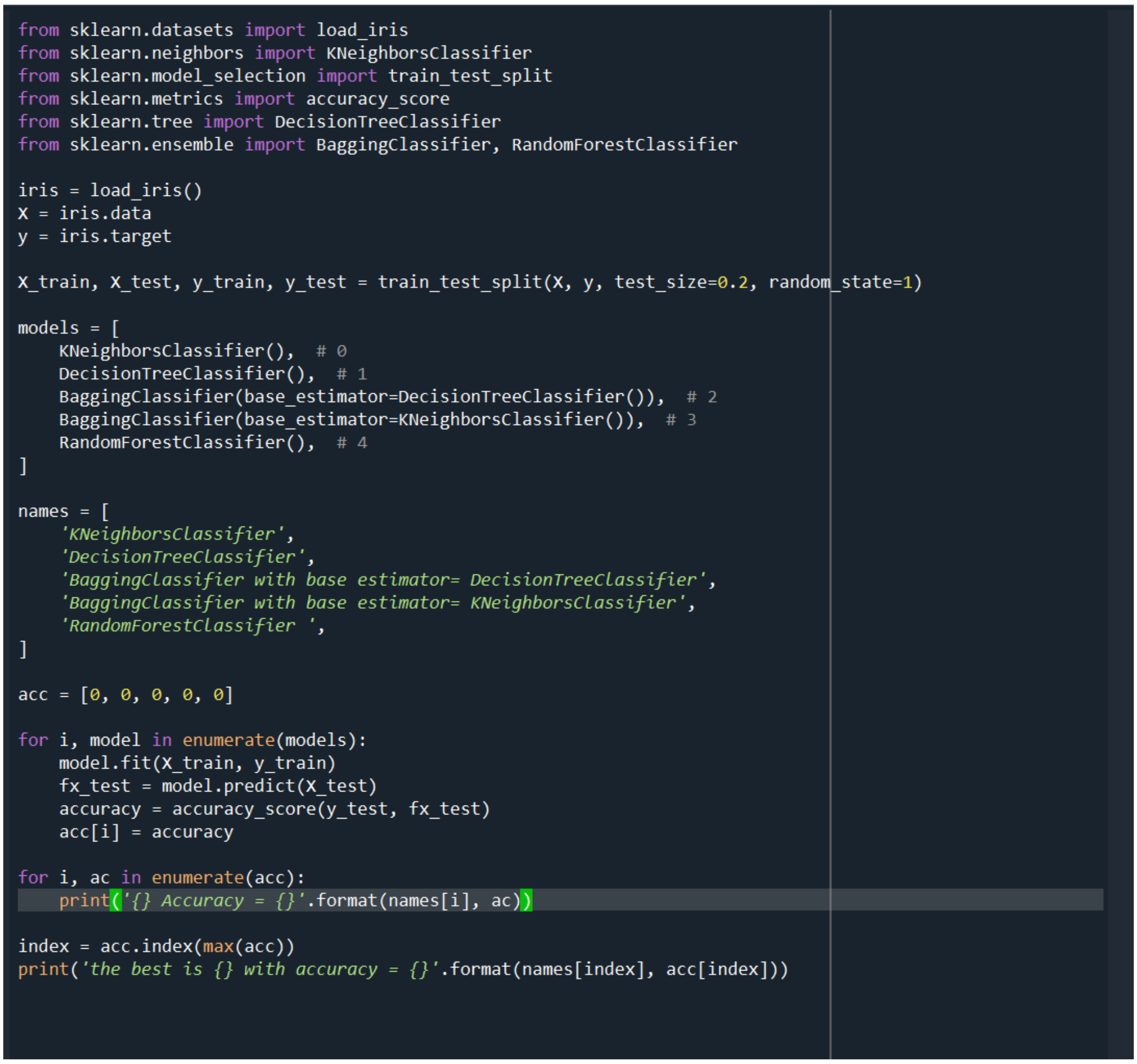
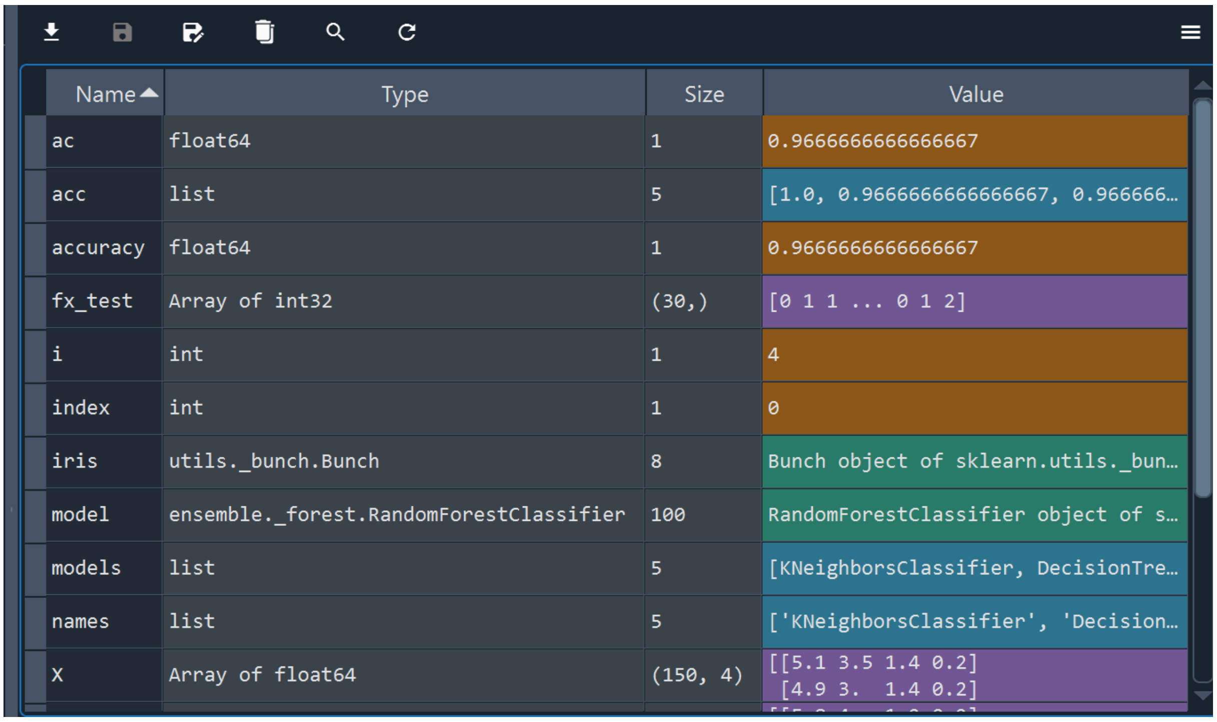
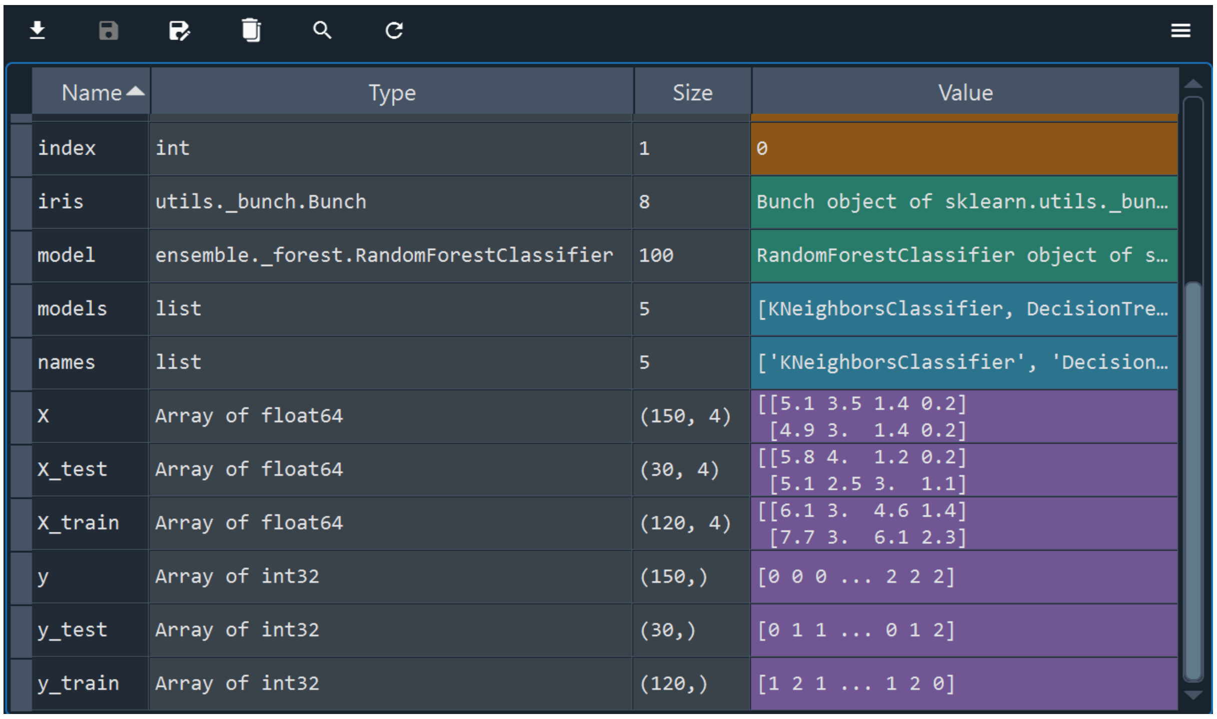


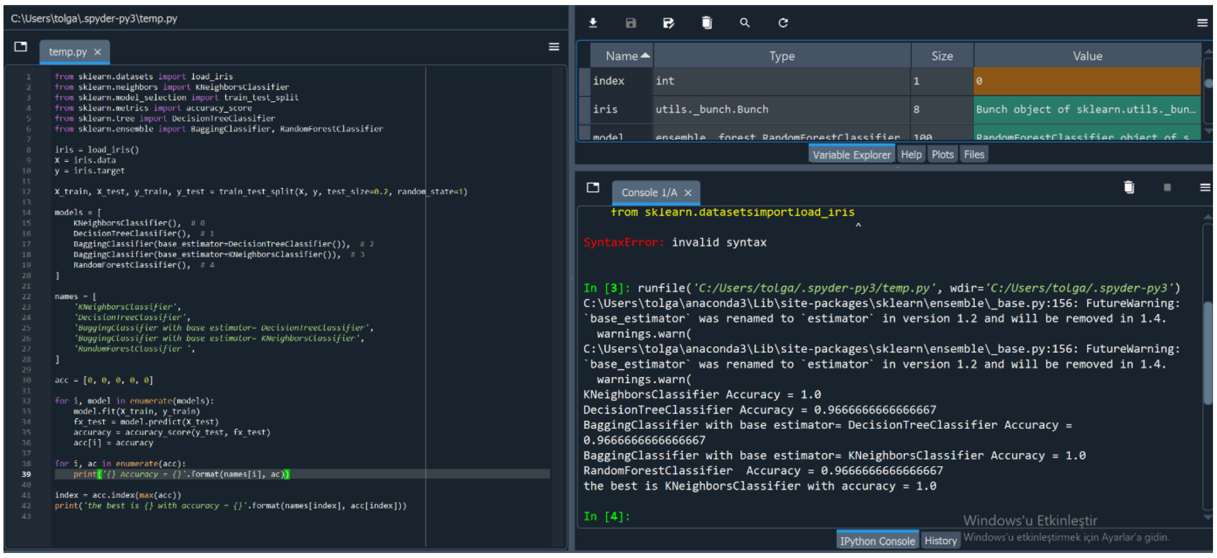
```python
from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier, RandomForestClassifier

iris = load_iris()
X = iris.data
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

models = [
    KNeighborsClassifier(),  # 0
    DecisionTreeClassifier(),  # 1
    BaggingClassifier(base_estimator=DecisionTreeClassifier()),  # 2
    BaggingClassifier(base_estimator=KNeighborsClassifier()),  # 3
    RandomForestClassifier(),  # 4
]

names = [
    'KNeighborsClassifier',
    'DecisionTreeClassifier',
    'BaggingClassifier with base estimator= DecisionTreeClassifier',
    'BaggingClassifier with base estimator= KNeighborsClassifier',
    'RandomForestClassifier ',
]

acc = [0, 0, 0, 0, 0]

for i, model in enumerate(models):
    model.fit(X_train, y_train)
    fx_test = model.predict(X_test)
    accuracy = accuracy_score(y_test, fx_test)
    acc[i] = accuracy

for i, ac in enumerate(acc):
    print('{} Accuracy = {}'.format(names[i], ac))

index = acc.index(max(acc))
print('the best is {} with accuracy = {}'.format(names[index], acc[index]))
