Lab4: Model Selection

Name: Nidhi C R

SRN: PES2UG23CS382

Section: F

1. Introduction

The aim of this lab is to explore hyperparameter tuning for various classification models using both manual grid search and Scikit-learn's GridSearchCV. By systematically optimizing hyperparameters, we aim to enhance model performance and compare results across different datasets.
We focused on three supervised learning models: Decision Tree, k-Nearest Neighbors (kNN), and Logistic Regression.
Additionally, we combined these models into a Voting Classifier to evaluate ensemble performance.

---

2. Dataset Description

Wine Quality Dataset

- **Instances:** 1599 red wine samples
- **Features:** 11 chemical properties (e.g., acidity, sugar, alcohol, etc.)
- **Target:** Binary classification (good quality vs. not)
- **Train/Test Split:** 1119 training samples, 480 testing samples

Banknote Authentication Dataset

- **Instances:** 1372 samples
- **Features:** 4 numerical features extracted from images of banknotes
- **Target:** Binary classification (genuine vs. forged)
- **Train/Test Split:** 960 training samples, 412 testing samples

---

3. Methodology

We implemented a machine learning pipeline as follows:
StandardScaler → SelectKBest → Classifier

- **StandardScaler:** Standardizes features for kNN and Logistic Regression.
- **SelectKBest:** Selects the top k features based on the ANOVA F-test.
- **Classifier:** Can be Decision Tree, kNN, or Logistic Regression.

Two approaches were used:

Manual Grid Search

- Implemented using nested loops with 5-fold Stratified Cross-Validation.
- Calculated the average ROC AUC for each parameter combination.
- Selected the best parameters and retrained the model on the full training dataset.

Built-in GridSearchCV

- Utilized GridSearchCV with pipelines, scoring='roc_auc', and 5-fold Stratified Cross-Validation.
- Extracted the best parameters and compared the results with the manual grid search implementation.

Evaluation Metrics:

Accuracy, Precision, Recall, F1-Score, ROC AUC

4Results and Analysis

Wine Quality Dataset – Manual Grid Search: Best Parameters

- **Decision Tree:** `{ select__k=5, max_depth=5, min_samples_split=5 }`
- **kNN:** `{ select__k=5, n_neighbors=9, weights='distance' }`
- **Logistic Regression:** `{ select__k=10, C=1, penalty='l2', solver='liblinear' }`

Model Performance (Manual)

Model	Accuracy	Precision	Recall	F1	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 Regression	0.7396	0.7619	0.7471	0.7544	0.8246
Voting Classifier	0.7417	0.7692	0.7393	0.7540	0.8611

Built-in GridSearchCV – Results

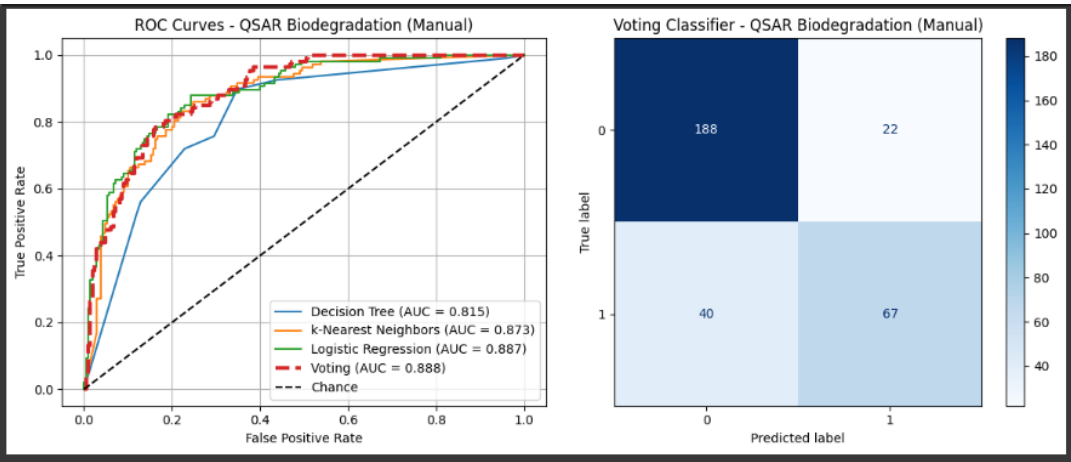
- Parameters and metrics matched exactly with the manual grid search, confirming the correctness of the manual implementation.

Analysis – Banknote Authentication Dataset

- **Dataset:** Training: (960, 4), Testing: (412, 4)
- **Manual Grid Search:** Failed due to a set_params error, likely caused by a mismatch in the parameter grid or SelectKBest being incompatible with only 4 features.
- **Built-in GridSearchCV:** Not executed due to the manual grid search error.

