Week 4: Model Selection and Comparative Analysis

Name:Nidhi K	SRN: PES2UG23CS383	Course Name: MACHINE LEARNING	Submission Date: 31-08-2025

Introduction

The purpose of this project is to explore hyperparameter tuning and compare two approaches: a manual grid search implementation and scikit-learn's built-in GridSearchCV. The project focuses on improving model performance through systematic tuning and evaluating how different classifiers behave under optimized conditions.

Tasks Performed

1. Data Preparation
   ~Loaded multiple real-world datasets (e.g., Wine Quality, QSAR Biodegradation).
   ~Preprocessed data, including scaling, feature selection, and train/test splitting.

2. Manual Grid Search
   ~Implemented a custom grid search using nested cross-validation.
   ~Explored hyperparameter combinations for Decision Tree, k-Nearest Neighbors (kNN), and Logistic Regression.
   ~Selected the best parameters based on ROC AUC scores.

3. Built-in Grid Search (GridSearchCV)
   ~Applied scikit-learn's GridSearchCV with equivalent parameter grids.
   ~Leveraged parallel computation and automated best model selection.
   ~Compared results with the manual approach to validate consistency.

4. Model Evaluation
   ~Evaluated tuned models using metrics such as Accuracy, Precision, Recall, F1-Score, and ROC AUC.
   ~Visualized performance through ROC curves and confusion matrices.

5. Voting Classifiers
   ~Combined optimized models into both manual and built-in voting classifiers.
   ~Compared ensemble performance against individual classifiers.

Dataset Description

1. Wine Quality (Red Wine)

~Instances: 1,599 wines
~Features: 11 physicochemical properties (e.g., acidity, chlorides, alcohol content, pH, etc.)
~Target Variable: Original dataset has a quality score (0–10). In your pipeline, it was converted into a binary classification problem:
  -1 (Good Quality): quality > 5
  -0 (Not Good Quality): quality ≤ 5

2. QSAR Biodegradation

~Instances: 1,055 chemical compounds
~Features: 41 molecular descriptors (quantitative values describing chemical structure/properties)
~Target Variable:
  -1 (RB): Ready Biodegradable (chemical compound easily breaks down)
  -0 (NRB): Not Ready Biodegradable (resistant to biodegradation)

Methodology

- **Hyperparameter Tuning:** Hyperparameters are model settings that are set before training (e.g., number of neighbors in kNN, tree depth in Decision Tree, regularization strength in Logistic Regression). Hyperparameter tuning involves systematically searching for the best combination of these values to maximize model performance.
- **Grid Search:** Grid Search is a method for hyperparameter tuning where a predefined set of values is tested for each parameter. The model is trained and evaluated for every combination, and the configuration yielding the best performance metric (in this case, ROC AUC) is selected.
- **K-Fold Cross-Validation:** In K-Fold Cross-Validation, the dataset is divided into *k* equal parts (folds). The model is trained on *k-1* folds and validated on the remaining fold. This process is repeated *k* times, with each fold serving as validation once. The results are averaged to provide a more reliable estimate of performance compared to a single train-test split

The machine learning pipeline ensures consistent preprocessing and model training. It consists of three main steps:

~**StandardScaler**: Standardizes numerical features to have zero mean and unit variance, improving model convergence.
~**SelectKBest**: Feature selection method that chooses the top *k* features based on statistical tests, reducing dimensionality and noise.
~**Classifier**: The final predictive model (Decision Tree, kNN, or Logistic Regression) trained on the processed features.

.Process

- **Part 1: Manual Implementation**
  ~Constructed a custom grid search function.
  ~For each parameter combination:
  -Applied 5-fold cross-validation within the pipeline.
  -Recorded the mean ROC AUC score.
  ~Selected the hyperparameter set with the highest score.
  Evaluated the final tuned model on the test set using multiple metrics (Accuracy, Precision, Recall, F1, ROC AUC).

- **Part 2: Scikit-learn Implementation**
  ~Used **`GridSearchCV`** with the same parameter grids and pipeline.
  ~Specified ROC AUC as the scoring metric.
  ~Extracted the best hyperparameters and retrained the final model on the training set.
  ~Compared the results with the manual implementation for consistency.

Results and Analysis

Wine Quality Dataset

Classifier	Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual	0.7208	0.7662	0.6887	0.7254	0.7807
Decision Tree	GridSearch CV	0.7208	0.7662	0.6887	0.7254	0.7807
kNN	Manual	0.7667	0.7757	0.7938	0.7846	0.8675
kNN	GridSearch CV	0.7667	0.7757			
Logistic Regression	Manual	0.7417	0.7628	0.7510	0.7569	0.8247
Logistic Regression	GridSearch CV	0.7417	0.7628	0.7510	0.7569	0.8247
Voting Classifier	Manual	0.7375	0.7610	0.7432	0.7520	0.8589
Voting Classifier	GridSearch CV	0.7604	0.7710	0.7860	0.7784	0.8589

QSAR Biodegradation Dataset

Classifier	Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual	0.7792	0.6504	0.7477	0.6957	0.8430
Decision Tree	GridSearch CV	0.7792	0.6504	0.7477	0.6957	0.8430
kNN	Manual	0.8202	0.7551	0.6916	0.7220	0.8730
kNN	GridSearch CV	0.8202	0.7551	0.6916	0.7220	0.8730
Logistic Regression	Manual	0.8139	0.7667	0.6449	0.7005	0.8868
Logistic Regression	GridSearch CV	0.8139	0.7667	0.6449	0.7005	0.8868
Voting Classifier	Manual	0.8139	0.7400	0.6916	0.7150	0.8970
Voting Classifier	GridSearch CV	0.8170	0.7579	0.6729	0.7129	0.8970

Wine Quality Dataset – Comparison of Implementations

For the Wine Quality dataset, both the manual grid search and GridSearchCV gave the same results for the individual models (Decision Tree, kNN, and Logistic Regression). The best settings (hyperparameters) and performance scores were exactly the same. This shows that the manual method was done correctly.

