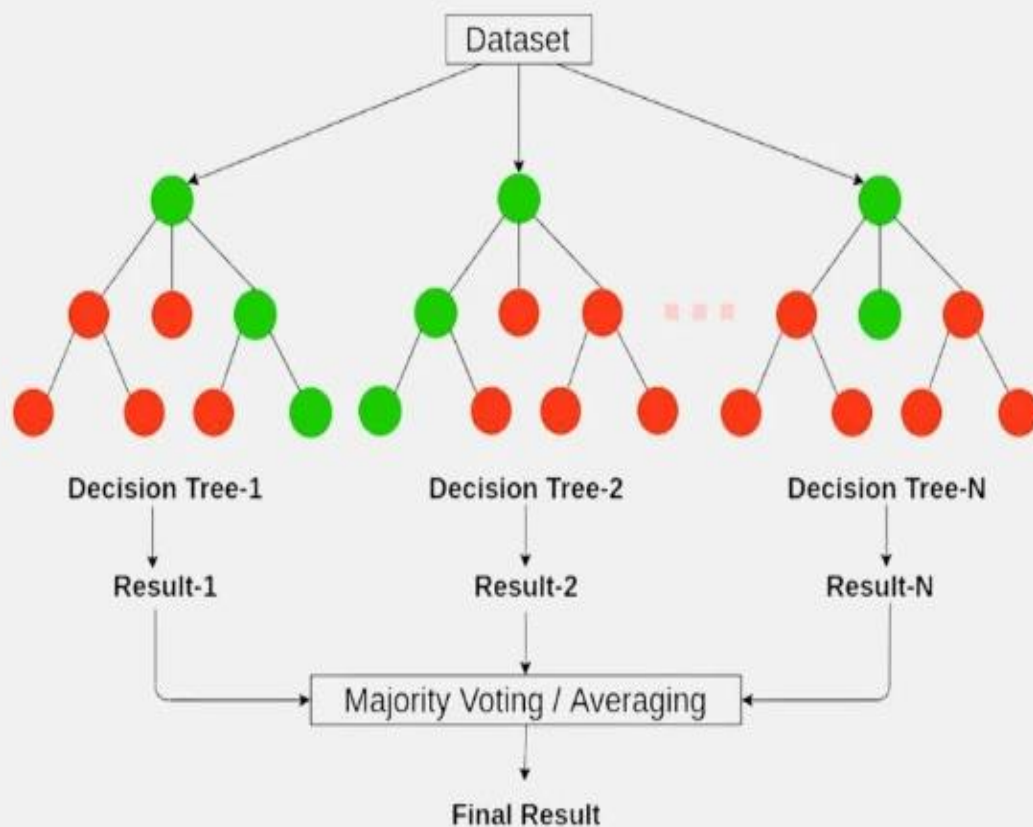
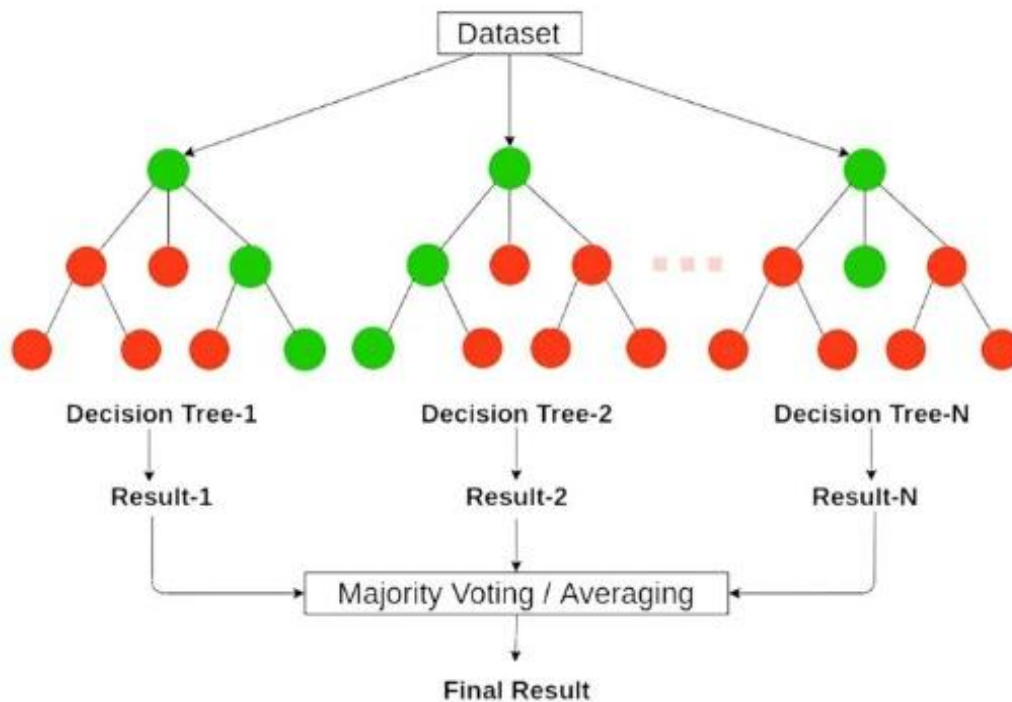


