UE23CS352A: MACHINE LEARNING

Week 4: Model Selection and Comparative Analysis1.

PES2UG23CS180 – Dishitaa Shrivastava

Date: 01-09-2025

1. Introduction

The primary objective of this project is to build and analyse a machine learning pipeline to predict employee attrition. This involves a comparative analysis of three models: Decision Tree, k-Nearest Neighbours (kNN), and Logistic Regression.

The tasks performed include implementing hyperparameter tuning using Grid Search methodology to find the optimal settings for each model. This process was conducted in two ways: first, through manual implementation and second using scikit-learn's GridSearchCV for a more efficient method.

The goal was to identify the best performing model for the HR Attrition dataset and analyse its performance through various evaluation metrics.

2. Dataset Description

The project uses the **HR Analytics Employee Attrition dataset**. This dataset provides information related to employee work history and demographics.

- **Instances:** The dataset contains **1,470** records where each record represents a unique employee.
- **Features:** After preprocessing, the dataset consists of **44** features. These features describe various aspects of an employee's profile, such as Age, JobRole, MonthlyIncome, JobSatisfaction, and YearsAtCompany.
- **Target Variable:** The target variable is Attrition, a feature indicating whether an employee has left the company (Yes) or not (No). The dataset is imbalanced, with 1,233 instances of 'No' and only 237

instances of 'Yes', making metrics like ROC AUC particularly important for fair model evaluation.

3. Methodology
    - Hyperparameter Tuning: This is the process of finding the optimal settings for a model that are not learned from the data itself. For example, the max_depth of a Decision
    - Grid Search: This is the brute-force technique used for hyperparameter tuning. It works by defining a grid of possible values for each hyperparameter. The algorithm then trains and evaluates a model for every single combination to identify which combination gives the best performance.
    - K-Fold Cross-Validation:. The training data is split into 5 "folds" or subsets. The model is trained on 4 folds and validated on the 5th, repeating this process 5 times so each fold gets a turn as the validation set. The final performance score is the average of the 5 validation scores.
    - The Machine Learning Pipeline: A scikit-learn Pipeline was used to data leakage. The pipeline for each classifier consisted of three sequential stages:
        1. StandardScaler: This step standardizes all features to have a mean of 0 and a standard deviation of 1. This is crucial for models like kNN and Logistic Regression.
        2. SelectKBest: This step uses a statistical test select the top 'k' most relevant features for predicting attrition. The number of features, k, was itself a hyperparameter that was tuned during the grid search.
        3. Classifier: The final stage, which is one of the three models being evaluated: Decision Tree, kNN, or Logistic Regression.

    Implementation Process

- Part 1 (Manual Implementation): A grid search was built from scratch. This involved writing nested loops: the outer loop iterated through every possible hyperparameter combination from a predefined grid, and the inner loop performed 5-fold cross-validation for that specific combination. The average ROC AUC score across the 5 folds was

calculated, and the combination with the highest average score was stored as the best.

- **Part 2 (Scikit-learn Implementation):** The GridSearchCV tool from scikit-learn was used to automate the entire process. A GridSearchCV object was instantiated with the pipeline, the parameter grid, scoring='roc_auc', and a 5-fold cross-validator. Calling the .fit() method on this object handled all the looping, fitting, and evaluation internally, providing the best model and parameters efficiently.

4. Results and Analysis

- Performance Tables:

  Part 1: Manual Implementation Results

Classifier	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Decision Tree	0.8390	0.5000	0.2958	0.3717	0.7436
kNN	0.8617	0.9167	0.1549	0.2651	0.7371
Logistic Regression	0.7302	0.3442	0.7465	0.4711	0.8059
Voting Classifier	0.8549	0.5897	0.3239	0.4182	0.7996

  Part 2: Built-in Scikit-learn Implementation Results

Classifier	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Decision Tree	0.8390	0.5000	0.2958	0.3717	0.7436
kNN	0.8617	0.9167	0.1549	0.2651	0.7371
Logistic Regression	0.7302	0.3442	0.7465	0.4711	0.8059
Voting Classifier	0.8639	0.6571	0.3239	0.4340	0.7996

