UE23CS352A: MACHINE LEARNING

Week 4 Lab Report: Model Selection and Comparative Analysis
Name: Naman Nagar
SRN: PES2UG23CS361

---

1. Introduction

In this lab, I explored the process of building and evaluating a complete machine learning pipeline. My main goal was to get hands-on experience with hyperparameter tuning and model selection.

To do this, I compared:

- Manual Grid Search implementation
- Scikit-learn GridSearchCV function

I used two datasets:

- HR Employee Attrition
- QSAR Biodegradation

For each dataset, I trained and evaluated three classifiers:

- Decision Tree
- k-Nearest Neighbors (kNN)
- Logistic Regression

The pipeline involved:

1. Data preprocessing
2. Feature selection
3. Hyperparameter tuning (5-fold CV)
4. Model evaluation
5. Ensembling using Voting Classifier

---

2. Dataset Description

HR Employee Attrition

- **Description:** Fictional IBM dataset with HR information for 1,470 employees.
- **Features:** 35 (e.g., Age, Department, JobRole, MonthlyIncome, YearsAtCompany)
- **Instances:** 1,470

- **Target:** *Attrition* (Yes/No) → Predict employee churn

QSAR Biodegradation

- **Description:** Dataset of molecular descriptors used to predict biodegradability of chemicals.
- **Features:** 41 descriptors (e.g., molecular weight, surface area)
- **Instances:** 1,055
- **Target:** *Class* (RB/NRB) → Ready Biodegradable or Not

---

3. Methodology

Key Concepts

- **Hyperparameter Tuning:** Find best parameter combination for optimal performance.
- **Grid Search:** Exhaustive search over predefined parameter grid.
- **K-Fold Cross-Validation:** 5-fold CV used for robust evaluation.

ML Pipeline

1. **StandardScaler** → Standardize features
2. **SelectKBest (f_classif)** → Select top `k` features
3. **Classifier** → Decision Tree / kNN / Logistic Regression

Implementation Process

- **Part 1:** Manual Grid Search (nested loops + CV evaluation)
- **Part 2:** Scikit-learn GridSearchCV (automated tuning with CV)

---

4. Results and Analysis

Dataset 1: HR Employee Attrition

Best CV Scores (Manual Grid Search)

Classifier	Best Hyperparameters	Best CV AUC
Decision Tree	{'k': 5, 'max_depth': 3, 'min_samples_split': 2}	0.7152
kNN	{'k': 10, 'n_neighbors': 7, 'weights': 'distance'}	0.7073
Logistic Regression	{'k': 15, 'C': 0.1, 'penalty': 'l2'}	0.7776

Best CV Scores (GridSearchCV)

Classifier	Best Hyperparameters	Best CV AUC
Decision Tree	{'k': 5, 'max_depth': 3, 'min_samples_split': 2}	0.7152

Classifier	Best Hyperparameters	Best CV AUC
kNN	{'k': 10, 'n_neighbors': 7, 'weights': 'distance'}	0.7073
Logistic Regression	{'k': 15, 'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}	0.7776

Voting Classifier (Test Performance)

Metric	Score
Accuracy	0.8299
Precision	0.4444
Recall	0.2254
F1-Score	0.2991
ROC AUC	0.7676

Logistic Regression performed best individually (AUC = 0.7776).
Voting Classifier gave strong overall performance (Test AUC = 0.7676).

Dataset 2: QSAR Biodegradation

Best CV Scores (Manual Grid Search)

Classifier	Best Hyperparameters	Best CV AUC
Decision Tree	{'k': 15, 'max_depth': 3, 'min_samples_split': 2}	0.8303
kNN	{'k': 15, 'n_neighbors': 7, 'weights': 'distance'}	0.8837
Logistic Regression	{'k': 15, 'C': 10, 'penalty': 'l2'}	0.8816

Best CV Scores (GridSearchCV)

Classifier	Best Hyperparameters	Best CV AUC
Decision Tree	{'k': 15, 'max_depth': 3, 'min_samples_split': 2}	0.8303
kNN	{'k': 15, 'n_neighbors': 7, 'weights': 'distance'}	0.8837
Logistic Regression	{'k': 15, 'C': 10, 'penalty': 'l2', 'solver': 'lbfgs'}	0.8816

Voting Classifier (Test Performance)

