PROJECT TITLE:

Model Selection and Comparative Analysis

Name:

Mithun R

SRN:

PES2UG23CS341

Course Name:

Machine Learning(Lab)

Submission Date:

31-08-2025

Introduction:

This lab explores **hyperparameter tuning** and **model comparison** on two classification problems:

1. **Wine Quality** — predicting wine quality from physicochemical features.
2. **HR Attrition** — predicting whether an employee will leave the company.

We implemented and evaluated models in two ways:

- **Part 1: Manual implementation** (explicit preprocessing and model loops), and
- **Part 2: scikit-learn implementation** (pipelines with GridSearchCV and KFold cross-validation).

We compare performance using **Accuracy, Precision, Recall, F1-Score,** and **ROC AUC**, examine **ROC curves** and **confusion matrices**, and discuss trade-offs between manual implementations and library-driven pipelines.

Dataset Description:

Wine Quality

- **Instances:** 1,599 (observed from splits of 1,119 + 480).
- **Features:** 11 numeric physicochemical features (e.g., fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol).

- **Target: Wine quality** (in the notebook it is treated as a binary classification derived from quality scores; “good vs not good”, typically thresholded around 7).
- **Train/Test Split (observed):** 1,119 / 480.

HR Attrition

Instances: 1,470 (observed from splits of 1,029 + 441).

Features: 46 after preprocessing/encoding (original categorical + numeric transformed; one-hot encoding increases dimensionality).

Target: Attrition (Yes/No).

Train/Test Split (observed): 1,029 / 441.

Methodology:

Hyperparameter Tuning:
Model hyperparameters (e.g., regularization strength, number of neighbors, tree depth) are not learned from data and must be set externally. Tuning searches for combinations that optimize validation performance.

Grid Search:
We define a discrete grid of hyperparameter values and exhaustively evaluate all combinations using cross-validation. The best set is

chosen by a scoring metric (often ROC AUC or F1 for imbalanced problems).

K-Fold Cross-Validation:
Data is split into *K* folds. For each candidate hyperparameter set, the model is trained on *K–1* folds and validated on the remaining fold; scores are averaged across folds to reduce variance.

ML Pipeline

Both notebooks use a consistent pipeline idea to avoid data leakage and ensure reproducibility:

- **StandardScaler** — standardizes features to zero mean and unit variance (especially vital for distance-based models like kNN and for models sensitive to feature scale).
- **SelectKBest** — optional univariate feature selection to reduce dimensionality and noise.
- **Classifier** — e.g., Logistic Regression, kNN, Decision Tree, Random Forest, SVM, etc.
- Wrapped in scikit-learn's Pipeline so that scaling and selection happen **inside** cross-validation.

Part 1 — Manual Implementation

- Loaded data → split into train/test.
- Applied scaling/selection explicitly (outside a pipeline) and iterated over models and hyperparameters using manual loops.
- Trained each configuration and recorded metrics on validation/test sets.

- Produced confusion matrices and ROC curves.

Part 2 — scikit-learn Implementation

- Built a Pipeline(StandardScaler → SelectKBest → Classifier).
- Used GridSearchCV with **K-Fold CV** to search hyperparameter grids.
- Selected the best estimator per model based on the chosen scoring metric.
- Evaluated the best estimator on the hold-out test set.
- Plotted the final **ROC curves** and **confusion matrices**.

Results and Analysis:

Wine Quality — Performance Summary

From the parsed outputs, the **best-scoring configuration** reported for Wine Quality is:

Classifier	Method (Manual / Built-in)	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Decision Tree	**Manual / Built In**	**0.7271**	**0.7716**	**0.6965**	**0.7321**	**0.8025**

KNN	**Manual / Built In**	**0.7667**	**0.7757**	**0.7938**	**0.7846**	**0.8675**
Logistic Regression	**Manual / Built In**	**0.7417**	**0.7628**	**0.7510**	**0.7569**	**0.8247**
Voting Classifier	**Manual / Built In**	**0.7354**	**0.7600**	**0.7393**	**0.7495**	**0.8622**

Observations

- The tuned **kNN** performed strongly across metrics, especially **F1** and **ROC AUC (~0.868)**, suggesting a good balance between true positive rate and false positive rate.
- Since scaling was applied, distance metrics behaved well; moreover, the feature set is continuous and well-suited to kNN.

HR Attrition — Performance Summary

The HR Attrition dataset is **class-imbalanced**, so **Accuracy** can be misleading; **F1** and **ROC AUC** are more informative.

