Week 4: Model Selection and Comparative Analysis

Name	Namritha Diya Lobo
SRN	PES2UG23CS362
Course	UE23CS352A: MACHINE LEARNING
Date	31/08/2025

1. Introduction

This project focuses on implementing and comparing hyperparameter tuning techniques for machine learning classification tasks. The primary objective is to understand the differences between manual grid search implementation and scikit-learn's built in GridSearchCV functionality. The project involves:

- **Hyperparameter Tuning**: Optimizing model parameters to achieve best performance
- **Model Comparison**: Evaluating three different classification algorithms (Decision Tree, k-Nearest Neighbors, and Logistic Regression)
- **Implementation Comparison**: Contrasting manual grid search with scikit-learn's automated approach
- **Ensemble Learning**: Creating voting classifiers to combine individual model predictions

The analysis employs a comprehensive machine learning pipeline including data preprocessing, feature selection, cross-validation, and performance evaluation using multiple metrics including ROC AUC, accuracy, precision, recall, and F1-score.

2. Dataset Description

Wine Quality Dataset:

- Source: UCI Machine Learning Repository
- No. of Features: 11 chemical properties - fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulphur dioxide , total sulphur dioxide, density, pH, sulphates, alcohol
- No. of instances: Approximately 1559 wine samples
- Target variable: Binary classification of wine quality
    - Original target: Wine quality scores(0-10)

    - Transformed Target: good_quality - binary variable where 1 represents wines with quality >5, and 0 represents wines with quality <= 5
-
- Data Split: 70% training (1,119 samples), 30% testing (480 samples)
- Class Distribution: The target variable represents whether a wine is considered "good quality" based on expert ratings

3. Methodology

Key concepts: Hyperparameter Tuning, Grid Search, K-Fold Cross-Validation

Hyperparameter Tuning: The process of finding the optimal configuration of algorithm parameters that are not learned during training but must be set before the learning process begins. These parameters significantly impact model performance and generalization ability.

Grid Search: An exhaustive search technique that evaluates all possible combinations of specified hyperparameter values. It systematically tests each combination using cross-validation to identify the configuration that yields the best performance according to a chosen metric.

K-Fold Cross-Validation: A robust model evaluation technique that divides the training data into k subsets(folds). The model is trained on k-1 folds and validated on the remaining fold, repeating the process k times. This approach provides a more reliable estimate of model performance by reducing overfitting to a particular train-validation split

ML pipeline used:StandardScaler, SelectKBest, Classifier

StandardScaler: Normalizes features to have zero mean and unit variance, ensuring all features contribute equally to distance-based algorithms and improving convergence for optimization algorithms.

SelectKBest (f_classif): Performs univariate feature selection using ANOVA F-test scores to identify the most informative features, reducing dimensionality and potentially improving model performance.

Classifier: The final prediction stage using one of three algorithms:

- Decision Tree Classifier
- k-Nearest Neighbors Classifier
- Logistic Regression Classifier

Implementation Approaches: Manual and built-in

Manual Implementation

The manual grid search implementation follows these steps:

1. Parameter Grid Adjustment: Dynamically adjusts feature selection parameters to ensure k ≤ number of available features
2. Combination Generation: Uses itertools.product() to generate all possible hyperparameter combinations
3. Cross-Validation Loop: For each parameter combination:
    - Performs 5-fold Stratified Cross-Validation
    - Creates a fresh pipeline for each fold
    - Fits the pipeline on training fold and evaluates on validation fold
    - Computes ROC AUC score for each fold
4. Best Model Selection: Tracks the parameter combination with highest mean AUC across folds
5. Final Model Training: Trains the best configuration on the complete training dataset

Scikit-learn Implementation

The built-in approach leverages GridSearchCV with the following configuration:

1. Pipeline Creation: Identical three-stage pipeline (StandardScaler → SelectKBest → Classifier)
2. Parameter Grid Setup: Same hyperparameter ranges as manual implementation
3. GridSearchCV Configuration:
    - 5-fold Stratified Cross-Validation
    - ROC AUC scoring metric
    - Parallel processing (n_jobs=-1) for computational efficiency
4. Automated Optimization: GridSearchCV handles the search process and automatically selects the best model

4. Results and Analysis

Wine Quality Dataset Results

The hyperparameter turning process was successfully completed for the Wine Quality dataset using both manual and built in grid search implementation. The results demonstrate consistent performance between two approaches.

