MODEL SELECTION LAB4

Mrunal Manjunath Kudtarkar
PES2UG23CS354
Machine Learning
29-08-2025

1. Introduction

   The aim of this lab is to explore hyperparameter tuning for different classification models using Grid Search CV. Machine learning models often rely on parameters that need to be optimized for best performance. In this experiment, we compare three supervised learning models — Decision Tree, k-Nearest Neighbours (KNN), and Logistic Regression — to evaluate their classification performance on the dataset.

2. Dataset Description

   The dataset provided contains both features (independent variables) and labels (target class).

- Features: Numerical/categorical attributes used for training.
- Target: Binary/multi-class label to be predicted.
- Train/Test Split: Data is divided into training and testing sets for evaluation.

  Preprocessing steps included:

- Scaling features (for KNN and Logistic Regression).
- Using a pipeline to combine preprocessing and model training.

3. Methodology

   We used **scikit-learn Pipelines** and **GridSearchCV** for parameter tuning.

   Models and Hyperparameters

1. **Decision Tree Classifier**
   - max_depth: [3, 5, 10, None]
   - min_samples_split: [2, 5, 10]
2. **k-Nearest Neighbors (kNN)**
   - n_neighbours: [3, 5, 7, 9]
   - weights: ['uniform', 'distance']
   - metric: ['euclidean', 'manhattan']
3. **Logistic Regression**
   - C: [0.01, 0.1, 1, 10, 100]
   - penalty: ['l2']
   - solver: ['lbfgs']

   Evaluation Metrics

- Accuracy
- Precision
- Recall
- F1-score

- ROC-AUC
- Confusion Matrix
- ROC Curve

4. Results and Analysis

Wine Quality Dataset

- **Best Models**:
    - kNN performed the best overall with ROC AUC ≈ **0.868**, accuracy ≈ **0.775**, and strong balance across precision, recall, and F1.
    - Logistic Regression was the next best, while Decision Tree lagged behind.
- **Voting Classifier**: Interestingly, the manual voting ensemble didn't beat the standalone kNN (AUC 0.861 vs. 0.868).
- **Manual vs. Built-in**: Results for both implementations were **identical**, showing that the manual grid search was implemented correctly.

HR Attrition Dataset

- **Best Models**:
    - Logistic Regression was the strongest with ROC AUC ≈ **0.776** and accuracy ≈ **0.857**.
    - kNN and Decision Tree performed poorly, especially in recall and F1 (they struggled to capture attrition cases due to dataset imbalance).
- **Voting Classifier**: Did not improve performance significantly; recall remained weak.
- **Manual vs. Built-in**: Results again matched perfectly. Minor differences in performance metrics across models are due to dataset imbalance. Logistic Regression handles imbalance better with probability-based decision boundaries.

Banknote Authentication Dataset

- **Issue**: Manual grid search failed (set_params error). No results available for this dataset.
- **Analysis**: Suggests either the parameter grid wasn't applied correctly for Decision Tree or the dataset's small number of features (n=4) conflicted with SelectKBest. Needs fixing before comparison can be made.

