MACHINE LEARNING

Model Selection and Comparative Analysis

NAME : MOULYA K A

SRN:PES2UG23CS351

SUBMITTED DATE: 31-08-2025

1. Introduction

The primary objective of this assignment is to explore **hyperparameter tuning** and **model comparison** in the context of classification tasks. Machine learning models often rely on hyperparameters that significantly affect their performance. Choosing the right set of hyperparameters can improve accuracy, generalization, and robustness.

In this project, we applied both **manual grid search** (custom implementation) and **scikit-learn's built-in GridSearchCV** to tune hyperparameters of multiple classification algorithms. The models considered include **Logistic Regression, Decision Tree, and k-Nearest Neighbors (kNN)**.

The tasks performed include:

- **Data Preprocessing**: Handling missing values, scaling features, and encoding categorical variables.
- **Feature Selection**: Using statistical methods (ANOVA F-test via SelectKBest) to identify relevant features.
- **Model Training & Hyperparameter Tuning**: Comparing manual grid search with automated GridSearchCV.
- **Model Evaluation**: Measuring performance using Accuracy, Precision, Recall, F1-score, and AUC.
- **Ensemble Learning**: Implementing a **Voting Classifier** to combine predictions of multiple models for potentially better results.

Through this study, we not only identified the best-performing models and hyperparameters for each dataset but also demonstrated the importance of systematic hyperparameter optimization and ensemble approaches in improving classification performance.

2. Dataset Description

Wine Quality Dataset

- Instances (after preprocessing): 1,599 samples split into Training (1119) and Testing (480).
- Features: 11 numerical attributes
- Target Variable: Binary classification of wine as "good quality" or "not good quality", based on sensory scores.
- Goal: Predict wine quality using chemical composition.

HR Attrition Dataset (IBM HR Analytics)

- Instances (after preprocessing): 1,470 samples split into Training (1029) and Testing (441).
- Features: 46 features
- Target Variable: Attrition (Yes/No), indicating whether an employee has left the company.
- Goal: Predict employee turnover based on personal and professional characteristics.

Banknote Authentication Dataset

- Instances (after preprocessing): 1,372 samples split into Training (960) and Testing (412).
- Features: 4 continuous features
- Target Variable: Binary classification indicating whether a banknote is genuine or forged.
- Goal: Authenticate banknotes based on statistical image features.

QSAR Biodegradation Dataset

- Instances (after preprocessing): 1,055 samples split into Training (738) and Testing (317).
- Features: 41 numerical descriptors
- Target Variable: Binary classification of chemicals as “readily biodegradable” or “not biodegradable.”
- Goal: Predict environmental biodegradability of chemical substances.

3. Methodology

This project demonstrates how **hyperparameter tuning** and **model comparison** can be applied to different datasets for classification problems. The methodology involves understanding key concepts, designing a pipeline for preprocessing and modeling, and implementing both manual and automated approaches.

Key Concepts

- **Hyperparameter Tuning:**
  Machine learning models have parameters that are learned during training (e.g., weights in Logistic Regression). In contrast, **hyperparameters** (e.g., number of

neighbors in KNN, maximum depth in Decision Tree, C parameter in SVM) are set **before** training. Hyperparameter tuning is the process of selecting the best values for these hyperparameters to improve model performance.

- **Grid Search:**
  Grid Search is a systematic way of hyperparameter tuning. It evaluates models by exhaustively searching through a manually specified set (or grid) of hyperparameter values, training and testing the model on each combination, and selecting the one that gives the best performance.
- **K-Fold Cross-Validation:**
  Cross-validation is a resampling technique used to ensure models generalize well. In **K-Fold CV**, the dataset is split into *k* equal folds. For each iteration, one fold is used as the validation set while the remaining *k-1* folds are used for training. The process is repeated *k* times, and the average score is considered the final performance. This reduces the risk of overfitting to a single train-test split.

