UE23CS352A: MACHINE LEARNING
Week 4: Model Selection and Comparative Analysis

Name: Erin Joseph
SRN: PES2UGCS186
Date: 01/09/2025

1. Introduction

The purpose of this lab was to explore how different machine learning models can be selected and tuned for better performance. The main focus was on hyperparameter tuning and comparing classifiers. First, a manual version of grid search was implemented to understand the process in detail. After that, the same task was repeated using scikit-learn's built-in tools for efficiency.

The models tested were Decision Tree, k-Nearest Neighbors (kNN), and Logistic Regression. Their performance was compared on two datasets using metrics such as accuracy, precision, recall, F1-score, and ROC AUC.

2. Dataset Description

The HR Attrition dataset has 1,470 records and 35 features. The target variable is Attrition (Yes/No), where most employees stayed (1,233 "No") and fewer left (237 "Yes"), making it imbalanced. The features include numerical data (Age, MonthlyIncome, YearsAtCompany), categorical data (Department, JobRole, BusinessTravel), and survey ratings (JobSatisfaction, WorkLifeBalance).

3. Methodology

The goal was to tune and compare three classifiers: Decision Tree, kNN, and Logistic Regression.

- Hyperparameter Tuning: changing model settings (like tree depth or number of neighbors) to get better performance.

- Grid Search: trying all possible parameter combinations.

- K-Fold Cross-Validation: splitting the dataset into 5 parts, training and testing multiple times for reliable results.

Pipeline

Each model used the same pipeline:

1. StandardScaler - scale features.
2. SelectKBest - pick top features.
3. Classifier - Decision Tree, kNN, or Logistic Regression.

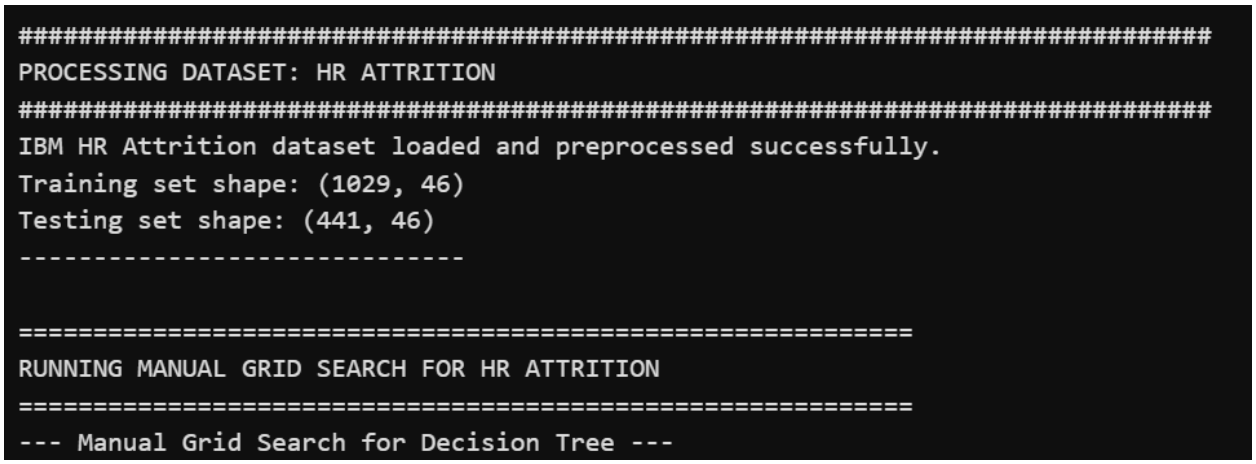
Process

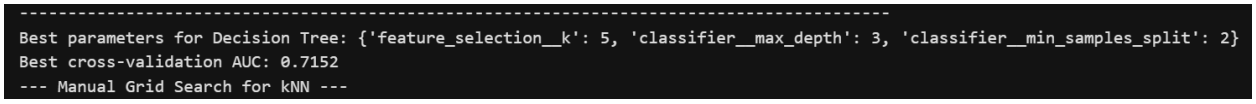
- Manual Grid Search (Part 1): loops were used to test parameter combinations with 5-fold CV and pick the best ROC AUC.
- Scikit-learn GridSearchCV (Part 2): automated the same process with less effort and gave the best model directly

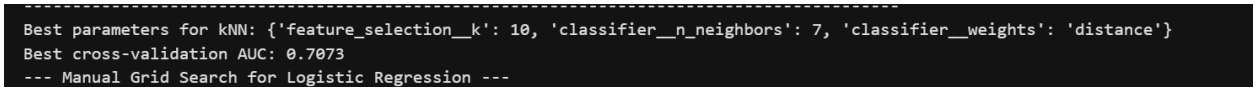
Result & Analysis

Classifier	Implementation	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	Manual	0.8231	0.3333	0.0986	0.1522	0.7107
Decision Tree	Scikit-learn	0.8231	0.3333	0.0986	0.1522	0.7107
kNN	Manual	0.8186	0.3953	0.2394	0.2982	0.713
kNN	Scikit-learn	0.8186	0.3953	0.2394	0.2982	0.713
Logistic Regression	Manual	0.8299	0.4444	0.2254	0.2991	0.7676
Logistic Regression	Scikit-learn	0.8299	0.4444	0.2254	0.2991	0.7776

