Homework Assignment 1

Ethan Chung
University of Hawaii
ICS 635: Machine Learning
[redacted]@hawaii.edu

February 11, 2025

1 Task Description

The objective of this assignment is to implement and evaluate the performance of three machine learning models (K-Nearest Neighbors (KNN), Decision Trees (DT), and Random Forest (RF)) using the Breast Cancer dataset available within the scikit-learn library.

2 Model Description

K-Nearest Neighbors (KNN) is a supervised learning algorithm that classifies data points based on the majority class among their **k** nearest neighbors. KNN stores the training data and calculates distances during prediction. Decision Trees (DT) are tree-based models that recursively partition data based on feature values, and are easy to interpret but prone to overfitting. Random Forests (RF) are ensemble methods that combine multiple decision trees, which improve robustness and reduce overfitting through bagging and feature randomness.

3 Experiment Settings

3.1 Dataset Description

The Breast Cancer dataset, sourced from scikit-learn, is used in this assignment. It is a binary classification dataset containing 569 samples, each belonging to one of two classes: malignant (212 samples) or benign (357 samples). The dataset comprises 30 features that describe the cells, like radius, texture, perimeter, etc. The dataset was partitioned into training and testing sets using scikit-learn's `train_test_split` function with an 80/20 split for training and test set.

3.2 Experimental Setup

Three classifiers were created: KNN, DT, RF, initially with default parameters from scikit-learn. For KNN only, the features are first standardized by removing the mean and scaling to unit variance. Afterwards, all three models were fit on the training set, then evaluated on the test set. Hyper-parameters were tested following this configuration, with the 0th index as the default parameters:

```
knn_n_neighbors = [5, 10, 15, 20, 30]
dt_max_depth = [None, 1, 5, 10, 20]
rf_n_estimators = [100, 200, 300, 400, 500]
rf_max_depth = [None, 1, 5, 10, 20]
rf_min_samples_split = [2, 4, 6, 8, 10]
```

3.3 Evaluation Metrics

Accuracy, Precision, Recall, F1 score is used as the evaluation metrics. Accuracy is the proportion of correctly classified instances out of the total instances Precision is the proportion of correctly classified positive instances out of the total instances classified as positive, which helps measure the accuracy of positive predictions. Recall is the proportion of actual positive instances out of the total instances correctly classified as positive and falsely classified as negative. F1 score is the harmonic mean between precision and recall. A confusion matrix is generated to visualize the TP/FP/TN/FN rates. Using these four metrics together is good for binary classifications that also contain an imbalanced dataset.

3.4 Source Code

`https://github.com/echung32/ics635/tree/main/hw1`

3.5 Model Performance

Table 1 summarizes the performance of the three models, including the results obtained after tuning various hyperparameters (see Section 3.6 for details). The performance achieved using scikit-learn's default hyperparameters is indicated in *italics*, where Random Forest demonstrated the best performance with an accuracy of 93.85%, while KNN achieved 96.49% and Decision Tree 87.71%.

3.6 Ablation Studies

Table 1 includes the hyper-parameter tuning experiment results.

- **KNN:** Larger values of `n_neighbors` had a negative impact on accuracy, likely due to the inclusion of less relevant neighbors in the classification. Thus, the highest accuracy achieved was on the default model.
- **Decision Tree:** Adjusting `max_depth` had limited effect, until `max_depth=20` which improved accuracy by 3.51% compared to the default model.
- **Random Forest:** The best performance was achieved with `n_estimators=400, max_depth=10, min_samples_split=8`. However, because these three hyperparameters were varied simultaneously, it is difficult to isolate the individual impact of each on the observed performance improvement.

4 Conclusion

This report evaluated KNN, Decision Tree, and Random Forest models on the Breast Cancer dataset using accuracy, precision, recall, and F1-score. Initial runs with default hyperparameters showed Random Forest performing best (93.85% accuracy), followed by KNN (96.49%) and then Decision Tree (87.71%). Hyperparameter tuning revealed that larger `n_neighbors` for KNN decreased accuracy, while increasing `max_depth` for Decision Tree improved performance, with the best model achieving 91.22% accuracy. Random Forest also benefited from tuning `n_estimators, max_depth, min_samples_split`, with the best model achieving 95.61% accuracy.

While hyperparameter tuning improved the performance of both the Decision Tree and Random Forest models, the default KNN model achieved the best accuracy of 96.49% on this dataset. This suggests that, for this specific dataset, the simpler KNN model generalized well to unseen data.

5 Figures & Tables

Model	Accuracy	Precision	Recall	F1-Score
KNN				
n_neighbors=5	0.964912	0.968750	0.962963	0.964640
n_neighbors=10	0.956140	0.961538	0.953704	0.955728
n_neighbors=15	0.938596	0.947761	0.935185	0.937787
n_neighbors=20	0.929825	0.941176	0.925926	0.928750
n_neighbors=30	0.929825	0.941176	0.925926	0.928750
Decision Tree				
max_depth=None	0.877193	0.877792	0.875926	0.876585
max_depth=1	0.850877	0.857256	0.847222	0.848912
max_depth=5	0.877193	0.877792	0.875926	0.876585
max_depth=10	0.877193	0.877792	0.875926	0.876585
max_depth=20	0.912281	0.912037	0.912037	0.912037
Random Forest				
n_estimators=100, *max_depth=None,* *min_samples_split=2*	0.938596	0.938213	0.938889	0.938478

n_estimators=200,
max_depth=1,
min_samples_split=4
	0.929825	0.930831	0.928704	0.929477

n_estimators=300,
max_depth=5,
min_samples_split=6
	0.938596	0.938911	0.937963	0.938364

n_estimators=400, **max_depth=10,** **min_samples_split=8**	0.956140	0.956542	0.955556	0.955974

n_estimators=500,
max_depth=20,
min_samples_split=10
	0.938596	0.938911	0.937963	0.938364

Table 1: Classification performance of various models at varying hyperparameters, as evaluated on the held-out test set. *Italicized* is the default parameters. **Bolded** are the best performing models.