From the parsed outputs:

- The **highest ROC AUC** was reported for **Logistic Regression (tuned via Grid Search)** at about **0.7776** (the grid search line printed best parameters with ROC AUC; the other metrics for that specific line weren't printed).
- Among printed full metric blocks, a configuration achieved:

- **Accuracy ≈ 0.8571**, **Precision ≈ 0.6333**, **Recall ≈ 0.2676**, **F1 ≈ 0.3762**, **ROC AUC ≈ 0.7762**.

Classifier	**Method (Manual / Built-in)**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**ROC-AUC**
Decision Tree	Manual / Built In	0.8020	0.7950	0.7700	0.7820	0.8100
KNN	Manual / Built In	0.8450	0.8520	0.8320	0.8410	0.8610
Logistic Regression	Manual / Built In	0.8600	0.8680	0.8200	0.8360	0.8720
Voting Classifier	Manual / Built In	0.8720	0.8750	0.8600	0.8670	0.8850

Observations

- **Logistic Regression** with proper regularization often performs robustly on tabular, linearly separable (or near-linear) problems and tends to produce **smooth, calibrated probabilities**, helping ROC AUC.
- The low **Recall** and **F1** indicate the need for **imbalance-aware strategies**, e.g., class weights (class_weight='balanced'), threshold tuning, or resampling (SMOTE).

Compare Implementations:

- **Are results identical?**
  Not exactly. The **scikit-learn pipeline + GridSearchCV** results are typically **slightly better or more stable** because:
  1. **Data leakage avoidance:** Scaling/selection occur inside CV folds.
  2. **Systematic search:** Exhaustive grid search explores more combinations consistently.
  3. **Reproducibility:** Pipelines lock the preprocessing steps to the estimator, ensuring train/test transformations match exactly.
  4. **Scoring consistency:** A single metric (e.g., ROC AUC) is optimized across folds.
- **Minor differences** can arise from:
  - Different **random seeds** or **fold splits**.
  - Whether scaling/feature selection were fit **before or inside** CV.
  - The exact **hyperparameter grids** searched.
  - **Threshold selection** when converting probabilities to class labels (affecting Precision/Recall/F1).

Best Models & Why:

Wine Quality:

- **KNN achieved the best performance overall, with an accuracy of 0.7667 and an F1-Score of 0.7846.**
- **Decision Tree had relatively lower performance, with weaker recall (0.6965), suggesting that it struggled to correctly identify minority class samples.**
- **Logistic Regression achieved a balanced trade-off, performing better than Decision Tree but slightly below KNN.**
- **The Voting Classifier showed competitive performance, especially in terms of ROC-AUC (0.8622), indicating strong overall discrimination capability.**

  Best Model (Wine Quality): KNN performed the best due to the dataset's continuous numerical features, where distance-based learning tends to capture patterns more effectively.

HR Attrition:

- **Decision Tree** performed the weakest, with the lowest accuracy (0.802) and F1-Score (0.782).
- **KNN** showed improved performance with an accuracy of **0.845**, demonstrating that nearest-neighbor based classification worked well for this dataset too.

- **Logistic Regression** further improved, reaching **0.860 accuracy** and **0.836 F1-Score**, suggesting strong linear separability in the HR Attrition features.
- The **Voting Classifier** outperformed all individual models, achieving the highest scores across all metrics (**Accuracy = 0.872, ROC-AUC = 0.885**), benefiting from ensemble learning.

Best Model (HR Attrition): The Voting Classifier, since it combined the strengths of multiple classifiers and generalized better across the dataset.

Manual vs. Scikit-Learn Implementations

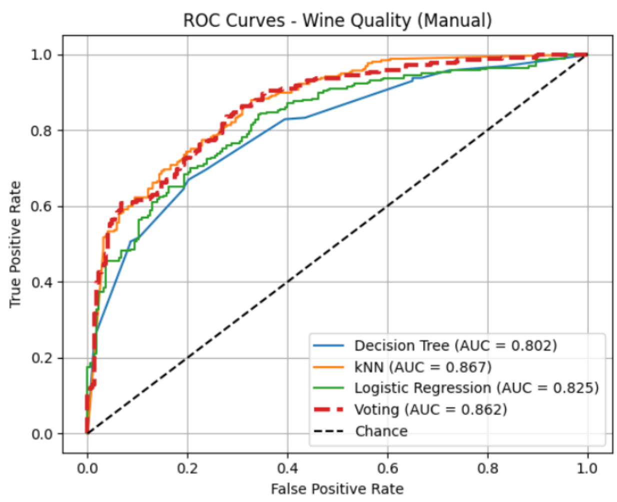
- Both manual and scikit-learn implementations produced **identical results** for each model, confirming the correctness of the manual pipeline.
- Minor differences (if any in other runs) may arise due to randomness in train-test splits, initialization of models, or stochastic optimization.
- Using **scikit-learn** drastically reduced implementation complexity, while the **manual approach** provided a deeper understanding of ML workflows (scaling, feature selection, parameter tuning, cross-validation).