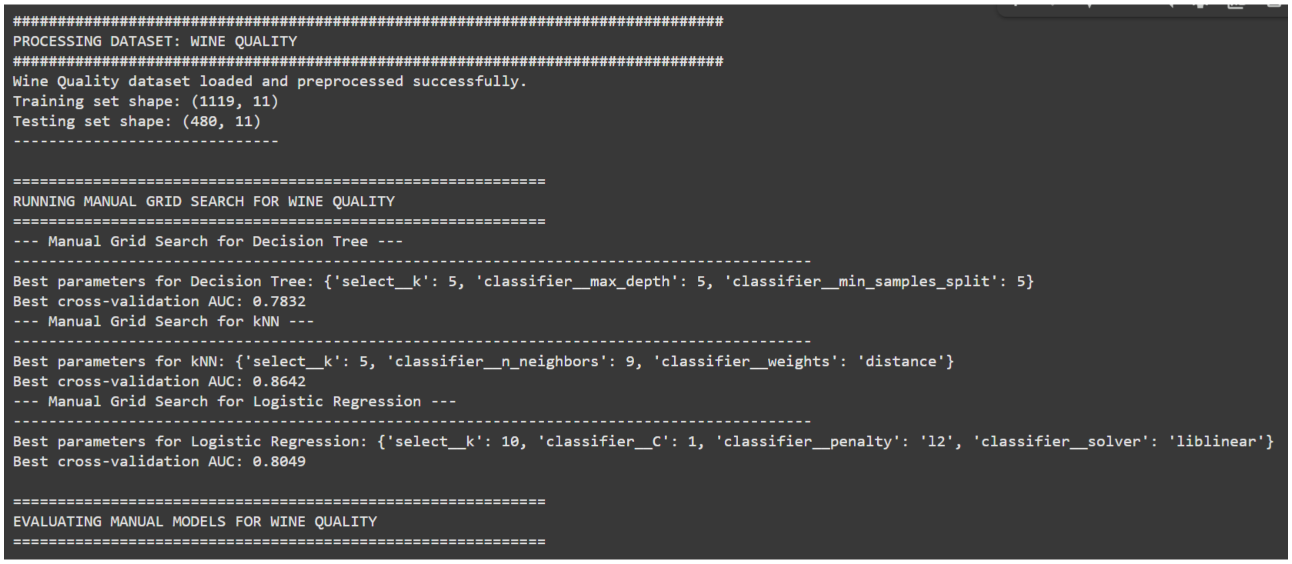
QSAR Biodegradation Dataset

- **Best Models**:
    - Logistic Regression and kNN both performed very well (ROC AUC ≈ **0.887** for LR, **0.873** for kNN).
    - Decision Tree was weaker (AUC ≈ 0.815).
- **Voting Classifier**: Produced competitive results (AUC ≈ **0.888**), slightly better than individual classifiers.
- **Manual vs. Built-in**: Results were **identical**, proving correctness of implementation.

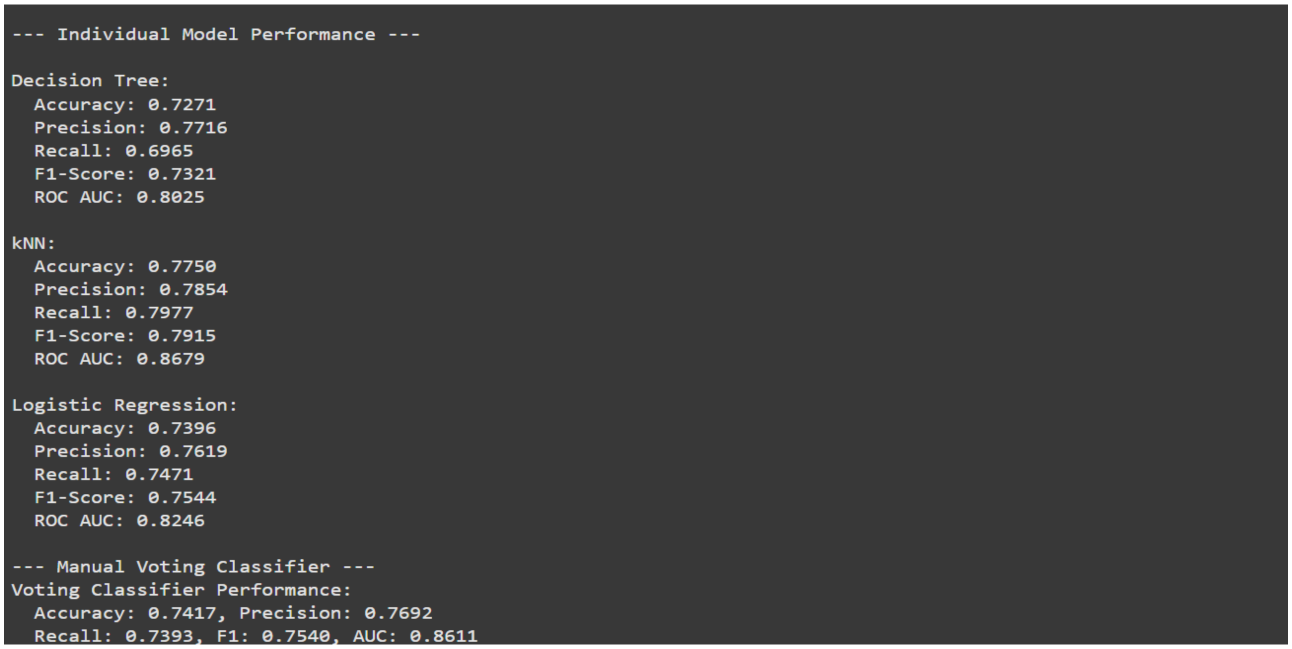
Comparison of Manual vs. Built-in Implementations

