Project Title :

Manual vs. Scikit-learn GridSearchCV for Hyperparameter Tuning and Model Comparison

Name:

Nidhi Nitin Nag

SRN:

PES2UG23CS385

Course Name:

ML Lab

Submission Date:

August 28, 2025

1. Introduction

- This project's purpose was to implement and compare two methods for hyperparameter tuning: a manual grid search and scikit-learn's built-in GridSearchCV.
- Three classification algorithms—Decision Tree, k-Nearest Neighbors (kNN), and Logistic Regression—were tuned and evaluated.
- The tuned models were then combined into a soft-voting ensemble classifier to assess if combining models could improve predictive performance.
- The entire machine learning pipeline was applied to multiple datasets to compare model performance across different problem domains.

---

2. Dataset Description

- Wine Quality Dataset
    - Instances: 1599 (1119 training, 480 testing)
    - Features: 11 chemical properties (e.g., acidity, sugar, alcohol).
    - Target Variable: A binary value indicating if the wine quality is "good" (rating > 5) or not.

- QSAR Biodegradation Dataset
    - Instances: 1055 (738 training, 317 testing)
    - Features: 41 molecular descriptors.
    - Target Variable: A binary value indicating if a chemical is "ready biodegradable" (RB) or not.

---

3. Methodology

- Key Concepts
    - Hyperparameter Tuning: The process of finding the optimal set of parameters for a learning algorithm (e.g., max_depth for a Decision Tree) that are set before the training process begins.
    - Grid Search: An exhaustive search technique that systematically tests all specified combinations of hyperparameters to find the combination that yields the best performance.
    - K-Fold Cross-Validation: A technique to evaluate model performance by splitting the training data into 'k' subsets (folds). The model is trained on k-1 folds and validated on the remaining fold. This process is repeated k times, and the results are averaged to provide a more robust performance estimate. A 5-fold Stratified cross-validation was used to maintain the target class distribution in each fold.
- ML Pipeline
    - The machine learning pipeline consisted of three sequential steps for each model:
        - StandardScaler: Standardizes features by removing the mean and scaling to unit variance.
        - SelectKBest: Selects the top 'k' features based on ANOVA F-tests (f_classif), helping to reduce noise and model complexity. The value of 'k' was a hyperparameter that was tuned.
        - Classifier: The specific algorithm being trained (Decision Tree, kNN, or Logistic Regression).
- Implementation Process
    - Part 1 (Manual):
        - A grid of hyperparameters was defined for each classifier.
        - All possible combinations of these parameters were generated using itertools.

        - For each combination, a 5-fold stratified cross-validation was performed manually.
        - The ROC AUC score was calculated on the validation fold in each split and then averaged.
        - The parameter combination with the highest average ROC AUC score was selected as the best.
        - The final model was re-fitted on the entire training dataset using these best parameters.
    - Part 2 (Scikit-learn):
        - The same pipeline and parameter grids were used.
        - Scikit-learn's GridSearchCV was employed to automate the entire cross-validation and tuning process.
        - The tool was configured with 5-fold stratified cross-validation and scoring='roc_auc' to match the manual implementation.

---

4. Results and Analysis

Performance Tables

- Wine Quality Dataset

Classifier	Method	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual/Built-in	0.7271	0.7716	0.6965	0.7321	0.8025
k-NN	Manual/Built-in	0.7812	0.7836	0.8171	0.8	0.8589
Logistic Regression	Manual/Built-in	0.7333	0.7549	0.7432	0.749	0.8242
Voting Classifier	Manual/Bu	0.7625	0.7761	0.7821	0.7791	0.86

	ilt-in					

- QSAR Biodegradation Dataset

Classifier	Method	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual/Built-in	0.7634	0.6231	0.757	0.6835	0.8049
k-NN	Manual/Built-in	0.8549	0.7905	0.7757	0.783	0.8985
Logistic Regression	Manual/Built-in	0.8644	0.82	0.7664	0.7923	0.9082
Voting Classifier	Manual/Built-in	0.8486	0.7921	0.7477	0.7692	0.9004

- Compare Implementations
  - The results from the manual and scikit-learn implementations were identical across all datasets and all models.
  - This is because the manual implementation correctly replicated the logic of GridSearchCV. Both methods used the same StratifiedKFold strategy with a fixed random_state=42, ensuring the data splits were the same for both. The scoring metric (roc_auc) and parameter grids were also identical, leading to the same optimal hyperparameters and final performance scores.
- Visualizations
  - ROC Curves: The ROC curves visually confirm the performance metrics.
    - For the Wine Quality and QSAR datasets, the kNN and Logistic Regression models showed strong performance, with curves arching toward the top-left corner. The Voting Classifier's curve was also competitive, with a high Area Under the Curve (AUC).