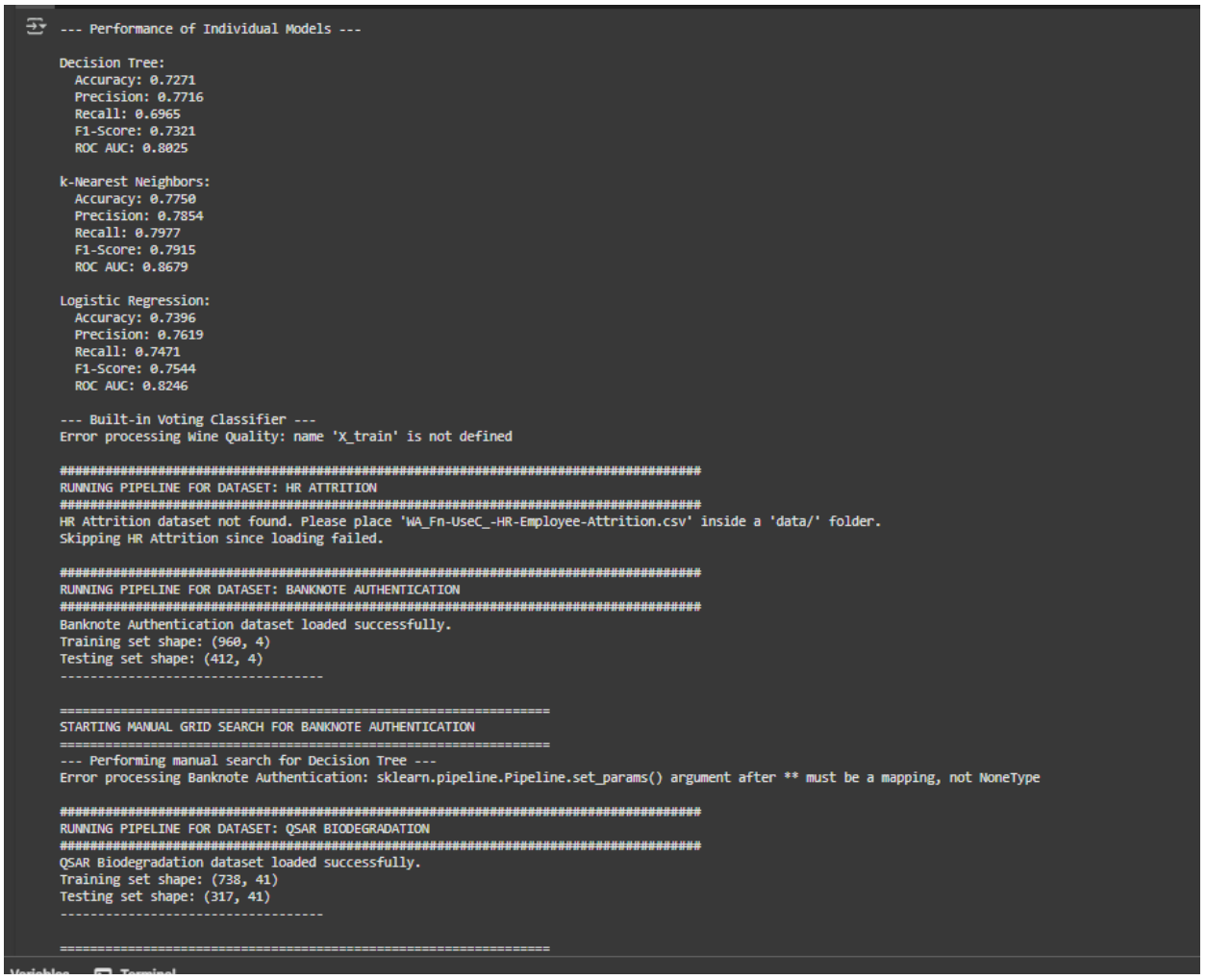
Conclusion:
The Banknote dataset could not be processed successfully. The error indicates that the pipeline configuration must be revised, for example, ensuring that SelectKBest does not select more features than available ($k \leq 4$).