```
]



Name	Type	Size	Value
ac	float64	1	0.9666666666666667
acc	list	5	[1.0, 0.9666666666666667, 0.966666…
accuracy	float64	1	0.9666666666666667
fx_test	Array of int32	(30,)	[0 1 1 ... 0 1 2]
i	int	1	4
index	int	1	0
iris	utils._bunch.Bunch	8	Bunch object of sklearn.utils._bun…
model	ensemble._forest.RandomForestClassifier	100	RandomForestClassifier object of s…
models	list	5	[KNeighborsClassifier, DecisionTre…
names	list	5	['KNeighborsClassifier', 'Decision…
X	Array of float64	(150, 4)	[[5.1 3.5 1.4 0.2] [4.9 3. 1.4 0.2]



Name	Type	Size	Value
index	int	1	0
iris	utils._bunch.Bunch	8	Bunch object of sklearn.utils._bun…
model	ensemble._forest.RandomForestClassifier	100	RandomForestClassifier object of s…
models	list	5	[KNeighborsClassifier, DecisionTre…
names	list	5	['KNeighborsClassifier', 'Decision…
X	Array of float64	(150, 4)	[[5.1 3.5 1.4 0.2] [4.9 3. 1.4 0.2]
X_test	Array of float64	(30, 4)	[[5.8 4. 1.2 0.2] [5.1 2.5 3. 1.1]
X_train	Array of float64	(120, 4)	[[6.1 3. 4.6 1.4] [7.7 3. 6.1 2.3]
y	Array of int32	(150,)	[0 0 0 ... 2 2 2]
y_test	Array of int32	(30,)	[0 1 1 ... 0 1 2]
y_train	Array of int32	(120,)	[1 2 1 ... 1 2 0]


C:\Users\tolga\.spyder-py3\temp.py

```python