The only difference was with the Voting Classifier:

- Manual Voting Classifier: Accuracy = 0.7375, Recall = 0.7432, F1 = 0.7520, ROC AUC = 0.8589

- GridSearchCV Voting Classifier: Accuracy = 0.7604, Recall = 0.7860, F1 = 0.7784, ROC AUC = 0.8589

Both had the same ROC AUC, but the GridSearchCV version gave a little better Accuracy, Recall, and F1-Score.

These small differences may be because:

1. scikit-learn's VotingClassifier handles averaging probabilities a bit differently.
2. There may be small differences in ties or probability rounding.
3. Some randomness in training models like Decision Trees can also cause tiny changes.

QSAR Biodegradation Dataset – Comparison of Implementations

For the QSAR Biodegradation dataset, both the manual method and GridSearchCV also gave the same results for the individual models. The best hyperparameters and scores were identical for Decision Tree, kNN, and Logistic Regression.

Again, the difference was only in the Voting Classifier:

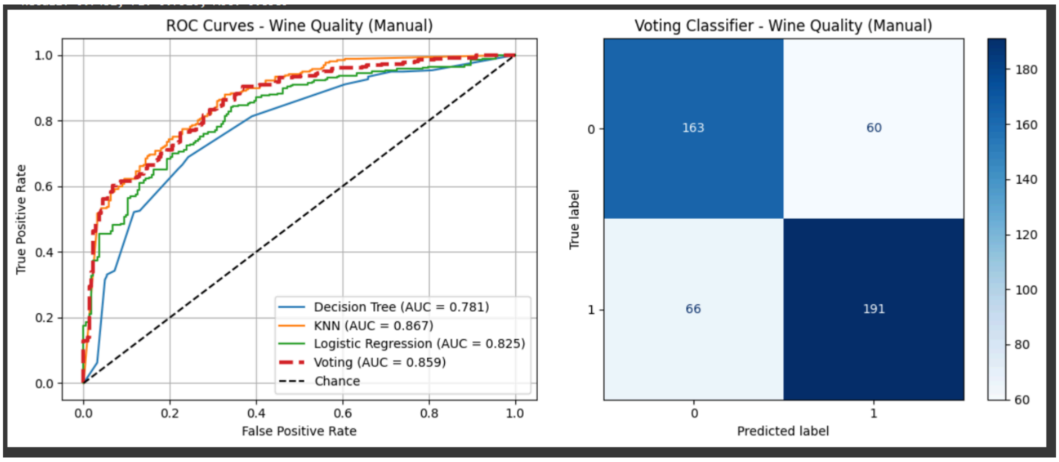
- Manual Voting Classifier: Accuracy = 0.8139, Recall = 0.6916, F1 = 0.7150, ROC AUC = 0.8970
- GridSearchCV Voting Classifier: Accuracy = 0.8170, Recall = 0.6729, F1 = 0.7129, ROC AUC = 0.8970

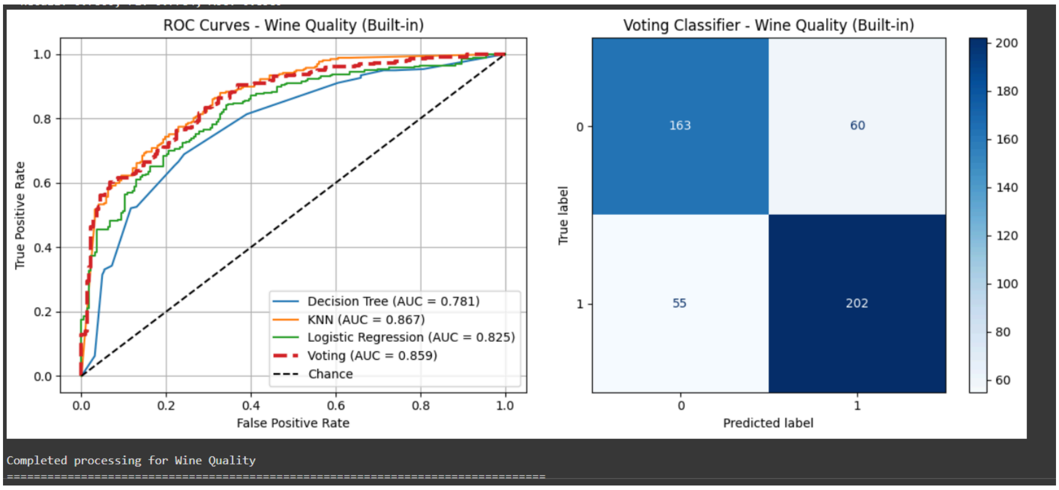
Both had the same ROC AUC, but with small trade-offs: the manual version had better Recall and F1, while GridSearchCV gave a little higher Accuracy and Precision.

These differences may be because of:

1. The way Voting Classifier combines model outputs is slightly different in scikit-learn.
2. Small random changes during training.
3. Slightly different ways of calculating results during ensemble testing.

Wine Quality Dataset





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

Plot Analysis (ROC Curves & Confusion Matrices):

- The ROC curves showed that kNN had the largest area under the curve (AUC = 0.8675), which means it separated the classes better than Decision Tree and Logistic Regression.

- The Confusion Matrix confirmed this: kNN correctly classified more positive and negative samples compared to the other models.