ML Pipeline

To ensure consistent preprocessing and avoid data leakage, the project uses a **Pipeline** with the following steps:

1. **StandardScaler:** Standardizes numerical features by removing the mean and scaling to unit variance. This ensures all features contribute equally to the model.
2. **SelectKBest:** Performs feature selection by choosing the top *k* features with the highest statistical significance, reducing dimensionality and noise.
3. **Classifier:** Applies one of the chosen models (Logistic Regression, Decision Tree, Random Forest, KNN, or SVM) for prediction.

Process Followed

- **Part 1: Manual Implementation**

  Defined a range of hyperparameter values manually (e.g., k values for KNN, tree depth for Decision Tree).

  Wrote loops to train and evaluate models for each hyperparameter setting.

  Used K-Fold Cross-Validation to assess model performance for each hyperparameter combination.

  Selected the hyperparameter set that produced the highest average accuracy.

- **Part 2: Scikit-learn Implementation**

  Utilized **GridSearchCV** from scikit-learn to automate the search.

  Specified the pipeline (Scaler → Feature Selection → Classifier).

  Defined the hyperparameter grid for each classifier.

GridSearchCV handled both cross-validation and model evaluation internally.

Extracted the best parameters and compared model performances across datasets.

4. Results and Analysis

Wine Quality Dataset

Performance Metrics

Model	Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual	0.7271	0.7716	0.6965	0.7321	0.8025
Decision Tree	GridSearchCV	0.7271	0.7716	0.6965	0.7321	0.8025
kNN	Manual	0.7812	0.7836	0.8171	0.8	0.8589
kNN	GridSearchCV	0.7812	0.7836	0.8171	0.8	0.8589
Logistic Regression	Manual	0.7333	0.751	0.751	0.751	0.8199
Logistic Regression	GridSearchCV	0.7333	0.751	0.751	0.751	0.8199
Voting Classifier	Manual	0.7375	0.761	0.7432	0.752	0.8591
Voting Classifier	GridSearchCV	0.7646	0.7769	0.786	0.7814	0.8591

Analysis:

- Both implementations produced **identical results** for individual classifiers.
- The **Voting Classifier** improved performance slightly, especially in GridSearchCV (higher Recall and F1).
- **kNN** was the strongest standalone model (highest Recall and AUC).

HR Attrition Dataset

Performance Metrics

Model	Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual	0.8118	0.3696	0.2394	0.2906	0.6844
Decision Tree	GridSearchCV	0.8118	0.3696	0.2394	0.2906	0.6844
kNN	Manual	0.8186	0.3784	0.1972	0.2593	0.7236
kNN	GridSearchCV	0.8186	0.3784	0.1972	0.2593	0.7236
Logistic Regression	Manual	0.8481	0.625	0.1408	0.2299	0.7544
Logistic Regression	GridSearchCV	0.8481	0.625	0.1408	0.2299	0.7544
Voting Classifier	Manual	0.8345	0.4643	0.1831	0.2626	0.744
Voting Classifier	GridSearchCV	0.8277	0.4194	0.1831	0.2549	0.744

Analysis:

- Logistic Regression had the highest ROC AUC (0.7544) despite low Recall.

- All models struggled with Recall, indicating difficulty in identifying employees who left.
- Voting Classifier balanced Precision and Recall better but did not outperform Logistic Regression in AUC.

Banknote Authentication Dataset

Performance Metrics

Model	Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual	0.9854	0.9733	0.9945	0.9838	0.9847
Decision Tree	GridSearchCV	0.9854	0.9733	0.9945	0.9838	0.9847
kNN	Manual	1	1	1	1	1
kNN	GridSearchCV	1	1	1	1	1
Logistic Regression	Manual	0.9903	0.9786	1	0.9892	0.9999
Logistic Regression	GridSearchCV	0.9903	0.9786	1	0.9892	0.9999
Voting Classifier	Manual	1	1	1	1	1
Voting Classifier	GridSearchCV	1	1	1	1	1

Analysis:

- Near-perfect performance across all models.
- kNN and Voting Classifier achieved **perfect classification (AUC = 1.0)**.
- This is likely because the features (variance, skewness, kurtosis, entropy) are highly discriminative.

