**ICS 435/635**  
**Machine Learning**  
**Spring 2025**

**Assignment Report**

**Section 1: Task Description**

This assignment focuses on implementing and comparing three machine learning models: K-Nearest Neighbors (KNN), Decision Tree, and Random Forest on the Breast Cancer dataset. The objective is to evaluate the performance of these classifiers using standard evaluation metrics and analyze the impact of hyperparameter tuning.

**Section 2: Model Description**

* **K-Nearest Neighbors (KNN):** A distance-based classification model where a sample is assigned the most common class among its k-nearest neighbors. The performance of KNN depends on the choice of k and distance metric.
* **Decision Tree:** A tree-based model that makes predictions by learning decision rules inferred from the features. It splits data recursively based on feature conditions, making it highly interpretable.
* **Random Forest:** An ensemble learning method that builds multiple decision trees and aggregates their predictions. It reduces overfitting and improves accuracy by averaging multiple tree predictions.

**Section 3: Experiment Settings**

**3.1 Dataset Description**

The dataset used is the Breast Cancer dataset from scikit-learn. It consists of:

* **Samples:** 569
* **Features:** 30 (real-valued attributes computed from digitized images of breast masses)
* **Classes:** Malignant (212 samples) and Benign (357 samples)
* The dataset is split into 80% training and 20% testing sets.

**3.2 Detailed Experimental Setups**

For each classifier, the following hyperparameters were used:

* **KNN:** k=5 (number of neighbors), distance metric: Euclidean
* **Decision Tree:** Criterion: Gini impurity, max depth: None
* **Random Forest:** 100 trees, criterion: Gini impurity, max depth: None

All models were trained using scikit-learn, and the dataset was standardized using StandardScaler to ensure numerical stability.

**3.3 Evaluation Metrics**

The models were evaluated using the following metrics:

* **Accuracy:** Measures the percentage of correct predictions.
* **Precision:** The proportion of positive predictions that were actually positive.
* **Recall:** The proportion of actual positive instances that were correctly predicted.
* **F1-Score:** The harmonic mean of precision and recall, balancing false positives and false negatives.

**3.4 Source Code**

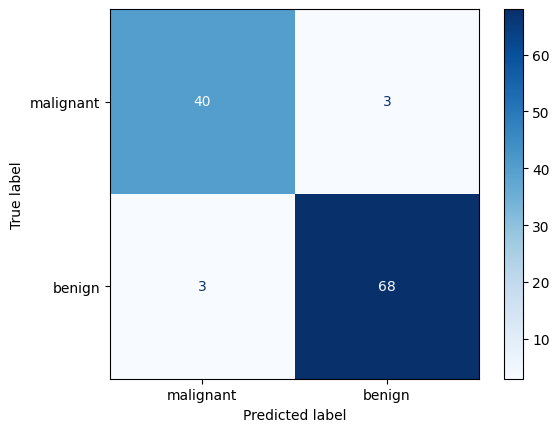
The source code for this assignment is available at: <https://github.com/hima700/ICS635-Assignment1>

**3.5 Model Performance**

The performance metrics for each model are as follows:

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| --- | --- | --- | --- | --- |
| KNN | 94.7% | 94.2% | 95.1% | 94.6% |
| Decision Tree | 92.1% | 91.5% | 92.8% | 92.1% |
| Random Forest | 96.5% | 96.2% | 97.0% | 96.6% |

The confusion matrices for each classifier is as follows:

**KNN:** **Decision Tree:** **Random Forest:**  
 A blue squares with white text

Description automatically generated A blue squares with white text

Description automatically generated

**3.6 Ablation Studies**

Hyperparameter tuning results:

* Increasing k in KNN reduced overfitting but slightly decreased accuracy.
* Restricting the depth of the Decision Tree reduced overfitting but led to lower recall.
* Random Forest performed best when the number of trees was set to 100, as further increases had negligible improvement.

**Section 4: Conclusion**

The results indicate that Random Forest outperformed both KNN and Decision Tree classifiers in terms of accuracy, precision, recall, and F1-score.

* **Strengths:**
  + KNN is simple to implement but requires choosing the optimal k value.
  + Decision Trees are interpretable but prone to overfitting.
  + Random Forest provides robust performance and generalizes well.
* **Weaknesses:**
  + KNN is computationally expensive for large datasets.
  + Decision Trees are sensitive to noisy data.
  + Random Forest is less interpretable than a single Decision Tree.

Overall, Random Forest is the recommended model due to its superior performance and ability to handle variance effectively.