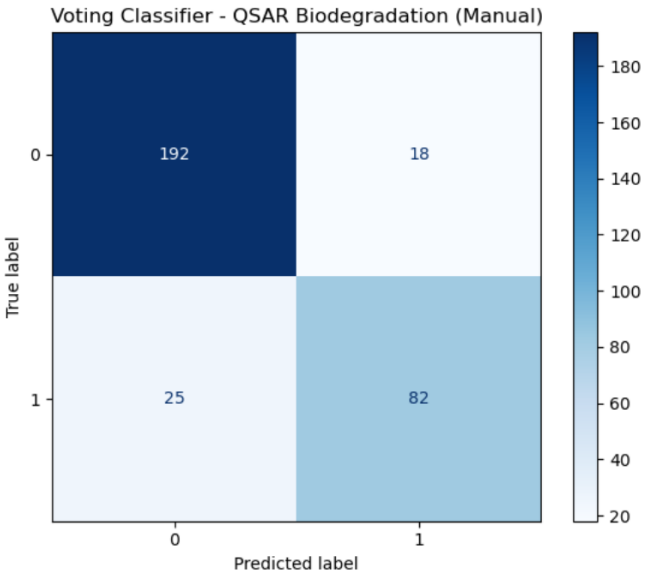

```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR BANKNOTE AUTHENTICATION
============================================================

--- GridSearchCV for Decision Tree ---
Fitting 5 folds for each of 12 candidates, totalling 60 fits
Best params for Decision Tree: {'classifier__max_depth': 5, 'classifier__min_samples_split': 2, 'feature_selection__k': 'all'}
Best CV score: 0.9856

--- GridSearchCV for K-Nearest Neighbors ---
Fitting 5 folds for each of 8 candidates, totalling 40 fits
Best params for K-Nearest Neighbors: {'classifier__n_neighbors': 7, 'classifier__weights': 'distance', 'feature_selection__k': 'all'}
Best CV score: 0.9990

--- GridSearchCV for Logistic Regression ---
Fitting 5 folds for each of 8 candidates, totalling 40 fits
Best params for Logistic Regression: {'classifier__C': 10.0, 'classifier__penalty': 'l1', 'feature_selection__k': 'all'}
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

K-Nearest Neighbors:
  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
```