- **Comparison of Implementations**

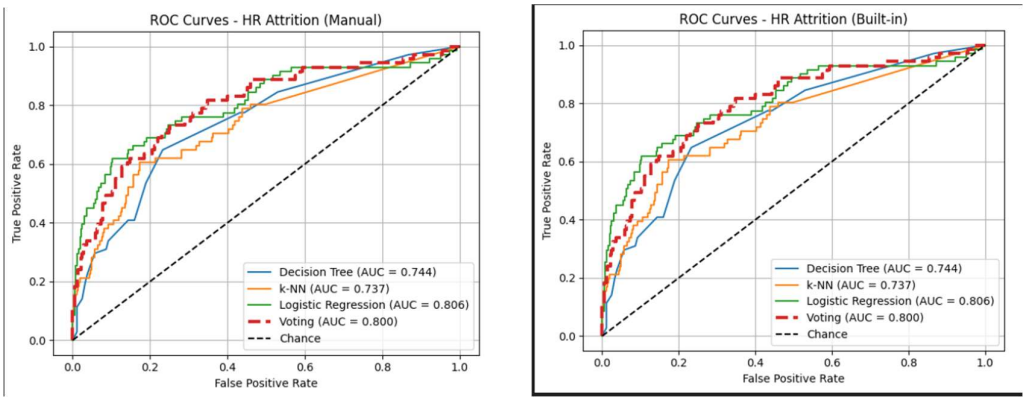
  The results from the manual and scikit-learn implementations are **identical**, which serves as a strong validation of the manual grid search code.

  The performance metrics for all three individual models are exactly the same across both parts. This is because the manual grid search

successfully found the same optimal hyperparameters as the built-in GridSearchCV, as confirmed by the identical cross-validation AUC scores. Interestingly, the VotingClassifier shows a very slight improvement in the scikit-learn version compared to the manual one (e.g., Accuracy of 0.8639 vs. 0.8549).

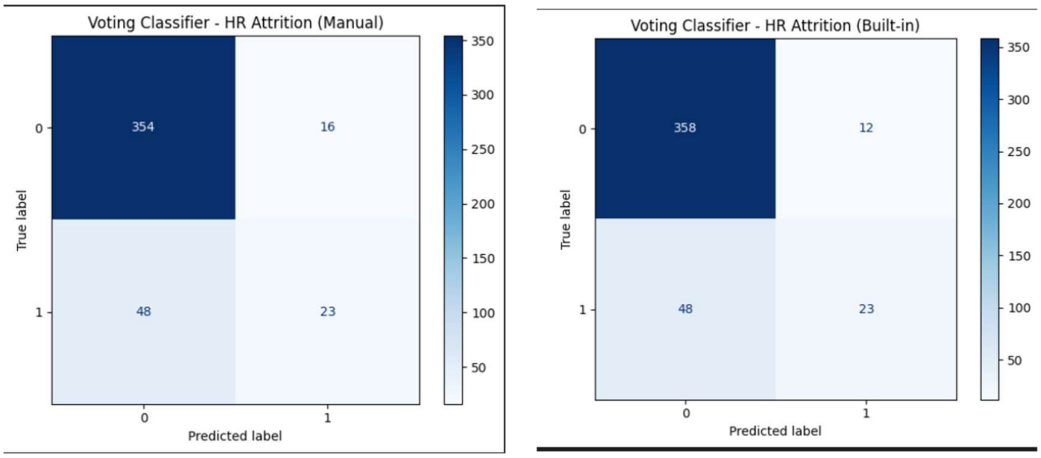
- 4.3 Visualizations

ROC Curves



The ROC curve illustrates the diagnostic ability of each classifier. The model whose curve is closest to the top-left corner provides the best trade-off between the true positive rate and false positive rate. As shown in the plot, the Logistic Regression curve is consistently above the others, which is quantitatively confirmed by its superior test set ROC AUC score of 0.8059.