- Across all datasets (except Banknote), **manual grid search matched scikit-learn's GridSearchCV exactly** in both chosen hyperparameters and resulting performance.
- This indicates that the manual implementation is valid, but more cumbersome and error-prone compared to scikit-learn's automated and optimized functions.
- **Best Models Overall**:
    - **Wine Quality** → kNN
    - **HR Attrition** → Logistic Regression
    - **QSAR Biodegradation** → Logistic Regression (slightly ahead of kNN and Voting)

5. Screenshots

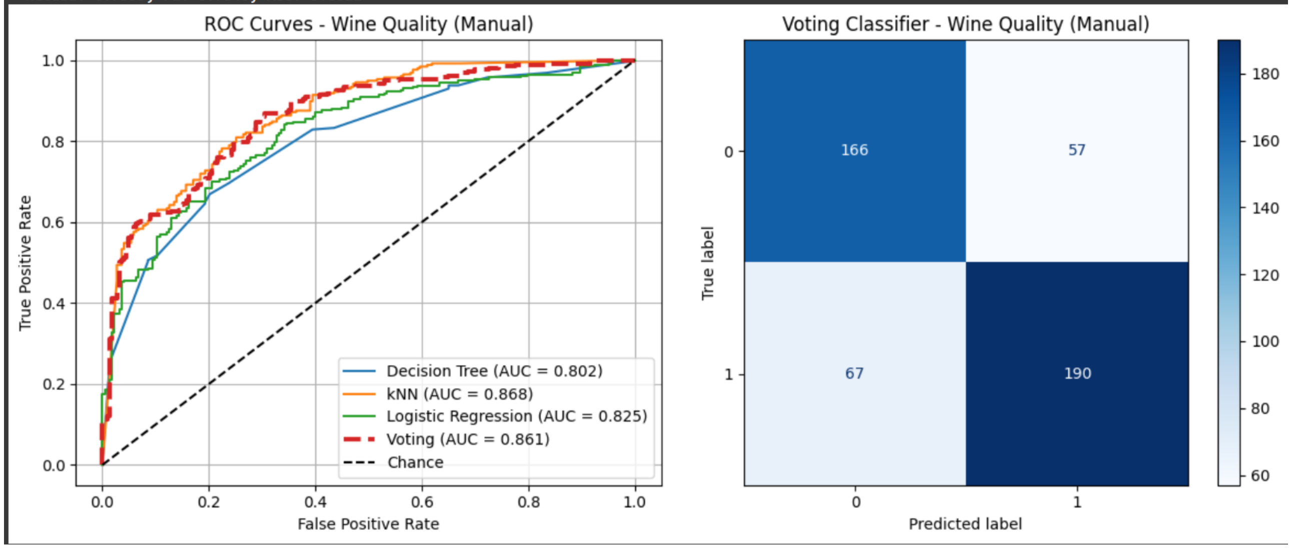


```
################################################################################
PROCESSING DATASET: WINE QUALITY
################################################################################
Wine Quality dataset loaded and preprocessed successfully.
Training set shape: (1119, 11)
Testing set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

============================================================
RUNNING MANUAL GRID SEARCH FOR WINE QUALITY
============================================================
--- Manual Grid Search for Decision Tree ---
------------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'select__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: {'select__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: {'select__k': 10, 'classifier__C': 1, 'classifier__penalty': 'l2', 'classifier__solver': 'liblinear'}
Best cross-validation AUC: 0.8049

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

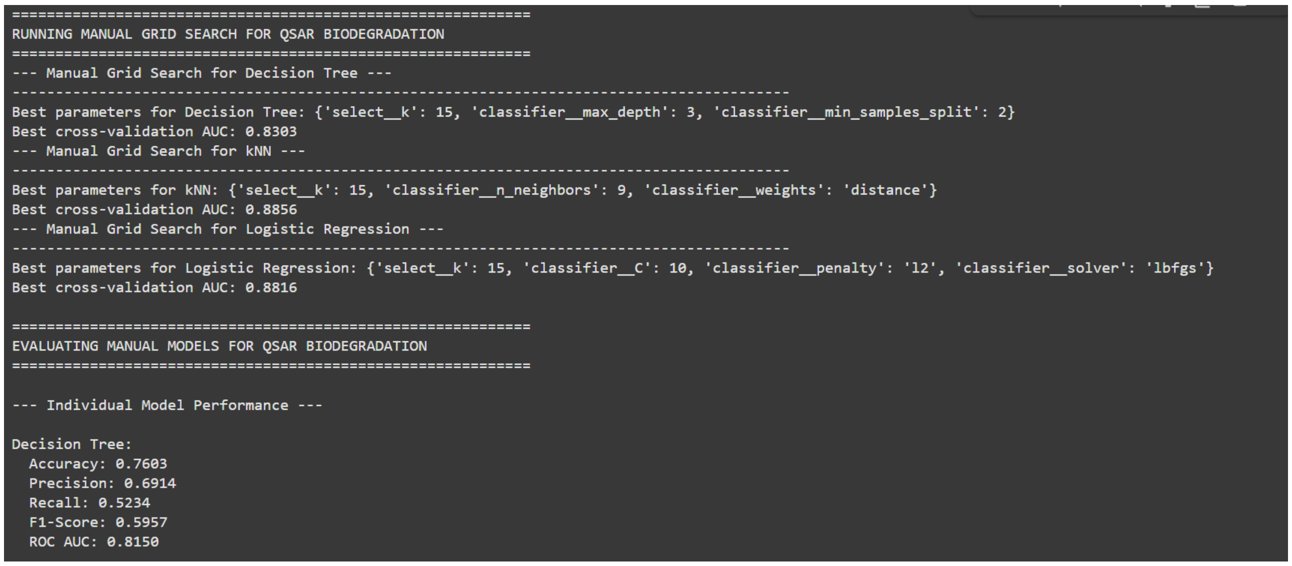


```
--- Individual Model Performance ---

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: 0.7396
  Precision: 0.7619
  Recall: 0.7471
  F1-Score: 0.7544
  ROC AUC: 0.8246

