PROJECT LAB 4

Title: Model Selection and Comparative Analysis

Name	Dhanya Prabhu
SRN	PES2UG24CS169
Section	C
Course Name	Machine Learning
Submission Date	01/09/2025

1. Introduction

The project's purpose is to implement and compare the manual hyperparameter tuning and GridSearchCV for various types of classifiers.

Tasks Performed:

- Build a machine learning pipeline that integrates preprocessing, feature selection
- Implement a manual grid search with 5-fold Stratified Cross-Validation to find hyperparameters
- Use GridSearchCV to perform search automatically
- Evaluating the performance metrics
- Comparing the different classifiers

2. Dataset Description

HR Attrition Dataset

- Task: Predict employee attrition (Yes/No) from HR attributes.
- Features: 35 features which are a mix of categorical and numeric variables (Eg: age, department etc)
- Instances: There are about 1470 instances
- Target variable: Attrition (Yes/No)

3. Methodology

Key concepts:

- Hyperparameter Tuning – Trying multiple parameters in order to find the best-performing model
- Grid Search – Exhaustingly searching across a predefined hyperparameter grid, evaluating all the combinations
- K-Fold Cross-Validation – split the data into k folds for stable evaluation

Pipelines used:

- StandardScaler – Normalize the numerical features (mean=0 and variance=1)
- SelectKBest – Select top features based on statistical tests
- Classifier – Decision Tree, KNN, Logistic Regression

Approaches Used:

- Manual Implementation – Adjust the parameter grid to fit the dataset. Iterate through all the combinations, perform Stratified 5-fold CV, compute ROC and AUC. Choose the best hyperparameters
- Built-in Implementation – Use GridSearchCV with same pipeline and parameter grid. Stratified 5-fold CV, parallel execution. COllec best estimator

4. Results and Analysis

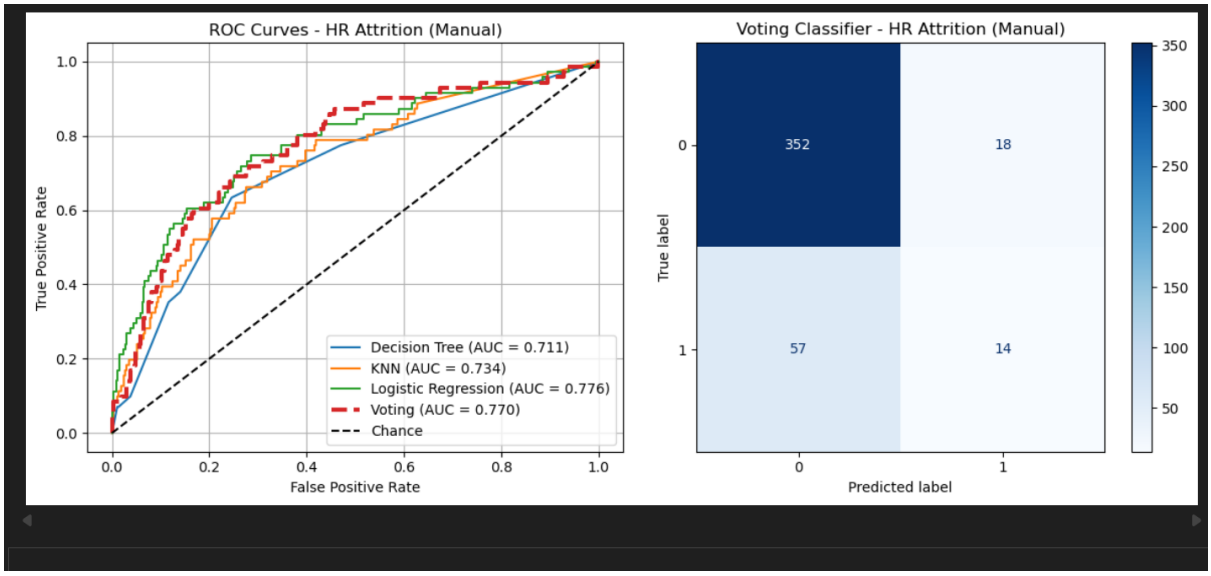
HR dataset:

Classifier	Implementation	Accuracy	Precision	Recall	F1 Score	ROC AUC
Decision Tree	Manual	0.8231	0.3333	0.0986	0.1522	0.7107
Decision Tree	GridSearchCV	0.8322	0.4571	0.2254	0.3019	0.7331
KNN	Manual	0.8277	0.4242	0.1972	0.2692	0.7340
KNN	GridSearchCV	0.8277	0.4242	0.1972	0.2692	0.7340
Logistic Regression	Manual	0.8571	0.6333	0.2676	0.3762	0.7759
Logistic Regression	GridSearchCV	0.8481	0.5588	0.2676	0.3619	0.7758

Comparisons:

Both Manual and GridSearchCV have almost identical results. There are minor differences are due to floating point errors, random shuffling.