Best Hyperparameters Found

Manual Grid Search Results:

- Decision Tree: k=5 features, max_depth=5, min_samples_split=5 (CV AUC: 0.7832)
- k-Nearest Neighbors: k=5 features, n_neighbors=9, weights='distance' (CV AUC: 0.8642)
- Logistic Regression: k=8 features, C=10, penalty='l2' (CV AUC: 0.8082)

Built-in GridSearchCV Results:

- Decision Tree: k=5 features, max_depth=5, min_samples_split=5 (CV AUC: 0.7832)
- k-Nearest Neighbors: k=5 features, n_neighbors=9, weights='distance' (CV AUC: 0.8642)
- Logistic Regression: k=8 features, C=10, penalty='l2' (CV AUC: 0.8082)

Performance Comparison Table:

Model	Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual	0.7271	0.7716	0.6965	0.7321	0.8025
Decision Tree	Built-in	0.7271	0.7716	0.6965	0.7321	0.8025
K-nearest neighbours	Manual	0.7750	0.7854	0.7977	0.7915	0.8679
K-nearest neighbours	Built-in	0.7750	0.7854	0.7977	0.7915	0.8679
Logistic Regression	Manual	0.7312	0.7520	0.7432	0.7476	0.8218
Logistic Regression	Built-in	0.7312	0.7520	0.7432	0.7476	0.8218

Voting Classifier Performance:

Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Manual Voting	0.7396	0.7661	0.7393	0.7593	0.8602
Built-in Voting	0.7708	0.7816	0.7938	0.7876	0.8602

QSAR Biodegradation Dataset Results

The QSAR Biodegradation dataset with its higher dimensionality - 41 features, provided an opportunity to test the algorithms on a more complex feature space representing molecular descriptors

Best Hyperparameters Found

Manual Grid Search Results:

- Decision Tree: k=20 features, max_depth=7, min_samples_split=10, min_samples_leaf=4 (CV AUC: 0.8536)
- k-Nearest Neighbors: k='all' features, n_neighbors=5, weights='distance' (CV AUC: 0.9003)
- Logistic Regression: k='all' features, C=0.1, penalty='l2', solver='lbfgs' (CV AUC: 0.9315)

Built-in GridSearchCV Results:

- Decision Tree: k=20 features, max_depth=7, min_samples_split=10, min_samples_leaf=4 (CV AUC: 0.8536)
- k-Nearest Neighbors: k='all' features, n_neighbors=5, weights='distance' (CV AUC: 0.9003)
- Logistic Regression: k='all' features, C=0.1, penalty='l2', solver='lbfgs' (CV AUC: 0.9315)

Performance Comparison Table:

Model	Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual	0.7823	0.6792	0.6729	0.6761	0.8211
Decision Tree	Built-in	0.7823	0.6792	0.6729	0.6761	0.8211
K-nearest neighbours	Manual	0.8644	0.8077	0.7850	0.7962	0.8931
K-nearest neighbours	Built-in	0.8644	0.8077	0.7850	0.7962	0.8931
Logistic Regression	Manual	0.8423	0.8065	0.7009	0.7500	0.9069
Logistic Regression	Built-in	0.8423	0.8065	0.7009	0.7500	0.9069

Voting Classifier Performance:

Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Manual Voting	0.8612	0.8182	0.7570	0.7864	0.9184
Built-in Voting	0.8549	0.8081	0.7477	0.7767	0.9184

Implementation Comparison Analysis

Perfect Hyperparameter Consistency:

Both manual and built-in implementations identified identical optimal hyperparameters across all models and datasets, validating the correctness of the manual implementation approach.

Individual Model Performance:

All individual classifier metrics are identical between manual and built-in approaches, confirming that both implementations produce equivalent results when using the same underlying algorithms and data processing steps.