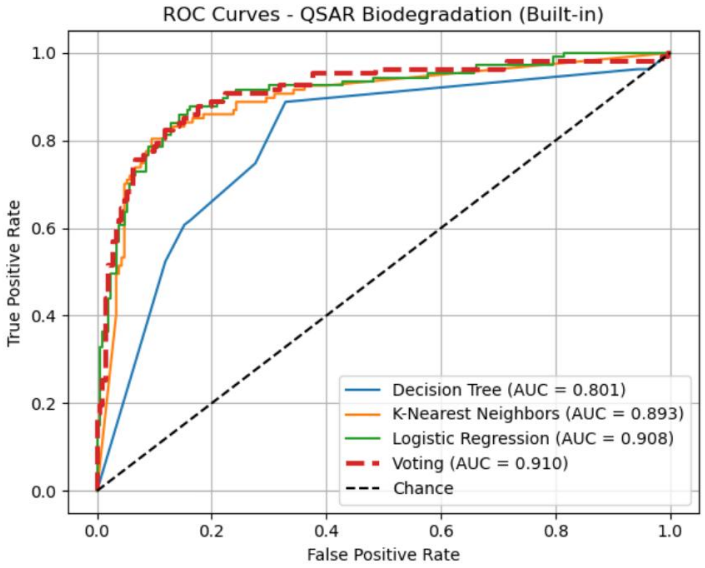


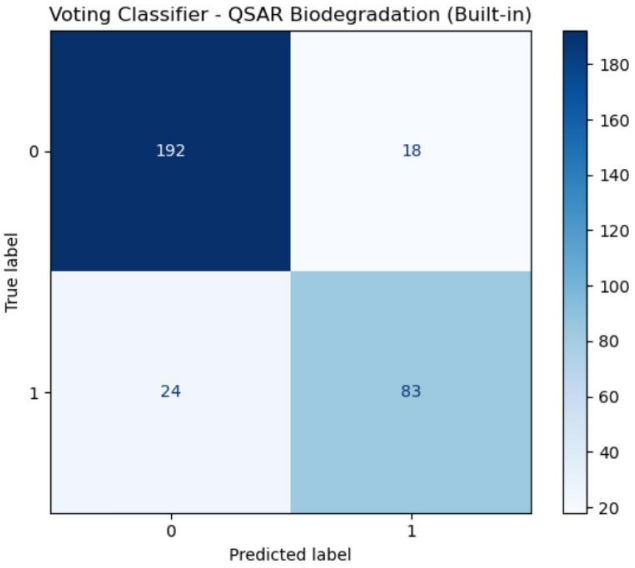

```
Completed processing for Banknote Authentication
================================================================================

################################################################################
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 ---
----------------------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'classifier__max_depth': 3, 'classifier__min_samples_split': 2, 'feature_selection__k': 'all'}
Best cross-validation AUC: 0.8369
--- Manual Grid Search for K-Nearest Neighbors ---
----------------------------------------------------------------------------------------------------
Best parameters for K-Nearest Neighbors: {'classifier__n_neighbors': 5, 'classifier__weights': 'distance', 'feature_selection__k': 'all'}
Best cross-validation AUC: 0.9002
--- Manual Grid Search for Logistic Regression ---
----------------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'classifier__C': 1.0, 'classifier__penalty': 'l1', 'feature_selection__k': 'all'}
Best cross-validation AUC: 0.9317

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7634
  Precision: 0.6600
  Recall: 0.6168
  F1-Score: 0.6377
  ROC AUC: 0.8007

K-Nearest Neighbors:
  Accuracy: 0.8644
  Precision: 0.8077
  Recall: 0.7850
  F1-Score: 0.7962
  ROC AUC: 0.8931

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: 0.8644, Precision: 0.8200
  Recall: 0.7664, F1: 0.7923, AUC: 0.9100
```






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

--- GridSearchCV for Decision Tree ---
Fitting 5 folds for each of 12 candidates, totalling 60 fits
Best params for Decision Tree: {'classifier__max_depth': 3, 'classifier__min_samples_split': 2, 'feature_selection__k': 'all'}
Best CV score: 0.8369

--- GridSearchCV for K-Nearest Neighbors ---
Fitting 5 folds for each of 8 candidates, totalling 40 fits
Best params for K-Nearest Neighbors: {'classifier__n_neighbors': 5, 'classifier__weights': 'distance', 'feature_selection__k': 'all'}
Best CV score: 0.9002

--- GridSearchCV for Logistic Regression ---
Fitting 5 folds for each of 8 candidates, totalling 40 fits
Best params for Logistic Regression: {'classifier__C': 1.0, 'classifier__penalty': 'l1', 'feature_selection__k': 'all'}
Best CV score: 0.9317

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7634
  Precision: 0.6600
  Recall: 0.6168
  F1-Score: 0.6377
  ROC AUC: 0.8007

K-Nearest Neighbors:
  Accuracy: 0.8644
  Precision: 0.8077
  Recall: 0.7850
  F1-Score: 0.7962
  ROC AUC: 0.8931

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: 0.8218
  Recall: 0.7757, F1: 0.7981, AUC: 0.9100
```






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

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

Conclusion

The results show that different models perform differently on the same dataset, and hyperparameter tuning plays a key role in improving outcomes. While simpler models like Logistic Regression and Decision Trees are easier to interpret, models such as KNN often achieve more balanced performance across metrics.

Manual grid search provides a clear understanding of how tuning impacts results, but it is time-consuming compared to the efficiency of scikit-learn's built-in functions. Overall, the main takeaway is that model selection involves trade-offs between accuracy, interpretability, and computational cost, and using libraries ensures both speed and reliability in practical applications.