CSE 572: Data Mining

Homework 2 Report

Name: Mohak Sharma

ASU ID: 1233869106

Date of Submission: October 20, 2025

Task 1 – Titanic Classification Challenge

The goal of this task was to predict passenger survival in the Titanic dataset using Decision Tree and Random Forest classifiers. The dataset contains both numerical and categorical features, which were preprocessed before training.

Data Preprocessing

Missing values in `Age` and `Embarked` were handled using the median and mode respectively. Categorical features were converted into dummy variables, and irrelevant columns such as `Name`, `Ticket`, `Cabin`, and `PassengerID` were removed. The cleaned dataset was then split into training and testing subsets for evaluation.

Model Training and Evaluation

Both models were trained on an 80–20 split of the data, followed by 5-fold cross-validation. The results are summarized below.

- **Decision Tree (5-Fold Mean Accuracy):** 0.8193
- **Random Forest (5-Fold Mean Accuracy):** 0.8271
- **Test Accuracy:** 0.7989

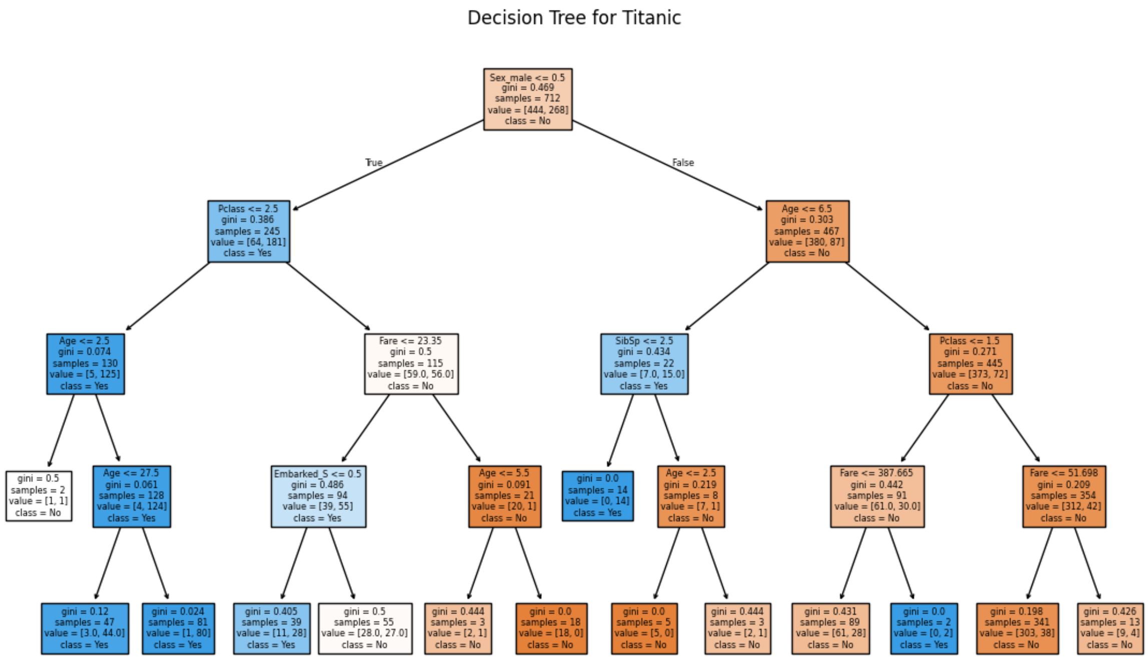

Figure 1: Decision Tree for Titanic Dataset.

