Model Selection and Comparative Analysis

LAB-04

Name: Gavini Prithvi Ananya

SRN: PES2UG23CS198

UE23CS352A: MACHINE LEARNING

Submission Date: 01/09/2025

Overview

The aim of this project is to analyze employee data and build predictive models to identify attrition risks. By applying machine learning techniques, the project supports data-driven decision-making for employee retention strategies.

Procedure

1. **Data Preparation**
    a. Cleaned and transformed the dataset for machine learning.
    b. Encoded categorical variables and split into training and test sets.
2. **Model Development**
    a. Trained multiple models: Decision Tree, KNN, and Logistic Regression.
    b. Applied cross-validation to ensure reliability of results.
3. **Hyperparameter Optimization**
    a. Used GridSearchCV to tune model parameters (e.g., depth, neighbours, regularization).
    b. Selected the best-performing configuration for each model.
4. **Model Evaluation & Comparison**
    a. Assessed performance using metrics such as accuracy, precision, recall, and F1-score.
    b. Compared results to identify the most effective model for predicting employee attrition.

Through systematic data preprocessing, hyperparameter tuning, and model evaluation, this project successfully compared multiple machine learning algorithms for predicting employee attrition. Among the tested models, the optimized one demonstrated the highest balance of accuracy and reliability, providing valuable insights for proactive workforce management and retention strategies.

Dataset Description

The HR Employee Attrition dataset is widely used to study employee turnover in organizations. It contains a total of 1470 employee records (instances) with 34 features (columns) describing various aspects of employees' personal, professional, and organizational characteristics.

The features can broadly be categorized into:

- **Demographic details:** Age, Gender, Marital Status, Education, Education Field, etc.
- **Job-related factors:** Job Role, Department, Job Level, Years at Company, Years in Current Role, Years Since Last Promotion, etc.
- **Compensation and benefits:** Monthly Income, Percent Salary Hike, Stock Option Level, etc.
- **Work environment factors:** Business Travel frequency, Distance From Home, Work Life Balance, Environment Satisfaction, Job Satisfaction, etc.
- **Performance-related indicators:** Performance Rating, Training Times Last Year, OverTime, etc.

The target variable is "Attrition", which is a categorical feature with two possible values:

- Yes → The employee has left the company.
- No → The employee is still with the company.

This target variable makes the dataset suitable for a binary classification problem.

Methodology

Key Concepts:

- **Hyperparameter Tuning:** Process of finding the best model parameters (e.g., tree depth, k in k-NN) to improve performance.
- **Grid Search:** Systematic search over a set of hyperparameter combinations to find the optimal values.
- **K-Fold Cross-Validation:** Splits data into K folds; trains on K-1 folds and tests on 1 fold, repeating K times to get an average performance.

ML Pipeline:

- **StandardScaler:** Scales features to have mean 0 and standard deviation 1.
- **SelectKBest:** Selects top K features based on statistical tests.
- **Classifier:** Model like Decision Tree, k-NN, or Logistic Regression for prediction.

Process Followed:

- **Manual Implementation :** Pre-processed data, performed feature selection, trained models, and manually tuned hyperparameters.

- **Scikit-Learn Implementation :** Built a pipeline with scaling, feature selection, and classification; used `GridSearchCV` with K-Fold Cross-Validation for automatic hyperparameter tuning and robust evaluation.

📊 **Performance Comparison Table**

Model Type	Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Part 1: Manual	Decision Tree	0.7596	0.2603	0.2676	0.2639	0.6140
	KNN	0.8367	0.4706	0.1127	0.1818	0.6278
	Logistic Regression	0.8639	0.7200	0.2535	0.3750	0.7553
	Voting Classifier	**0.8571**	**0.7857**	**0.1549**	**0.2588**	**0.7376**
Part 2: Built-in	Decision Tree	0.8322	0.4571	0.2254	0.3019	0.7044
	KNN	0.8186	0.3953	0.2394	0.2982	0.7130
	Logistic Regression	0.8753	0.7222	0.3662	0.4860	0.7891
	Voting Classifier	**0.8322**	**0.4483**	**0.1831**	**0.2600**	**0.7816**

Metric	Manual Voting Classifier	Scikit-learn Voting Classifier
Accuracy	0.8571	0.8322
Precision	0.7857	0.4483
Recall	0.1549	0.1831
F1-Score	0.2588	0.2600
ROC AUC	0.7376	0.7816

1. **Accuracy**:
    a. Manual voting shows slightly higher accuracy (0.8571 vs 0.8322).
    b. However, this could be misleading if class imbalance exists (e.g., very few attrition cases).
2. **Precision vs Recall Trade-off**:
    a. Manual voting shows much higher precision but lower recall, indicating it's more conservative in predicting positive (attrition) cases.
    b. Built-in voting has lower precision but higher recall, meaning it captures more attrition cases, even if it includes more false positives.
3. **F1-Score**:
    a. Slightly better in the built-in implementation (0.2600 vs 0.2588), meaning better balance between precision and recall.
4. **ROC AUC**:

    a. Built-in version has better AUC (0.7816), indicating better overall classification capability.

