PROJECT TITLE : Model Selection and Comparative Analysis

NAME : CHARAN M REDDY

SRN : PES2UG23CS146

COURSE : MACHINE LEARNING

DATE : 1/09/2025

1

In this project we are trying to design a machine learning pipeline that can train, tune, and evaluate classification models across different datasets.

Here we first preprocess the data . Then we use both manual grid search and scikit-learn's built-in GridSearchCV for hyperparameter tuning, after which the models are evaluated on test data to measure their performance.

The results from different models are compared .

2

Number of instances : 1470

Number of features : 35

Target Variable: Attrition

Yes or No Value , which tells whether an employee has left the company

3

Hyperparameter tuning is finding the best parameter values for a model, such as the depth of a decision tree or the number of neighbors in KNN, on which performance is depended.

Grid search is approach where multiple combinations of hyperparameters are tested to find the most optimal one

K-fold cross-validation ensures fair evaluation by splitting the dataset into multiple folds, training on some while testing on the remaining, and averaging the results .

The machine learning pipeline in this lab consists of three main steps: StandardScaler, SelectKBest, and a classifier.

StandardScaler standardizes the features to have zero mean and unit variance, ensuring that models like K-Nearest Neighbors and Logistic Regression are not biased by feature scales.

SelectKBest then selects the top k most relevant features based on the AnovaF-value

Finally, the classifier , Decision Tree, K-Nearest Neighbors, or Logistic Regression , learns to predict the target variable. These classifiers are combined in a voting setup .

In the manual implementation , hyperparameter tuning was done by iterating over all parameter combinations for each classifier and evaluating them using 5-fold Cross-Validation. The combination with the highest mean AUC was selected, and the best pipelines were used to predict on the test set. Predictions from all classifiers were then combined using majority voting and averaged probabilities.

In the scikit-learn implementation , GridSearchCV automated the same process, performing cross-validation for all parameter combinations and selecting the best based on mean AUC. The final pipelines were fitted on the full training data, and predictions were combined with a voting classifier .

4

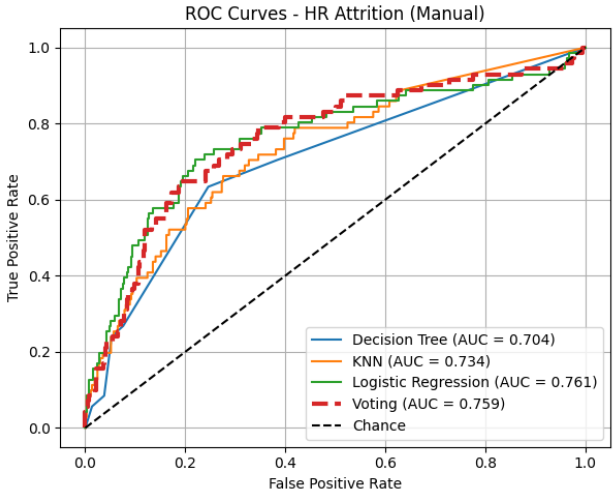
Performance Table

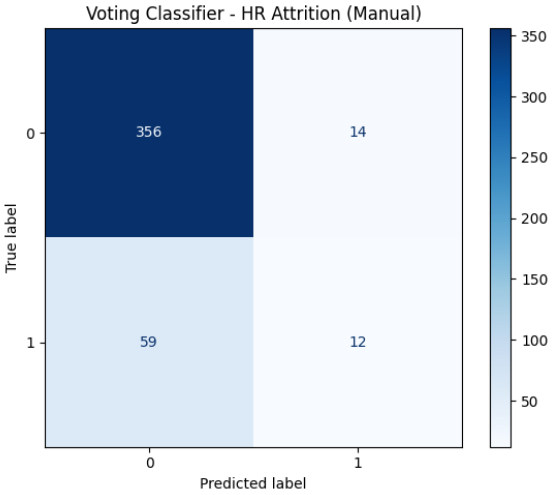
Manual :

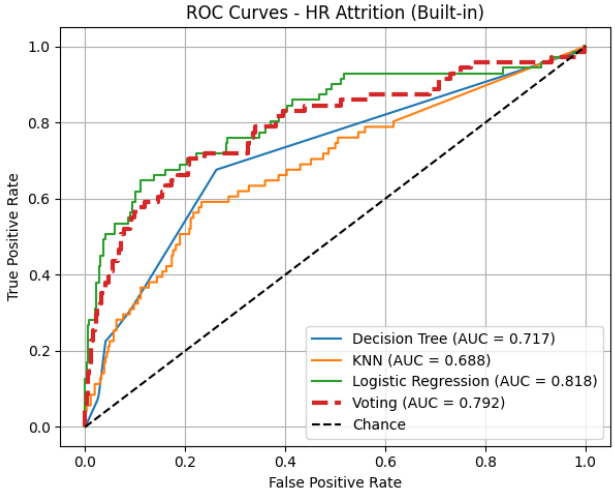
AlgoName ->	Decision Tree	KNN	Logistic Regression
Accuracy	0.8322	0.8277	0.8458
Precision	0.4571	0.4242	0.5600
Recall	0.2254	0.1972	0.1972
F-1 Score	0.3019	0.2692	0.2917
ROC ACU	0.7044	0.7340	0.7615