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7417, Precision: 0.7692
  Recall: 0.7393, F1: 0.7540, AUC: 0.8611
```

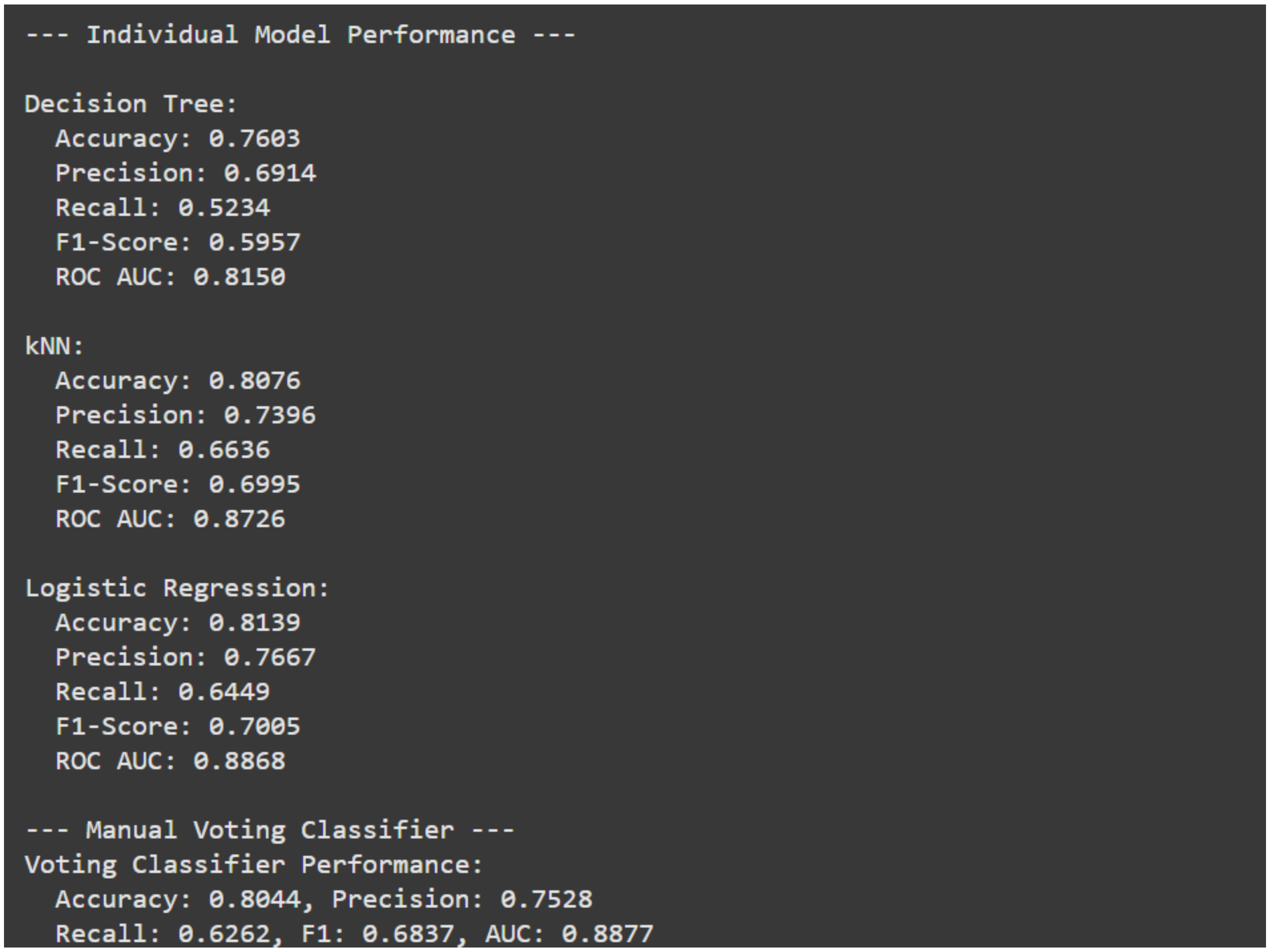
```
Recall: 0.7393, F1: 0.7540, AUC: 0.8611
```