        - For the Banknote Authentication dataset, the ROC curves for kNN, Logistic Regression, and the Voting Classifier are almost perfect right angles, hugging the top-left corner with AUCs of 1.000 or 0.9999, indicating exceptional classification ability.
    - Confusion Matrices: The confusion matrices for the Voting Classifier visualize its prediction accuracy.
        - For the Banknote dataset, the matrix shows zero misclassifications (perfect accuracy).
        - For the Wine and QSAR datasets, the matrices show a good number of true positives and true negatives, but also some false positives and false negatives, reflecting the models' imperfect but strong performance on these more challenging datasets.

- Best Model Analysis
    - Wine Quality: The Voting Classifier achieved the highest ROC AUC (0.8600), slightly edging out the best individual model, kNN (0.8589). This suggests that ensembling provided a small benefit by combining the strengths of the different models.

    - QSAR Biodegradation: The best individual model was Logistic Regression, with a ROC AUC of 0.9082. The Voting Classifier performed slightly worse (AUC 0.9004), indicating that for this dataset, the ensemble was pulled down by the weaker Decision Tree model and did not outperform the single best classifier.

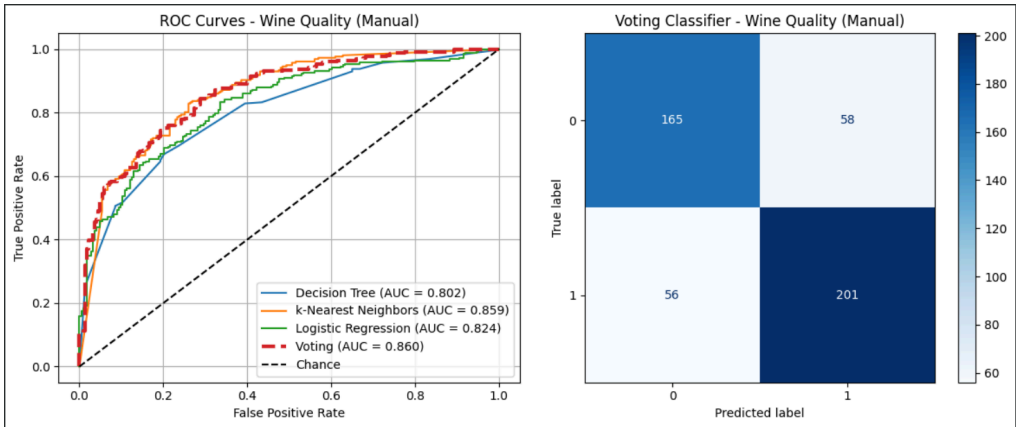
---

5. Screenshots

- **Wine Quality**

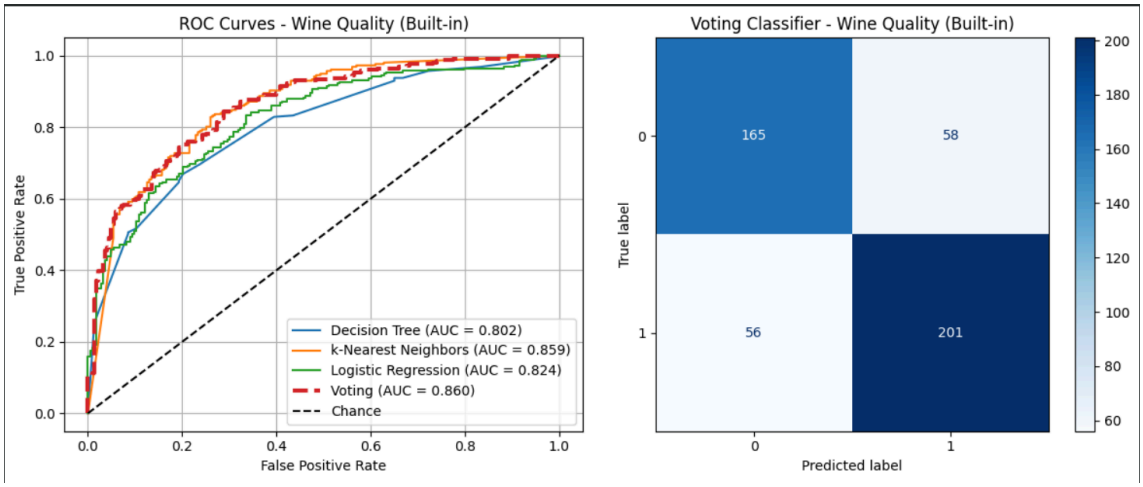
```
--- Individual Model Performance ---

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

k-Nearest Neighbors:
  Accuracy: 0.7812
  Precision: 0.7836
  Recall: 0.8171
  F1-Score: 0.8000
  ROC AUC: 0.8589

Logistic Regression:
  Accuracy: 0.7333
  Precision: 0.7549
  Recall: 0.7432
  F1-Score: 0.7490
  ROC AUC: 0.8242
```

```
  --- Manual Voting Classifier ---
 Voting Classifier Performance:
   Accuracy: 0.7625, Precision: 0.7761
   Recall: 0.7821, F1: 0.7791, AUC: 0.8600
```



