UE23CS352A : Machine Learning

Lab 4 – Model Selection and Comparative Analysis

Navya G N	PES2UG23CS372	Section F

Date : 31-082025

1. Introduction

The purpose of this project was to understand model selection and comparative analysis using hyperparameter tuning and ensemble methods. The main tasks included:

- Implementing a manual grid search with cross-validation to tune hyperparameters.
- Using scikit-learn's GridSearchCV to automate tuning.
- Comparing the results from manual and automated approaches.
- Evaluating classifiers with performance metrics and visualizations.

This exercise deepened the understanding of trade-offs between manual coding (flexibility, clarity) and library-based tools (efficiency, reliability).

2. Dataset Description

Dataset 1: Wine Quality

- Instances: ~1600 red wine samples.
- Features: 11 chemical properties (alcohol, acidity, sulphates, etc.).
- Target: Binary variable – whether the wine is of "good quality" or not.

Dataset 2: QSAR Biodegradation

- Instances: 1,055 chemical compounds
- Features: 41 molecular descriptors (continuous values)
- Target Variable: Binary label indicating biodegradability:
  - "Ready Biodegradable" (356 samples)
  - "Not Ready Biodegradable" (699 samples)

3. Methodology

Key Concepts

- Hyperparameter Tuning: It is the process of systematically selecting the best set of hyperparameters (settings that control how a model learns) to maximize model performance. Unlike model parameters learned during training, hyperparameters are predefined and must be optimized externally.
- Grid Search: A method that tries all possible combinations of hyperparameters to pick the best one.
- K-Fold Cross-Validation: It is a model evaluation technique where the dataset is split into *k* equal folds; the model is trained on *k-1* folds and tested on the remaining fold, repeating the process *k* times. This reduces variance in evaluation and provides a more reliable estimate of model performance.

ML Pipeline

Each classifier was trained inside a Pipeline consisting of:

1. StandardScaler: Standardizes features to have a mean of 0 and a standard deviation of 1.
2. SelectKBest: Selects the top 'k' features using a statistical test (f_classif). The number of features, k, is a key hyperparameter you will tune.
3. Classifier: The final modeling step (Decision Tree, kNN, or Logistic Regression).

The process

Part 1: Manual Grid Search Implementation

In the manual grid search, I first defined parameter grids for Decision Tree, KNN, and Logistic Regression with names matching the pipeline steps. I then generated all possible hyperparameter combinations and performed 5-fold stratified cross-validation for each, building a pipeline with scaling, feature selection, and the classifier. For every

fold, I trained and validated the model, calculated the average ROC AUC score, and tracked the best-performing parameter set. Finally, the best model was refitted on the entire training dataset.

Part 2: Built-in GridSearchCV Implementation
In the built-in approach, I created the same pipeline structure and used scikit-learn's GridSearchCV with the defined parameter grids, scoring set to ROC AUC, and 5-fold StratifiedKFold. GridSearchCV automatically trained and validated models across all parameter combinations, selected the best estimator, and reported the best parameters and cross-validation score. This automated much of the process performed manually in Part 1.

4. Results and Analysis

- **Performance Tables:**

Dataset 1: Wine Quality

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 Reg.	0.7312	0.7520	0.7432	0.7476	0.8219
Voting Classifier	0.7396	0.7661	0.7393	0.7525	0.8602

Change in Built in

Voting Classifier	0.7708	0.7816	0.7938	0.7876	0.8601

Observations:

- Best single model: kNN clearly outperforms both Decision Tree and Logistic Regression across all metrics, especially Recall (0.7977) and ROC AUC (0.8679).
- Voting Classifier: The manual voting version underperforms compared to kNN alone, but the built-in GridSearchCV tuned version nearly matches kNN (Accuracy 0.7708 vs. 0.7750).

- Conclusion for Dataset 1: kNN is the best classifier, but the built-in Voting Classifier becomes competitive after tuning, suggesting hyperparameter optimization plays a big role.