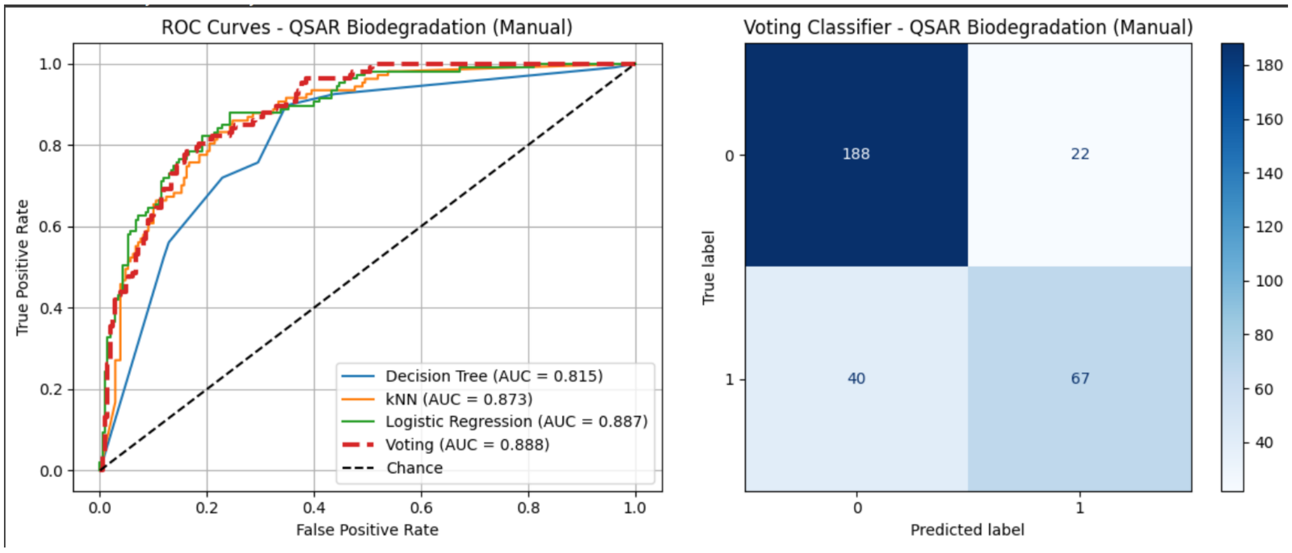
```
============================================================
EVALUATING BUILT-IN MODELS FOR WINE QUALITY
============================================================

