```
import numpy as np
import pandas as pd
from sklearn.datasets import make_classification


X,y = make_classification(n_features=5, n_redundant=0, n_informative=5,n_clusters_per_class=1)


df = pd.DataFrame(X,columns=['col1','col2','col3','col4','col5'])
df['target'] = y
print(df.shape)
df.head()
```

```
(100, 6)
```

	col1	col2	col3	col4	col5	target
0	0.551380	-1.876815	-1.807618	-1.557932	-0.039809	0
1	-0.148084	-3.129695	0.387900	-1.322573	-1.448057	0
2	-3.874607	-1.727474	-0.943985	-2.895265	0.601189	1
3	1.709298	-0.825889	2.392230	-1.987155	1.819194	0
4	1.802057	-0.911493	0.022074	-2.545045	2.194047	0

```
# function for row sampling

def sample_rows(df,percent):
  return df.sample(int(percent*df.shape[0]),replace=True)


# function for feature sampling
def sample_features(df,percent):
  cols = random.sample(df.columns.tolist()[:-1],int(percent*(df.shape[1]-1)))
  new_df = df[cols]
  new_df['target'] = df['target']
  return new_df


# function for combined sampling

def combined_sampling(df,row_percent,col_percent):
  new_df = sample_rows(df,row_percent)
  return sample_features(new_df,col_percent)


df1 = combined_sampling(df,0.5,0.5)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
  """
```

```
df2 = combined_sampling(df,0.5,0.5)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
  """
```

```
df3 = combined_sampling(df,0.5,0.5)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
  """
```

```
print(df1.columns)
print(df2.columns)
print(df3.columns)
```

```
Index(['col5', 'col1', 'target'], dtype='object')
Index(['col5', 'col1', 'target'], dtype='object')
Index(['col3', 'col2', 'target'], dtype='object')
```

```
df3
```

	col3	col2	target
2	-0.943985	-1.727474	1
53	2.879522	-1.179645	0
17	-0.601854	0.240279	1
12	-1.209724	0.077615	1
8	-0.147061	-0.352877	0
6	-1.029536	-0.770751	1
94	-0.930578	-1.265605	1
4	0.022074	-0.911493	0
38	2.538559	-1.042250	0
40	0.595038	-0.162759	0
99	1.558165	-2.026716	0
25	-0.256852	-3.206283	1
91	-1.182347	-1.528637	0
12	-1.209724	0.077615	1
11	-0.679621	-1.275311	1
32	-1.343188	-0.141246	1
54	-0.384640	-0.270707	1
79	-0.558081	-1.985678	0
70	-0.438380	-1.497126	1
89	1.305171	-0.374577	0
10	-0.513609	-1.946675	1
22	-1.774446	-0.503487	1
87	-1.347947	-0.196919	1
96	-1.153825	-0.494453	1
14	-1.039782	-0.106427	1
84	1.986095	-1.284953	0
49	0.578898	-1.649036	0
74	1.125979	-2.884012	0
1	0.387900	-3.129695	0
73	-0.580873	-0.734898	1
86	1.411655	1.288265	0
66	0.885978	-3.087130	0
40	0.595038	-0.162759	0
91	-1.182347	-1.528637	0
48	-1.042498	-0.064323	1
92	-1.406179	-1.957413	1
8	-0.147061	-0.352877	0
89	1.305171	-0.374577	0
50	0.534164	-1.161223	0
98	-0.158598	-1.798861	0
83	-1.550002	-1.126191	1
51	-0.986968	-2.296564	1
54	-0.384640	-0.270707	1
56	-1.061683	-0.685227	1
91	-1.182347	-1.528637	0
94	-0.930578	-1.265605	1
49	0.578898	-1.649036	0
91	-1.182347	-1.528637	0
2	-0.943985	-1.727474	1

78	-1.013512	-1.587964	1

```
from sklearn.tree import DecisionTreeClassifier
clf1 = DecisionTreeClassifier()
clf2 = DecisionTreeClassifier()
clf3 = DecisionTreeClassifier()
```