from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier, RandomForestClassifier

iris = load_iris()
X = iris.data
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

models = [
    KNeighborsClassifier(),  # 0
    DecisionTreeClassifier(),  # 1
    BaggingClassifier(base_estimator=DecisionTreeClassifier()),  # 2
    BaggingClassifier(base_estimator=KNeighborsClassifier()),  # 3
    RandomForestClassifier(),  # 4
]

names = [
    'KNeighborsClassifier',
    'DecisionTreeClassifier',
    'BaggingClassifier with base estimator= DecisionTreeClassifier',
    'BaggingClassifier with base estimator= KNeighborsClassifier',
    'RandomForestClassifier ',
]

acc = [0, 0, 0, 0, 0]

for i, model in enumerate(models):
    model.fit(X_train, y_train)
    fx_test = model.predict(X_test)
    accuracy = accuracy_score(y_test, fx_test)
    acc[i] = accuracy

for i, ac in enumerate(acc):
    print('{} Accuracy = {}'.format(names[i], ac))

index = acc.index(max(acc))
print('the best is {} with accuracy = {}'.format(names[index], acc[index]))
```

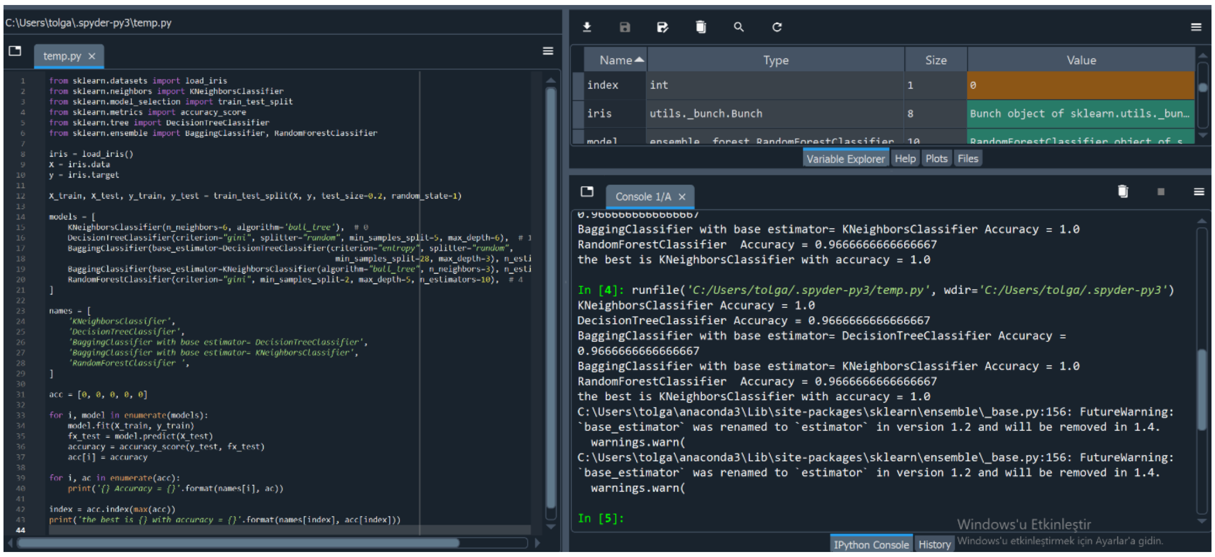
Name	Type	Size	Value
index	int	1	0
iris	utils._bunch.Bunch	8	Bunch object of sklearn.utils._bun…
