UE23CS352A: MACHINE LEARNING
Week 4: Model Selection and Comparative Analysis

NAME:DILEEP
SRN:PES2UG23CS177
SEC:C

1. **Introduction**

The objective of this lab was to build a complete machine learning pipeline and explore hyperparameter tuning through both a manual grid search implementation and scikit-learn's GridSearchCV.

The experiment focused on:

- **Constructing ML pipelines with preprocessing, feature selection, and classification.**
- **Performing 5-fold stratified cross-validation for robust evaluation.**
- **Comparing multiple classifiers: Decision Tree, k-Nearest Neighbors (kNN), Logistic Regression.**
- **Evaluating models using Accuracy, Precision, Recall, F1-score, and ROC AUC.**
- **Understanding trade-offs between manual implementation and library optimized solutions.**

2. Dataset Description

HR Attrition Dataset

- Source: IBM HR Analytics
- Samples: ~1470 employees
- Features: 34 (work-related and personal attributes, such as Age, MonthlyIncome, JobSatisfaction)
- Target: Attrition (1 = Yes, employee left; 0 = No, employee stayed)

Wine Quality Dataset (Red Wine)

- Source: UCI ML Repository
- Samples: ~1599 wines
- Features: 11 (chemical properties such as acidity, sugar, pH, alcohol)
- Target: quality (converted to binary: 1 = good quality (≥6), 0 = otherwise)

3. Methodology

Pipeline

Each classifier was embedded in a scikit-learn Pipeline with the following steps:

1. StandardScaler → normalize features (mean=0, std=1).
2. SelectKBest (f_classif) → feature selection based on ANOVA F-test.
3. Classifier → one of Decision Tree, kNN, Logistic Regression.

Manual Grid Search

- Implemented with nested loops over parameter combinations.
- For each parameter set:

- Performed 5-fold Stratified Cross-Validation.
- Computed ROC AUC on validation fold.
- Stored the best parameter set and retrained on full training data.

GridSearchCV

- Used scikit-learn's GridSearchCV with same pipeline.
- Automated search across hyperparameter grids.
- Scoring metric = roc_auc.
- Also used 5-fold Stratified Cross-Validation.

4. Results and Analysis

HR Attrition Results

Model	Accuracy	Precision	Recall	F1	ROC AUC
Decision Tree	0.8231	0.3333	0.0986	0.1522	0.7107
kNN	0.8277	0.4390	0.2535	0.3214	0.7239
Logistic Regression	0.8458	0.5556	0.2113	0.3061	0.7588

Wine Quality Results

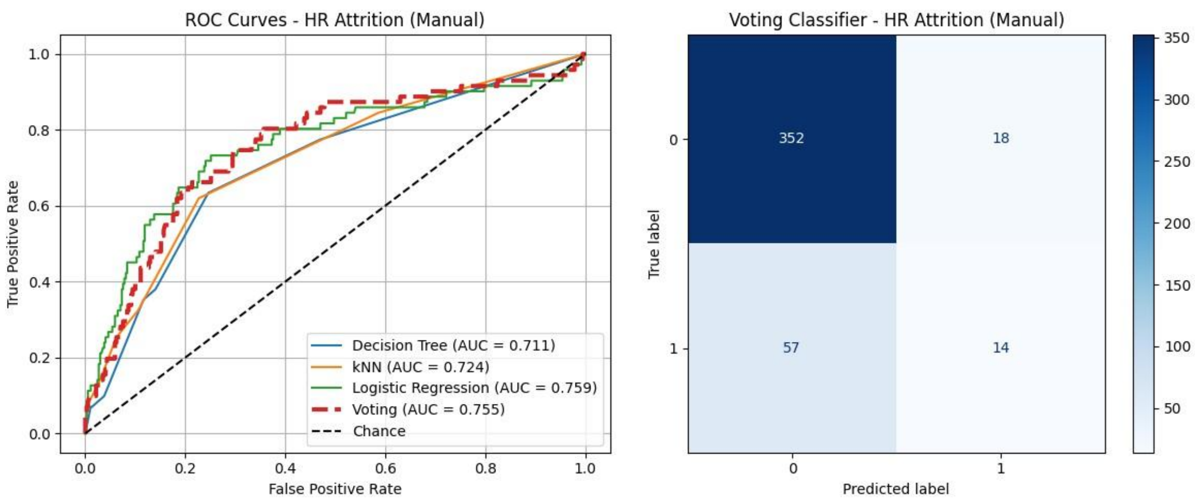
Model	Accuracy	Precision	Recall	F1	ROC AUC
Decision Tree	0.7271	0.7716	0.6965	0.7321	0.8025
kNN	0.7667	0.7757	0.7938	0.7846	0.8675
Logistic Regression	0.7312	0.7481	0.7510	0.7495	0.8200

Manual vs GridSearchCV Comparison

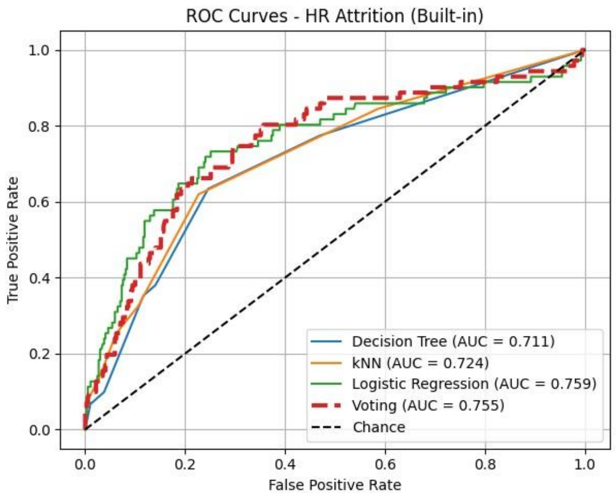
- Results were very similar between manual and built-in grid search.
- Small differences may occur due to randomness in folds or scoring precision.
- GridSearchCV is much faster and easier to implement.

