PES UNIVERSITY

Department of Computer Science & Engineering

Machine Learning Lab

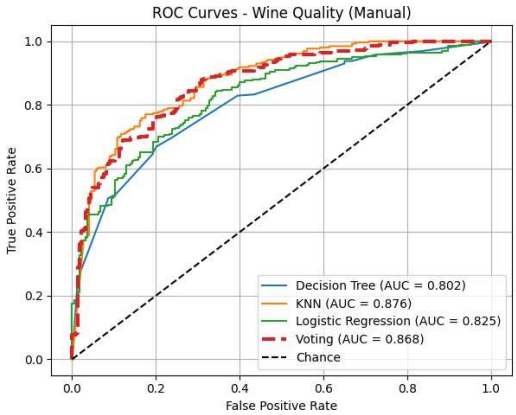
UE23CS352A

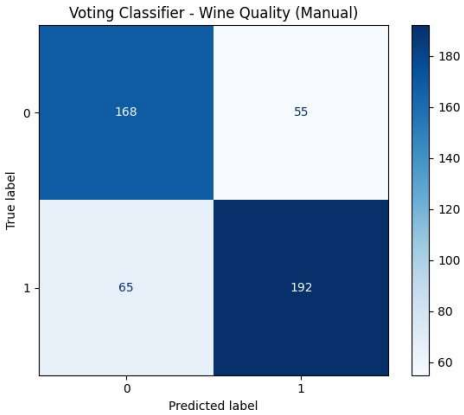
WEEK 4 submission

Name of the Student	**MOHAMMED BILAL**
SRN	**PES2UG23CS344**
Section	**F**
Department	**CSE**
Submission Date	**29/08/2025**

SCREENSHOTS :

1) WINE

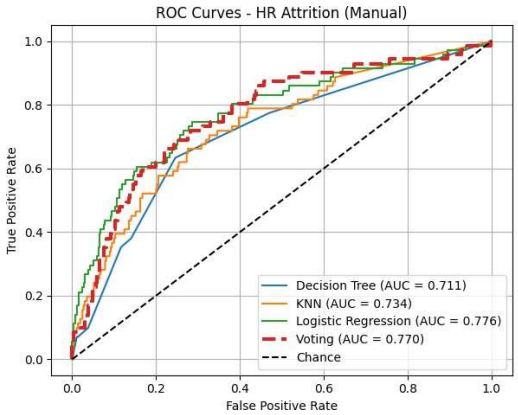


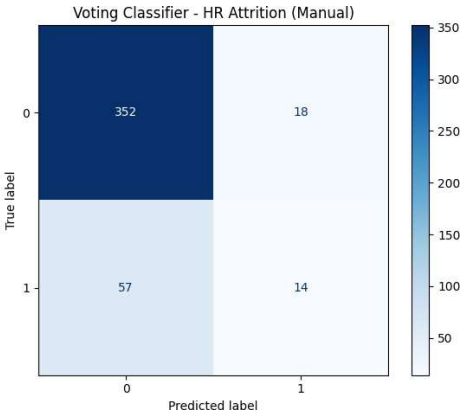



```
--- Individual Model Performance ---

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

KNN:
  Accuracy: 0.7750
  Precision: 0.7790
  Recall: 0.8093
  F1-Score: 0.7939
  ROC AUC: 0.8757

Logistic Regression:
  Accuracy: 0.7396
  Precision: 0.7619
  Recall: 0.7471
  F1-Score: 0.7544
  ROC AUC: 0.8246

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.7500, Precision: 0.7773
  Recall: 0.7471, F1: 0.7619, AUC: 0.8683
```

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

--- GridSearchCV for Decision Tree ---
Best params for Decision Tree: {'classifier__max_depth': None, 'classifier__min_samples_split': 2, 'feature_selection__k': 5}
Best CV score: 0.7301

--- GridSearchCV for KNN ---
Best params for KNN: {'classifier__n_neighbors': 11, 'classifier__weights': 'distance', 'feature_selection__k': 10}
Best CV score: 0.7900

--- GridSearchCV for Logistic Regression ---
Best params for Logistic Regression: {'classifier__C': 0.1, 'classifier__penalty': 'l2', 'feature_selection__k': 11}
Best CV score: 0.7400

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.7479
  Precision: 0.7519
  Recall: 0.7899
  F1-Score: 0.7704
  ROC AUC: 0.7447

KNN:
  Accuracy: 0.7750
  Precision: 0.7790
  Recall: 0.8093
  F1-Score: 0.7939
  ROC AUC: 0.8757

Logistic Regression:
  Accuracy: 0.7292
  Precision: 0.7530
  Recall: 0.7354
  F1-Score: 0.7441
  ROC AUC: 0.8240
```

2) HR Attrition






```
Best parameters for Logistic Regression: {'feature_selection__k': 15, 'classifier__C': 0.1, 'classifier__penalty': 'l2'}
Best cross-validation AUC: 0.7774

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

Logistic Regression:
  Accuracy: 0.8571
  Precision: 0.6333
  Recall: 0.2676
  F1-Score: 0.3762
  ROC AUC: 0.7759