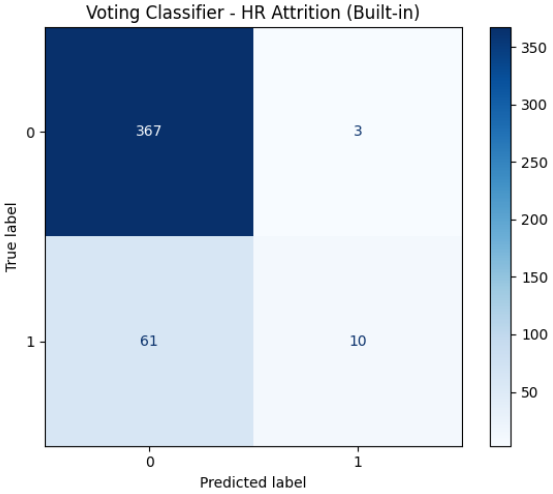
SciKit Learn

AlgoName ->	Decision Tree	KNN	Logistic Regression
Accuracy	0.8413	0.8413	0.8798
Precision	0.5161	0.5556	0.7368
Recall	0.2254	0.0704	0.3944
F-1 Score	0.3137	0.1250	0.5138
ROC ACU	0.7165	0.6883	0.8177









The manual implementation and GridSearchCV results are similar but not identical. Minor differences are from variations in cross-validation splits, handling of invalid parameter combinations, and slight differences in scoring or averaging methods.

Logistic Regression appears to be the best model overall for the HR Attrition dataset. Hypothesis - features in the dataset are largely linearly separable with respect to attrition making Logistic Regression very effective.

The dataset likely contains many small, numeric features that scale well, which suits Logistic Regression after standardization.

The ensemble (voting) classifier likely further improves robustness by combining strengths across models.

5

Output

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

============================================================
RUNNING MANUAL GRID SEARCH FOR HR ATTRITION
============================================================
--- Manual Grid Search for Decision Tree ---
--------------------------------------------------------------------------------
Best parameters for Decision Tree: {'classifier__max_depth': 5, 'classifier__min_samples_split': 10}
Best cross-validation AUC: 0.6885
--- Manual Grid Search for KNN ---
--------------------------------------------------------------------------------
Best parameters for KNN: {'classifier__n_neighbors': 11, 'classifier__weights': 'distance'}
Best cross-validation AUC: 0.7245
--- Manual Grid Search for Logistic Regression ---
--------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'classifier__C': 0.1, 'classifier__penalty': 'l2'}
Best cross-validation AUC: 0.8328

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

--- GridSearchCV for Decision Tree ---
Best params for Decision Tree: {'classifier__max_depth': 3, 'classifier__min_samples_split': 2}
Best CV score: 0.6916
Best params for KNN: {'classifier__n_neighbors': 11, 'classifier__weights': 'distance'}
Best CV score: 0.7245

Best params for Logistic Regression: {'classifier__C': 0.1, 'classifier__penalty': 'l2'}
Best CV score: 0.8328
```

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8322
  Precision: 0.4571
  Recall: 0.2254
  F1-Score: 0.3019
  ROC AUC: 0.7044

KNN:
  Accuracy: 0.8277
  Precision: 0.4242
  Recall: 0.1972
  F1-Score: 0.2692
  ROC AUC: 0.7340

Logistic Regression:
  Accuracy: 0.8458
  Precision: 0.5600
  Recall: 0.1972
  F1-Score: 0.2917
  ROC AUC: 0.7615

--- Voting Classifier (Manual) ---

Voting Classifier Performance:
  Accuracy: 0.8345, Precision: 0.4615
  Recall: 0.1690, F1: 0.2474, AUC: 0.7594
```

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8413
  Precision: 0.5161
  Recall: 0.2254
  F1-Score: 0.3137
  ROC AUC: 0.7165

KNN:
  Accuracy: 0.8413
  Precision: 0.5556
  Recall: 0.0704
  F1-Score: 0.1250
  ROC AUC: 0.6883

Logistic Regression:
  Accuracy: 0.8798
  Precision: 0.7368
  Recall: 0.3944
  F1-Score: 0.5138
  ROC AUC: 0.8177

--- Voting Classifier (Built-in) ---

Voting Classifier Performance:
  Accuracy: 0.8549, Precision: 0.7692
  Recall: 0.1408, F1: 0.2381, AUC: 0.7919
```

6

From this lab, I learned that model selection involves tuning hyperparameters and evaluating models using consistent metrics like ROC AUC. Manual implementation provides a deep understanding of the process, showing how cross-validation, feature selection, and parameter combinations interact, but it is time-consuming and may lead to small errors.

Using a library like scikit-learn automates tasks, handles errors making it faster and more reliable. The trade-off is that manual implementation offers more learning and flexibility, while scikit-learn gives rise to efficiency and ease of use.