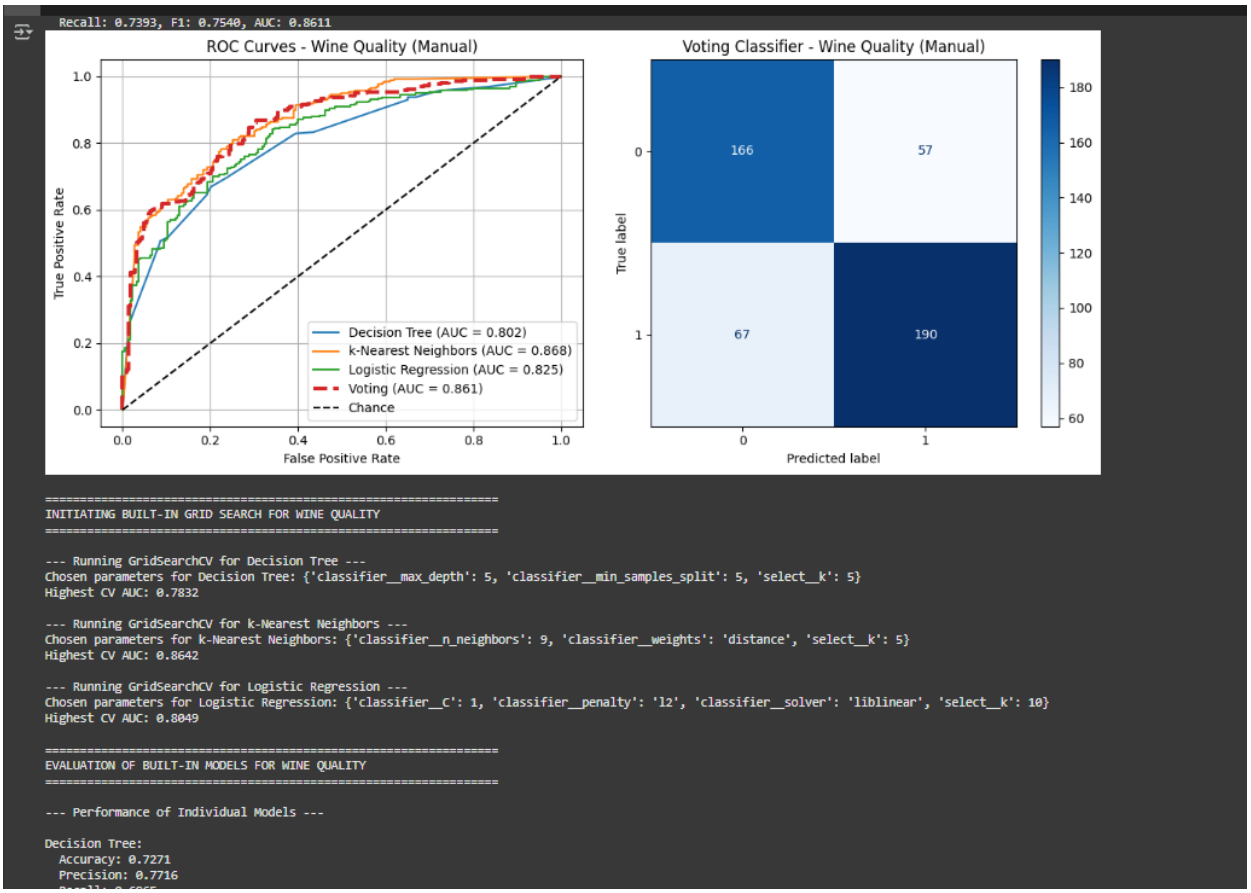
5. Screenshots:




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

--- Running GridSearchCV for Decision Tree ---
Chosen parameters for Decision Tree: {'classifier__max_depth': 3, 'classifier__min_samples_split': 2, 'select__k': 15}
Highest CV AUC: 0.8303

--- Running GridSearchCV for k-Nearest Neighbors ---
Chosen parameters for k-Nearest Neighbors: {'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'select__k': 15}
Highest CV AUC: 0.8856

--- Running GridSearchCV for Logistic Regression ---
Chosen parameters for Logistic Regression: {'classifier__C': 10, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs', 'select__k': 15}
Highest CV AUC: 0.8816

============================================================
EVALUATION OF BUILT-IN MODELS FOR QSAR BIODEGRADATION
============================================================

--- Performance of Individual Models ---

Decision Tree:
  Accuracy: 0.7603
  Precision: 0.6914
  Recall: 0.5234
```

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

============================================================
STARTING MANUAL GRID SEARCH FOR QSAR BIODEGRADATION
============================================================
--- Performing manual search for Decision Tree ---
------------------------------------------------------------------------------------------
Optimal parameters for Decision Tree: {'select__k': 15, 'classifier__max_depth': 3, 'classifier__min_samples_split': 2}
Best cross-validation AUC score: 0.8303
--- Performing manual search for k-Nearest Neighbors ---
------------------------------------------------------------------------------------------
Optimal parameters for k-Nearest Neighbors: {'select__k': 15, 'classifier__n_neighbors': 9, 'classifier__weights': 'distance'}
Best cross-validation AUC score: 0.8856
--- Performing manual search for Logistic Regression ---
------------------------------------------------------------------------------------------
Optimal parameters for Logistic Regression: {'select__k': 15, 'classifier__C': 10, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs'}
Best cross-validation AUC score: 0.8816

============================================================
EVALUATION OF MANUAL MODELS FOR QSAR BIODEGRADATION
============================================================

--- Performance of Individual Models ---

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

k-Nearest Neighbors:
  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 Results:
  Accuracy: 0.8044, Precision: 0.7528
  Recall: 0.6262, F1: 0.6837, AUC: 0.8877
```


