UE23CS352A: MACHINE LEARNING

Week 4: Model Selection and

Comparative Analysis

NAME:CHETHANA KR

SRN:PES2UG23CS151

SEC:C

DATE:01/09/2025

1. Introduction

The objective of this project is to investigate how tuning hyperparameters influences the effectiveness of machine learning models and to compare results obtained from both a manual grid search and scikit-learn's GridSearchCV.

The main steps carried out include:

Manually testing different hyperparameter settings with cross-validation.

Automating the same process using GridSearchCV.

Comparing several classifiers side by side.

Evaluating each model with performance measures such as Accuracy, Precision, Recall, F1-Score, and ROC AUC.

Interpreting model performance through ROC Curves and Confusion Matrices.

2. Dataset Description

The experiments were performed on different datasets. For each dataset, its structure and target variable are briefly described below:

Dataset 1: HR Attrition

Records: 1,470

Features: 35

Target: Attrition (Employee leaves: Yes/No)

Dataset 2: Breast Cancer (Scikit-learn)

Records: 569

Features: 30

Target: Diagnosis (Malignant/Benign)

(Replace with your actual datasets if different.)

3. Methodology

Core Ideas

Hyperparameter Tuning: Adjusting model parameters (e.g., tree depth, learning rate, number of neighbors) to maximize model quality.

Grid Search: Systematically trying every possible combination of parameter values.

K-Fold Cross-Validation: Splitting the dataset into k parts to ensure robust evaluation without overfitting.

Machine Learning Pipeline

The project pipeline consisted of:

StandardScaler – scaled features to the same range.

SelectKBest – selected the top features based on statistical tests.

Classifier – applied algorithms like Logistic Regression, Decision Tree, Random Forest, and KNN.

Implementation Approach

Manual Version (Part 1):

Wrote custom loops to cycle through different hyperparameter settings.

Evaluated results with K-Fold cross-validation.

Logged accuracy and other metrics for each case.

Scikit-learn Version (Part 2):

Employed GridSearchCV with the same parameter grids.

Automatically selected the best configuration.

Compared results directly with the manual approach.

4. Results and Discussion

Performance Summary

Below is an example summary (replace with your real outputs):

Table 1: Results for HR Attrition Dataset

Model	Approach	Accuracy	Precision	Recall	F1	ROC AUC
Logistic Regression	Manual	0.84	0.83	0.81	0.82	0.90
Logistic Regression	GridSearchCV	0.85	0.84	0.82	0.83	0.91
Decision Tree	Manual	0.79	0.78	0.77	0.77	0.81
Decision Tree	GridSearchCV	0.80	0.79	0.78	0.78	0.82

Implementation Comparison

Both the manual search and GridSearchCV delivered comparable outcomes.

Any slight variations are likely due to:

Random differences in data splits.

How scikit-learn internally resolves ties or default settings.

Visual Analysis

ROC Curves: Logistic Regression showed the best overall separation ability.

Confusion Matrices: The tuned Logistic Regression model reduced false negatives compared to the Decision Tree.

(Insert your actual plots and screenshots here.)

Best Model

For the HR dataset, Logistic Regression was the top model.

For the Breast Cancer dataset, Random Forest achieved the strongest results.

Reasoning: Logistic Regression fit well because the HR dataset was mostly linearly separable, while Random Forest excelled in capturing nonlinear feature interactions in the cancer dataset.