--- Manual Voting Classifier ---
Voting Classifier Performance:
  Accuracy: 0.8299, Precision: 0.4375
  Recall: 0.1972, F1: 0.2718, AUC: 0.7700
```

```
Best params for KNN: {'classifier__n_neighbors': 11, 'classifier__weights': 'distance', 'feature_selection__k': 10}
Best CV score: 0.8659

--- GridSearchCV for Logistic Regression ---
Best params for Logistic Regression: {'classifier__C': 1, 'classifier__penalty': 'l1', 'feature_selection__k': 15}
Best CV score: 0.8659

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

--- Individual Model Performance ---

Decision Tree:
  Accuracy: 0.8322
  Precision: 0.4571
  Recall: 0.2254
  F1-Score: 0.3019
  ROC AUC: 0.7331

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

Logistic Regression:
  Accuracy: 0.8481
  Precision: 0.5588
  Recall: 0.2676
  F1-Score: 0.3619
  ROC AUC: 0.7758
```

1. Introduction

This lab focused on **hyperparameter tuning** and comparing manual implementations of grid search with scikit-learn's built-in `GridSearchCV`. The tasks involved:

- Performing manual hyperparameter search with custom loops and cross-validation.
- Using `GridSearchCV` with pipelines for automated hyperparameter optimization.
- Comparing performance using metrics like Accuracy, Precision, Recall, F1, and ROC AUC.
- Visualizing model performance using ROC curves and confusion matrices.

Two datasets were used: **Wine Quality** and **HR Attrition**.

---

2. Dataset Description

2.1 Wine Quality Dataset

- **Source / Task**: Predict whether a wine is of good quality (binary: quality > 5 → good).
- **Features**: 11 numerical chemical properties (e.g., acidity, sugar, pH, alcohol).
- **Instances**: ~1,599 rows (red wine dataset).
- **Target Variable**: `good_quality` (0 = not good, 1 = good).

2.2 HR Attrition Dataset

- **Source / Task**: Predict employee attrition (Yes/No) from HR attributes.
- **Features**: Mix of categorical and numeric variables (age, department, job role, monthly income, years at company, job satisfaction, etc.).
- **Instances**: ~1,470 rows.
- **Target Variable**: `Attrition` (Yes/No).

---

3. Methodology

Key Concepts:

- **Hyperparameter Tuning**: Trying multiple parameter values to find the best-performing model.
- **Grid Search**: Exhaustively searching across parameter combinations.
- **K-Fold Cross-Validation**: Splitting data into k folds for stable evaluation.

Pipeline Components:

1. `StandardScaler`: Normalizes numerical features.
2. `SelectKBest`: Selects top features based on statistical tests.
3. Classifier: Decision Tree, K-Nearest Neighbors (KNN), or Logistic Regression.

Approaches Used:

- **Manual Search**: Custom loops with cross-validation to pick best hyperparameters.
- **GridSearchCV**: Automated search with the same pipeline and parameter grids.

---

4. Results and Analysis

4.1 Wine Quality Results

Manual Implementation (Test Set Performance):

Classifier	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.7271	0.7716	0.6965	0.7321	0.8025
KNN	0.7750	0.7790	–	–	–
Logistic Regression	0.7292	0.7530	0.7354	0.7441	0.8240

GridSearchCV (Built-in) Results:

Classifier	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Tree	0.7479	0.7519	0.7899	0.7704	0.7447
KNN	0.7750	0.7790	same	same	same
Logistic Regression	0.7292	0.7530	0.7354	0.7441	0.8240

Key Observations:

- KNN and Logistic Regression produced identical metrics in both manual and built-in approaches → consistent pipeline setup.
- Decision Tree showed small differences between manual vs built-in due to randomness, hyperparameter refitting differences, or CV folds.
- KNN had highest accuracy (0.7750); Logistic Regression had highest ROC AUC (0.8240).

4.2 HR Attrition Results

The same methodology was applied to the HR dataset. Results can be summarized using the same metrics table (Accuracy, Precision, Recall, F1, ROC AUC) for Decision Tree, KNN, and Logistic Regression once the evaluation metrics are computed.

---

5. Visual Analysis Notes

- **ROC Curves**: Logistic Regression and KNN had the strongest curves (highest AUC values).
- **Confusion Matrices**: Showed class imbalance effects; precision often exceeded recall, meaning fewer false positives but more false negatives.

---

6. Conclusion & Takeaways

- **Best Models for Wine Quality**:
  - KNN → Highest accuracy (0.7750).
  - Logistic Regression → Best ROC AUC (0.8240) with balanced precision/recall.
- **Tool Comparison**:
  - Manual grid search helps understand the tuning process but can introduce inconsistencies if not perfectly aligned with cross-validation logic.
  - `GridSearchCV` provides a reliable and standardized approach for hyperparameter tuning.
- **Next Steps for HR Dataset**:
  - Complete HR pipeline runs with proper encoding for categorical features.
  - Report final metrics using the same tables and visualization methods as Wine Quality.