# **Random Forest**

Random forests are popularly applied to both data science competitions and practical problems. They are often accurate, do not require feature scaling, categorical feature encoding, and need little parameter tuning. They can also be more interpretable than other complex models such as neural networks.

A random forest consists of multiple random decision trees. Two types of randomness’s are built into the trees. First, each tree is built on a random sample from the original data. Second, at each tree node, a subset of features is randomly selected to generate the best split.

**Step1:**

First, start with the selection of random samples from a given dataset.

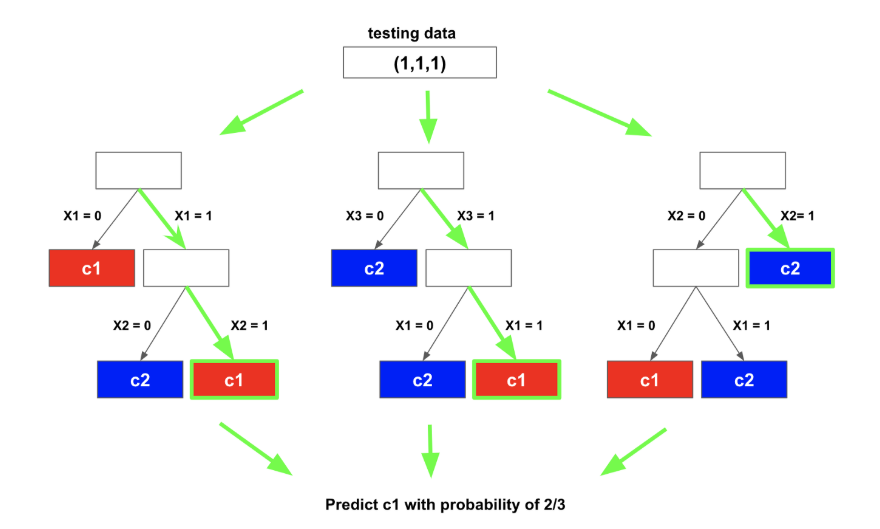
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | TimeOfAcc | AccLocation | NatureAccident | Classification  Of  Accident | Causes | ……. |
| 13-01-2018 | 1:50 AM | 162.050KM,OTH,LHS | 7 | 1 | 4 | ……. |
| 12/1/2018 | 3:12 AM | 153.900KM,MCW,LHS | 2 | 1 | 2 | ……. |
| 12/1/2018 | 4:50 AM | 112.600KM LHS | 7 | 4 | 2 | ……. |
| 12/1/2018 | 9:10 AM | 142.300KM,MCW,RHS | 7 | 4 | 2 | ……. |
| 20-10-2015 | 5:00 AM | 128+000 | 2 | 1 | 3 | ……. |
| 15-10-2015 | 9:00 AM | 125+000 | 2 | 1 | 2 | ……. |
| 8/9/2015 | 5:00 AM |  | 1 | 2 | 2 | ……. |
| 14-08-2015 | 9:00 AM | 86+000 | 7 | 1 | 2 | ……. |
| 14-08-2015 | 5:00 AM | 86+000 | 1 | 1 | 2 | ……. |

You can then capture this data in Python using [**pandas Data Frame**](https://datatofish.com/create-pandas-dataframe/):

data = pd.read\_csv("Accident\_data.csv")

**Step2:**

Next, this algorithm will construct a decision tree by specifying the condition for every sample. Then it will get the prediction result from every decision tree.



**Step3:**

Assign each data point to the closest cluster by calculating its distance with respect to each centroid

Finding the centroids

Once you created the DataFrame based on the above data, you’ll need to import 2 additional Python modules:

* **matplotlib** – for [creating charts in Python](https://datatofish.com/scatter-line-bar-charts-using-matplotlib/)
* **sklearn** – for applying the K-Means Clustering in Python

**Step 4:**

Compute cluster centroids: The centroid of data points is calculated by considering all parameters.

In this step, voting will be performed for every predicted result.

**Step 5:**

Re-assign each point to the closest cluster centroid: Note that only the data point at the bottom is assigned to the cluster even though its closer to the centroid of the cluster. Thus, we assign that data point into cluster

**Step 6 :**

At last, select the most voted prediction result as the final prediction result.

**K-Means Clustering**

K-Means Clustering is a concept that falls under Unsupervised Learning. This algorithm can be used to find groups within unlabelled data.

**Step1:**

Creating the Data Frame for two-dimensional dataset

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | TimeOfAcc | AccLocation | NatureAccident | Classification  Of  Accident | Causes | ……. |
| 13-01-2018 | 1:50 AM | 162.050KM,OTH,LHS | 7 | 1 | 4 | ……. |
| 12/1/2018 | 3:12 AM | 153.900KM,MCW,LHS | 2 | 1 | 2 | ……. |
| 12/1/2018 | 4:50 AM | 112.600KM LHS | 7 | 4 | 2 | ……. |
| 12/1/2018 | 9:10 AM | 142.300KM,MCW,RHS | 7 | 4 | 2 | ……. |
| 20-10-2015 | 5:00 AM | 128+000 | 2 | 1 | 3 | ……. |
| 15-10-2015 | 9:00 AM | 125+000 | 2 | 1 | 2 | ……. |
| 8/9/2015 | 5:00 AM |  | 1 | 2 | 2 | ……. |
| 14-08-2015 | 9:00 AM | 86+000 | 7 | 1 | 2 | ……. |
| 14-08-2015 | 5:00 AM | 86+000 | 1 | 1 | 2 | ……. |

You can then capture this data in Python using [**pandas Data Frame**](https://datatofish.com/create-pandas-dataframe/):

data = pd.read\_csv("Accident\_data.csv")

**Step2:**

Finding the centroids

Once you created the DataFrame based on the above data, you’ll need to import 2 additional Python modules:

* **matplotlib** – for [creating charts in Python](https://datatofish.com/scatter-line-bar-charts-using-matplotlib/)
* **[sklearn](https://scikit-learn.org/stable/" \t "_blank)** - to find the centroids
* **sklearn** – for applying the K-Means Clustering in Python

**Step3:**

Randomly assign each data point to a cluster: Let’s assign random points to a clusters

Compute cluster centroids: The centroid of data points are calculate using distance formula

**Step4:**

Re-assign each point to the closest cluster centroid.

Re-compute cluster centroids, Now re-computing the centroids for the clusters.

**Step5:**

Repeat steps 4 and 5 until no improvements are possible: Similarly, we’ll repeat the 4th and 5th steps until we’ll reach global optima. When there will be no further switching of data points between clusters for successive repeats. It will mark the termination of the algorithm if not explicitly mentioned.

**Decision Tree**

Decision Trees are a type of Supervised Machine Learning where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, decision tree algorithm can be used for solving regression and classification problems.



**Step1:**

Place the best attribute of the dataset at the root of the tree.

Each individual tree in the Decision tree spits out a class prediction and the class with the most votes becomes our model’s prediction

**Step2:**

Split the data set into subsets. Subsets should be made in such a way that each subset contains data with the same value for an attribute.ie Make that attribute a decision node and breaks the dataset into smaller subsets.

**Step3:**

In this step, voting will be performed for every predicted result from the data subset or we can say this step is to choose the attribute that gives us highest possible Information Gain

**Step3:**

Repeat step 1 and step 2 on each subset until you find leaf nodes in all the branches of the tree and select the most voted prediction result as the final prediction result.