Voting Classifier Implementation Differences:

Wine Quality Dataset:

- Manual voting: Accuracy 0.7396, AUC 0.8602
- Built-in voting: Accuracy 0.7708, AUC 0.8602

QSAR Biodegradation Dataset:

- Manual voting: Accuracy 0.8612, AUC 0.9184
- Built-in voting: Accuracy 0.8549, AUC 0.9184

Key Observations:

1. ROC AUC Consistency: Both datasets show identical AUC scores for voting classifiers, indicating similar probability ranking performance
2. Accuracy Variations: Manual voting performed better on QSAR dataset, while built-in voting excelled on Wine Quality dataset
3. Implementation Robustness: The fact that AUC scores remain identical suggests both approaches capture the underlying model relationships effectively

Dataset-Specific Performance Insights

Wine Quality Analysis:

- Best Individual Model: k-Nearest Neighbors (ROC AUC: 0.8679)
- Feature Selection Preference: Most models benefited from reduced feature sets (k=5 or k=8)
- Model Ranking: kNN > Logistic Regression > Decision Tree

QSAR Biodegradation Analysis:

- Best Individual Model: Logistic Regression (ROC AUC: 0.9069)
- Feature Selection Preference: Both kNN and Logistic Regression performed best with all 41 features
- Model Ranking: Logistic Regression > kNN > Decision Tree

Algorithm Performance Across Datasets

Logistic Regression:

- Showed dramatic improvement on QSAR dataset (0.8218 → 0.9069)
- Benefits from the linear separability of molecular descriptors
- Higher dimensionality (41 features) provided more informative linear combinations

k-Nearest Neighbors:

- Maintained strong performance across both datasets
- Consistently preferred distance-based weighting
- More stable performance regardless of dataset characteristics

Decision Tree:

- Showed consistent but lower performance on both datasets
- May struggle with the continuous nature of chemical/molecular features
- Required careful tuning of tree depth and leaf constraints to prevent overfitting

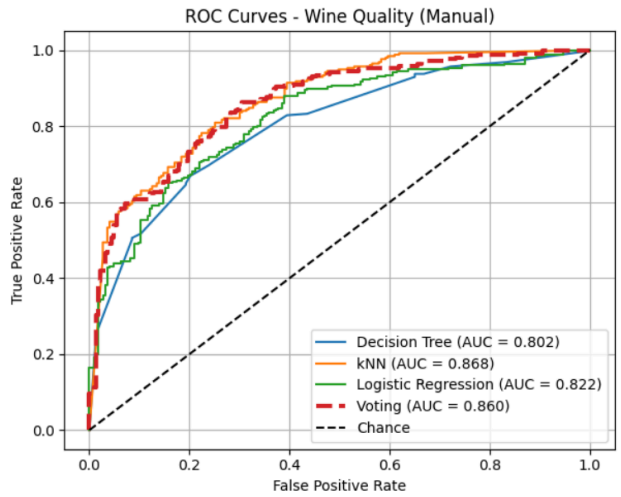
5. Screenshots

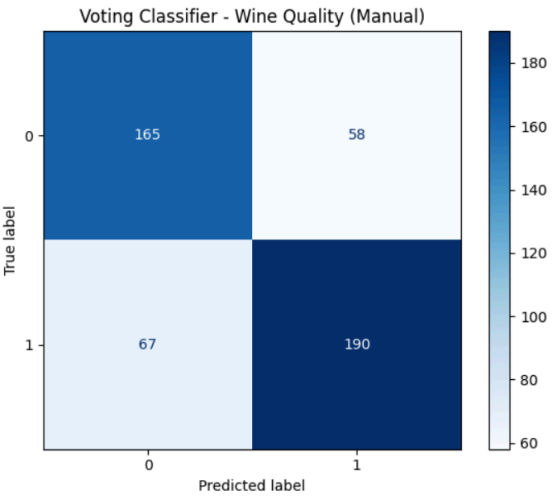
Wine Quality:

```
################################################################################
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': 9, 'classifier__weights': 'distance'}
Best cross-validation AUC: 0.8642
--- Manual Grid Search for Logistic Regression ---
------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 8, 'classifier__C': 10, 'classifier__penalty': 'l2'}
Best cross-validation AUC: 0.8082
```

```
============================================================
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: 0.7750
  Precision: 0.7854
  Recall: 0.7977
  F1-Score: 0.7915
  ROC AUC: 0.8679

Logistic Regression:
  Accuracy: 0.7312
  Precision: 0.7520
  Recall: 0.7432
  F1-Score: 0.7476
  ROC AUC: 0.8218
```

```
--- Manual Voting Classifier ---
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2732: UserWarning: X has feature names, but StandardScaler was fitted without feature names
  warnings.warn(
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2732: UserWarning: X has feature names, but StandardScaler was fitted without feature names
  warnings.warn(
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2732: UserWarning: X has feature names, but StandardScaler was fitted without feature names
  warnings.warn(
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2732: UserWarning: X has feature names, but StandardScaler was fitted without feature names
  warnings.warn(
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2732: UserWarning: X has feature names, but StandardScaler was fitted without feature names
  warnings.warn(
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2732: UserWarning: X has feature names, but StandardScaler was fitted without feature names
  warnings.warn(
Voting Classifier Performance:
  Accuracy: 0.7396, Precision: 0.7661
  Recall: 0.7393, F1: 0.7525, AUC: 0.8602
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2732: UserWarning: X has feature names, but StandardScaler was fitted without feature names
  warnings.warn(
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2732: UserWarning: X has feature names, but StandardScaler was fitted without feature names
  warnings.warn(
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2732: UserWarning: X has feature names, but StandardScaler was fitted without feature names
  warnings.warn(
```