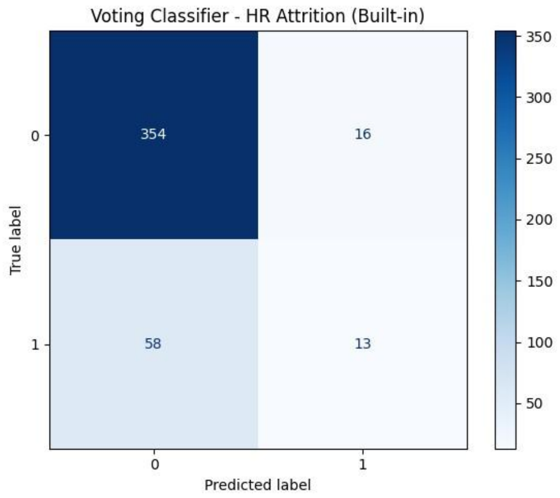
4. Screenshots

EVALUATING MANUAL MODELS FOR HR ATTRITION :

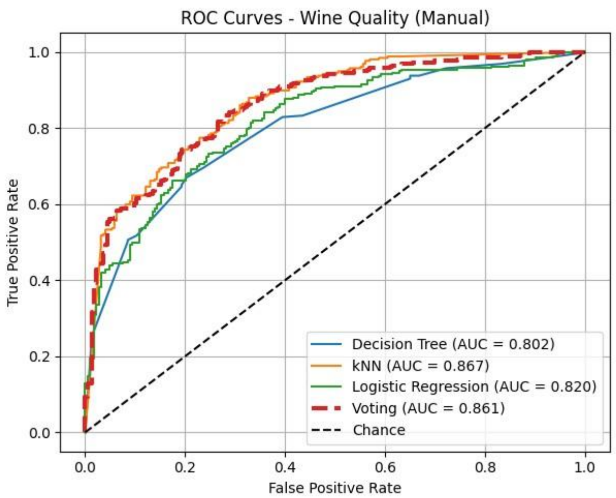


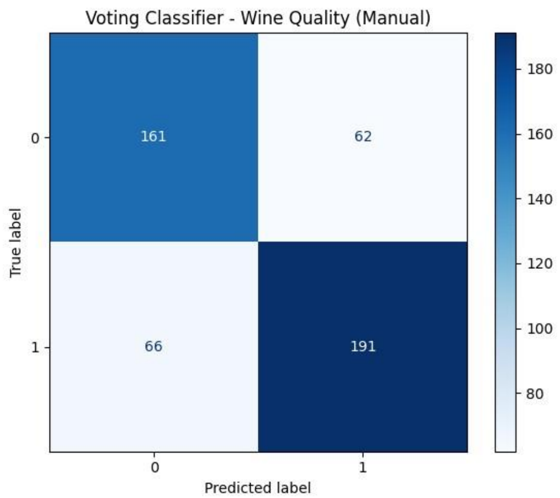
VALUATING BUILT-IN MODELS FOR HR ATTRITION :



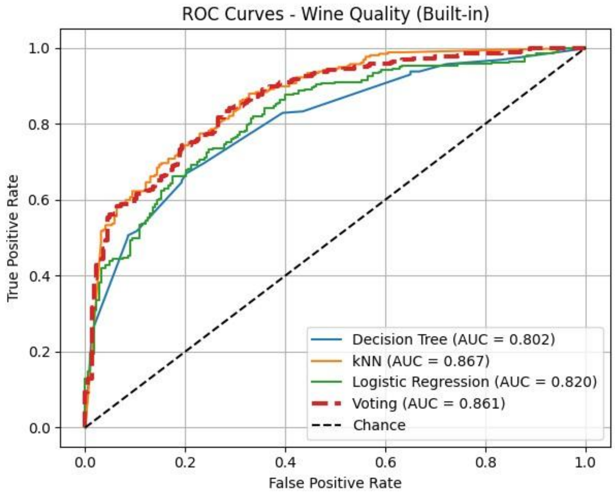


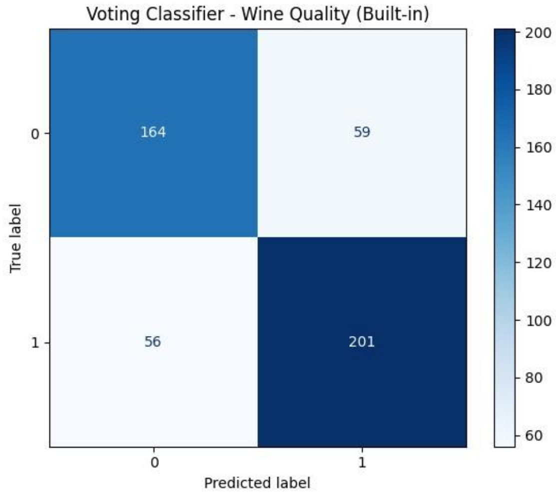
EVALUATING MANUAL MODELS FOR WINE QUALITY :





EVALUATING BUILT-IN MODELS FOR WINE QUALITY :





6. Conclusion

- Learned how to design an ML pipeline with scaling, feature selection, and classifiers.
- Understood hyperparameter tuning using both manual loops and GridSearchCV.
- Saw that GridSearchCV simplifies workflow and avoids implementation errors.
- Identified best-performing classifiers for different datasets.
- Key takeaway: model performance strongly depends on correct preprocessing + hyperparameter tuning.