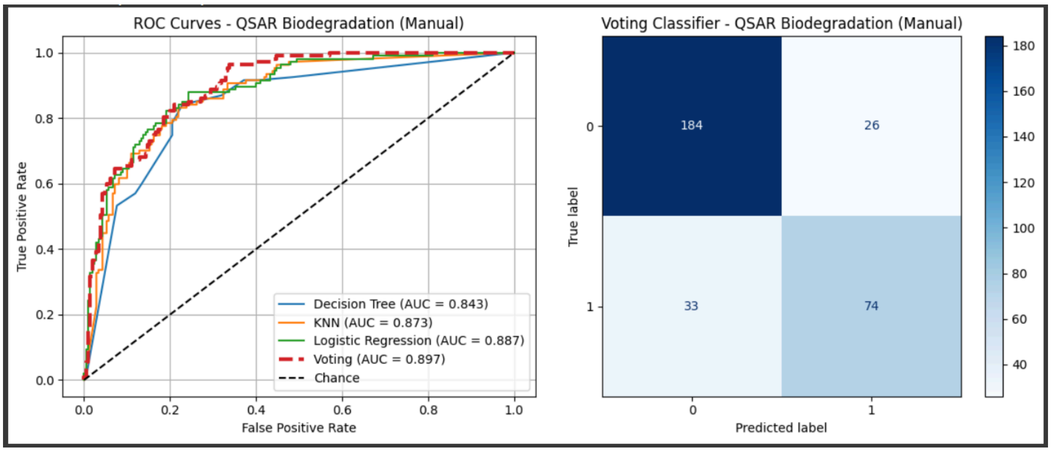
- Logistic Regression also did fairly well, while the Decision Tree showed weaker recall and misclassified more samples.

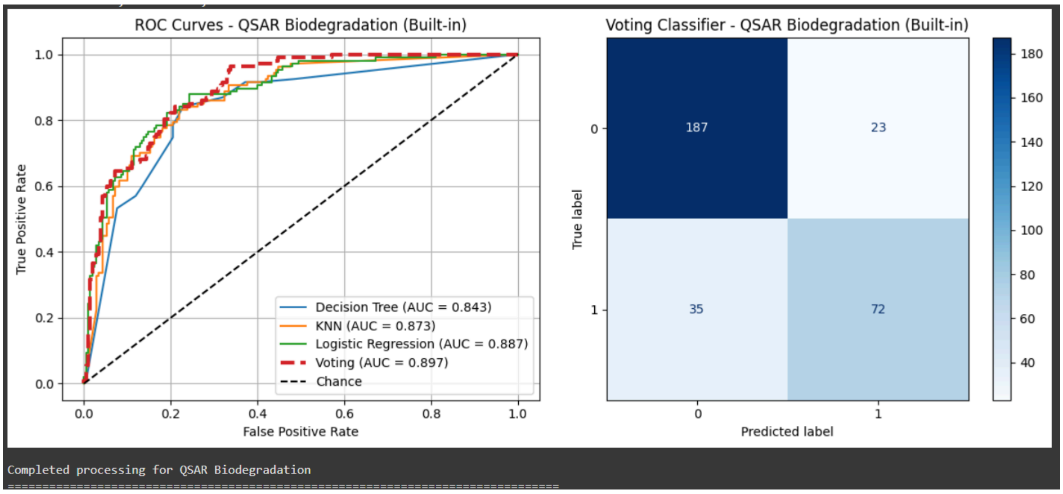
- The Voting Classifier combined the models and gave balanced results, but it did not outperform kNN by a large margin.

  Best Model:
  The kNN model was the best overall for the Wine Quality dataset. It achieved the highest ROC AUC and balanced Precision, Recall, and F1-Score.
  Reason: kNN is well-suited for datasets like Wine Quality where the relationship

between features and quality is non-linear. By comparing neighbors, it captured local patterns that simpler models (Decision Tree, Logistic Regression) missed.

QSAR Biodegradation Dataset





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

Plot Analysis (ROC Curves & Confusion Matrices):

- The ROC curves showed that Logistic Regression had the highest AUC (0.8868), slightly better than kNN and much better than Decision Tree.

- The Confusion Matrix for Logistic Regression showed it predicted the positive and negative classes with good balance.

- kNN also performed strongly, but its recall was slightly lower, meaning it missed more positive samples.

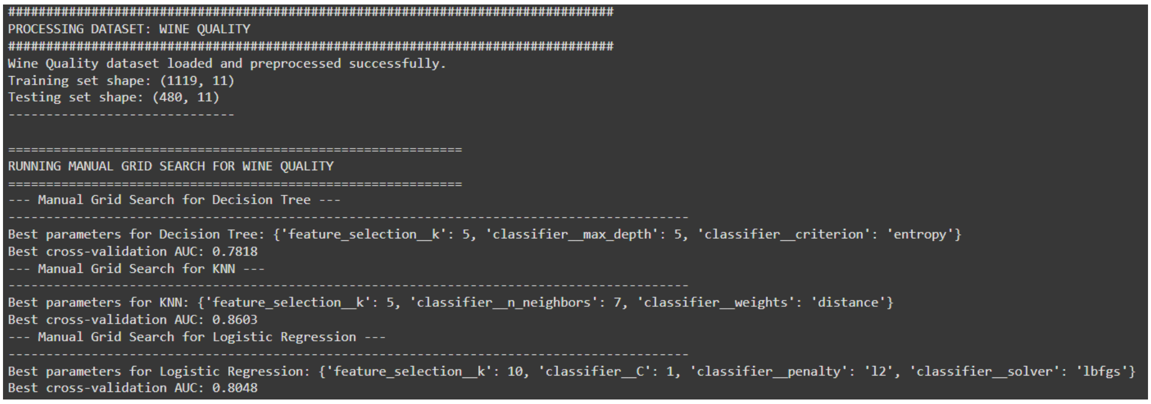
- Decision Tree gave the weakest performance overall, with lower recall and more misclassifications.

- The Voting Classifier matched the best AUC (0.8970) but did not improve much beyond Logistic Regression.