Dataset 2: QSAR Biodegradation

Classifier	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.7918	0.6990	0.6729	0.6857	0.8347
kNN	0.8423	0.7664	0.7664	0.7664	0.9059
Logistic Reg.	0.8644	0.8200	0.7664	0.7923	0.9082
Voting Classifier	0.8675	0.8218	0.7757	0.7981	0.9189

Change in Built in

Voting Classifier	0.8612	0.8182	0.7570	0.7864	0.9189

Observations:

- Best single model: Logistic Regression slightly edges out kNN, with higher Accuracy (0.8644 vs. 0.8423), Precision (0.8200), and ROC AUC (0.9082).
- Voting Classifier: Both manual and built-in Voting Classifiers surpass all individual classifiers in ROC AUC (0.9189). The manual implementation performs slightly better than the built-in one in Accuracy and Recall.
- Conclusion for Dataset 2: Ensemble methods provide the strongest performance, proving that combining models helps capture different patterns in QSAR data.

- Compare Implementations:

Dataset 1: Wine Quality

- Manual Voting Classifier: Accuracy = 0.7396, F1 = 0.7525, ROC AUC = 0.8602
- Built-in Voting Classifier: Accuracy = 0.7708, F1 = 0.7876, ROC AUC = 0.8601

Observation: The built-in implementation performs noticeably better in terms of Accuracy, Precision, Recall, and F1-score, while ROC AUC is nearly identical

Dataset 2: QSAR Biodegradation

- Manual Voting Classifier: Accuracy = 0.8675, F1 = 0.7981, ROC AUC = 0.9189
- Built-in Voting Classifier: Accuracy = 0.8612, F1 = 0.7864, ROC AUC = 0.9189

Observation: The two implementations produce very close results, with the manual version slightly better in Accuracy and Recall, while ROC AUC is identical.

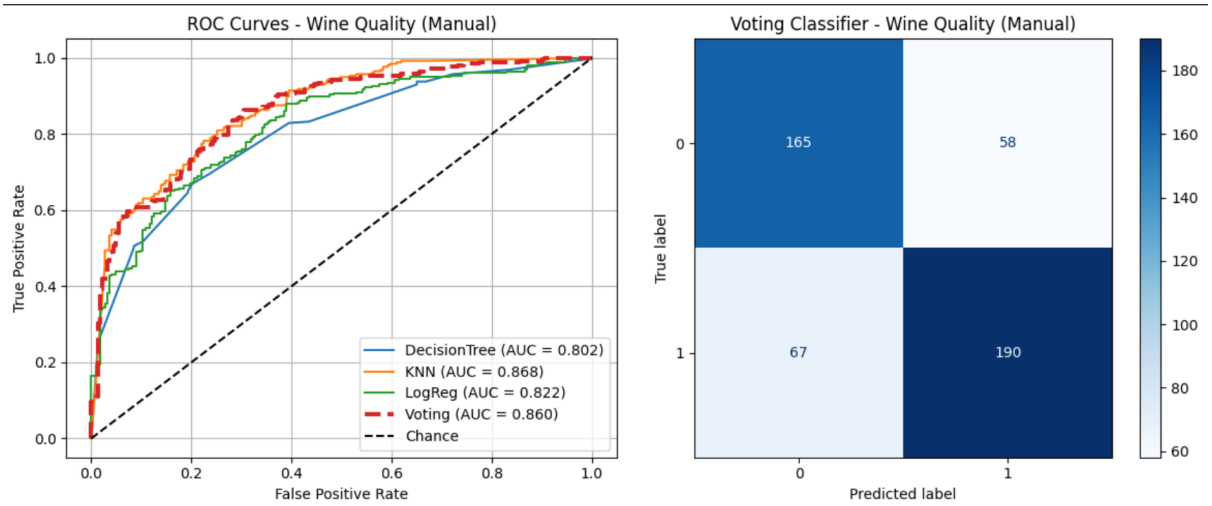
No, they are not identical.