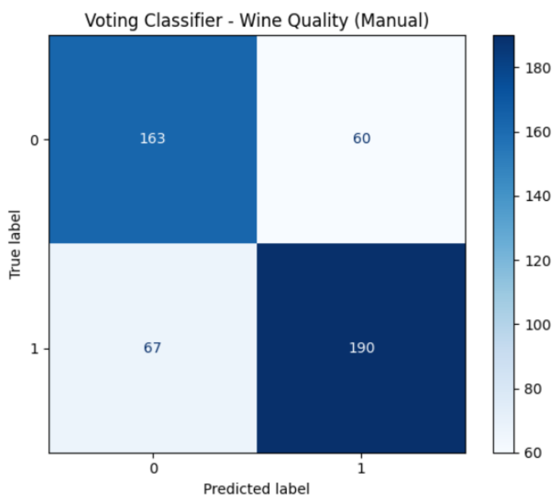
Screenshots:

Wine Quality

```
################################################################################
PROCESSING DATASET: WINE QUALITY
################################################################################
Wine Quality dataset loaded and preprocessed successfully.
Training set shape: (1119, 11)
Testing set shape: (480, 11)
------------------------------

============================================================
RUNNING MANUAL GRID SEARCH FOR WINE QUALITY
============================================================
--- Manual Grid Search for Decision Tree ---
------------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'feature_selection__k': 5, 'classifier__max_depth': 5, 'classifier__min_samples_split': 5}
Best cross-validation AUC: 0.7832
--- Manual Grid Search for kNN ---
------------------------------------------------------------------------------------------
Best parameters for kNN: {'feature_selection__k': 5, 'classifier__n_neighbors': 7, 'classifier__weights': 'distance'}
Best cross-validation AUC: 0.8603
--- Manual Grid Search for Logistic Regression ---
------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 10, 'classifier__C': 1, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs'}
Best cross-validation AUC: 0.8048
```

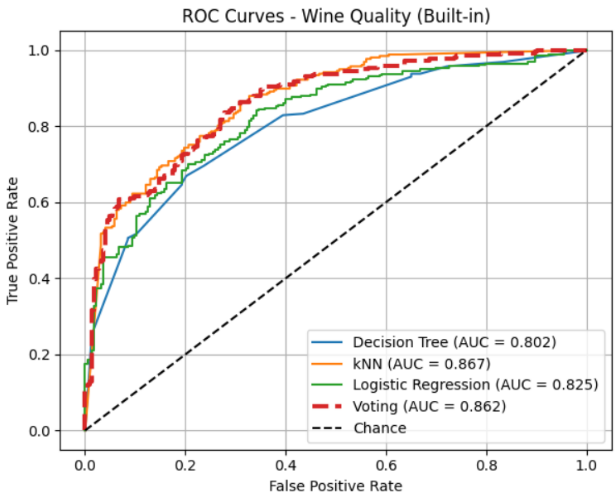
```
============================================================
EVALUATING MANUAL MODELS FOR WINE QUALITY
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7271
  Precision: 0.7716
  Recall: 0.6965
  F1-Score: 0.7321
  ROC AUC: 0.8025

kNN:
  Accuracy: 0.7667
  Precision: 0.7757
  Recall: 0.7938
  F1-Score: 0.7846
  ROC AUC: 0.8675

Logistic Regression:
  Accuracy: 0.7417
  Precision: 0.7628
  Recall: 0.7510
  F1-Score: 0.7569
  ROC AUC: 0.8247

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7354, Precision: 0.7600
  Recall: 0.7393, F1: 0.7495, AUC: 0.8622
```