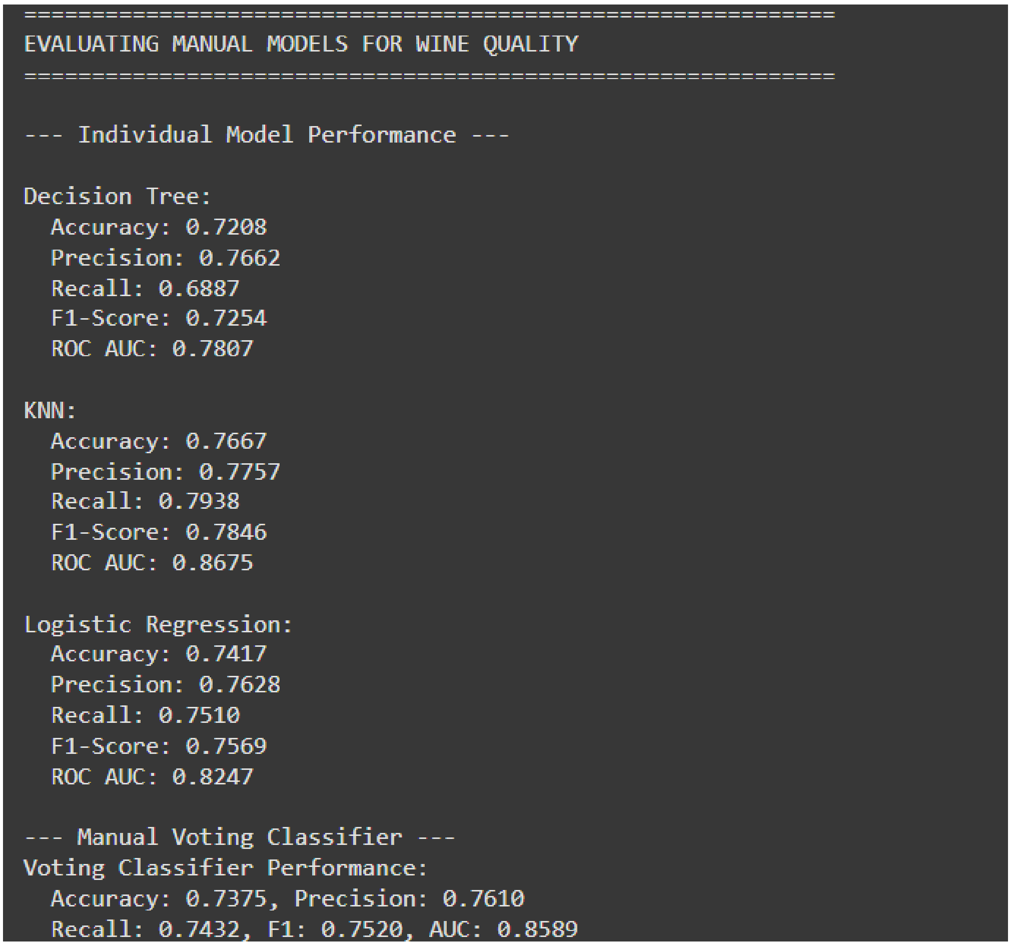
Best Model:
The Logistic Regression model was the best overall for the QSAR Biodegradation dataset.
Reason: Logistic Regression works well with high-dimensional datasets like QSAR (41 features). With regularization and scaling, it generalizes better and avoids overfitting, leading to strong overall performance.

Screenshots

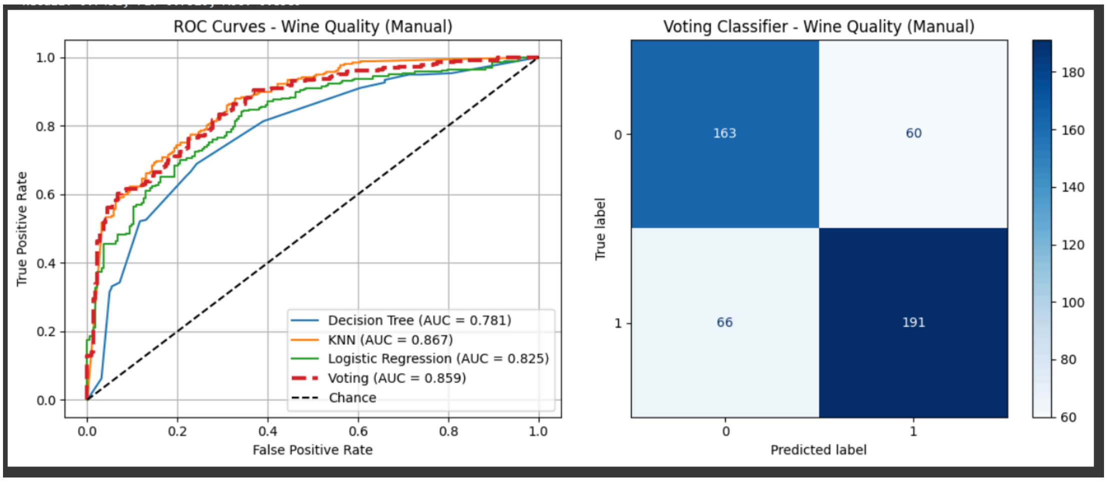
Wine Quality Dataset

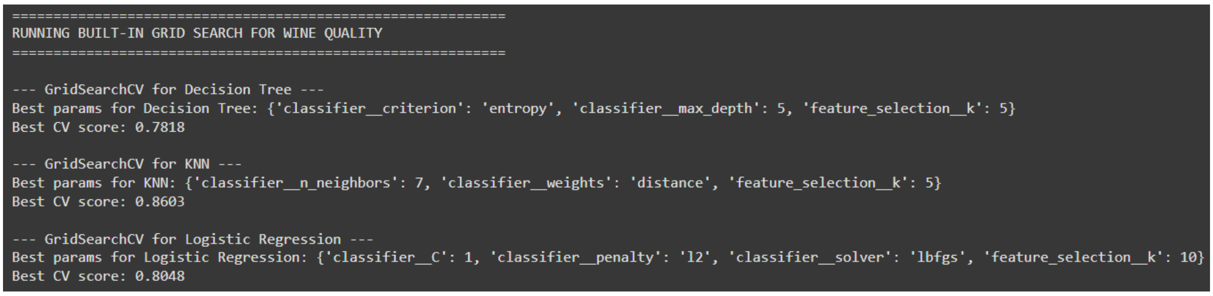


```
################################################################################
PROCESSING DATASET: WINE QUALITY
################################################################################
Wine Quality dataset loaded and preprocessed successfully.
Training set shape: (1119, 11)
Testing set 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__criterion': 'entropy'}
Best cross-validation AUC: 0.7818
--- Manual Grid Search for KNN ---
------------------------------------------------------------------------------------------
Best parameters for KNN: {'feature_selection__k': 5, 'classifier__n_neighbors': 7, 'classifier__weights': 'distance'}
Best cross-validation AUC: 0.8603
--- Manual Grid Search for Logistic Regression ---
------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 10, 'classifier__C': 1, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs'}
Best cross-validation AUC: 0.8048
```