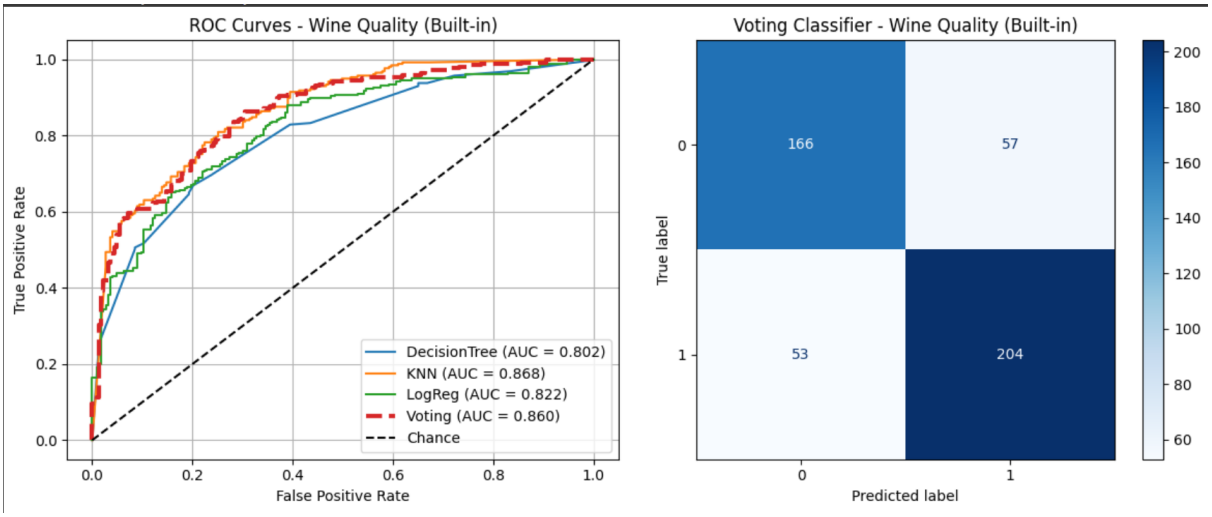
Reasons for differences:

First, hyperparameter coverage may differ, as the manual grid might not explore the same combinations as GridSearchCV. Second, cross-validation handling in GridSearchCV is more systematic, while the manual version may introduce slight inconsistencies in fold splitting. Third, randomness in data shuffling or fold assignment can cause small variations in results. Finally, GridSearchCV automatically refits the best model on the full training set,which can improve generalization compared to a manual refit.

- **Visualizations**:

Dataset 1: Wine Quality



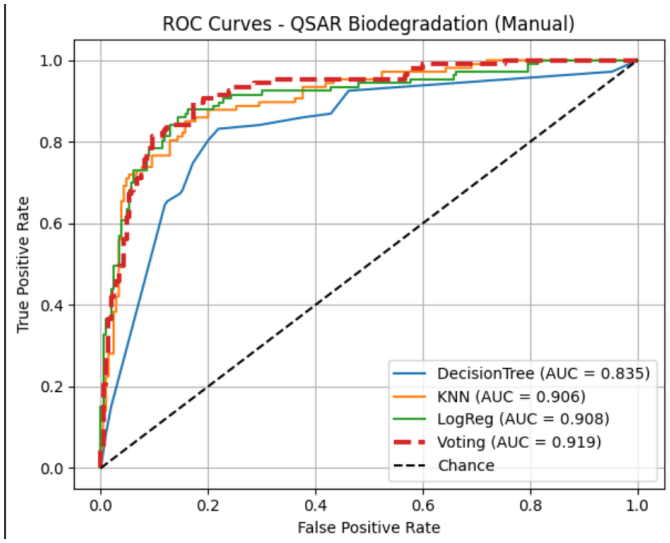


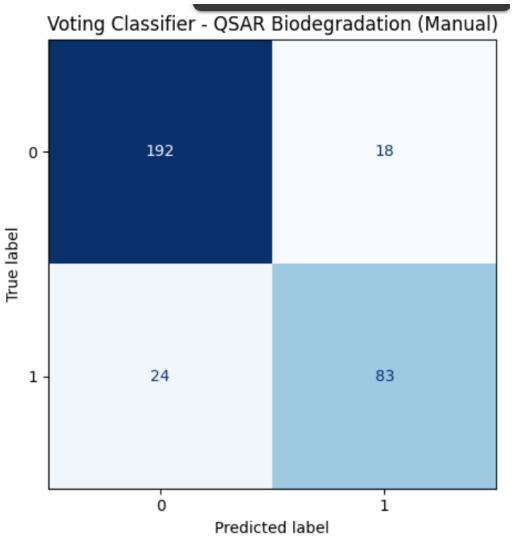
- ROC Curves:
  - Decision Tree again is weakest (AUC ≈ 0.802).
  - kNN achieves the best single-model performance (AUC ≈ 0.868).
  - Logistic Regression is moderate (AUC ≈ 0.822).
  - Voting Classifier (AUC ≈ 0.860) is competitive but doesn't surpass kNN significantly.
  - Manual vs. built-in curves are almost overlapping, suggesting consistent optimization.
- Confusion Matrices:
  - Built-in Voting Classifier: TP = 204, TN = 166, FP = 57, FN = 53
  - Manual Voting Classifier: TP = 190, TN = 165, FP = 58, FN = 67