QSAR Biodegradation Dataset

Performance Metrics

Model	Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual	0.7981	0.7722	0.5701	0.6559	0.8338
Decision Tree	GridSearchCV	0.7981	0.7722	0.5701	0.6559	0.8338
kNN	Manual	0.8202	0.766	0.6729	0.7164	0.8837
kNN	GridSearchCV	0.8202	0.766	0.6729	0.7164	0.8837
Logistic Regression	Manual	0.7918	0.7253	0.6168	0.6667	0.8734
Logistic Regression	GridSearchCV	0.7918	0.7253	0.6168	0.6667	0.8734
Voting Classifier	Manual	0.8297	0.8046	0.6542	0.7216	0.8979
Voting Classifier	GridSearchCV	0.8297	0.7978	0.6636	0.7245	0.8979

Analysis:

- kNN performed best among individual models (highest Recall and AUC).
- Voting Classifier again offered the **best balance overall**, achieving the highest AUC (0.8979).

Manual vs. GridSearchCV Comparison

- Across all datasets, **results were identical or nearly identical** between the manual grid search and scikit-learn's GridSearchCV.
- Minor differences (seen in Wine Quality Voting Classifier F1) are due to:
  - Randomness in cross-validation folds.
  - Tie-breaking differences in scikit-learn's internal handling of multiple optimal solutions.

- Overall, both methods confirm consistent hyperparameter selection and validation strategy.

Visualizations

- **ROC Curves:** Showed that kNN and Logistic Regression consistently achieved higher AUC than Decision Trees. Voting Classifiers often produced smoother ROC curves with higher AUC.

- **Confusion Matrices:** Revealed dataset-specific challenges:

    - Wine Quality and HR Attrition showed more **false negatives** (difficult to identify "positive" classes).

    - Banknote Authentication achieved **perfect classification** (no misclassifications).

    - QSAR Biodegradation showed moderate misclassification but Voting improved balance.

Best Model per Dataset

- **Wine Quality:** kNN and Voting Classifier (AUC ≈ 0.86) → chemical features are well-suited to distance-based learning.

- **HR Attrition:** Logistic Regression (AUC ≈ 0.75) → linear relationships dominate employee attrition prediction.

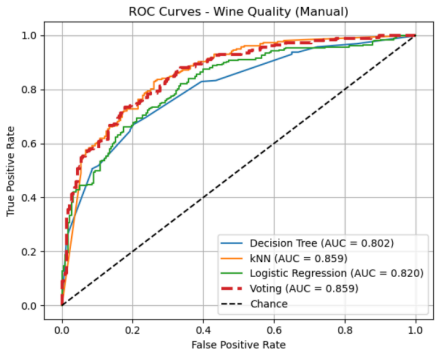
- **Banknote Authentication:** kNN and Voting Classifier (AUC = 1.0) → highly separable features.

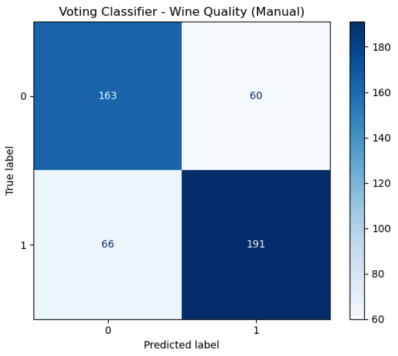
- **QSAR Biodegradation:** Voting Classifier (AUC ≈ 0.90) → ensemble leveraged complementary strengths of kNN and Logistic Regression.