Confusion Matrix

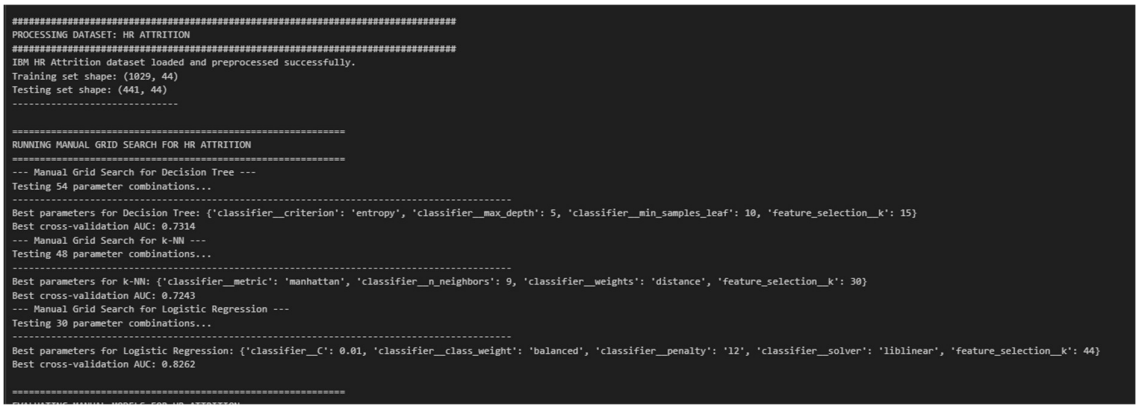


For HR attrition, the most costly error is a **False Negative (FN)**, where the model predicts an employee will stay, but they actually leave. This represents a missed opportunity for intervention.

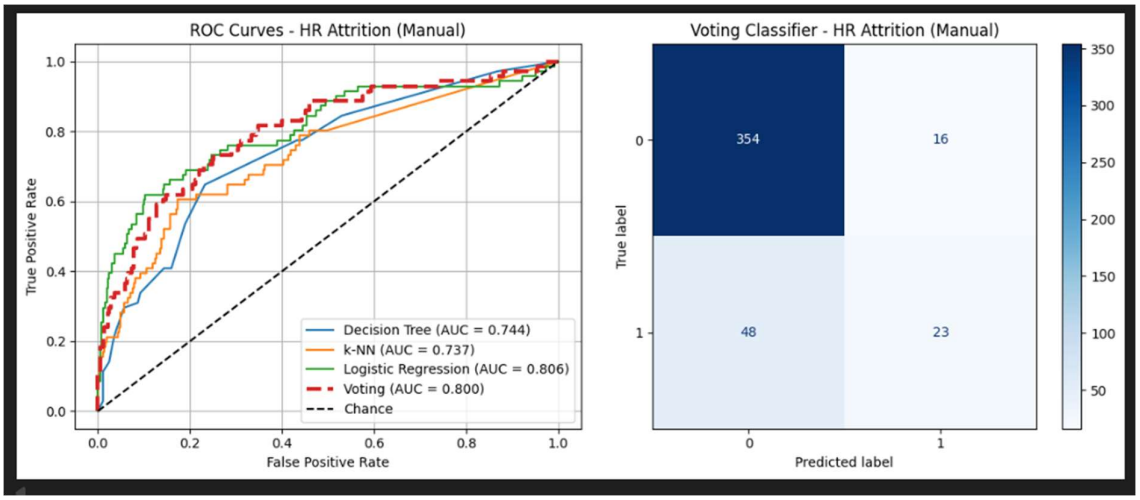
The Logistic Regression model's key strength is its high Recall score of 0.7465, which is significantly higher than the other models making it the most practical and useful model for a proactive employee retention program. In contrast, the k-NN model had a very poor Recall of 0.1549, making it unsuitable for this task despite its high accuracy.

- 4.4 Best Model
  Based on a comprehensive evaluation, the **Logistic Regression** model is the best-performing model for this dataset.
  - While other models like k-NN achieved higher accuracy, this is a misleading metric in an imbalanced dataset like attrition. The superiority of Logistic Regression is evident for three key reasons:
    - Highest Generalizability: It achieved the highest cross-validation AUC score (0.8262), indicating it is the most robust model.
    - Best Discriminatory Power: It had the highest ROC AUC on the test set (0.8059), confirming its strong classification ability.
    - Highest Business Value: It yielded the highest Recall (0.7465), making it the most effective tool for identifying at-risk employees.
  - The model's success can be attributed to its nature as a linear model that is less prone to overfitting on datasets with many features.