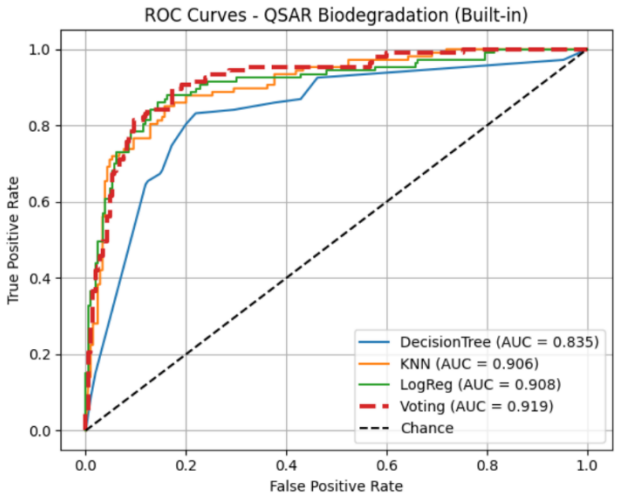
- The built-in version performs better, correctly classifying more positives (higher Recall and Accuracy). The manual version misses more true positives (higher FN count).

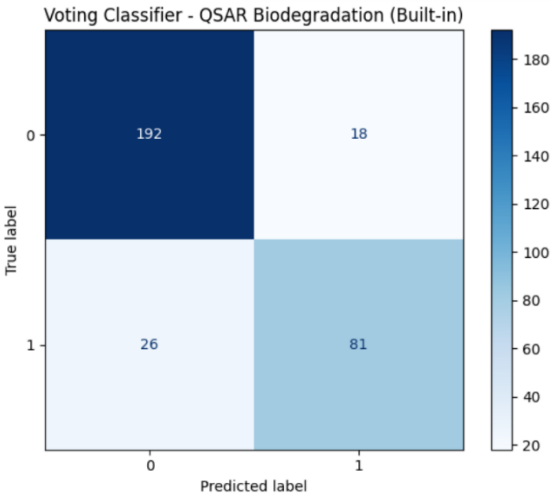
Conclusion: For Wine Quality, the built-in Voting Classifier outperforms the manual version, highlighting the advantage of GridSearchCV in tuning hyperparameters more effectively.

Dataset 2: QSAR Biodegradation









- ROC Curves:
  - Decision Tree lags behind (AUC ≈ 0.835).
  - kNN and Logistic Regression perform similarly (AUC ≈ 0.906–0.908).

        - The Voting Classifier outperforms all (AUC ≈ 0.919), showing the benefit of combining models.
        - Manual and built-in versions produce nearly identical ROC curves, confirming consistent performance.
- Confusion Matrices:
        - Built-in Voting Classifier: TP = 81, TN = 192, FP = 18, FN = 26
        - Manual Voting Classifier: TP = 83, TN = 192, FP = 18, FN = 24
        - Both perform almost the same, but the manual version catches slightly more positives (higher Recall) while the built-in one makes fewer false negatives but at the cost of slightly lower sensitivity.

Conclusion: Both methods achieve strong and very similar performance, with the Voting Classifier being the best overall model.

- Best Model: Analyze which model performed best overall for each dataset and offer a hypothesis as to why.

