```
import pandas as pd
import numpy as np

df = pd.DataFrame(pd.Series([12,14,1,8,4,33,0,24,17,10]),
columns=['sample'])

print(df.sample(10,replace=True))
print(df.sample(10,replace=True))
# 8, 10, 24

   sample
8      17
1      14
5      33
0      12
9      10
6       0
4       4
1      14
0      12
4       4
   sample
5      33
5      33
2       1
5      33
1      14
0      12
0      12
5      33
6       0
6       0

from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Create a synthetic dataset
X, y = make_classification(n_samples=1000, n_features=20,
n_informative=15, n_redundant=5, random_state=42)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Initialize the classifier with oob_score=True
rf = RandomForestClassifier(oob_score=True, random_state=42)

# Train the classifier
```

```
rf.fit(X_train, y_train)

# Predict the labels of the test set
predictions = rf.predict(X_test)

# Compute the accuracy of the classifier
accuracy = accuracy_score(y_test, predictions)

# Get the OOB score
oob_score = rf.oob_score_

accuracy, oob_score
```

```
(0.9, 0.9075)
```

```
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier,
ExtraTreesClassifier
from sklearn.metrics import accuracy_score

# Create a synthetic dataset
X, y = make_classification(n_samples=1000, n_features=20,
n_informative=15, n_redundant=5, random_state=42)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Initialize the classifiers
rf = RandomForestClassifier(random_state=42)
et = ExtraTreesClassifier(random_state=42)

# Train the classifiers
rf.fit(X_train, y_train)
et.fit(X_train, y_train)

# Predict the labels of the test set
rf_predictions = rf.predict(X_test)
et_predictions = et.predict(X_test)

# Compute the accuracy of the classifiers
rf_accuracy = accuracy_score(y_test, rf_predictions)
et_accuracy = accuracy_score(y_test, et_predictions)

rf_accuracy, et_accuracy
```

```
(0.9, 0.935)
```