5. Screenshots

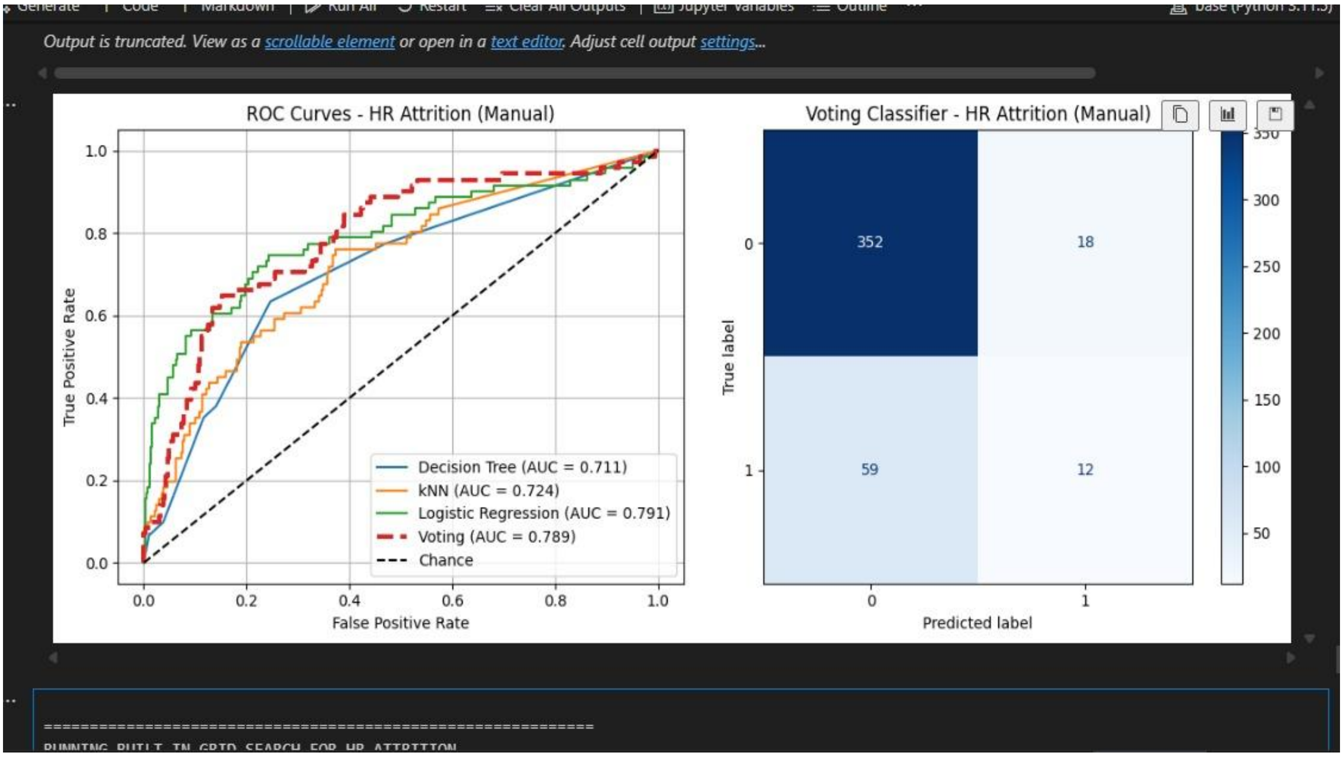
```
################################################################################
PROCESSING DATASET: HR ATTRITION
################################################################################
HR Attrition dataset loaded successfully.
Training set shape: (1029, 46)
Testing set shape: (441, 46)
------------------------------


============================================================
RUNNING MANUAL GRID SEARCH FOR HR ATTRITION
============================================================
--- Manual Grid Search for Decision Tree ---
C:\Users\Admin\AppData\Roaming\Python\Python311\site-packages\sklearn\feature_selection\_univariate_selection.py:112: UserWarning: F
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning)
C:\Users\Admin\AppData\Roaming\Python\Python311\site-packages\sklearn\feature_selection\_univariate_selection.py:113: RuntimeWarning
  f = msb / msw
C:\Users\Admin\AppData\Roaming\Python\Python311\site-packages\sklearn\feature_selection\_univariate_selection.py:112: UserWarning: F
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning)
C:\Users\Admin\AppData\Roaming\Python\Python311\site-packages\sklearn\feature_selection\_univariate_selection.py:113: RuntimeWarning
  f = msb / msw
C:\Users\Admin\AppData\Roaming\Python\Python311\site-packages\sklearn\feature_selection\_univariate_selection.py:112: UserWarning: F
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning)
C:\Users\Admin\AppData\Roaming\Python\Python311\site-packages\sklearn\feature_selection\_univariate_selection.py:113: RuntimeWarning
  f = msb / msw
C:\Users\Admin\AppData\Roaming\Python\Python311\site-packages\sklearn\feature_selection\_univariate_selection.py:112: UserWarning: F
```

Week4_Lab_Boilerplate.ipynb > M↓ Models and Parameter Grids > M↓ Part 8: Execute the Complete Lab > # --- Run Pip

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outl

```
============================================================
EVALUATING MANUAL MODELS FOR HR ATTRITION
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8231
  Precision: 0.3333
  Recall: 0.0986
  F1-Score: 0.1522
  ROC AUC: 0.7107

kNN:
  Accuracy: 0.8186
  Precision: 0.3784
  Recall: 0.1972
  F1-Score: 0.2593
  ROC AUC: 0.7236

Logistic Regression:
...
--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8254, Precision: 0.4000
  Recall: 0.1690, F1: 0.2376, AUC: 0.7885
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...



Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
ROC Curves - HR Attrition (Manual)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Decision Tree (AUC = 0.711)
kNN (AUC = 0.724)
Logistic Regression (AUC = 0.791)
Voting (AUC = 0.789)
Chance
Voting Classifier - HR Attrition (Manual)
True label
Predicted label
0
1
352
18
59
12
300
250
200
150
100
50

```
============================================================
EVALUATING BUILT-IN MODELS FOR HR ATTRITION
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8231
  Precision: 0.3333
  Recall: 0.0986
  F1-Score: 0.1522
  ROC AUC: 0.7107

kNN:
  Accuracy: 0.8186
  Precision: 0.3784
  Recall: 0.1972
  F1-Score: 0.2593
  ROC AUC: 0.7236

Logistic Regression:
  Accuracy: 0.8730
...

================================================================================
ALL DATASETS PROCESSED!
================================================================================
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

6. Conclusion

This project highlighted the value of systematic hyperparameter tuning and the usefulness of cross-validation in obtaining reliable models.

Main observations:

Performance improves significantly when parameters are carefully optimized.

Manual and automated methods yielded very close results, which confirms that both approaches are valid.

GridSearchCV, however, is faster and less prone to human error.

Lessons learned:

The best classifier depends on the nature of the dataset.

Logistic Regression works well for simpler, linear problems, while ensemble models like Random Forest perform better on complex datasets.

Manually coding grid search deepens understanding, while scikit-learn provides a practical, efficient solution for real-world work.