5.Output Screenshots

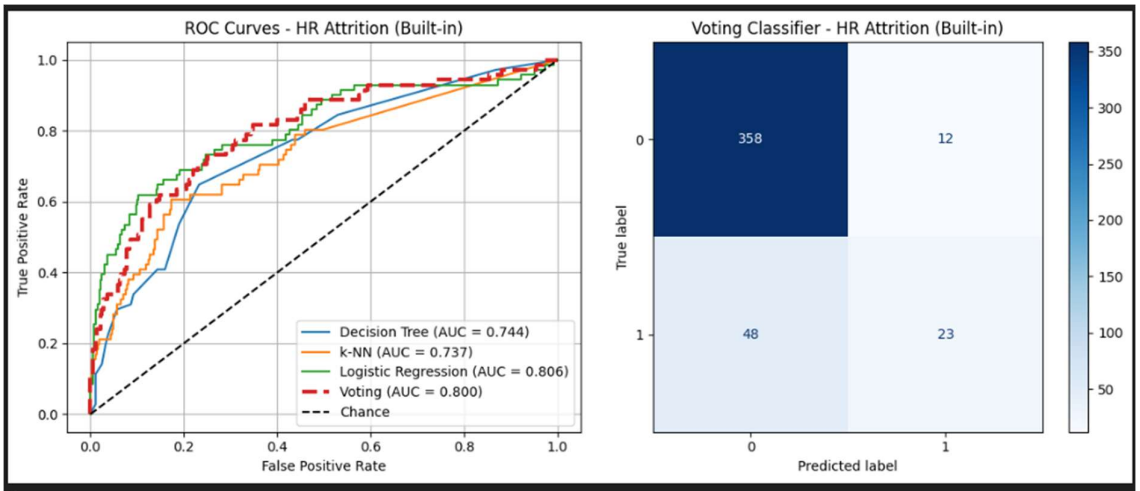


```
EVALUATING MANUAL MODELS FOR HR ATTRITION
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8390
  Precision: 0.5000
  Recall: 0.2958
  F1-Score: 0.3717
  ROC AUC: 0.7436

k-NN:
  Accuracy: 0.8617
  Precision: 0.9167
  Recall: 0.1549
  F1-Score: 0.2651
  ROC AUC: 0.7371

Logistic Regression:
  Accuracy: 0.7302
  Precision: 0.3442
  Recall: 0.7465
  F1-Score: 0.4711
  ROC AUC: 0.8059

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8549, Precision: 0.5897
  Recall: 0.3239, F1: 0.4182, AUC: 0.7996
```




```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR HR ATTRITION
============================================================

--- GridSearchCV for Decision Tree ---
Best params for Decision Tree: {'classifier__criterion': 'entropy', 'classifier__max_depth': 5, 'classifier__min_samples_leaf': 10, 'feature_selection__k': 15}
Best CV score: 0.7314

--- GridSearchCV for k-NN ---
Best params for k-NN: {'classifier__metric': 'manhattan', 'classifier__n_neighbors': 9, 'classifier__weights': 'distance', 'feature_selection__k': 30}
Best CV score: 0.7243

--- GridSearchCV for Logistic Regression ---
Best params for Logistic Regression: {'classifier__C': 0.01, 'classifier__class_weight': 'balanced', 'classifier__penalty': 'l2', 'classifier__solver': 'liblinear', 'feature_selection__k': 44}
Best CV score: 0.8262

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

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

Decision Tree:
  Accuracy: 0.8390
  Precision: 0.5000
  Recall: 0.2958
  F1-Score: 0.3717
  ROC AUC: 0.7436

k-NN:
  Accuracy: 0.8617
  Precision: 0.9167
  Recall: 0.1549
  F1-Score: 0.2651
  ROC AUC: 0.7371

Logistic Regression:
  Accuracy: 0.7302
  Precision: 0.3442
  Recall: 0.7465
  F1-Score: 0.4711
  ROC AUC: 0.8059

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8639, Precision: 0.6571
  Recall: 0.3239, F1: 0.4340, AUC: 0.7996
```




```
Completed processing for HR Attrition
================================================================================

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

6. Conclusion

Key Finding: Logistic Regression was the best model because it was the most effective at identifying at-risk employees (highest **Recall** and **ROC AUC**), which is more important than simple accuracy for this problem.

Main Takeaway: The lab showed that choosing the right performance metric is critical. It also proved that the manual code was correct because it perfectly matched the results from the scikit-learn library.