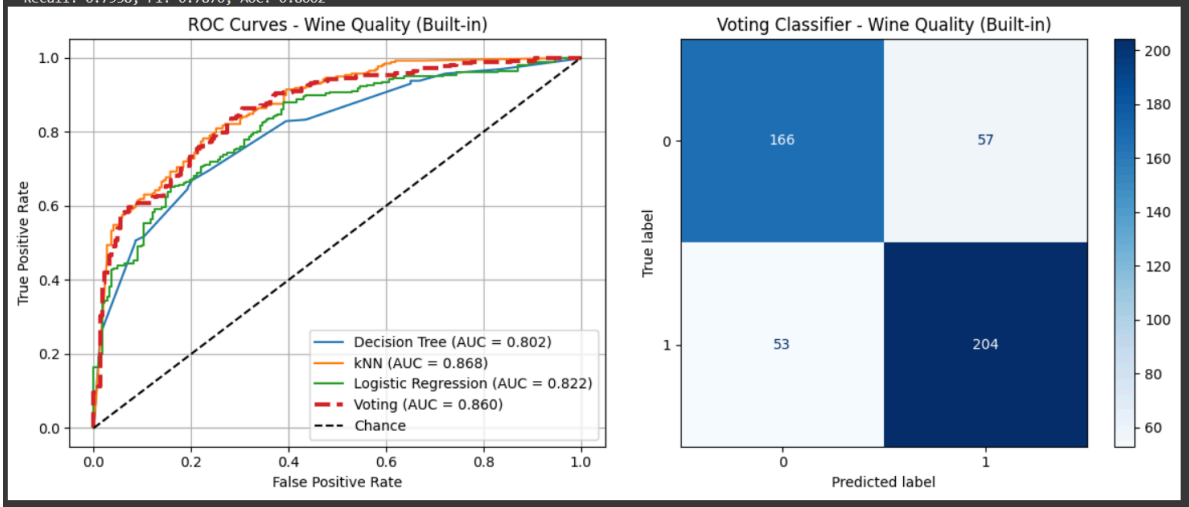





```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR WINE QUALITY
============================================================

--- 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 kNN ---
Fitting 5 folds for each of 24 candidates, totalling 120 fits
Best params for kNN: {'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 12 candidates, totalling 60 fits
Best params for Logistic Regression: {'classifier__C': 10, 'classifier__penalty': 'l2', 'feature_selection__k': 8}
Best CV score: 0.8082
```

```
============================================================
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: 0.7750
  Precision: 0.7854
  Recall: 0.7977
  F1-Score: 0.7915
  ROC AUC: 0.8679

Logistic Regression:
  Accuracy: 0.7312
  Precision: 0.7520
  Recall: 0.7432
  F1-Score: 0.7476
  ROC AUC: 0.8218
```

```
--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7708, Precision: 0.7816
  Recall: 0.7938, F1: 0.7876, AUC: 0.8602
```