model	ensemble._forest.RandomForestClassifier	100	RandomForestClassifier object of s

```
    from sklearn.datasetsimportload_iris
                                       ^
SyntaxError: invalid syntax


In [3]: runfile('C:/Users/tolga/.spyder-py3/temp.py', wdir='C:/Users/tolga/.spyder-py3')
C:\Users\tolga\anaconda3\Lib\site-packages\sklearn\ensemble\_base.py:156: FutureWarning:
`base_estimator` was renamed to `estimator` in version 1.2 and will be removed in 1.4.
  warnings.warn(
C:\Users\tolga\anaconda3\Lib\site-packages\sklearn\ensemble\_base.py:156: FutureWarning:
`base_estimator` was renamed to `estimator` in version 1.2 and will be removed in 1.4.
  warnings.warn(
KNeighborsClassifier Accuracy = 1.0
DecisionTreeClassifier Accuracy = 0.9666666666666667
BaggingClassifier with base estimator= DecisionTreeClassifier Accuracy =
0.9666666666666667
BaggingClassifier with base estimator= KNeighborsClassifier Accuracy = 1.0
RandomForestClassifier  Accuracy = 0.9666666666666667
the best is KNeighborsClassifier with accuracy = 1.0

In [4]:
```

Best Parameters of each algorithm Using the same code above for each algorithm while playing with parameters i obtained the following optimum parameters:

Algorithm	Parameters	Accuracy	note
KNN	n_neighbors < 7 (1,2,3,4,5 or 6) algorithm="auto" , “ball_tree” or “kd_tree” (same result)	1.0	Increasing the n_neghbors reduces generalization so keeping it relatively low is better
Decision Tree Classifier	criterion="entropy" or ”gini” splitter="random",min_samples_split between 3 and 7 max_depth=6	Average of 0.98	Setting a fixed max-depth value enhanced this algorithm very well because it increased the generalization of the algorithm
Bagging Classifier with Decision Tree Classifier as the base estimator	base_estimator = DecisionTreeClassifier(criterion="entropy", splitter="random", min_samples_split=28, max_depth=3), n_estimators=5	Average of 0.973	Increased min_samples_s plit so the algorithm better generalize a larger amount of data
Bagging Classifier with KNN as the base estimator	base_estimator = KNeighborsClassifier(algorithm="ball_tree",n_neighbors=3) n_estimators=5	1.0	Reduced n_neigbors as above
Random Forest Classifier	criterion="gini", min_samples_split=2, max_depth=5, n_estimators=10	0.967	The Random Forest Classifier achieved 96.7% accuracy on the iris dataset using the “gini” criterion, at least 2 replicate splits, a maximum depth of 5, and 10 predictors.

