

#100DaysOfMLCode

Day 33

©Avik Jain

# RANDOM FOREST

## AN INTUITION TO RANDOM FOREST

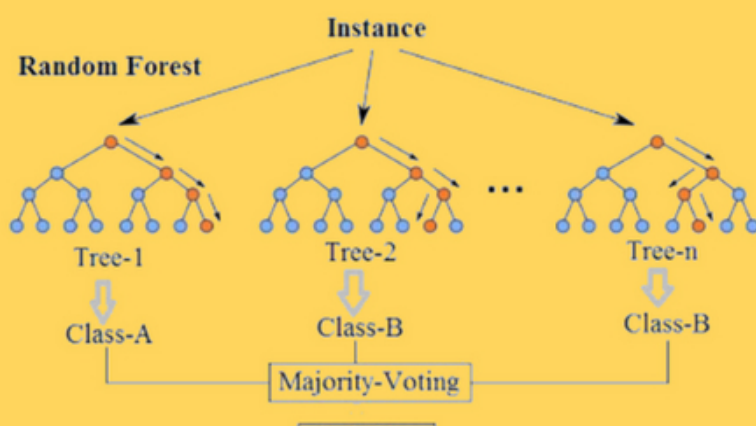
**RANDOM FORESTS ARE SUPERVISED ENSEMBLE-LEARNING MODELS USED FOR CLASSIFICATION AND REGRESSION.**

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

## WHAT IS THE RANDOM FOREST ALGORITHM?

Ensemble learning models aggregate multiple machine learning models, allowing for overall better performance.

The logic behind this is that each of the models used is weak when employed on its own, but strong when put together in an ensemble. In the case of Random Forests, a large number of Decision Trees, acting as the “weak” factors, are used and their outputs are aggregated, with the result representing the “strong” ensemble.



There are two steps in the Random Forest algorithm, one is random forest creation, the other is to make a prediction from the random forest classifier created in the first step.

**THE DIFFERENCE BETWEEN THE RANDOM FOREST ALGORITHM AND THE DECISION TREE ALGORITHM IS THAT IN RANDOM FOREST, THE PROCESSES OF FINDING THE ROOT NODE AND SPLITTING THE FEATURE NODES WILL RUN RANDOMLY.**

## HOW DOES IT WORK?



### CREATION

Each tree is grown as follows:

1. If the number of cases in the training set is  $N$ , sample  $N$  cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree.
2. If there are  $M$  input variables, a number is specified such that at each node,  $m$  variables are selected at random out of the  $M$  and the best split on this  $m$  is used to split the node.

### PREDICTION

The random forest prediction is broken down in the below steps :

1. Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)
2. Calculate the votes for each predicted target
3. Consider the high voted predicted target as the final prediction from the random forest algorithm

**CHECK OUT THE REPOSITORY AT – [GITHUB.COM/AVIK-JAIN/100-DAYS-OF-ML-CODE](https://github.com/Avik-Jain/100-Days-Of-ML-Code)**

Follow Me For More Updates



### 🔗 Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

### 🔗 Importing the dataset

```
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
```

## 🔗 Splitting the dataset into the Training set and Test set

```
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25,
random_state = 0)
```

## 🔗 Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

## 🔗 Fitting Random Forest to the Training set

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy',
random_state = 0)
classifier.fit(X_train, y_train)
```

## 🔗 Predicting the Test set results

```
y_pred = classifier.predict(X_test)
```

## 🔗 Making the Confusion Matrix

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

## 🔗 Visualising the Training set results

```
from matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:,
0].max() + 1, step = 0.01),
                    np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:,
1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(),
X2.ravel()]).T).reshape(X1.shape),
            alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
```

```

plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
            c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Random Forest Classification (Training set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()

```

## 🔗 Visualising the Test set results

```

from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:,
0].max() + 1, step = 0.01),
                    np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:,
1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(),
X2.ravel()]).T).reshape(X1.shape),
            alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Random Forest Classification (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()

```