@abhishekkaddipudi

RandomForest From Scratch

Using previously implemented DecisionTree

one

```python
from DecisionTreeClasifier import DecisionTreeClassifier
import numpy as np
from collections import Counter

class RandomForest:

    def  __init__(self,n_trees=10,max_depth=10,min_sample_split=2,n_features=None) -> None:
        self.n_features=n_features
        self.min_sample_split=min_sample_split
        self.max_depth=max_depth
        self.n_trees=n_trees
        self.Trees=[]

    def fit(self,X,y):
        n_sample=X.shape[1]
        for _ in range(self.n_trees):
            tree=DecisionTreeClassifier(self.n_features,self.min_sample_split,self.max_depth)
            idxs=np.random.choice(n_sample,n_sample,replace=True)
            X_sample,y_sample=X[idxs],y[idxs]
            tree.fit(X_sample,y_sample)
            self.Trees.append(tree)
```

two

```
    def most_common(self,y):
        return Counter(y).most_common(1)[0][0]


    def predict(self,X):
        predictions=np.array([tree.predict(X) for tree in self.Trees])
        tree_pred=np.swapaxes(predictions,0,1)
        return np.array([self.most_common(pred) for pred in tree_pred])
```

three

```python
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from RandomForest import RandomForest
dataset=load_breast_cancer()
X=dataset['data']
Y=dataset['target']
X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.1,random_state=1234)
for i in range(10,15):
    DR=RandomForest(max_depth=i)
    DR.fit(X_train,y_train)
    y_pred=DR.predict(X_test)
    from sklearn.metrics import accuracy_score
    print(accuracy_score(y_test,y_pred))

#0.9122807017543859
#0.9473684210526315
#0.9122807017543859
#0.9298245614035088
#0.9473684210526315
```