

Practical No. 5

Title: Implementation of Bagging Algorithm: Random Forest

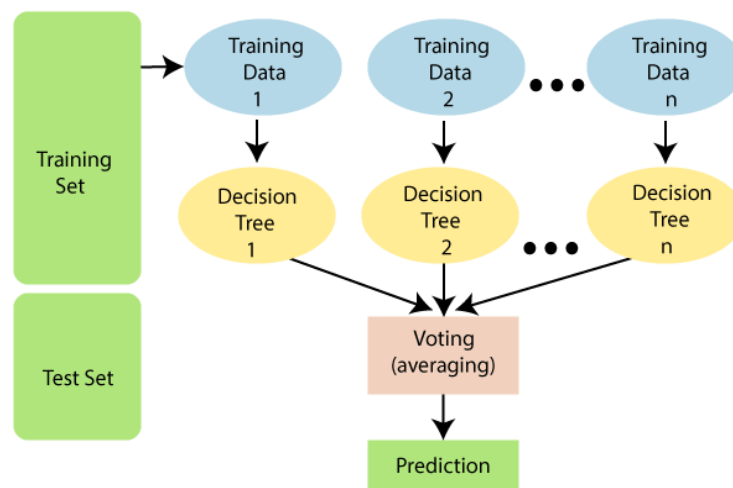
Aim: Understanding basics of Bagging Algorithm: Random Forest

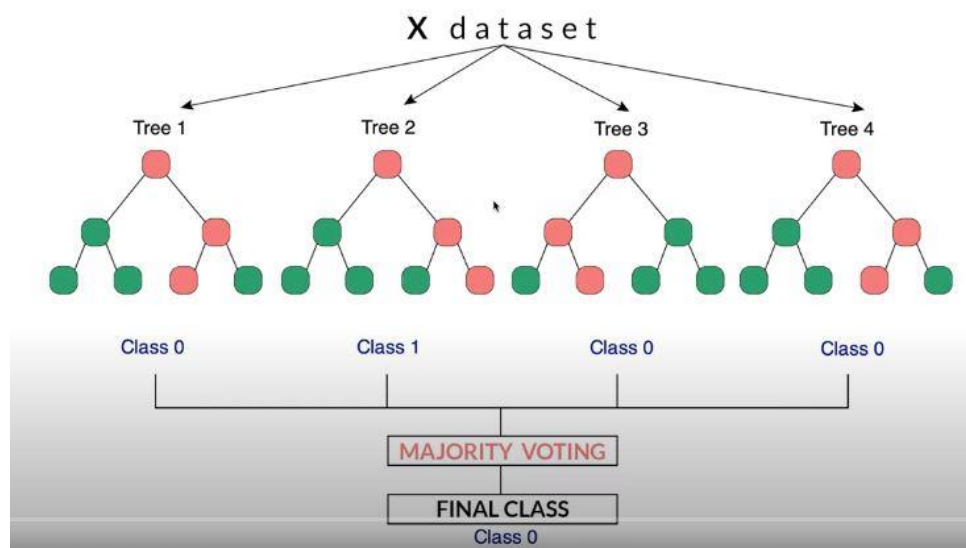
Introduction: Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. The below diagram explains the working of the Random Forest algorithm:





Exercise -

1. Using iris dataset properties like sepal length, sepal width, petal length and petal width, predict the class of flower.

Implementation:**Program And Output :**

```
from sklearn import tree
tree_model = tree.DecisionTreeRegressor()
tree_model.fit(inp_new,trgt)
```

```
DecisionTreeRegressor()
```

```
tree_model.predict([[2,1,1]])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but DecisionTreeRegressor
"X does not have valid feature names, but"
array([1.])
```

```
import pandas as pd
import numpy as np
url = 'http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
iris = pd.read_csv(url, names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'class'])
```

```
iris.head()
```

	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
from sklearn.model_selection import train_test_split
X_train,X_test, y_train, y_test = train_test_split(iris.drop(['class'],axis='columns'),iris[['class']],train_size=0.8)
```

```
from sklearn.ensemble import RandomForestClassifier
RFC = RandomForestClassifier()
RFC.fit(X_train,y_train)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: DataConversionWarning: A column-vector y was passed when a 1d array was
This is separate from the ipykernel package so we can avoid doing imports until
RandomForestClassifier()
```

```
RFC.predict([[4.8,3.0,1.5,0.3]])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but RandomForestClassifier
"X does not have valid feature names, but"
array(['Iris-setosa'], dtype=object)
```