Applying the table optimum parameters again to the previous code



C:\Users\tolga\.spyder-py3\temp.py

```python
from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier, RandomForestClassifier

iris = load_iris()
X = iris.data
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

models = [
    KNeighborsClassifier(n_neighbors=6, algorithm='ball_tree'),  # 0
    DecisionTreeClassifier(criterion="gini", splitter="random", min_samples_split=5, max_depth=6),  # 1
    BaggingClassifier(base_estimator=DecisionTreeClassifier(criterion="entropy", splitter="random",
                                                            min_samples_split=28, max_depth=3), n_esti
    BaggingClassifier(base_estimator=KNeighborsClassifier(algorithm="ball_tree", n_neighbors=3), n_esti
    RandomForestClassifier(criterion="gini", min_samples_split=2, max_depth=5, n_estimators=10),  # 4
]

names = [
    'KNeighborsClassifier',
    'DecisionTreeClassifier',
    'BaggingClassifier with base estimator= DecisionTreeClassifier',
    'BaggingClassifier with base estimator= KNeighborsClassifier',
    'RandomForestClassifier ',
]

acc = [0, 0, 0, 0, 0]

for i, model in enumerate(models):
    model.fit(X_train, y_train)
    fx_test = model.predict(X_test)
    accuracy = accuracy_score(y_test, fx_test)
    acc[i] = accuracy

for i, ac in enumerate(acc):
    print('{} Accuracy = {}'.format(names[i], ac))

index = acc.index(max(acc))
print('the best is {} with accuracy = {}'.format(names[index], acc[index]))
```

Name	Type	Size	Value
index	int	1	0
iris	utils._bunch.Bunch	8	Bunch object of sklearn.utils._bun…
model	ensemble._forest.RandomForestClassifier	10	RandomForestClassifier object of s…

```
0.9666666666666667
BaggingClassifier with base estimator= KNeighborsClassifier Accuracy = 1.0
RandomForestClassifier  Accuracy = 0.9666666666666667
the best is KNeighborsClassifier with accuracy = 1.0

In [4]: runfile('C:/Users/tolga/.spyder-py3/temp.py', wdir='C:/Users/tolga/.spyder-py3')
KNeighborsClassifier Accuracy = 1.0
DecisionTreeClassifier Accuracy = 0.9666666666666667
BaggingClassifier with base estimator= DecisionTreeClassifier Accuracy =
0.9666666666666667
BaggingClassifier with base estimator= KNeighborsClassifier Accuracy = 1.0
RandomForestClassifier  Accuracy = 0.9666666666666667
the best is KNeighborsClassifier with accuracy = 1.0
C:\Users\tolga\anaconda3\Lib\site-packages\sklearn\ensemble\_base.py:156: FutureWarning:
`base_estimator` was renamed to `estimator` in version 1.2 and will be removed in 1.4.
  warnings.warn(
C:\Users\tolga\anaconda3\Lib\site-packages\sklearn\ensemble\_base.py:156: FutureWarning:
`base_estimator` was renamed to `estimator` in version 1.2 and will be removed in 1.4.
  warnings.warn(

In [5]:
```