

IMPLEMENT ANY TWO ENSEMBLE CLASSIFIERS

ADABOOST

```
import pandas as pd
from sklearn import model_selection
from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score

df = pd.read_csv('iris.csv')
array = df.values

X = df.iloc[:, :-1]
y = df.iloc[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

```
from sklearn.ensemble import AdaBoostClassifier

seed = 10
num_trees = 15

model = AdaBoostClassifier(n_estimators=num_trees, random_state=seed)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

```

print('Accuracy = ',accuracy_score(y_pred, y_test))

y_true = y_test
print('\nConfusion Matrix: \n', confusion_matrix(y_true, y_pred))

matrix = classification_report(y_true,y_pred)
print('\nClassification report : \n',matrix)

```

Accuracy = 0.9

Confusion Matrix:

```

[[12  0  0]
 [ 0  5  2]
 [ 0  1 10]]

```

Classification report :

	precision	recall	f1-score	support
1	1.00	1.00	1.00	12
2	0.83	0.71	0.77	7
3	0.83	0.91	0.87	11
accuracy			0.90	30
macro avg	0.89	0.87	0.88	30
weighted avg	0.90	0.90	0.90	30

BAGGING CLASSIFIER

```

import pandas as pd
from sklearn import model_selection
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score

df = pd.read_csv('iris.csv')
array = df.values

X = df.iloc[:, :-1]
y = df.iloc[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

cart = DecisionTreeClassifier()
num_trees = 50
model = BaggingClassifier(base_estimator=cart, n_estimators=num_trees)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

```

```

print('Accuracy = ',accuracy_score(y_pred, y_test))

y_true = y_test
print('\nConfusion Matrix: \n', confusion_matrix(y_true, y_pred))

matrix = classification_report(y_true,y_pred)
print('\nClassification report : \n',matrix)

```

Accuracy = 0.9333333333333333

Confusion Matrix:

```

[[10  0  0]
 [ 0  7  1]
 [ 0  1 11]]

```

Classification report :

	precision	recall	f1-score	support
1	1.00	1.00	1.00	10
2	0.88	0.88	0.88	8
3	0.92	0.92	0.92	12
accuracy			0.93	30
macro avg	0.93	0.93	0.93	30
weighted avg	0.93	0.93	0.93	30