

Random Forest

Random Forest is an ensemble learning algorithm that combines multiple Decision Trees to improve accuracy and reduce overfitting. It is widely used for classification and regression tasks.

1.How Random Forest Works

1. **Creates multiple Decision Trees** from random subsets of data.
2. Each tree makes an **independent prediction**.
3. The **final prediction** is made by:
 - **Classification** → Majority vote (most common class).
 - **Regression** → Average of all predictions.

- ✓ **More trees = Better performance**
 - ✓ **Reduces overfitting** compared to a single Decision Tree
 - ✓ **Handles missing values and noisy data well**
-

2.Random Forest Implementation (Classification)

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Sample dataset (features & labels)
X = [[5, 120, 70], [2, 140, 80], [8, 130, 75], [6, 125, 72], [3, 145, 85]]
```

```
y = [0, 1, 0, 0, 1] # Labels (0 = No Disease, 1 = Disease)

# Split data into training & test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Initialize & train Random Forest model
rf_model = RandomForestClassifier(n_estimators=100, random_state=42) # 100 trees
rf_model.fit(X_train, y_train)

# Make predictions
predictions = rf_model.predict(X_test)

# Check accuracy
accuracy = accuracy_score(y_test, predictions)

print(f"Random Forest Accuracy: {accuracy * 100:.2f}%")
```

3.Key Hyperparameters in RandomForestClassifier()

Parameter	Description
n_estimators	Number of trees (default = 100)
max_depth	Maximum depth of each tree (prevents overfitting)
max_features	Number of features per split (auto = sqrt(features))
bootstrap	Whether to use random sampling (default = True)
random_state	Ensures reproducibility

4.Random Forest vs. Decision Tree

Feature	Decision Tree	Random Forest
Overfitting	High risk	Low risk
Stability	Unstable (sensitive to data)	More stable
Performance	Can be good	Usually better
Training Speed	Faster	Slower (more trees)

6.When to Use Random Forest?

When you want **higher accuracy** than a single Decision Tree.

When you have **a lot of data** and need a **robust model**.

When you want to **reduce overfitting** in a complex dataset.