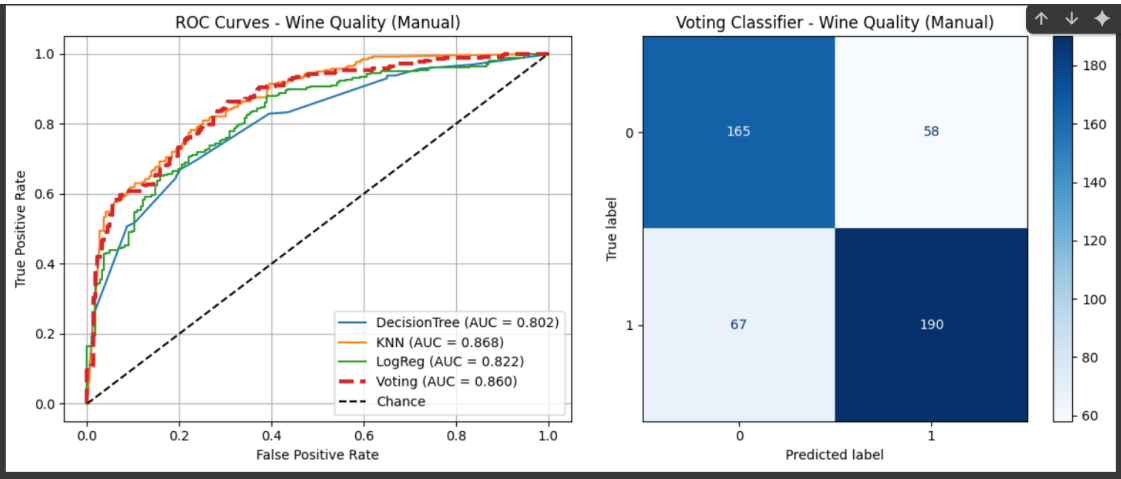
- Wine Quality → kNN is the best model, likely due to its ability to capture local non-linear structures.
- QSAR Biodegradation → Voting Classifier is the best model, as ensembling provides a stronger and more balanced performance on a dataset with mixed patterns.

5. Screenshots :

For wine quality dataset:

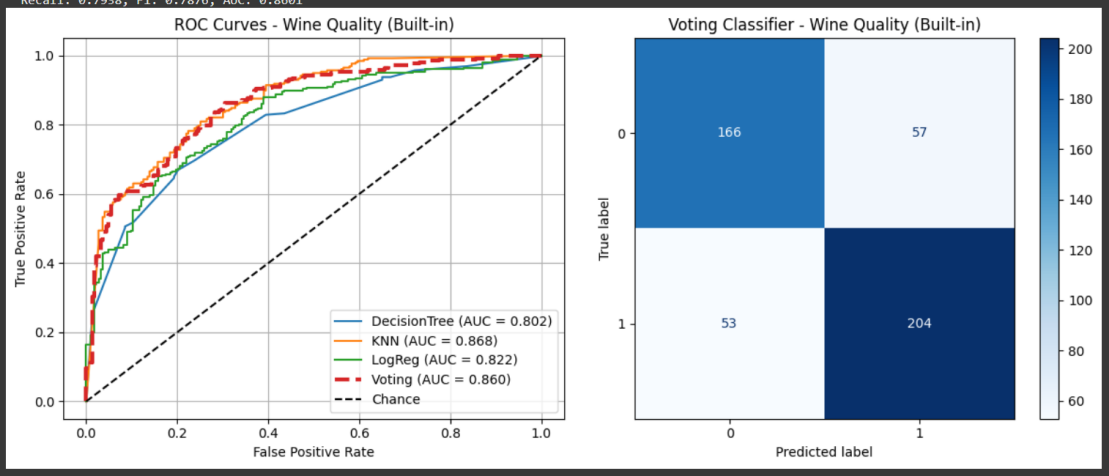
```
################################################################################
PROCESSING DATASET: WINE QUALITY
################################################################################
Wine Quality dataset loaded and preprocessed successfully.
Training set shape: (1119, 11)
Testing set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

============================================================
RUNNING MANUAL GRID SEARCH FOR WINE QUALITY
============================================================
--- Manual Grid Search for DecisionTree ---
----------------------------------------------------------------------------------------------
Best parameters for DecisionTree: {'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 LogReg ---
----------------------------------------------------------------------------------------------
Best parameters for LogReg: {'feature_selection__k': 8, 'classifier__C': 10, 'classifier__penalty': 'l1'}
Best cross-validation AUC: 0.8081

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

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

DecisionTree:
  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

LogReg:
  Accuracy: 0.7312
  Precision: 0.7520
  Recall: 0.7432
  F1-Score: 0.7476
  ROC AUC: 0.8219

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7396, Precision: 0.7661
  Recall: 0.7393, F1: 0.7525, AUC: 0.8602
```




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

--- GridSearchCV for DecisionTree ---
Best params for DecisionTree: {'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__n_neighbors': 9, 'classifier__weights': 'distance', 'feature_selection__k': 5}
Best CV score: 0.8642