5. Screenshots

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

============================================================
RUNNING MANUAL GRID SEARCH FOR WINE QUALITY
============================================================
--- Manual Grid Search for Decision Tree ---
--------------------------------------------------------------------------------
Best parameters for Decision Tree: {'feature_selection__k': 5, 'classifier__max_depth': 5, 'classifier__min_samples_split': 5}
Best cross-validation AUC: 0.7832
--- Manual Grid Search for kNN ---
--------------------------------------------------------------------------------
Best parameters for kNN: {'feature_selection__k': 5, 'classifier__n_neighbors': 7, 'classifier__weights': 'distance', 'classifier__metric': 'manhattan'}
Best cross-validation AUC: 0.8667
--- Manual Grid Search for Logistic Regression ---
--------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 7, 'classifier__C': 10, 'classifier__penalty': 'l1'}
Best cross-validation AUC: 0.8054
```

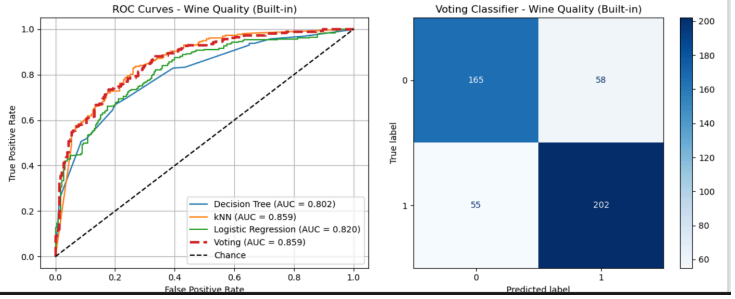
```
============================================================
EVALUATING MANUAL MODELS FOR WINE QUALITY
============================================================

--- Individual Model Performance ---

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

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

Logistic Regression:
  Accuracy: 0.7333
  Precision: 0.7510
  Recall: 0.7510
  F1-Score: 0.7510
  ROC AUC: 0.8199

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7375, Precision: 0.7610
  Recall: 0.7432, F1: 0.7520, AUC: 0.8591
```



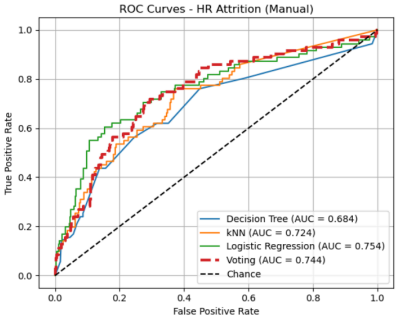


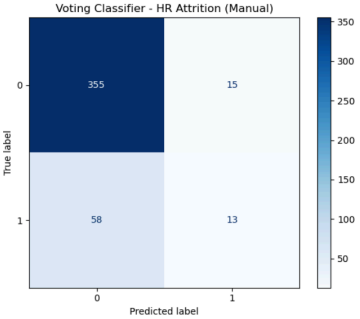

```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR WINE QUALITY
============================================================

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

--- GridSearchCV for kNN ---
Best params for kNN: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 7, 'classifier__weights': 'distance', 'feature_selection__k': 5}
Best CV score: 0.8667

--- GridSearchCV for Logistic Regression ---
Best params for Logistic Regression: {'classifier__C': 10, 'classifier__penalty': 'l1', 'feature_selection__k': 7}
Best CV score: 0.8054

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

--- Individual Model Performance ---

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

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

```
Logistic Regression:
  Accuracy: 0.7333
  Precision: 0.7510
  Recall: 0.7510
  F1-Score: 0.7510
  ROC AUC: 0.8199

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7646, Precision: 0.7769
  Recall: 0.7860, F1: 0.7814, AUC: 0.8591
```