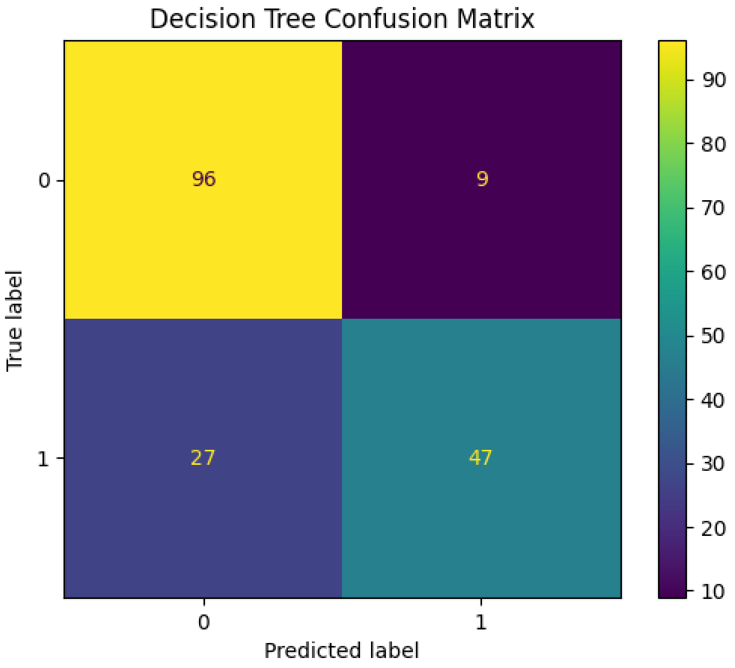

Figure 2: Decision Tree Confusion Matrix.

The Random Forest performed slightly better than the Decision Tree because ensemble learning reduces variance by averaging the outputs of multiple independent trees. This results in a more robust and generalized model. The confusion matrix confirms that the model correctly predicts most survivors, with a few misclassifications due to overlapping passenger characteristics.

Task 2 – Understanding Training Error and Testing

Each leaf in a decision tree predicts the majority class of the samples reaching it. The training error for the tree is calculated by summing the misclassified samples (the smaller class in each leaf) and dividing that total by the number of training records.
From the figure:

$$\begin{aligned} D(0) &: (+14, -5) \Rightarrow 5 \text{ errors}, \\ D(1) &: (+6, -7) \Rightarrow 6 \text{ errors}, \\ E(0) &: (+2, -10) \Rightarrow 2 \text{ errors}, \\ E(1) &: (+8, -6) \Rightarrow 6 \text{ errors}, \\ C(0) &: (+5, -17) \Rightarrow 5 \text{ errors}, \\ C(1) &: (+15, -5) \Rightarrow 5 \text{ errors}. \end{aligned}$$

Total errors = 29, total samples = 100.

$$\text{Training Error Rate} = \frac{29}{100} = 0.29 = 29\%$$

Hence, the model correctly classifies 71 percent of the training data.

For the test instance $T = \{A = 0, B = 1, C = 1, D = 1, E = 0\}$, we trace the path: $A = 0 \to B$, $B = 1 \to E$, $E = 0$, which leads to the leaf $(+2, -10)$. The majority class at this leaf is negative, so the model predicts **negative** for this instance.

Task 3 – Understanding the Splitting Process

The dataset contains ten records with four positive and six negative samples. The overall Gini impurity before any split is:

$$Gini_{parent} = 1 - (p^2 + q^2) = 1 - (0.4^2 + 0.6^2) = 0.48$$

Split on A: For $A = T$: 7 records $(+4, -3)$, $Gini = 1 - (4/7)^2 - (3/7)^2 = 0.4898$ For $A = F$:

3 records $(+0, -3)$, $Gini = 0$ Weighted impurity $= (7/10) \times 0.4898 + (3/10) \times 0 = 0.3429$ Gain $= 0.48 - 0.3429 = 0.1371$

Split on B: For $B = T$: 4 records $(+3, -1)$, $Gini = 0.375$ For $B = F$: 6 records $(+1, -5)$, $Gini = 0.2778$ Weighted impurity $= (4/10) \times 0.375 + (6/10) \times 0.2778 = 0.3167$ Gain $= 0.48 - 0.3167 = 0.1633$

Since $Gain(B) > Gain(A)$, the decision tree chooses attribute B for the first split because it achieves a greater reduction in impurity.

Task 4 – Decision Tree Properties

Q1: Are decision trees linear classifiers? Decision trees are not linear classifiers. They create axis-aligned splits that partition the data space into rectangles rather than forming straight lines or planes. This enables them to learn complex and non-linear relationships between features. As a result, they perform well even when the underlying data cannot be separated by a single linear boundary.