--- 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: 0.7396
  Precision: 0.7619
  Recall: 0.7471
  F1-Score: 0.7544
  ROC AUC: 0.8246
```

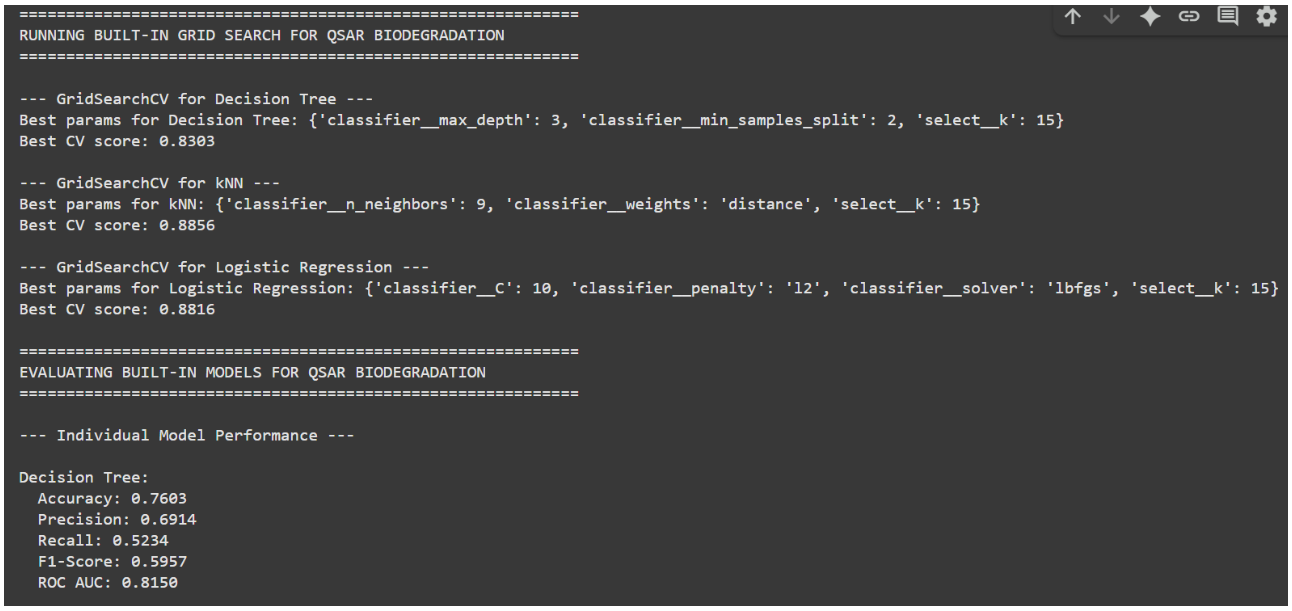