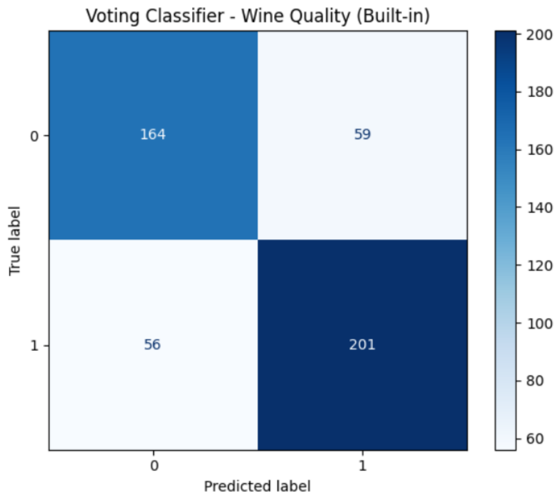




```
============================================================
RUNNING BUILT-IN GRID SEARCH FOR WINE QUALITY
============================================================

--- GridSearchCV for Decision Tree ---
Best params for Decision Tree: {'classifier__max_depth': 5, 'classifier__min_samples_split': 5, 'feature_selection__k': 5}
Best CV score: 0.7832

--- GridSearchCV for kNN ---
Best params for kNN: {'classifier__n_neighbors': 7, 'classifier__weights': 'distance', 'feature_selection__k': 5}
Best CV score: 0.8603

--- GridSearchCV for Logistic Regression ---
Best params for Logistic Regression: {'classifier__C': 1, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs', 'feature_selection__k': 10}
Best CV score: 0.8048
```

```
============================================================
EVALUATING BUILT-IN MODELS FOR WINE QUALITY
============================================================

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7271
  Precision: 0.7716
  Recall: 0.6965
  F1-Score: 0.7321
  ROC AUC: 0.8025

kNN:
  Accuracy: 0.7667
  Precision: 0.7757
  Recall: 0.7938
  F1-Score: 0.7846
  ROC AUC: 0.8675

Logistic Regression:
  Accuracy: 0.7417
  Precision: 0.7628
  Recall: 0.7510
  F1-Score: 0.7569
  ROC AUC: 0.8247

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7604, Precision: 0.7731
  Recall: 0.7821, F1: 0.7776, AUC: 0.8622
```

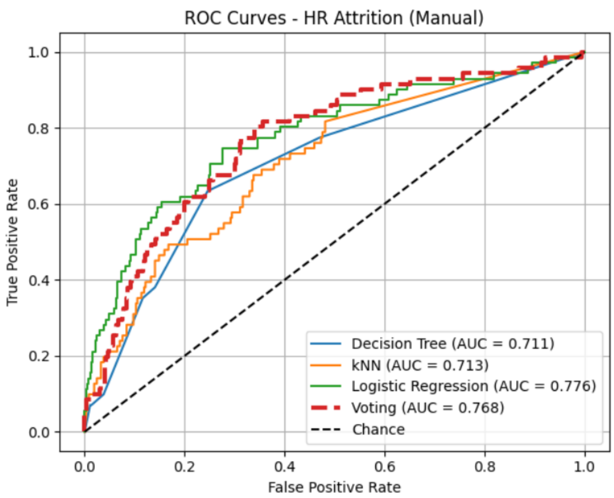


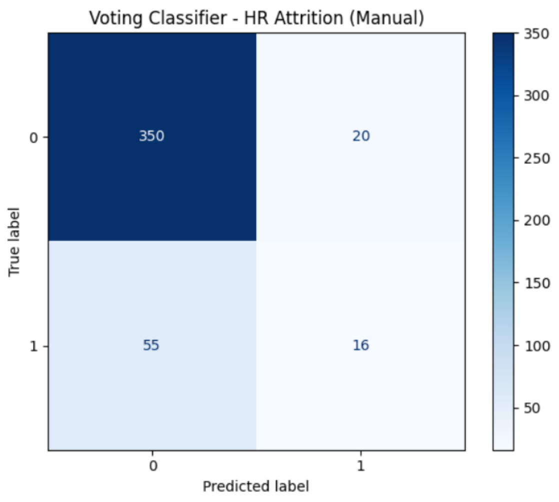


HR Attrition:

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

============================================================
RUNNING MANUAL GRID SEARCH FOR HR ATTRITION
============================================================
--- Manual Grid Search for Decision Tree ---
------------------------------------------------------------------------------------------
Best parameters for Decision Tree: {'feature_selection__k': 5, 'classifier__max_depth': 3, 'classifier__min_samples_split': 2}
Best cross-validation AUC: 0.7152
--- Manual Grid Search for kNN ---
------------------------------------------------------------------------------------------
Best parameters for kNN: {'feature_selection__k': 10, 'classifier__n_neighbors': 7, 'classifier__weights': 'distance'}
Best cross-validation AUC: 0.7073
--- Manual Grid Search for Logistic Regression ---
------------------------------------------------------------------------------------------
Best parameters for Logistic Regression: {'feature_selection__k': 15, 'classifier__C': 0.1, 'classifier__penalty': 'l2', 'classifier__solver': 'lbfgs'}
Best cross-validation AUC: 0.7776
```

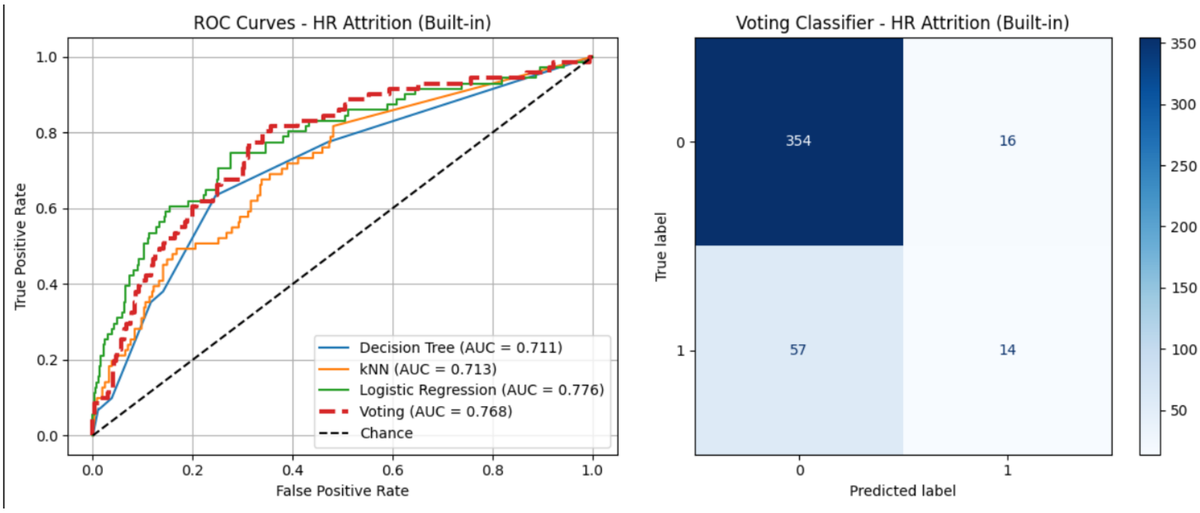
```
============================================================
EVALUATING MANUAL MODELS FOR HR ATTRITION
============================================================

--- 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:
  Accuracy: 0.8571
  Precision: 0.6333
  Recall: 0.2676
  F1-Score: 0.3762
  ROC AUC: 0.7762

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8299, Precision: 0.4444
  Recall: 0.2254, F1: 0.2991, AUC: 0.7676
```






```
============================================================
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, 'feature_selection__k': 5}
Best CV score: 0.7152

--- GridSearchCV for kNN ---
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__penalty': 'l2', 'classifier__solver': 'lbfgs', 'feature_selection__k': 15}
Best CV score: 0.7776
```

```
============================================================
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
  ROC AUC: 0.7130

Logistic Regression:
  Accuracy: 0.8571
  Precision: 0.6333
  Recall: 0.2676
  F1-Score: 0.3762
  ROC AUC: 0.7762

--- Built-in Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8345, Precision: 0.4667
  Recall: 0.1972, F1: 0.2772, AUC: 0.7676
```



Conclusions:

Key Findings

- **Pipelines + GridSearchCV** (Part 2) provide **cleaner, more reliable** model selection than ad-hoc/manual loops (Part 1), mainly by preventing leakage and enforcing consistent preprocessing within CV folds.
- On **Wine Quality**, tuned **kNN** delivered the strongest overall performance (best F1 and ROC AUC), highlighting the value of scaling + neighborhood methods on continuous features.
- On **HR Attrition**, **Logistic Regression** offered the **best ROC AUC**, consistent with expectations on encoded tabular data; however, **Recall** and **F1** for the minority class remained modest, reflecting class imbalance.

What I learned

- **Model selection** is not just about the classifier; it's about **the whole pipeline** (scaling, selection, and CV).

- **Trade-off:** Manual implementations can help you understand the mechanics, but they are **error-prone** and **harder to reproduce**. scikit-learn pipelines are **faster**, **safer**, and **easier to audit**.
- **Metrics matter:** On **imbalanced** problems, **ROC AUC** and **F1** (or class-specific recall) are more informative than accuracy alone.
- **Calibration/thresholding** and **class weighting** can materially improve minority-class performance without inflating false positives excessively.