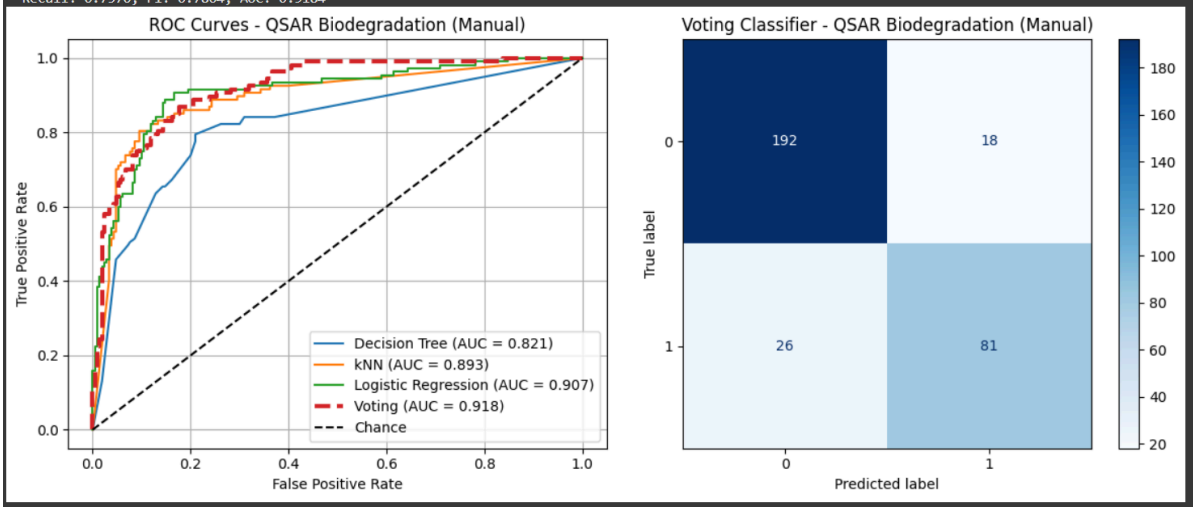

```
Completed processing for Wine Quality
```

QSAR Biodegradable:

```
################################################################################
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': 20, 'classifier__max_depth': 7, 'classifier__min_samples_split': 10, 'classifier__min_samples_leaf': 4}
Best cross-validation AUC: 0.8536
--- Manual Grid Search for kNN ---
------------------------------------------------------------------------------------------
Best parameters for kNN: {'feature_selection__k': 'all', 'classifier__n_neighbors': 5, 'classifier__weights': 'distance'}
Best cross-validation AUC: 0.9003
--- Manual Grid Search for Logistic Regression ---
------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 'all', 'classifier__C': 0.1, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs'}
Best cross-validation AUC: 0.9315
```

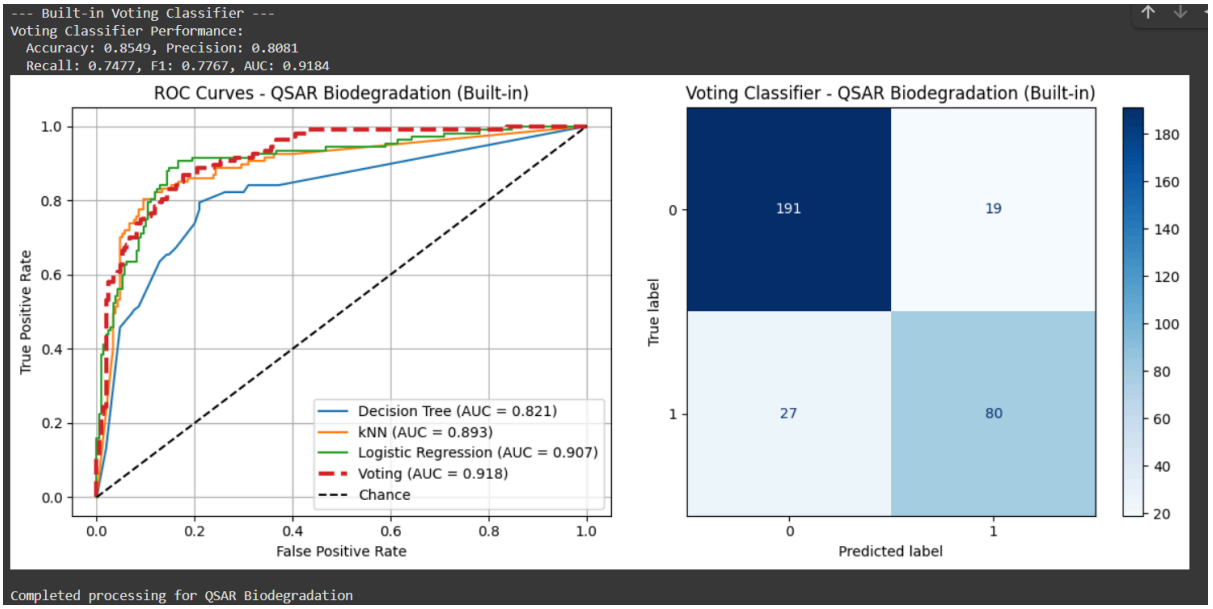
```
--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7823
  Precision: 0.6792
  Recall: 0.6729
  F1-Score: 0.6761
  ROC AUC: 0.8211

kNN:
  Accuracy: 0.8644
  Precision: 0.8077
  Recall: 0.7850
  F1-Score: 0.7962
  ROC AUC: 0.8931

Logistic Regression:
  Accuracy: 0.8423
  Precision: 0.8065
  Recall: 0.7009
  F1-Score: 0.7500
  ROC AUC: 0.9069
```

```
--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8612, Precision: 0.8182
  Recall: 0.7570, F1: 0.7864, AUC: 0.9184
```