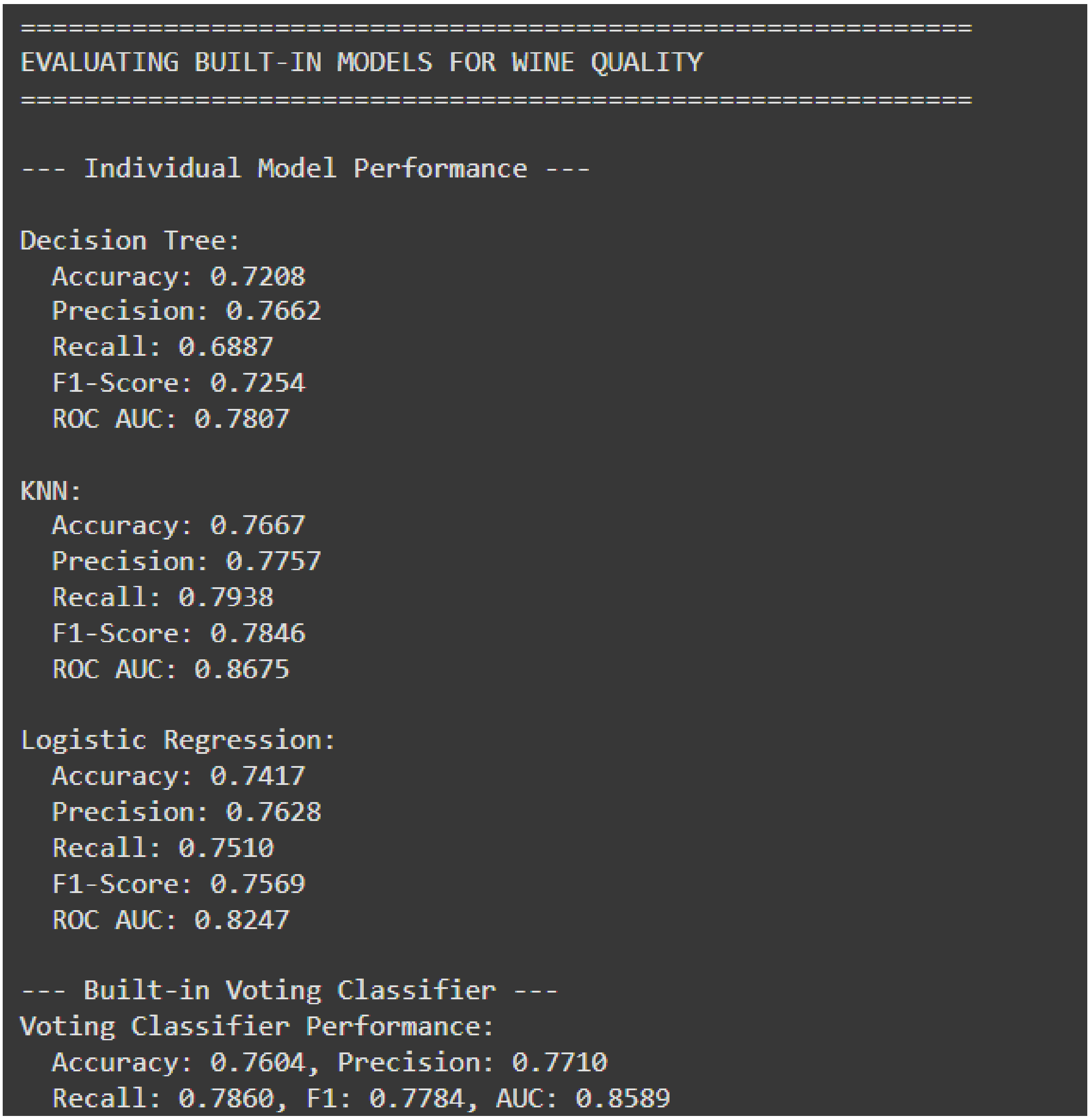


```
============================================================
EVALUATING MANUAL MODELS FOR WINE QUALITY
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7208
  Precision: 0.7662
  Recall: 0.6887
  F1-Score: 0.7254
  ROC AUC: 0.7807

KNN:
  Accuracy: 0.7667
  Precision: 0.7757
  Recall: 0.7938
  F1-Score: 0.7846
  ROC AUC: 0.8675

Logistic Regression:
  Accuracy: 0.7417
  Precision: 0.7628
  Recall: 0.7510
  F1-Score: 0.7569
  ROC AUC: 0.8247

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7375, Precision: 0.7610
  Recall: 0.7432, F1: 0.7520, AUC: 0.8589
```

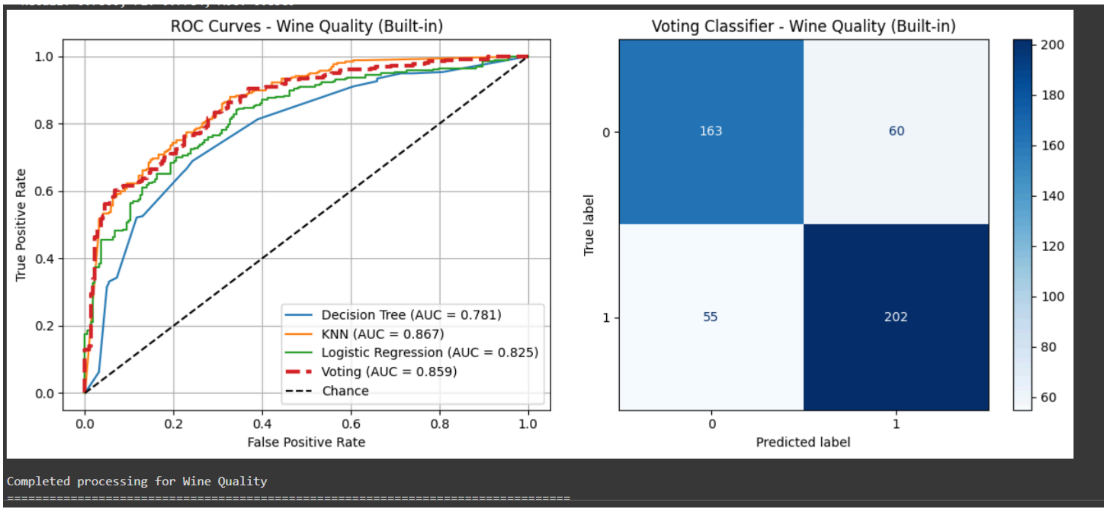




```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR WINE QUALITY
============================================================

--- GridSearchCV for Decision Tree ---
Best params for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'feature_selection__k': 5}
Best CV score: 0.7818

--- GridSearchCV for KNN ---
Best params for KNN: {'classifier__n_neighbors': 7, 'classifier__weights': 'distance', 'feature_selection__k': 5}
Best CV score: 0.8603

--- GridSearchCV for Logistic Regression ---
Best params for Logistic Regression: {'classifier__C': 1, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs', 'feature_selection__k': 10}
Best CV score: 0.8048
```