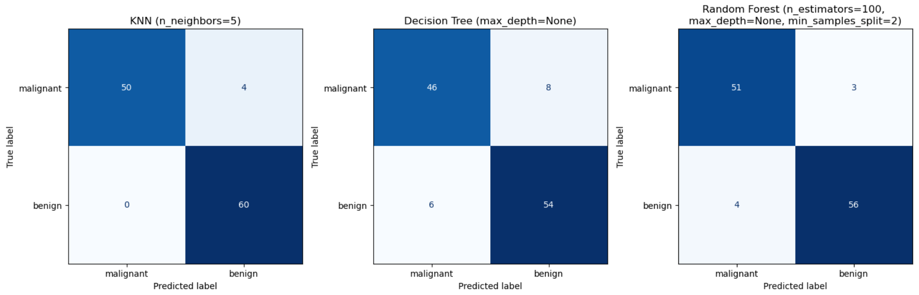

Figure 1: Default Model Performance

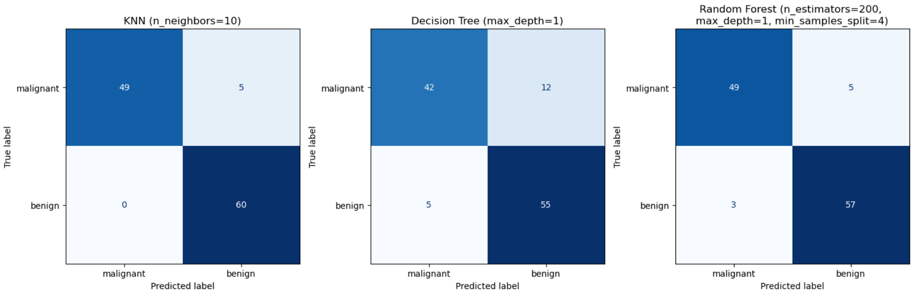

Figure 2: Ablation Model 1

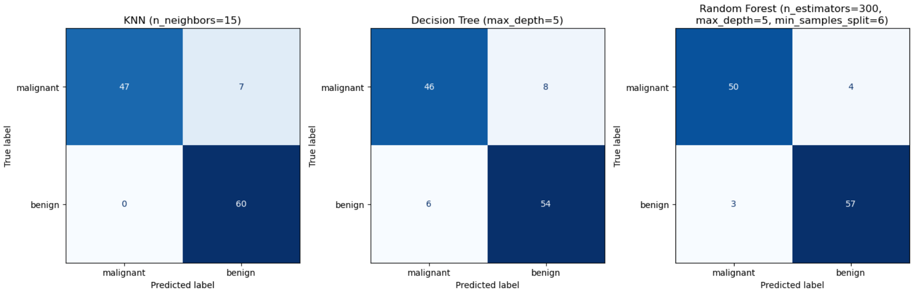

Figure 3: Ablation Model 2

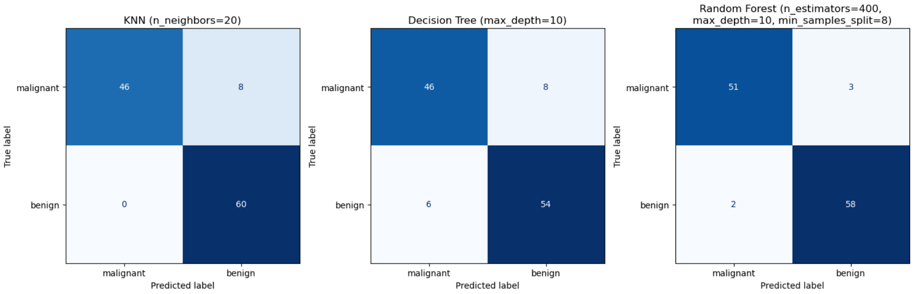

Figure 4: Ablation Model 3

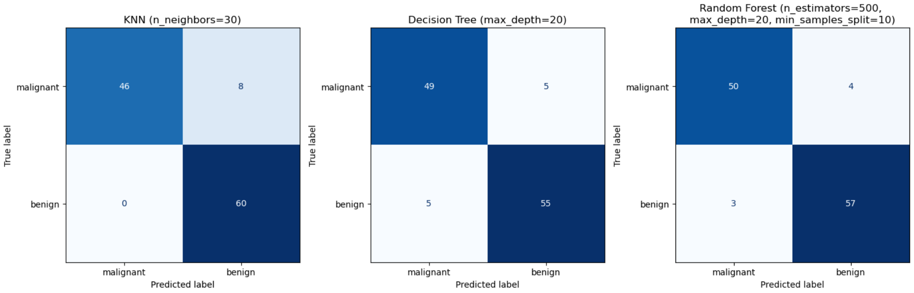

Figure 5: Ablation Model 4