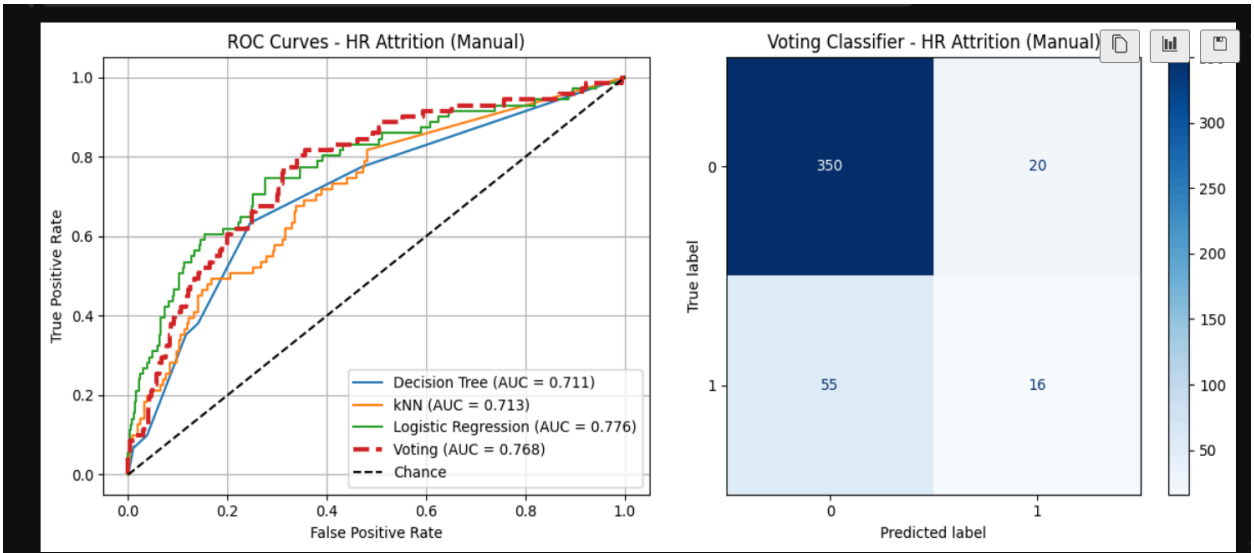
- Manual and scikit-learn gave almost the same results.
- Logistic Regression was best overall with highest AUC (~0.77).
- Decision Tree and kNN had okay accuracy but weak recall.
- ROC curves confirmed Logistic Regression separated classes better, but recall stayed low due to imbalance.







```
--------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 15, 'classifier__C': 0.1, 'classifier__penalty': 'l2', 'classif
Best cross-validation AUC: 0.7776

============================================================
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: 0.3953
  Recall: 0.2394
  F1-Score: 0.2982
  ROC AUC: 0.7130

Logistic Regression:
...
--- Manual Voting Classifier ---
Voting Classifier Performance:
```




```
--- GridSearchCV for Decision Tree ---
C:\Users\liss\AppData\Roaming\Python\Python312\site-packages\sklearn\feature_selection\_univariate_selection.py:110: UserWarnin
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning)
C:\Users\liss\AppData\Roaming\Python\Python312\site-packages\sklearn\feature_selection\_univariate_selection.py:111: RuntimeWar
  f = msb / msw
Best params for Decision Tree: {'classifier__max_depth': 3, 'classifier__min_samples_split': 2, 'feature_selection__k': 5}
Best CV score: 0.7152

--- GridSearchCV for kNN ---
C:\Users\liss\AppData\Roaming\Python\Python312\site-packages\sklearn\feature_selection\_univariate_selection.py:110: UserWarnin
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning)
C:\Users\liss\AppData\Roaming\Python\Python312\site-packages\sklearn\feature_selection\_univariate_selection.py:111: RuntimeWar
  f = msb / msw
Best params for kNN: {'classifier__n_neighbors': 7, 'classifier__weights': 'distance', 'feature_selection__k': 10}
Best CV score: 0.7073

--- GridSearchCV for Logistic Regression ---
Best params for Logistic Regression: {'classifier__C': 0.1, 'classifier__max_iter': 200, 'classifier__penalty': 'l2', 'classifi
Best CV score: 0.7776

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

```
============================================================
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: 0.3953
  Recall: 0.2394
  F1-Score: 0.2982
...

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

6. Conclusion

The key finding was that Logistic Regression performed best on the HR Attrition dataset with the highest ROC AUC (~0.77). Decision Tree and kNN had similar accuracy but much weaker recall.

The main takeaway is that model selection depends on both metrics and dataset balance, not just accuracy. The manual grid search gave a good understanding of how hyperparameter tuning works, but scikit-learn's GridSearchCV was faster, cleaner, and more reliable for larger experiments.