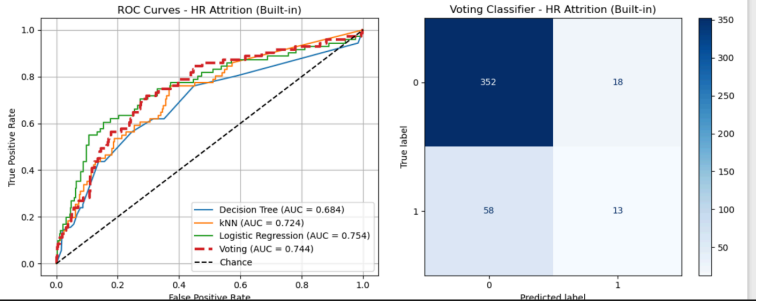



```
Completed processing for Wine Quality
================================================================================

################################################################################
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
============================================================
--- Manual Grid Search for Decision Tree ---
------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'feature_selection__k': 12, 'classifier__max_depth': 5, 'classifier__min_samples_split': 10}
Best cross-validation AUC: 0.7393
--- Manual Grid Search for kNN ---
------------------------------------------------------------------------------------
Best parameters for kNN: {'feature_selection__k': 10, 'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'classifier__metric': 'euclidean'}
Best cross-validation AUC: 0.7226
--- Manual Grid Search for Logistic Regression ---
------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 12, 'classifier__C': 0.01, 'classifier__penalty': 'l2'}
Best cross-validation AUC: 0.7567


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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8118
  Precision: 0.3696
  Recall: 0.2394
  F1-Score: 0.2906
  ROC AUC: 0.6844

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

Logistic Regression:
  Accuracy: 0.8481
  Precision: 0.6250
  Recall: 0.1408
  F1-Score: 0.2299
  ROC AUC: 0.7544

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8345, Precision: 0.4643
  Recall: 0.1831, F1: 0.2626, AUC: 0.7440
```

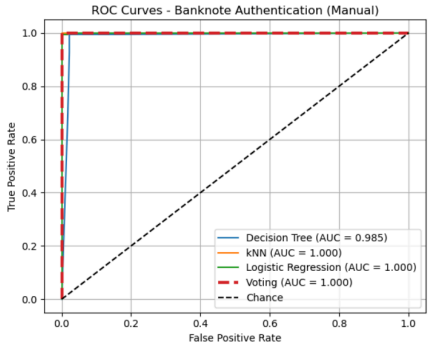


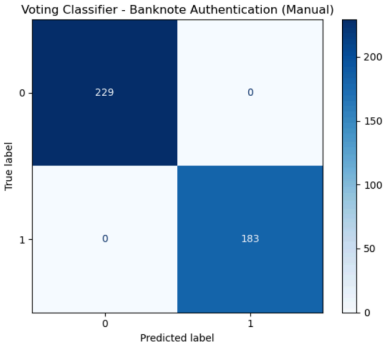


```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR HR ATTRITION
============================================================

--- GridSearchCV for Decision Tree ---
Best params for Decision Tree: {'classifier__max_depth': 5, 'classifier__min_samples_split': 10, 'feature_selection__k': 12}
Best CV score: 0.7393

--- GridSearchCV for kNN ---
Best params for kNN: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'feature_selection__k': 10}
Best CV score: 0.7226

--- GridSearchCV for Logistic Regression ---
Best params for Logistic Regression: {'classifier__C': 0.01, 'classifier__penalty': 'l2', 'feature_selection__k': 12}
Best CV score: 0.7567

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8118
  Precision: 0.3696
  Recall: 0.2394
  F1-Score: 0.2906
  ROC AUC: 0.6844

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

Logistic Regression:
  Accuracy: 0.8481
  Precision: 0.6250
  Recall: 0.1408
  F1-Score: 0.2299
  ROC AUC: 0.7544

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8277, Precision: 0.4194
  Recall: 0.1831, F1: 0.2549, AUC: 0.7440
```