Random Forest

Explained with Code



What is Random Forest



Random Forest is an **ensemble learning technique** used in both **classification** and **regression** tasks. It operates by constructing a **multitude of decision trees** during the training phase and **outputs the mode** (for classification) **or mean prediction** (for regression) of the individual trees as the final prediction.

Real Life Example

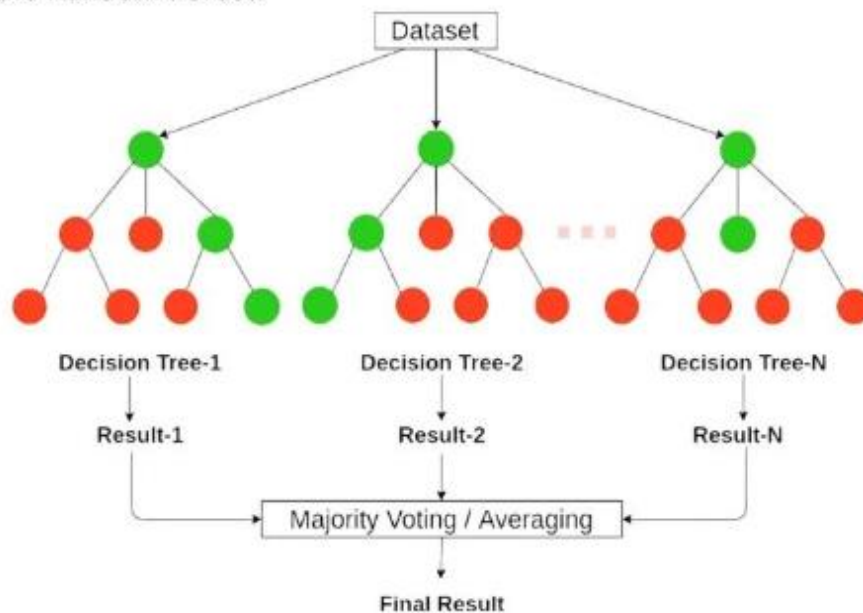
Imagine you have a big decision to make, and instead of asking just one person for advice, you decide to consult a group of diverse individuals. Each person has their own strengths and expertise, and by combining their opinions, you hope to make a better decision.



How it works

Random Forest works somewhat similarly in the world of machine learning:

1. **Decision Trees:** Think of each person as a "Decision Tree." A Decision Tree is like a flowchart with a series of yes/no questions that lead to a decision or prediction.
2. **Ensemble of Trees:** Instead of relying on just one Decision Tree, Random Forest creates a "forest" by combining many Decision Trees. Each tree is trained on a random subset of the data and makes its own predictions.
3. **Voting System:** When you have a new data point that needs a prediction, each Decision Tree in the forest gives its opinion. The most popular prediction (the one that gets the most votes) becomes the final prediction of the Random Forest.



How it works

The idea behind Random Forest is that **by combining the opinions of many individual decision-makers (trees), you can often make more accurate predictions** and avoid relying too much on the quirks or biases of a single decision-maker (tree). It's like taking a vote among a diverse group to make a collective, more robust decision.



Code example

```

# Import necessary libraries
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.datasets import load_iris

# Load the Iris dataset (a built-in dataset in scikit-learn)
iris = load_iris()
X = iris.data # Features
y = iris.target # Target variable (species)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Create a Random Forest classifier with 100 trees
rf_classifier = RandomForestClassifier(n_estimators=100,
random_state=42)

# Train the classifier on the training data
rf_classifier.fit(X_train, y_train)

# Make predictions on the testing data
y_pred = rf_classifier.predict(X_test)

# Evaluate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```