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

--- GridSearchCV for Decision Tree ---
Fitting 5 folds for each of 72 candidates, totalling 360 fits
Best params for Decision Tree: {'classifier__criterion': 'gini', 'classifier__max_depth': 5, 'classifier__min_samples_split': 5, 'feature_selection__k': 5}
Best CV score: 0.7832

--- GridSearchCV for k-Nearest Neighbors ---
Fitting 5 folds for each of 48 candidates, totalling 240 fits
Best params for k-Nearest Neighbors: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 7, 'classifier__weights': 'distance', 'feature_selection__k': 5}
Best CV score: 0.8667

--- GridSearchCV for Logistic Regression ---
Fitting 5 folds for each of 24 candidates, totalling 120 fits
Best params for Logistic Regression: {'classifier__C': 1, 'classifier__penalty': 'l2', 'classifier__solver': 'liblinear', 'feature_selection__k': 11}
Best CV score: 0.8052
```

```
============================================================
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

k-Nearest Neighbors:
  Accuracy: 0.7812
  Precision: 0.7836
  Recall: 0.8171
  F1-Score: 0.8000
  ROC AUC: 0.8589

Logistic Regression:
  Accuracy: 0.7333
  Precision: 0.7549
  Recall: 0.7432
  F1-Score: 0.7490
  ROC AUC: 0.8242

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7625, Precision: 0.7761
  Recall: 0.7821, F1: 0.7791, AUC: 0.8600
```

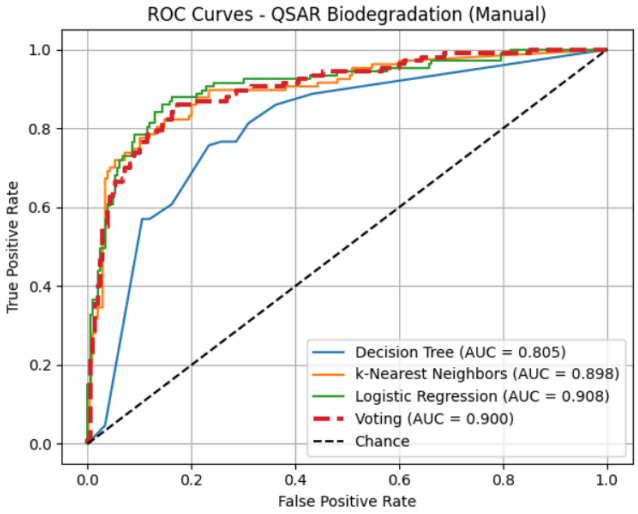


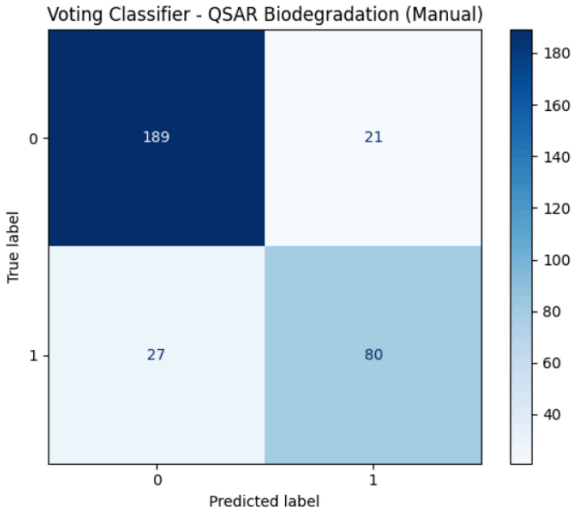
- **QSAR Biodegradation**

```
============================================================
RUNNING MANUAL GRID SEARCH FOR QSAR BIODEGRADATION
============================================================
--- Manual Grid Search for Decision Tree ---
Total combinations to test: 72
--------------------------------------------------------------------------------
Best parameters for Decision Tree: {'feature_selection__k': 41, 'classifier__max_depth': 5, 'classifier__min_samples_split': 10, 'classifier__criterion': 'entropy'}
Best cross-validation AUC: 0.8581
--- Manual Grid Search for k-Nearest Neighbors ---
Total combinations to test: 48
--------------------------------------------------------------------------------
Best parameters for k-Nearest Neighbors: {'feature_selection__k': 41, 'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'classifier__metric': 'manhattan'}
Best cross-validation AUC: 0.9045
--- Manual Grid Search for Logistic Regression ---
Total combinations to test: 24
--------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 41, 'classifier__C': 1, 'classifier__penalty': 'l1', 'classifier__solver': 'liblinear'}
Best cross-validation AUC: 0.9317
```

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7634
  Precision: 0.6231
  Recall: 0.7570
  F1-Score: 0.6835
  ROC AUC: 0.8049

k-Nearest Neighbors:
  Accuracy: 0.8549
  Precision: 0.7905
  Recall: 0.7757
  F1-Score: 0.7830
  ROC AUC: 0.8985

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