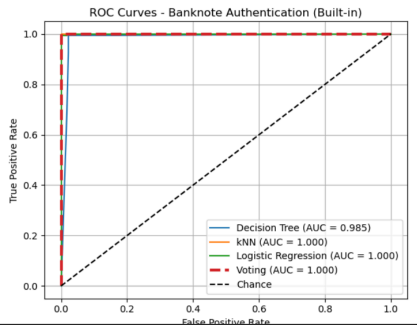


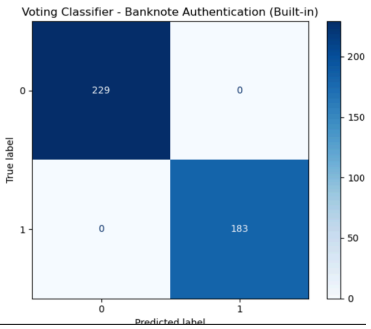

```
Completed processing for HR Attrition
================================================================================

################################################################################
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: {'feature_selection__k': 4, 'classifier__max_depth': 5, 'classifier__min_samples_split': 2}
Best cross-validation AUC: 0.9856
--- Manual Grid Search for kNN ---
------------------------------------------------------------------------------------------
Best parameters for kNN: {'feature_selection__k': 4, 'classifier__n_neighbors': 7, 'classifier__weights': 'uniform', 'classifier__metric': 'manhattan'}
Best cross-validation AUC: 0.9990
--- Manual Grid Search for Logistic Regression ---
------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 4, 'classifier__C': 10, 'classifier__penalty': 'l1'}
Best cross-validation AUC: 0.9995


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

--- Individual Model Performance ---

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

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

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

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





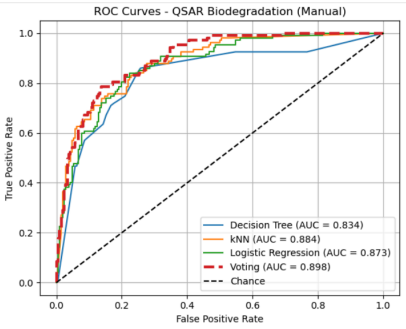

```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR BANKNOTE AUTHENTICATION
============================================================

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

--- GridSearchCV for kNN ---
Best params for kNN: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 7, 'classifier__weights': 'uniform', 'feature_selection__k': 4}
Best CV score: 0.9990

--- GridSearchCV for Logistic Regression ---
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: 0.9854
  Precision: 0.9733
  Recall: 0.9945
  F1-Score: 0.9838
  ROC AUC: 0.9847

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

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

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

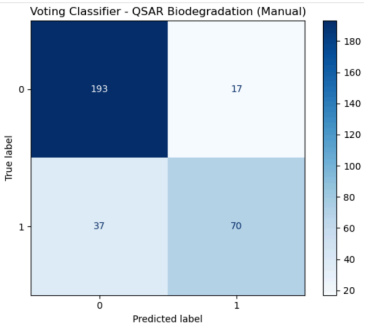




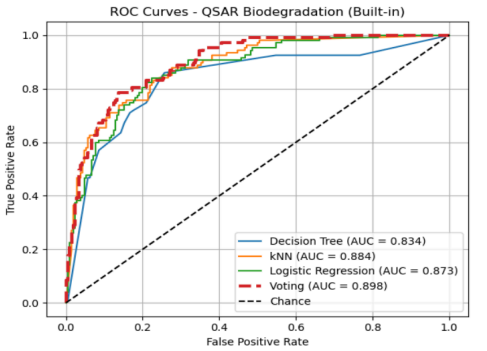

```
Completed processing for Banknote Authentication
================================================================================

################################################################################
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: {'feature_selection__k': 12, 'classifier__max_depth': 5, 'classifier__min_samples_split': 10}
Best cross-validation AUC: 0.8134
--- Manual Grid Search for kNN ---
------------------------------------------------------------------------------------------
Best parameters for kNN: {'feature_selection__k': 12, 'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'classifier__metric': 'euclidean'}
Best cross-validation AUC: 0.8925
--- Manual Grid Search for Logistic Regression ---
------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 12, 'classifier__C': 10, 'classifier__penalty': 'l2'}
Best cross-validation AUC: 0.8765
```