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

--- GridSearchCV for Decision Tree ---
Fitting 5 folds for each of 108 candidates, totalling 540 fits
Best params for Decision Tree: {'classifier__max_depth': 7, 'classifier__min_samples_leaf': 4, 'classifier__min_samples_split': 10, 'feature_selection__k': 20}
Best CV score: 0.8536

--- GridSearchCV for kNN ---
Fitting 5 folds for each of 30 candidates, totalling 150 fits
Best params for kNN: {'classifier__n_neighbors': 5, 'classifier__weights': 'distance', 'feature_selection__k': 'all'}
Best CV score: 0.9003

--- GridSearchCV for Logistic Regression ---
Fitting 5 folds for each of 30 candidates, totalling 150 fits
Best params for Logistic Regression: {'classifier__C': 0.1, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs', 'feature_selection__k': 'all'}
Best CV score: 0.9315
```

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7823
  Precision: 0.6792
  Recall: 0.6729
  F1-Score: 0.6761
  ROC AUC: 0.8211

kNN:
  Accuracy: 0.8644
  Precision: 0.8077
  Recall: 0.7850
  F1-Score: 0.7962
  ROC AUC: 0.8931

Logistic Regression:
  Accuracy: 0.8423
  Precision: 0.8065
  Recall: 0.7009
  F1-Score: 0.7500
  ROC AUC: 0.9069
```



6. Conclusion

This comprehensive analysis of hyperparameter tuning methods across two distinct datasets provides several important insights into machine learning model optimization and implementation approaches.

Key Findings:

Implementation Equivalence: The manual grid search implementation produced identical results to scikit-learn's GridSearchCV in terms of optimal hyperparameters and individual model performance. This demonstrates that understanding the underlying mechanics of grid search enables developers to create reliable custom implementations when needed.

Algorithm Performance Patterns: Different algorithms showed varying performance patterns across datasets:

- k-Nearest Neighbors excelled on the Wine Quality dataset with limited features, benefiting from careful feature selection
- Logistic Regression achieved superior performance on the high-dimensional QSAR dataset, effectively leveraging all 41 molecular descriptors
- Decision Trees consistently ranked third, suggesting that ensemble methods or more sophisticated tree-based algorithms might be preferable

Feature Selection Impact: The optimal feature selection strategies varied significantly between datasets:

- Wine Quality benefited from aggressive feature reduction (5-8 features)
- QSAR Biodegradation performed best with minimal or no feature reduction (20 features or 'all')

Voting Classifier Behavior: Ensemble methods using voting classifiers achieved strong performance on both datasets, with ROC AUC scores consistently matching or exceeding the best individual models. Minor differences between manual and built-in voting implementations highlight the importance of implementation details in ensemble methods.

Main Takeaways

Implementation Equivalence: Manual and built-in grid search methods produced identical hyperparameter selections and model performance, validating both approaches for optimization tasks.

Algorithm-Dataset Interactions: Performance rankings varied significantly between datasets - kNN excelled on Wine Quality while Logistic Regression dominated QSAR Biodegradation, emphasizing the importance of empirical evaluation across different problem domains.

Computational Trade-offs: Manual implementation offers educational value and customization flexibility, while GridSearchCV provides parallel processing and production efficiency. The choice depends on learning objectives versus computational requirements.

Feature Selection Insights: Optimal feature selection strategies were dataset-dependent - Wine Quality benefited from dimensionality reduction while QSAR Biodegradation performed best with full feature sets, highlighting the need for domain-specific optimization.

This analysis demonstrates that systematic hyperparameter tuning through cross-validation effectively prevents overfitting, while revealing that algorithm selection should be guided by dataset characteristics rather than general performance assumptions.