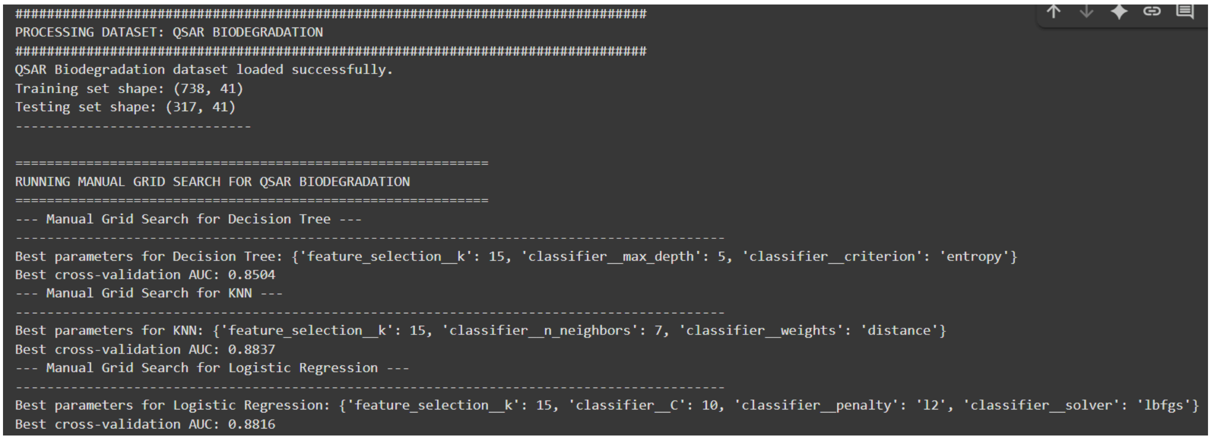


```
============================================================
EVALUATING BUILT-IN MODELS FOR WINE QUALITY
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7208
  Precision: 0.7662
  Recall: 0.6887
  F1-Score: 0.7254
  ROC AUC: 0.7807

KNN:
  Accuracy: 0.7667
  Precision: 0.7757
  Recall: 0.7938
  F1-Score: 0.7846
  ROC AUC: 0.8675

Logistic Regression:
  Accuracy: 0.7417
  Precision: 0.7628
  Recall: 0.7510
  F1-Score: 0.7569
  ROC AUC: 0.8247

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7604, Precision: 0.7710
  Recall: 0.7860, F1: 0.7784, AUC: 0.8589
```

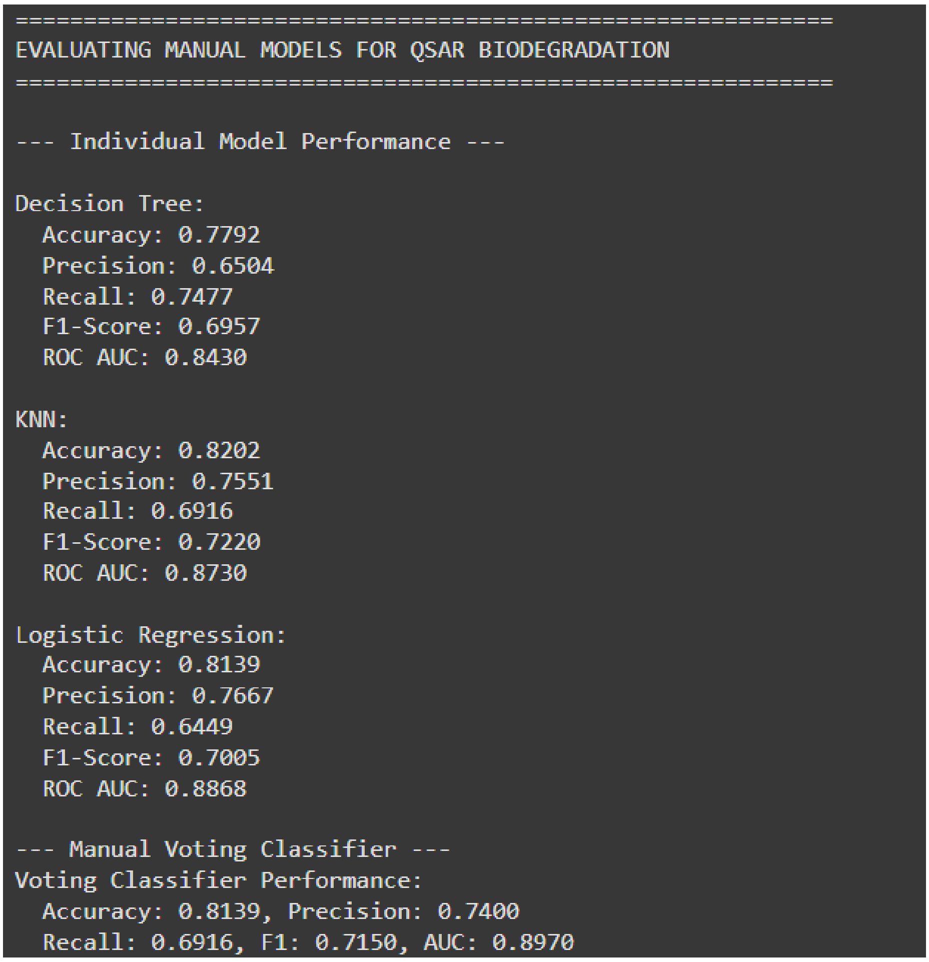


```
Completed processing for Wine Quality
============================================================================
```