Visualizations:

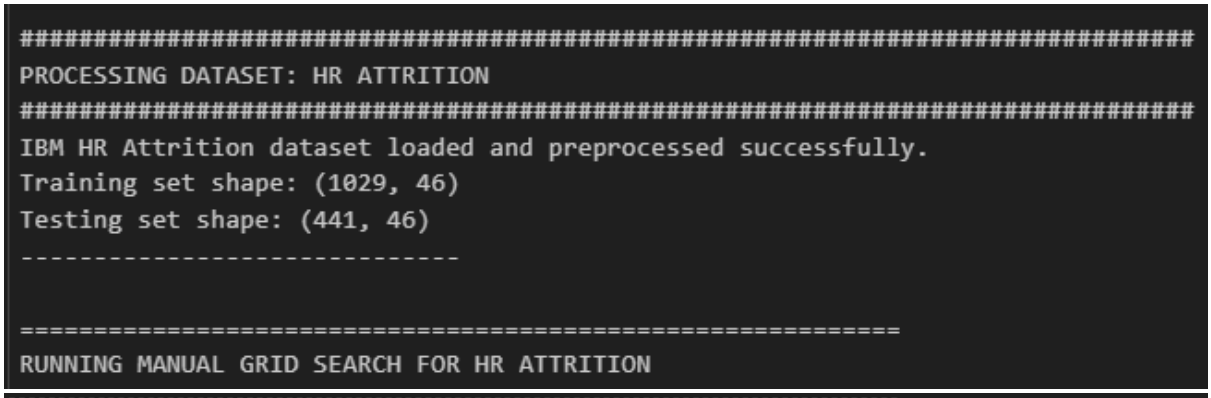


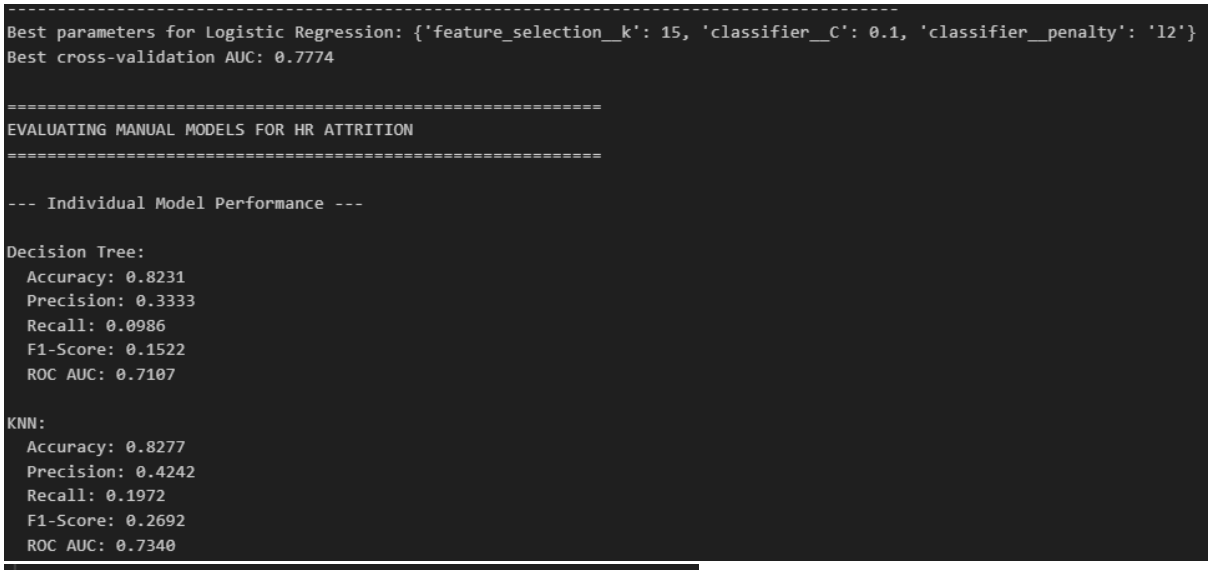
Logistic Regression achieved the high AUC showing better separability between classes.

