

## Random Forest Algorithm

The following are the basic steps involved in performing the random forest algorithm:

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

```
import pandas as pd
```

```
import numpy as np
```

```
dataset = pd.read_csv('C:/Users/Admin/Desktop/petrol_consumption.csv')
```

```
dataset.head()
```

```
X = dataset.iloc[:, 0:4].values
```

```
y = dataset.iloc[:, 4].values
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
from sklearn.preprocessing import StandardScaler
```

```
sc = StandardScaler()
```

```
X_train = sc.fit_transform(X_train)
```

```
X_test = sc.transform(X_test)
```

```
from sklearn.ensemble import RandomForestRegressor
```

```
regressor = RandomForestRegressor(n_estimators=20, random_state=0)
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)

from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

## Random Forest for Classification

```
import pandas as pd

dataset = pd.read_csv('C:/Users/Admin/Desktop/bill_authentication.csv')

dataset.head()

X = dataset.iloc[:, 0:4].values
y = dataset.iloc[:, 4].values

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
classifier= RandomForestClassifier(n_estimators=20, random_state=0)  
classifier.fit(X_train, y_train)
```

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

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

```
print(confusion_matrix(y_test,y_pred))
```

```
print(classification_report(y_test,y_pred))
```

```
print(accuracy_score(y_test, y_pred))
```