Q2: Is the misclassification error better than the Gini index? Misclassification error is not as effective because it only changes when the majority class label in a node flips. It cannot capture small improvements in node purity, which limits its usefulness during training. The Gini index, on the other hand, is more sensitive to distribution changes and helps the model find splits that gradually improve purity, leading to more balanced and accurate trees.

Task 5 – Bagging and Random Forests

The main weakness of bagging is that it can produce correlated models. Although it reduces variance by combining multiple bootstrap samples, if each base model focuses on the same dominant features, their predictions will still be similar. This correlation limits the overall benefit of averaging.

Random forests overcome this limitation by adding feature randomness. During training, each tree in the ensemble considers only a random subset of features at each split. This prevents all trees from relying on the same variables and encourages diversity. The result is a collection of less correlated models, which improves stability, lowers variance, and enhances predictive accuracy compared to plain bagging.

Task 6 – Support Vector Machine and Margin

Given four input points and their labels:

$$[-1,-1] \to -, \quad [-1,1] \to +, \quad [1,-1] \to +, \quad [1,1] \to -$$

We map them into a new space using $\phi(x) = (x_1, x_1x_2)$. In this transformed space, the data becomes linearly separable along the second dimension $z_2 = x_1x_2$. The decision boundary is $z_2 = 0$, and the distance from this boundary to the nearest support vectors is one unit. Thus, the margin is **1**. This example shows that with a suitable feature mapping, an SVM can transform non-linear relationships into a linearly separable form.

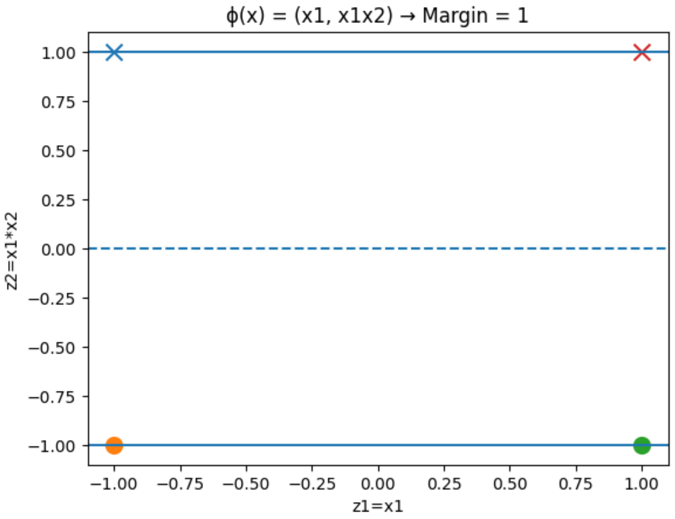

Figure 3: SVM Feature Mapping $\phi(x)$ and Margin = 1.

Task 7 – Linear Separability of Circles

The equation of a circle is:

$$(x_1 - a)^2 + (x_2 - b)^2 - r^2 = 0$$

Expanding gives:

$$x_1^2 + x_2^2 - 2ax_1 - 2bx_2 + (a^2 + b^2 - r^2) = 0$$

By mapping data into the feature space $[x_1, x_2, x_1^2, x_2^2, 1]$, the circular relationship becomes linear in these new dimensions. Therefore, any circular boundary becomes linearly separable in this quadratic feature space. This concept illustrates how non-linear patterns can be handled effectively through proper feature transformations.

Task 8 – Ellipses and Polynomial Kernels

The general equation of an ellipse is:

$$c(x_1 - a)^2 + d(x_2 - b)^2 - 1 = 0$$

Expanding gives:

$$cx_1^2 + dx_2^2 - 2acx_1 - 2bdx_2 + (ca^2 + db^2 - 1) = 0$$

An SVM using a polynomial kernel of degree two,

$$K(u, v) = (1 + u \cdot v)^2,$$

implicitly maps data to a feature space $[1, x_1, x_2, x_1^2, x_2^2, x_1x_2]$. In this transformed space, the ellipse equation becomes a linear relation, allowing the SVM to separate any elliptic region using a simple linear boundary. This demonstrates the power of kernel functions in converting non-linear patterns into linearly separable ones.

Code Repository Link

All source code, Jupyter notebook, and generated plots for this homework are available at:

github.com/mohaksharma2507/CSE572_HW2