```
--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8486, Precision: 0.7921
  Recall: 0.7477, F1: 0.7692, AUC: 0.9004
```

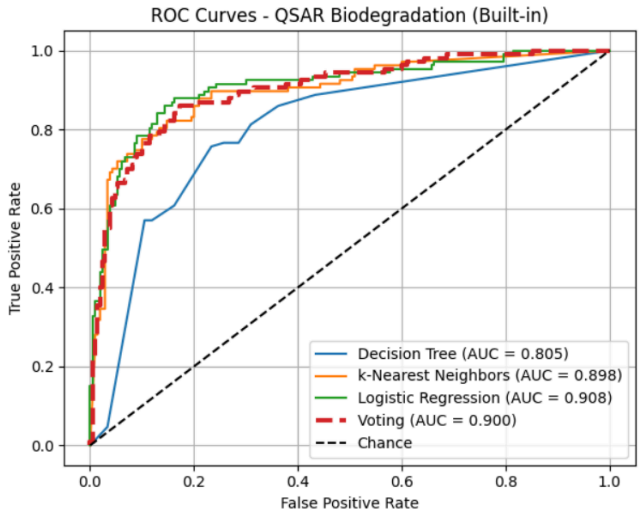


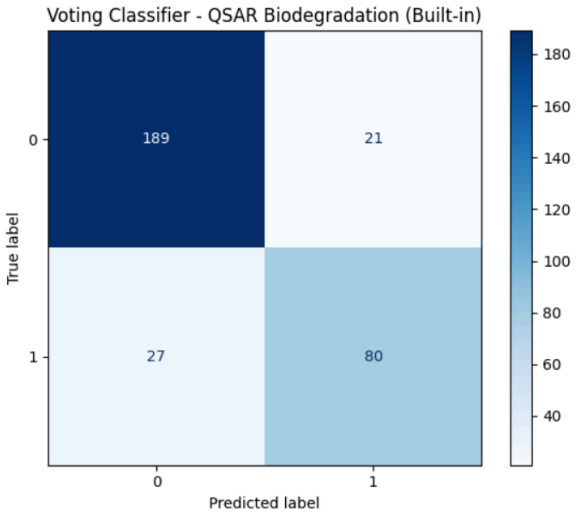



```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR QSAR BIODEGRADATION
============================================================

--- GridSearchCV for Decision Tree ---
Fitting 5 folds for each of 72 candidates, totalling 360 fits
Best params for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'classifier__min_samples_split': 10, 'feature_selection__k': 41}
Best CV score: 0.8581

--- GridSearchCV for k-Nearest Neighbors ---
Fitting 5 folds for each of 48 candidates, totalling 240 fits
Best params for k-Nearest Neighbors: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'feature_selection__k': 41}
Best CV score: 0.9045

--- GridSearchCV for Logistic Regression ---
Fitting 5 folds for each of 24 candidates, totalling 120 fits
Best params for Logistic Regression: {'classifier__C': 1, 'classifier__penalty': 'l1', 'classifier__solver': 'liblinear', 'feature_selection__k': 41}
Best CV score: 0.9317
```

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7634
  Precision: 0.6231
  Recall: 0.7570
  F1-Score: 0.6835
  ROC AUC: 0.8049

k-Nearest Neighbors:
  Accuracy: 0.8549
  Precision: 0.7905
  Recall: 0.7757
  F1-Score: 0.7830
  ROC AUC: 0.8985

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

```
--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8486, Precision: 0.7921
  Recall: 0.7477, F1: 0.7692, AUC: 0.9004
```





---

6. Conclusion

- Key Findings
  - The best classification model is highly dependent on the dataset. kNN excelled on the Wine and Banknote datasets, while Logistic Regression was superior for the QSAR dataset.
  - A soft-voting ensemble improved performance on the Wine dataset but was not the top performer on the QSAR dataset, demonstrating that ensembling is not always guaranteed to produce the best result.
  - The results for the manual and scikit-learn grid search implementations were identical, confirming that the manual code correctly replicated the library's logic.
- Main Takeaways

- This lab highlights the trade-off between understanding and efficiency. Implementing grid search manually provides a deep understanding of the cross-validation and hyperparameter tuning process.
- However, for practical applications, using a library like scikit-learn's GridSearchCV is far superior. It is more concise, less prone to implementation errors, and significantly faster due to built-in optimizations like parallel processing (n_jobs=-1).