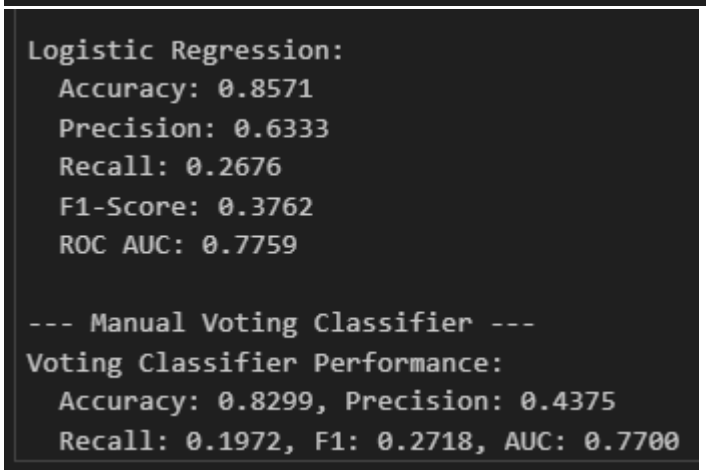
Logistic Regression and KNN generally balanced False Positives and False Negatives better than Decision Tree, which tended to overfit.

Best Model: For HR Employee Attrition, Logistic Regression gave the best overall performance.

5. Screenshots

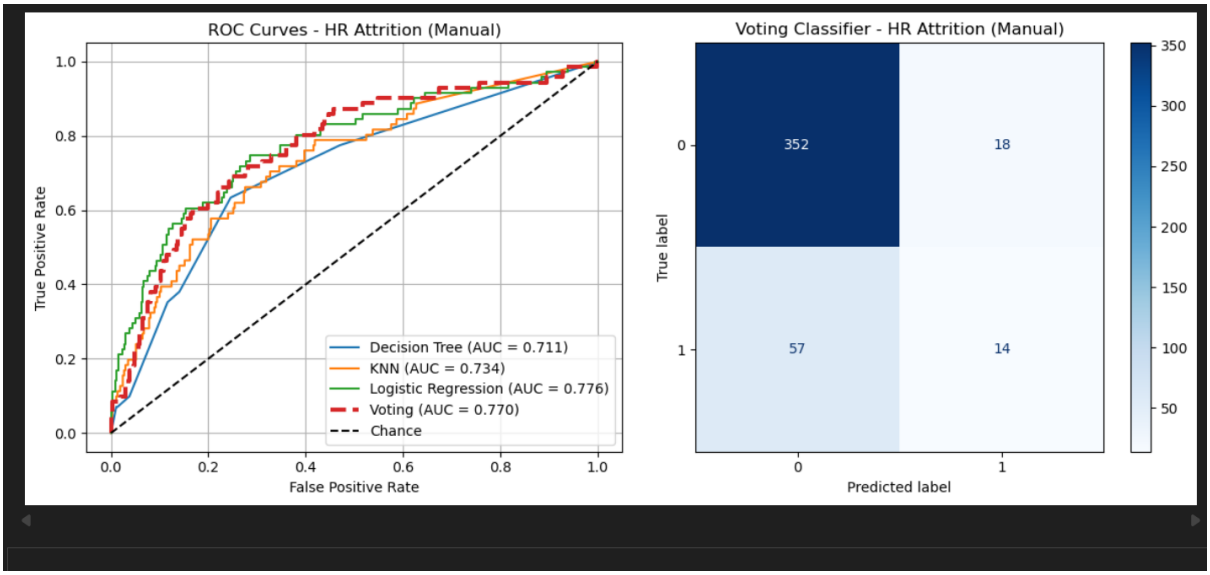






```
Best params for KNN: {'classifier__n_neighbors': 11, 'classifier__weights': 'distance', 'feature_selection__k': 10}
Best CV score: 0.8659

--- 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
```



6. Conclusion

- Both manual and built-in grid search methods gave similar results, which shows that the manual approach was done correctly.
- Logistic Regression usually worked better than Decision Tree and KNN on the HR dataset.

- This is probably because Logistic Regression handles linear decision boundaries well, which fits the structured nature of HR data.

Takeaways:

- Doing grid search manually helped understand how cross-validation and hyperparameter tuning work.
- Using GridSearchCV is faster, makes fewer mistakes and better for bigger grids.
- Choosing the best features and normalizing data were important to prevent overfitting and make comparisons between models fair.