--- GridSearchCV for LogReg ---
Best params for LogReg: {'classifier__C': 10, 'classifier__penalty': 'l2', 'feature_selection__k': 8}
Best CV score: 0.8081

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

--- Individual Model Performance ---

DecisionTree:
  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

LogReg:
  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: 0.8601
```

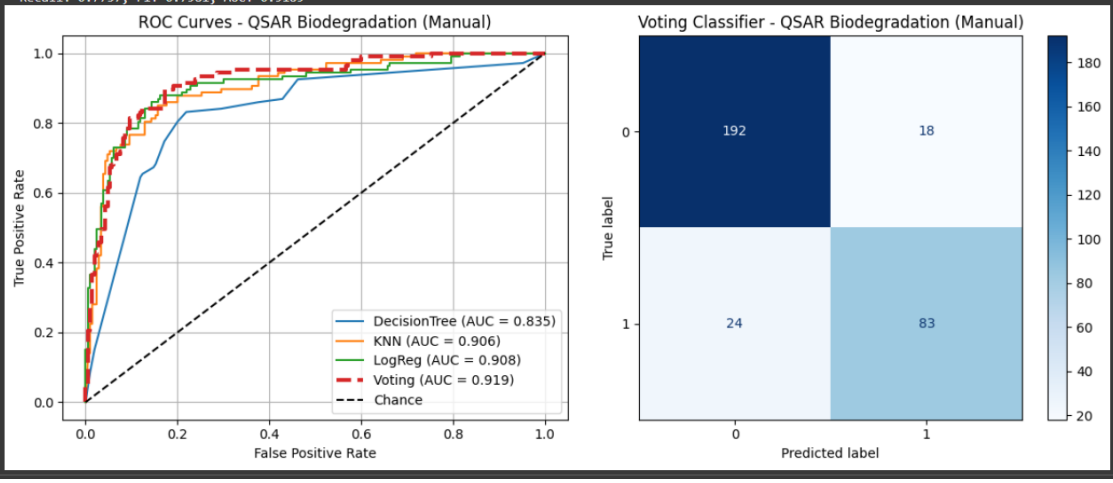



```
Completed processing for Wine Quality
================================================================================
```

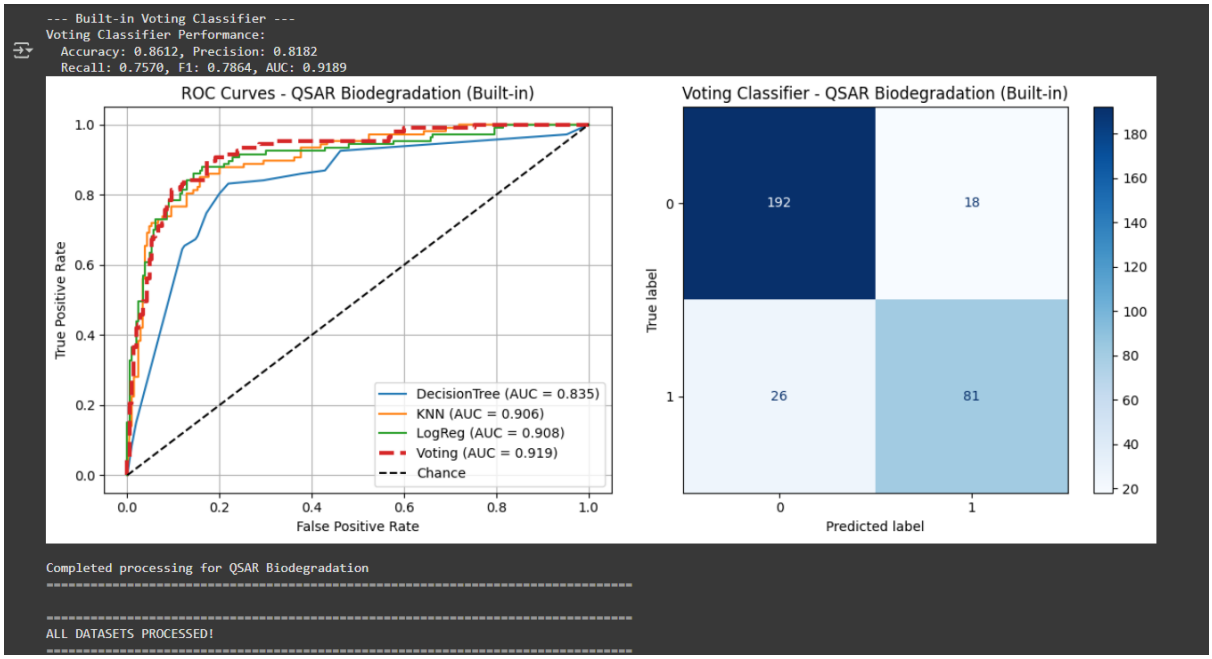
For QSAR Biodegradation:

```
Commands | + Code | + Text | Run all

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

============================================================
RUNNING MANUAL GRID SEARCH FOR QSAR BIODEGRADATION
============================================================
--- Manual Grid Search for DecisionTree ---
----------------------------------------------------------------------------------------------------
Best parameters for DecisionTree: {'feature_selection__k': 30, 'classifier__max_depth': 5, 'classifier__min_samples_split': 10}
Best cross-validation AUC: 0.8505
--- Manual Grid Search for KNN ---
----------------------------------------------------------------------------------------------------
Best parameters for KNN: {'feature_selection__k': 30, 'classifier__n_neighbors': 15, 'classifier__weights': 'distance'}
Best cross-validation AUC: 0.9056
--- Manual Grid Search for LogReg ---
----------------------------------------------------------------------------------------------------
Best parameters for LogReg: {'feature_selection__k': 'all', 'classifier__C': 1, 'classifier__penalty': 'l1'}
Best cross-validation AUC: 0.9318

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

--- Individual Model Performance ---

DecisionTree:
  Accuracy: 0.7918
  Precision: 0.6990
  Recall: 0.6729
  F1-Score: 0.6857
  ROC AUC: 0.8347
```

```
KNN:
  Accuracy: 0.8423
  Precision: 0.7664
  Recall: 0.7664
  F1-Score: 0.7664
  ROC AUC: 0.9059

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

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8675, Precision: 0.8218
  Recall: 0.7757, F1: 0.7981, AUC: 0.9189
```




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

--- GridSearchCV for DecisionTree ---
Best params for DecisionTree: {'classifier__max_depth': 5, 'classifier__min_samples_split': 10, 'feature_selection__k': 30}
Best CV score: 0.8505

--- GridSearchCV for KNN ---
Best params for KNN: {'classifier__n_neighbors': 15, 'classifier__weights': 'distance', 'feature_selection__k': 30}
Best CV score: 0.9056

--- GridSearchCV for LogReg ---
Best params for LogReg: {'classifier__C': 1, 'classifier__penalty': 'l1', 'feature_selection__k': 'all'}
Best CV score: 0.9318

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

--- Individual Model Performance ---

DecisionTree:
  Accuracy: 0.7918
  Precision: 0.6990
  Recall: 0.6729
  F1-Score: 0.6857
  ROC AUC: 0.8347

KNN:
  Accuracy: 0.8423
  Precision: 0.7664
  Recall: 0.7664
  F1-Score: 0.7664
  ROC AUC: 0.9059

LogReg:
  Accuracy: 0.8644
  Precision: 0.8200
  Recall: 0.7664
  F1-Score: 0.7923
  ROC AUC: 0.9082
```



Conclusion :

From this lab, we learned how different models perform on different datasets and how important proper model selection is. For the Wine Quality dataset, the k-Nearest Neighbors (kNN) model worked best, most likely because the data had non-linear patterns that kNN could capture by comparing neighborhoods of samples. On the other hand, for the QSAR Biodegradation dataset, the Voting Classifier gave the best results since it combined the strengths of Logistic Regression and kNN, leading to a more balanced and reliable prediction.

Another key lesson was the impact of hyperparameter tuning. The manual grid search helped us understand how cross-validation works and how model parameters affect performance. However, scikit-learn's GridSearchCV was faster, more systematic, and often slightly more accurate, especially for Wine Quality.

The main takeaway is that while manual implementation builds a solid foundation for understanding, using tools like GridSearchCV is much more practical when working with larger datasets or more complex problems. It shows the trade-off between learning the "behind-the-scenes" process versus relying on efficient libraries for real-world applications.