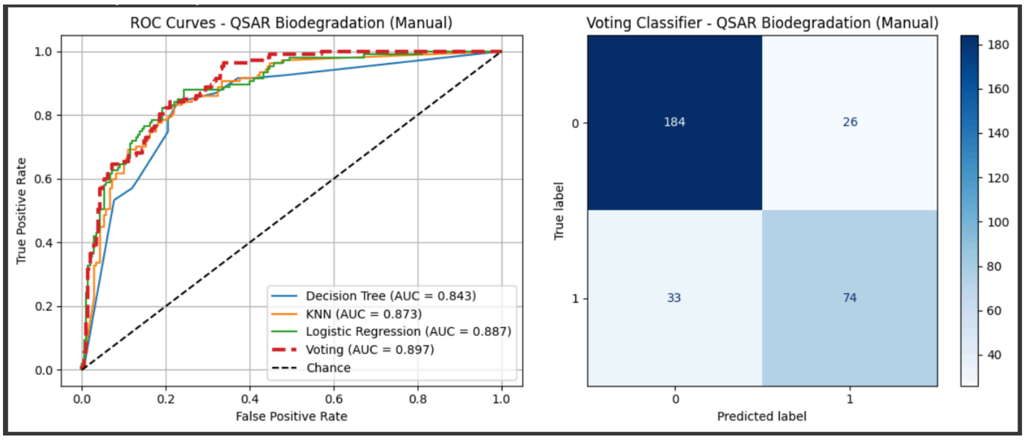
QSAR Biodegradation Dataset

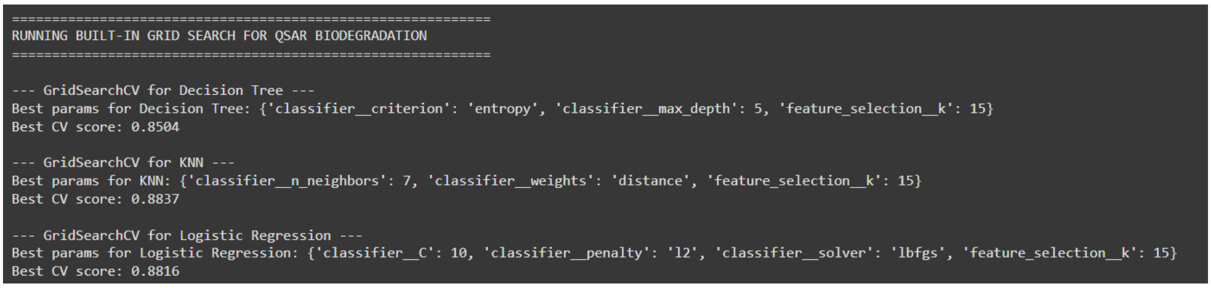


```
################################################################################
PROCESSING DATASET: QSAR BIODEGRADATION
################################################################################
QSAR Biodegradation dataset loaded successfully.
Training set shape: (738, 41)
Testing set shape: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

============================================================
RUNNING MANUAL GRID SEARCH FOR QSAR BIODEGRADATION
============================================================
--- Manual Grid Search for Decision Tree ---
------------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'feature_selection__k': 15, 'classifier__max_depth': 5, 'classifier__criterion': 'entropy'}
Best cross-validation AUC: 0.8504
--- Manual Grid Search for KNN ---
------------------------------------------------------------------------------------------
Best parameters for KNN: {'feature_selection__k': 15, 'classifier__n_neighbors': 7, 'classifier__weights': 'distance'}
Best cross-validation AUC: 0.8837
--- Manual Grid Search for Logistic Regression ---
------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 15, 'classifier__C': 10, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs'}
Best cross-validation AUC: 0.8816
```