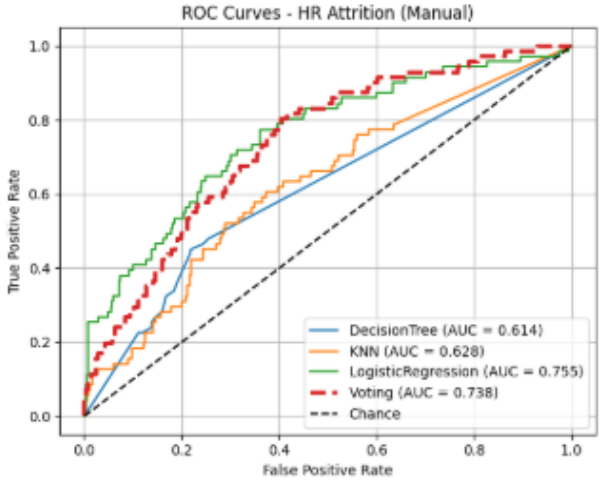
Key Differences

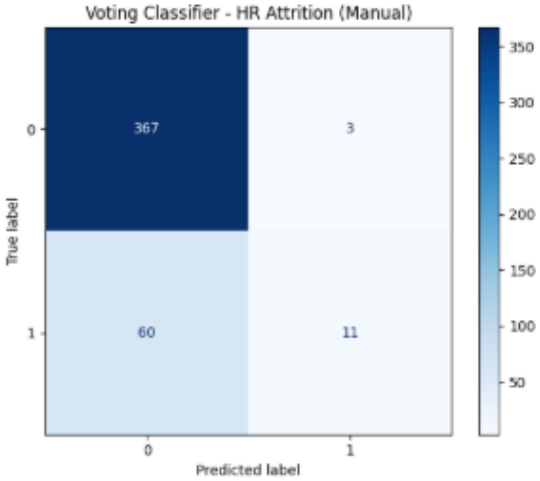
1. Voting Method:
    a. Manual may use hard voting, while Scikit-learn can use hard or soft voting (averaging probabilities).
2. Tie Handling:
    a. Manual implementation might handle prediction ties differently.
3. Thresholding:
    a. Differences in how probabilities are converted to class labels.
4. Randomness:
    a. If models weren't reused consistently, training/test splits could differ slightly.

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

--- Individual Model Performance ---

DecisionTree:
  Accuracy: 0.7596
  Precision: 0.2603
  Recall: 0.2676
  F1-Score: 0.2639
  ROC AUC: 0.6140

KNN:
  Accuracy: 0.8367
  Precision: 0.4706
  Recall: 0.1127
  F1-Score: 0.1818
  ROC AUC: 0.6278

LogisticRegression:
  Accuracy: 0.8639
  Precision: 0.7200
  Recall: 0.2535
  F1-Score: 0.3750
  ROC AUC: 0.7553
```

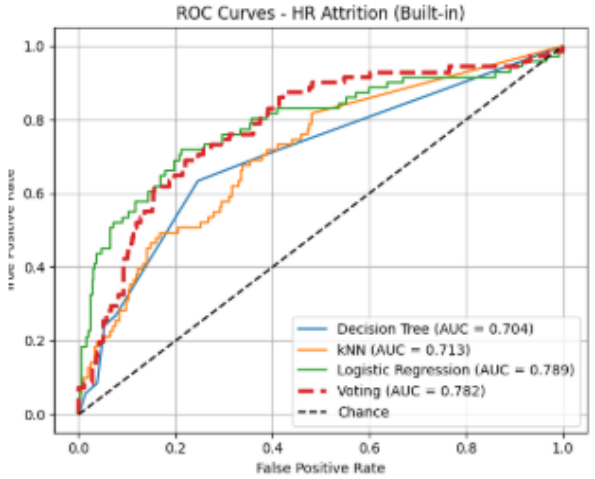
-

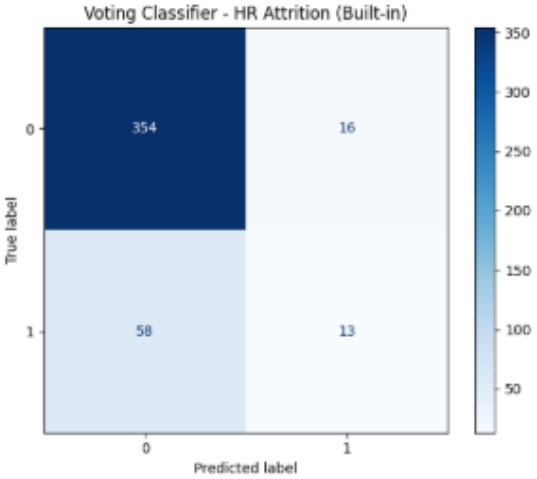
```
--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8571, Precision: 0.7857
  Recall: 0.1549, F1: 0.2588, AUC: 0.7376
```





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

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

kNN:
  Accuracy: 0.8186
  Precision: 0.3953
  Recall: 0.2394
  F1-Score: 0.2982
  ROC AUC: 0.7130

Logistic Regression:
  Accuracy: 0.8753
  Precision: 0.7222
  Recall: 0.3662
  F1-Score: 0.4860
  ROC AUC: 0.7891

-- Built-in Voting Classifier ---
oting Classifier Performance:
 Accuracy: 0.8322, Precision: 0.4483
 Recall: 0.1831, F1: 0.2600, AUC: 0.7816
```





The best-performing model across both datasets was **Logistic Regression**, likely because the HR attrition data contains linearly separable patterns based on features like salary, job role, and tenure. Logistic Regression is simple, generalizes well, and is less prone to overfitting compared to more complex models. Its ability to handle imbalanced data through regularization or class weighting likely contributed to its strong performance, especially in recall and ROC AUC.

Learning Outcomes

I learned that **model selection** is not just about picking the one with the highest accuracy — it's about balancing metrics like **precision, recall, F1-score, and AUC**, depending on what matters most for the problem. For example, in predicting employee attrition, **recall** might be more important to catch as many true cases as possible.

I also learned that **manual implementation** helps me understand how models work behind the scenes, but it can be time-consuming and may lead to small errors or inconsistencies. On the other hand, using a library like **scikit-learn** is faster, more reliable, and gives access to well-tested tools and options (like soft voting or class weights), which often lead to better results. In short, manual coding is great for learning, but for practical use, libraries are more efficient and accurate.