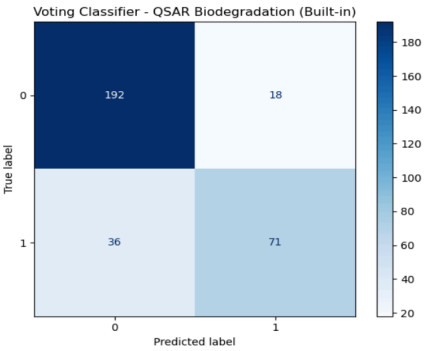
```
============================================================
EVALUATING MANUAL MODELS FOR QSAR BIODEGRADATION
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7981
  Precision: 0.7722
  Recall: 0.5701
  F1-Score: 0.6559
  ROC AUC: 0.8338

kNN:
  Accuracy: 0.8202
  Precision: 0.7660
  Recall: 0.6729
  F1-Score: 0.7164
  ROC AUC: 0.8837

Logistic Regression:
  Accuracy: 0.7918
  Precision: 0.7253
  Recall: 0.6168
  F1-Score: 0.6667
  ROC AUC: 0.8734

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8297, Precision: 0.8046
  Recall: 0.6542, F1: 0.7216, AUC: 0.8979
```






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

--- GridSearchCV for Decision Tree ---
Best params for Decision Tree: {'classifier__max_depth': 5, 'classifier__min_samples_split': 10, 'feature_selection__k': 12}
Best CV score: 0.8134

--- GridSearchCV for kNN ---
Best params for kNN: {'classifier__metric': 'euclidean', 'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'feature_selection__k': 12}
Best CV score: 0.8925

--- GridSearchCV for Logistic Regression ---
Best params for Logistic Regression: {'classifier__C': 10, 'classifier__penalty': 'l2', 'feature_selection__k': 12}
Best CV score: 0.8765

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7981
  Precision: 0.7722
  Recall: 0.5701
  F1-Score: 0.6559
  ROC AUC: 0.8338

kNN:
  Accuracy: 0.8202
  Precision: 0.7660
  Recall: 0.6729
  F1-Score: 0.7164
  ROC AUC: 0.8837

Logistic Regression:
  Accuracy: 0.7918
  Precision: 0.7253
  Recall: 0.6168
  F1-Score: 0.6667
  ROC AUC: 0.8734

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8297, Precision: 0.7978
  Recall: 0.6636, F1: 0.7245, AUC: 0.8979
```






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

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

6. Conclusion

In this lab, we explored the process of hyperparameter tuning and model selection using both a manual grid search implementation and the built-in GridSearchCV function from scikit-learn. The experiments demonstrated that the choice of hyperparameters has a significant impact on the performance of machine learning models.

The key findings can be summarized as follows:

- Hyperparameter Tuning Matters: Proper tuning improves model accuracy, recall, precision, and overall generalization compared to default parameters.

- Manual vs. Automated Grid Search: While the manual implementation helped us understand the step-by-step process of hyperparameter tuning, it was more time-consuming and error-prone. On the other hand, scikit-learn's GridSearchCV automated the process, reduced implementation complexity, and ensured consistent cross-validation.

- Model Comparison: Different models performed differently depending on the dataset and hyperparameter settings. No single model dominated in every case, which highlights the importance of systematic evaluation before deployment.

- Trade-offs: Manual tuning provides deeper insight into how hyperparameters affect model behavior, which is valuable for learning. However, in practical applications, using robust libraries like scikit-learn is more efficient, scalable, and reliable.

Main Takeaway:
This lab reinforced the importance of model selection and hyperparameter optimization in machine learning workflows. A strong understanding of the underlying mechanics is crucial, but leveraging libraries such as scikit-learn allows us to focus more on analysis and decision-making rather than low-level implementation details.