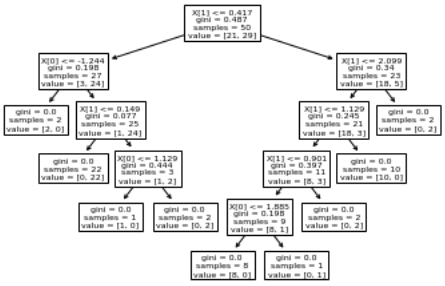
```
clf1.fit(df1.iloc[:,0:2],df1.iloc[:,-1])
clf2.fit(df2.iloc[:,0:2],df2.iloc[:,-1])
clf3.fit(df3.iloc[:,0:2],df3.iloc[:,-1])
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

```
from sklearn.tree import plot_tree
```

```
plot_tree(clf1)
```

```
[Text(167.4, 199.32, 'X[1] <= 0.417\ngini = 0.487\nsamples = 50\nvalue = [21, 29]'),
 Text(55.800000000000004, 163.07999999999998, 'X[0] <= -1.244\ngini = 0.198\nsamples = 27\nvalue = [3, 24]'),
 Text(27.900000000000002, 126.83999999999999, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(83.7, 126.83999999999999, 'X[1] <= 0.149\ngini = 0.077\nsamples = 25\nvalue = [1, 24]'),
 Text(55.800000000000004, 90.6, 'gini = 0.0\nsamples = 22\nvalue = [0, 22]'),
 Text(111.60000000000001, 90.6, 'X[0] <= 1.129\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(83.7, 54.359999999999985, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(139.5, 54.359999999999985, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(279.0, 163.07999999999998, 'X[1] <= 2.099\ngini = 0.34\nsamples = 23\nvalue = [18, 5]'),
 Text(251.10000000000002, 126.83999999999999, 'X[1] <= 1.129\ngini = 0.245\nsamples = 21\nvalue = [18, 3]'),
 Text(223.20000000000002, 90.6, 'X[1] <= 0.901\ngini = 0.397\nsamples = 11\nvalue = [8, 3]'),
 Text(195.3, 54.359999999999985, 'X[0] <= 1.885\ngini = 0.198\nsamples = 9\nvalue = [8, 1]'),
 Text(167.4, 18.119999999999976, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(223.20000000000002, 18.119999999999976, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(251.10000000000002, 54.359999999999985, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(279.0, 90.6, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
 Text(306.90000000000003, 126.83999999999999, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]')]
```




```
plot_tree(clf2)
```

```
[Text(209.25, 201.90857142857143, 'X[1] <= -0.391\ngini = 0.5\nsamples = 50\nvalue = [25, 25]'),
 Text(167.4, 170.84571428571428, 'gini = 0.0\nsamples = 17\nvalue = [0, 17]'),
 Text(251.10000000000002, 170.84571428571428, 'X[0] <= 3.267\ngini = 0.367\nsamples = 33\nvalue = [25, 8]'),
 Text(209.25, 139.78285714285715, 'X[1] <= 0.588\ngini = 0.278\nsamples = 30\nvalue = [25, 5]'),
 Text(167.4, 108.72, 'X[1] <= 0.357\ngini = 0.496\nsamples = 11\nvalue = [6, 5]'),
 Text(125.55000000000001, 77.65714285714284, 'X[1] <= -0.15\ngini = 0.245\nsamples = 7\nvalue = [6, 1]'),
 Text(83.7, 46.59428571428572, 'X[1] <= -0.162\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(41.85, 15.531428571428563, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(125.55000000000001, 15.531428571428563, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(167.4, 46.59428571428572, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(209.25, 77.65714285714284, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(251.10000000000002, 108.72, 'gini = 0.0\nsamples = 19\nvalue = [19, 0]'),
 Text(292.95, 139.78285714285715, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]')]
```