Experiment: Classification Analysis using Decision Tree and Random Forest

1. Dataset Source

- **Dataset Name:** Airline Passenger Satisfaction Dataset
- **Source:** Kaggle - Airline Passenger Satisfaction
- **Origin:** This dataset contains an airline passenger satisfaction survey. It is widely used to study the factors that lead to customer satisfaction or dissatisfaction in the aviation industry.

2. Dataset Description

The dataset aims to predict whether a future customer will be satisfied with their service based on historical data.

- **Size:** The uploaded file (`test.csv`) contains **25,976 instances** (rows) and **25 features** (columns).
- **Target Variable:**
  - `satisfaction`: A categorical variable with two classes: `'satisfied'` and `'neutral or dissatisfied'`. This makes it a binary classification problem.
- **Key Features:**
  - **Categorical:** `Gender`, `Customer Type` (Loyal/Disloyal), `Type of Travel` (Business/Personal), `Class` (Business, Eco, Eco Plus).
  - **Numerical (Continuous):** `Age`, `Flight Distance`, `Departure Delay in Minutes`, `Arrival Delay in Minutes`.
  - **Numerical (Ordinal/Likert Scale 0-5):** Various satisfaction factors including `Inflight wifi service`, `Departure/Arrival time convenient`, `Ease of Online booking`, `Gate location`, `Food and drink`, `Online boarding`, `Seat comfort`, `Inflight entertainment`, `On-board service`, `Leg room service`, `Baggage handling`, `Checkin service`, `Inflight service`, and `Cleanliness`.

3. Mathematical Formulation of the Algorithm

A. Decision Tree Classifier

A Decision Tree splits the data into subsets based on the value of input features. The goal is to create "pure" leaf nodes (containing only one class).

Splitting Criteria: The algorithm selects the best feature to split the data by maximizing **Information Gain** or minimizing **Gini Impurity**.

1. **Entropy (H):** A measure of randomness or disorder in the dataset S.
   $H(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$
   Where p_i is the probability of an instance belonging to class i, and c is the number of classes.
2. **Information Gain (IG):** The reduction in entropy after splitting dataset S on attribute A.
   $IG(S,A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)$
   Where S_v is the subset of S for which attribute A has value v.
3. **Gini Impurity (G):** An alternative to entropy (often faster to compute). It measures the likelihood of an incorrect classification of a new instance if it was randomly classified according to the distribution of class labels.
   $G(S) = 1 - \sum_{i=1}^{c} (p_i)^2$

B. Random Forest Classifier

Random Forest is an ensemble learning method that constructs a multitude of decision trees at training time.

1. **Bootstrapping (Bagging):** Given a training set X of size n, Random Forest generates B new training sets $X_1, ... X_B$ by sampling with replacement (bootstrap samples).
2. **Random Feature Selection:** When splitting a node during the construction of the tree, the split is chosen from a random subset of k features, rather than evaluating all features. This reduces correlation between trees.
3. **Aggregation (Voting):** For classification, the final prediction $\hat{Y}$ is the mode (majority vote) of the classes predicted by individual trees.
   $\hat{Y} = mode\{h_1(x), h_2(x), ..., h_B(x)\}$
   Where $h_b(x)$ is the prediction of the b-th tree.

4. Algorithm Limitations

Decision Trees:

- **Overfitting:** Decision trees can create overly complex trees that memorize the training data but fail to generalize to new data. This is handled by pruning or setting a maximum depth.

- **Instability:** Small variations in the data can result in a completely different tree being generated.
- **Bias:** Trees can be biased if one class dominates. (The dataset should be balanced before training).

Random Forest:

- **Computational Cost:** Random Forests create many trees (often 100+), making them slower to train and predict compared to a single Decision Tree.
- **Interpretability:** Unlike a single Decision Tree, which is easy to visualize and explain ("If X > 5, then Class A"), a Random Forest is a "black box" model because it aggregates hundreds of decisions.
- **Memory Usage:** Storing a large ensemble of trees requires significant memory.

5. Methodology / Workflow

The experiment follows this standard Machine Learning pipeline:

1. **Data Preprocessing:**
    - **Handling Missing Values:** Impute missing values in the `Arrival Delay in Minutes` column (e.g., using the mean or median).
    - **Encoding:** Convert categorical variables (`Gender`, `Customer Type`, `Type of Travel`, `Class`) into numerical format using **One-Hot Encoding** or **Label Encoding**. Target variable `satisfaction` is mapped to binary (0/1).
    - **Scaling:** While trees are scale-invariant, normalization (e.g., StandardScaler) ensures consistency, especially if comparison with other algorithms is needed later.
2. **Data Splitting:**
    - Split the dataset into a **Training Set (80%)** for model building and a **Testing Set (20%)** for evaluation.
3. **Model Training:**
    - Initialize the **Decision Tree Classifier** (e.g., using Gini impurity).
    - Initialize the **Random Forest Classifier** (e.g., with 100 estimators).
    - Fit both models to the Training Set.
4. **Prediction:**
    - Use the trained models to predict labels for the Testing Set.
5. **Evaluation:**
    - Compare predicted labels against actual labels using performance metrics.

6. Performance Analysis

To evaluate the models, the following metrics are analyzed:

1. **Accuracy:** The ratio of correctly predicted observations to the total observations. Accuracy=TP+TN+FP+FNTP+TN
2. **Confusion Matrix:** A table showing True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). This reveals if the model is confusing one class for another.
3. **Precision & Recall:** Important if false positives/negatives have different costs.
    - Precision=TP+FPTP
    - Recall=TP+FNTP
4. **F1-Score:** The harmonic mean of Precision and Recall, useful if the dataset has an uneven class distribution.
   F1=2⋅Precision+RecallPrecision⋅Recall

Interpretation: Typically, Random Forest is expected to outperform a single Decision Tree in terms of Accuracy and F1-Score because the ensemble approach reduces variance and overfitting, resulting in a more robust model.

7. Hyperparameter Tuning

Hyperparameter tuning optimizes the model configurations to improve performance. This is often done using **Grid Search** or **Random Search**.

For Decision Tree:

- **`max_depth`:** Controls the maximum depth of the tree. Limiting this prevents overfitting.
- **`min_samples_split`:** The minimum number of samples required to split an internal node. Higher values constrain the model.
- **`criterion`:** Choosing between `'gini'` and `'entropy'`.

For Random Forest:

- **`n_estimators`:** The number of trees in the forest. More trees generally increase performance but also computation time.
- **`max_features`:** The number of features to consider when looking for the best split.
- **`bootstrap`:** Whether bootstrap samples are used when building trees.