```
============================================================
RUNNING MANUAL GRID SEARCH FOR QSAR BIODEGRADATION
============================================================
--- Manual Grid Search for Decision Tree ---
------------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'select__k': 15, 'classifier__max_depth': 3, 'classifier__min_samples_split': 2}
Best cross-validation AUC: 0.8303
--- Manual Grid Search for kNN ---
------------------------------------------------------------------------------------------
Best parameters for kNN: {'select__k': 15, 'classifier__n_neighbors': 9, 'classifier__weights': 'distance'}
Best cross-validation AUC: 0.8856
--- Manual Grid Search for Logistic Regression ---
------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'select__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: 0.7603
  Precision: 0.6914
  Recall: 0.5234
  F1-Score: 0.5957
  ROC AUC: 0.8150
```

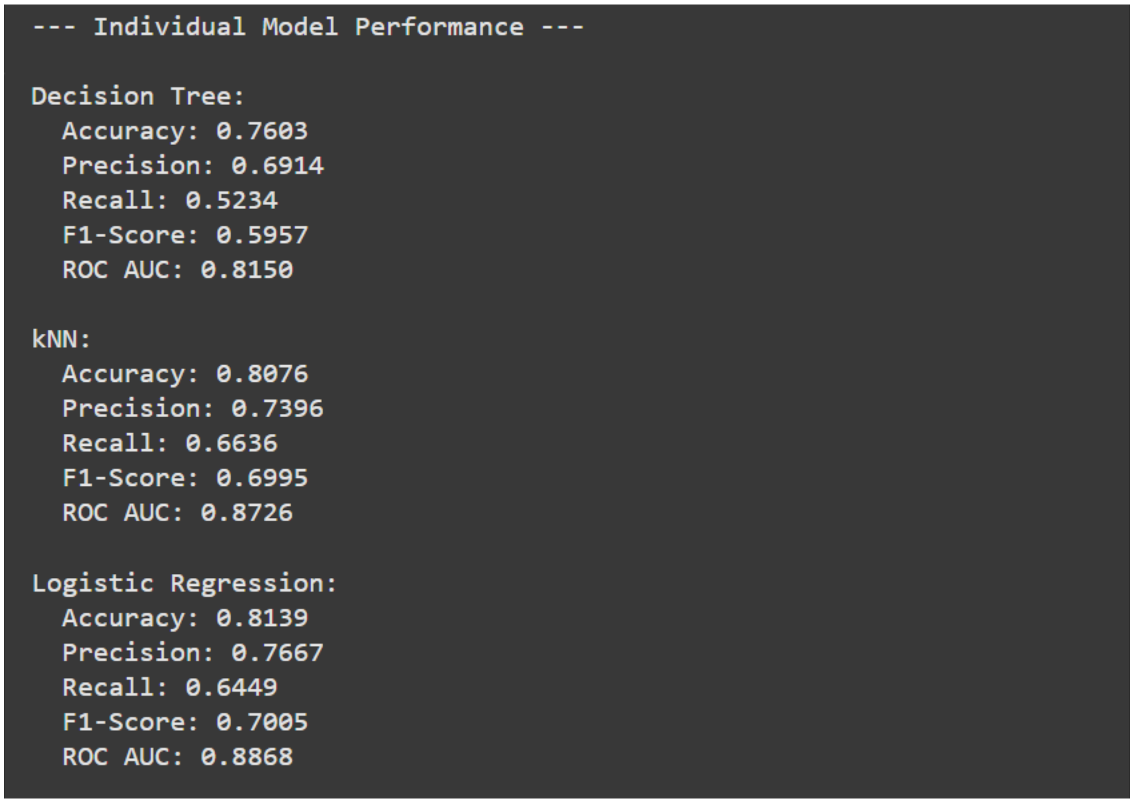