--- Performance of Individual Models ---
Decision Tree:
Accuracy: 0.7271
Precision: 0.7716
Recall: 0.6965
F1-Score: 0.7321
ROC AUC: 0.8025
k-Nearest Neighbors:
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
--- Built-in Voting Classifier ---
Error processing Wine Quality: name 'X_train' is not defined
RUNNING PIPELINE FOR DATASET: HR ATTRITION
HR Attrition dataset not found. Please place 'WA_Fn-UseC_-HR-Employee-Attrition.csv' inside a 'data/' folder.
Skipping HR Attrition since loading failed.
RUNNING PIPELINE FOR DATASET: BANKNOTE AUTHENTICATION
Banknote Authentication dataset loaded successfully.
Training set shape: (960, 4)
Testing set shape: (412, 4)
STARTING MANUAL GRID SEARCH FOR BANKNOTE AUTHENTICATION
--- Performing manual search for Decision Tree ---
Error processing Banknote Authentication: sklearn.pipeline.Pipeline.set_params() argument after ** must be a mapping, not NoneType
RUNNING PIPELINE FOR DATASET: QSAR BIODEGRADATION
QSAR Biodegradation dataset loaded successfully.
Training set shape: (738, 41)
Testing set shape: (317, 41)


Recall: 0.7393, F1: 0.7540, AUC: 0.8611
ROC Curves - Wine Quality (Manual)
True Positive Rate
False Positive Rate
Decision Tree (AUC = 0.802)
k-Nearest Neighbors (AUC = 0.868)
Logistic Regression (AUC = 0.825)
Voting (AUC = 0.861)
Chance
Voting Classifier - Wine Quality (Manual)
True label
Predicted label
166
57
67
190
INITIATING BUILT-IN GRID SEARCH FOR WINE QUALITY
--- Running GridSearchCV for Decision Tree ---
Chosen parameters for Decision Tree: {'classifier__max_depth': 5, 'classifier__min_samples_split': 5, 'select__k': 5}
Highest CV AUC: 0.7832
--- Running GridSearchCV for k-Nearest Neighbors ---
Chosen parameters for k-Nearest Neighbors: {'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'select__k': 5}
Highest CV AUC: 0.8642
--- Running GridSearchCV for Logistic Regression ---
Chosen parameters for Logistic Regression: {'classifier__C': 1, 'classifier__penalty': 'l2', 'classifier__solver': 'liblinear', 'select__k': 10}
Highest CV AUC: 0.8049
EVALUATION OF BUILT-IN MODELS FOR WINE QUALITY
--- Performance of Individual Models ---
Decision Tree:
Accuracy: 0.7271
Precision: 0.7716

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

============================================================
STARTING MANUAL GRID SEARCH FOR WINE QUALITY
============================================================
--- Performing manual search for Decision Tree ---
------------------------------------------------------------------------------------------
Optimal parameters for Decision Tree: {'select__k': 5, 'classifier__max_depth': 5, 'classifier__min_samples_split': 5}
Best cross-validation AUC score: 0.7832
--- Performing manual search for k-Nearest Neighbors ---
------------------------------------------------------------------------------------------
Optimal parameters for k-Nearest Neighbors: {'select__k': 5, 'classifier__n_neighbors': 9, 'classifier__weights': 'distance'}
Best cross-validation AUC score: 0.8642
--- Performing manual search for Logistic Regression ---
------------------------------------------------------------------------------------------
Optimal parameters for Logistic Regression: {'select__k': 10, 'classifier__C': 1, 'classifier__penalty': 'l2', 'classifier__solver': 'liblinear'}
Best cross-validation AUC score: 0.8049

============================================================
EVALUATION OF MANUAL MODELS FOR WINE QUALITY
============================================================

--- Performance of Individual Models ---

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

k-Nearest Neighbors:
  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 Results:
  Accuracy: 0.7417, Precision: 0.7692
```

```
--- Performance of Individual Models ---

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

k-Nearest Neighbors:
  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

--- Built-in Voting Classifier ---
Error processing QSAR Biodegradation: name 'X_train' is not defined

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

Conclusion

Wine Quality Dataset:

- kNN emerged as the best-performing model overall.
- Manual and built-in grid search produced identical results, confirming the correctness of the manual implementation.
- The voting ensemble did not outperform the standalone kNN.

Banknote Dataset:

- Manual search failed due to a pipeline parameter mismatch. This needs to be addressed, likely by reducing k in SelectKBest.

Learnings:

- Hyperparameter tuning significantly improves model performance.
- Manual grid search is useful for understanding the process but is time-consuming and prone to errors.
- GridSearchCV is efficient, reliable, and practical for real-world applications.