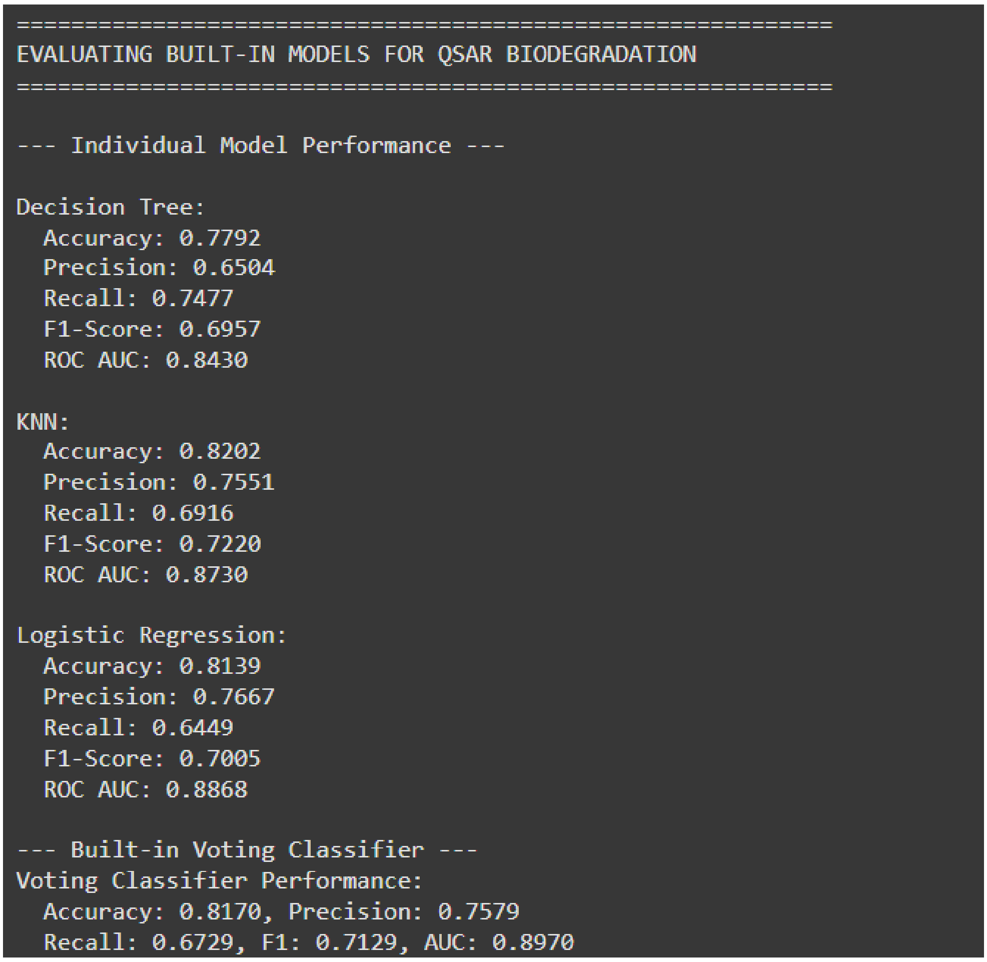


```
============================================================
EVALUATING MANUAL MODELS FOR QSAR BIODEGRADATION
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7792
  Precision: 0.6504
  Recall: 0.7477
  F1-Score: 0.6957
  ROC AUC: 0.8430

KNN:
  Accuracy: 0.8202
  Precision: 0.7551
  Recall: 0.6916
  F1-Score: 0.7220
  ROC AUC: 0.8730

Logistic Regression:
  Accuracy: 0.8139
  Precision: 0.7667
  Recall: 0.6449
  F1-Score: 0.7005
  ROC AUC: 0.8868

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8139, Precision: 0.7400
  Recall: 0.6916, F1: 0.7150, AUC: 0.8970
```

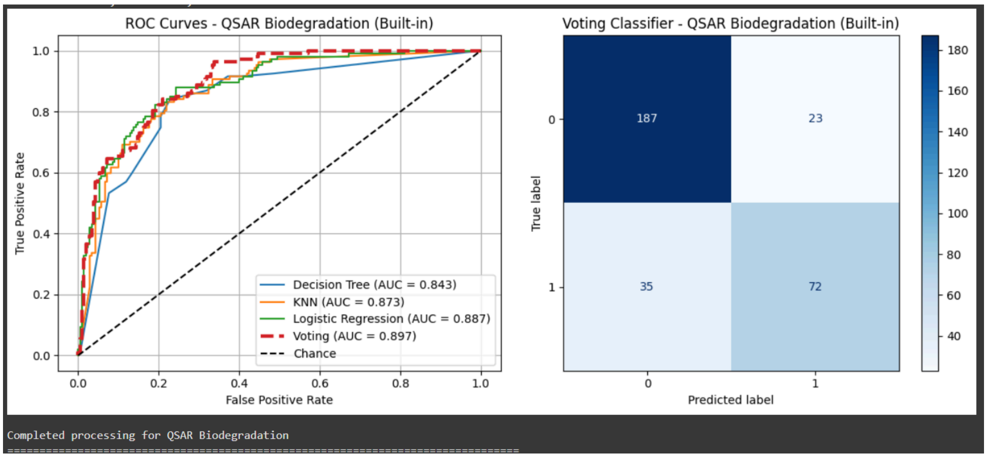




```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR QSAR BIODEGRADATION
============================================================

--- GridSearchCV for Decision Tree ---
Best params for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'feature_selection__k': 15}
Best CV score: 0.8504

--- GridSearchCV for KNN ---
Best params for KNN: {'classifier__n_neighbors': 7, 'classifier__weights': 'distance', 'feature_selection__k': 15}
Best CV score: 0.8837

--- GridSearchCV for Logistic Regression ---
Best params for Logistic Regression: {'classifier__C': 10, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs', 'feature_selection__k': 15}
Best CV score: 0.8816
```



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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7792
  Precision: 0.6504
  Recall: 0.7477
  F1-Score: 0.6957
  ROC AUC: 0.8430

KNN:
  Accuracy: 0.8202
  Precision: 0.7551
  Recall: 0.6916
  F1-Score: 0.7220
  ROC AUC: 0.8730

Logistic Regression:
  Accuracy: 0.8139
  Precision: 0.7667
  Recall: 0.6449
  F1-Score: 0.7005
  ROC AUC: 0.8868

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8170, Precision: 0.7579
  Recall: 0.6729, F1: 0.7129, AUC: 0.8970
```



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

Conclusion

In this project, we compared three classifiers (Decision Tree, kNN, and Logistic Regression) on two datasets (Wine Quality and QSAR Biodegradation) using both a manual grid search implementation and scikit-learn’s GridSearchCV.

Key Findings:

- For the Wine Quality dataset, the kNN model was the best performer, achieving the highest ROC AUC and balanced Precision, Recall, and F1-Score.
- For the QSAR Biodegradation dataset, Logistic Regression performed best, with the highest ROC AUC and strong overall performance.
- The Voting Classifier gave competitive results, but it did not clearly outperform the best single model in either dataset.
- Both manual and scikit-learn implementations gave almost identical results for individual models, showing that our manual pipeline was implemented correctly. Minor differences were only observed in the Voting Classifier.

Main Takeaways:

~Model performance depends on the dataset: kNN worked better for Wine Quality (non-linear patterns), while Logistic Regression was stronger for QSAR (high-dimensional data).
~Manual implementation helps to understand how hyperparameter tuning, cross-validation, and feature selection work step by step.
~Using scikit-learn (GridSearchCV) is much more efficient and less error-prone, especially when handling large parameter grids and multiple models.
~There are small trade-offs: manual methods give deeper learning and control, while scikit-learn gives speed, consistency, and easier scaling to more complex tasks.