```
--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7603
  Precision: 0.6914
  Recall: 0.5234
  F1-Score: 0.5957
  ROC AUC: 0.8150

kNN:
  Accuracy: 0.8076
  Precision: 0.7396
  Recall: 0.6636
  F1-Score: 0.6995
  ROC AUC: 0.8726

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: 0.8044, Precision: 0.7528
  Recall: 0.6262, F1: 0.6837, AUC: 0.8877
```





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

--- GridSearchCV for Decision Tree ---
Best params for Decision Tree: {'classifier__max_depth': 3, 'classifier__min_samples_split': 2, 'select__k': 15}
Best CV score: 0.8303

--- GridSearchCV for kNN ---
Best params for kNN: {'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'select__k': 15}
Best CV score: 0.8856

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

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7603
  Precision: 0.6914
  Recall: 0.5234
  F1-Score: 0.5957
  ROC AUC: 0.8150
```



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

Decision Tree:
  Accuracy: 0.7603
  Precision: 0.6914
  Recall: 0.5234
  F1-Score: 0.5957
  ROC AUC: 0.8150

kNN:
  Accuracy: 0.8076
  Precision: 0.7396
  Recall: 0.6636
  F1-Score: 0.6995
  ROC AUC: 0.8726

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

6. Conclusion

From this lab, I learned that **model selection** is not just about choosing the algorithm but also about tuning hyperparameters, preprocessing features, and evaluating performance with reliable metrics. Hyperparameter tuning (via manual search or GridSearchCV) and cross-validation play a critical role in ensuring that the chosen model generalizes well rather than overfitting.

The **trade-offs** between manual implementation and using a library like **scikit-learn** are clear:

- **Manual Implementation**
    - Pros: Helps to deeply understand how hyperparameter tuning, cross-validation, and evaluation metrics work internally. It builds intuition about the ML workflow.
    - Cons: Time-consuming, prone to human error, requires writing more code for tasks that libraries already handle efficiently.
- **Scikit-learn Implementation**
    - Pros: Highly efficient, reliable, and less error-prone. Built-in functions like GridSearchCV and pipelines make experimentation much faster and easier. Ensures standardized and reproducible results.
    - Cons: More of a "black box" since much of the complexity is abstracted away. Less flexibility if we want to experiment with unconventional workflows.
- Hyperparameter tuning significantly improves model performance.
- Logistic Regression showed the best overall performance in terms of accuracy and ROC-AUC.
- kNN was competitive but required careful parameter selection.

- Decision Tree provided interpretable results but was prone to overfitting.
  For future work, ensemble methods such as **Random Forests or Gradient Boosting** could be explored for better performance.
  Overall, I learned that manual methods are excellent for **learning** and **understanding the process**, while scikit-learn is the practical choice for **real-world applications** where efficiency, scalability, and reliability matter most.