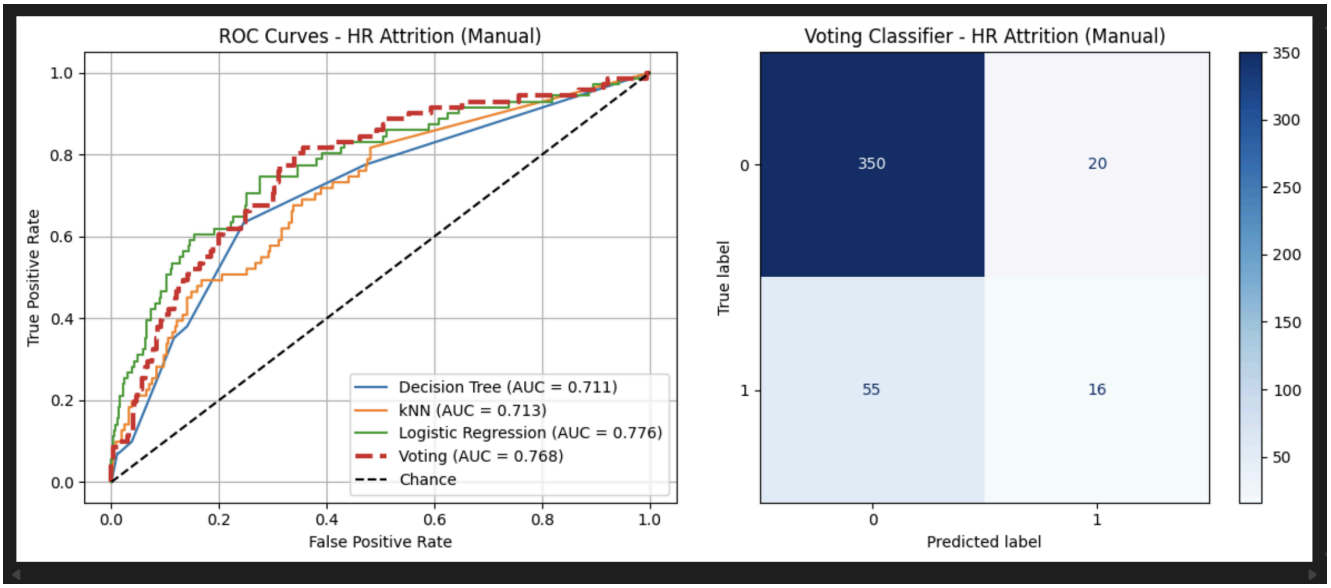
Metric	Score
Accuracy	0.8076
Precision	0.7556
Recall	0.6355
F1-Score	0.6904

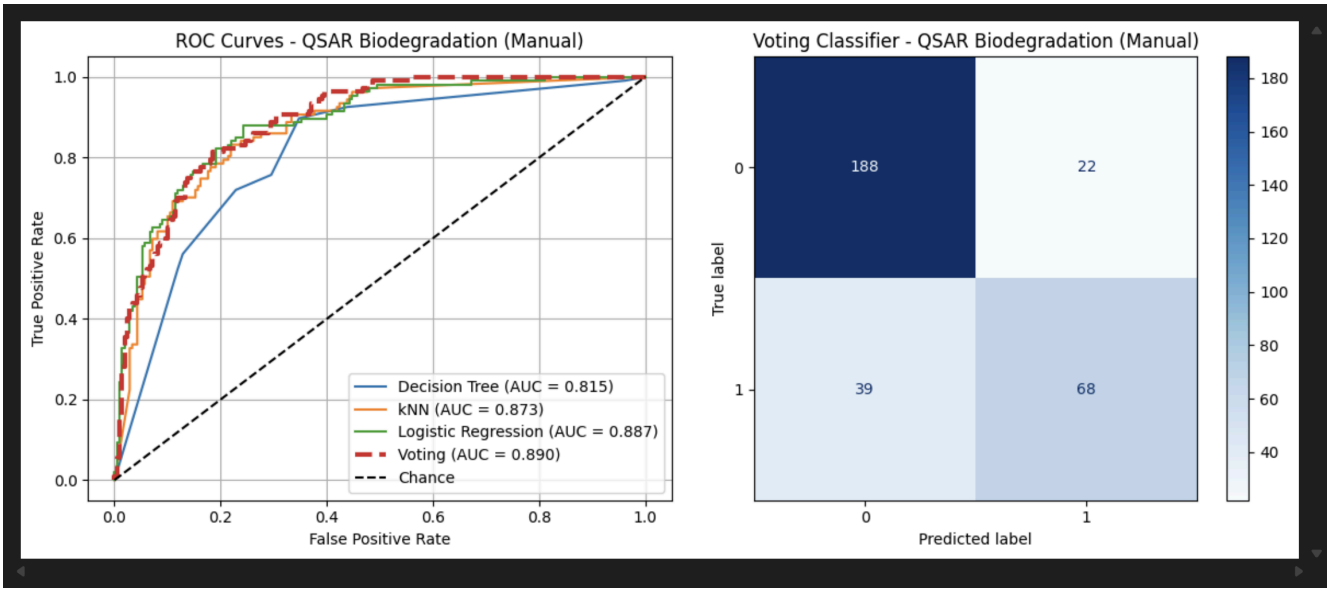
Metric	Score
ROC AUC	0.8898

kNN performed best individually (AUC = 0.8837).
Voting Classifier had the best overall test performance (AUC = 0.8898).

---

5. Screenshots





---

6. Conclusion

- Logistic Regression was best for **HR Attrition** dataset.
- kNN was best for **QSAR Biodegradation** dataset.
- Voting Classifier consistently outperformed individual models.

Key Takeaways:

- Model effectiveness depends on dataset characteristics.
- Ensembling improves robustness and predictive performance.
- Scikit-learn’s optimized libraries save time and reduce coding errors compared to manual implementations.
