In [62]:

```
# multiple model ensemble
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [63]:

```
df = sns.load_dataset("tips")
df.head()
```

Out[63]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [64]:

```
# predict what its times if its is lunch or dinner
```

In [65]:

```
df.time.unique()
```

Out[65]:

```
['Dinner', 'Lunch']
Categories (2, object): ['Lunch', 'Dinner']
```

In [66]:

```
#since time is nominal value we will use labelencoder
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
df['time']= encoder.fit_transform(df['time'])
df
```

Out[66]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	0	2
1	10.34	1.66	Male	No	Sun	0	3
2	21.01	3.50	Male	No	Sun	0	3
3	23.68	3.31	Male	No	Sun	0	2
4	24.59	3.61	Female	No	Sun	0	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	0	3
240	27.18	2.00	Female	Yes	Sat	0	2
241	22.67	2.00	Male	Yes	Sat	0	2
242	17.82	1.75	Male	No	Sat	0	2
243	18.78	3.00	Female	No	Thur	0	2

244 rows × 7 columns

In [67]:

```
df.time.unique()
```

Out[67]:

```
array([0, 1])
```

In [68]:

```
x = df.drop('time',axis= 1)
y = df['time']
```

In [69]:

```
x
```

Out[69]:

	total_bill	tip	sex	smoker	day	size
0	16.99	1.01	Female	No	Sun	2
1	10.34	1.66	Male	No	Sun	3
2	21.01	3.50	Male	No	Sun	3
3	23.68	3.31	Male	No	Sun	2
4	24.59	3.61	Female	No	Sun	4
...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	3
240	27.18	2.00	Female	Yes	Sat	2
241	22.67	2.00	Male	Yes	Sat	2
242	17.82	1.75	Male	No	Sat	2
243	18.78	3.00	Female	No	Thur	2

244 rows × 6 columns

In [70]:

```
y
```

Out[70]:

```
0      0
1      0
2      0
3      0
4      0
      ..
239    0
240    0
241    0
242    0
243    0
Name: time, Length: 244, dtype: int32
```

In [71]:

```
from sklearn.model_selection import train_test_split
```

In [72]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size =0.20,random_state = 1)
```

In [73]:

```
x_train
```

Out[73]:

	total_bill	tip	sex	smoker	day	size
0	16.99	1.01	Female	No	Sun	2
154	19.77	2.00	Male	No	Sun	4
167	31.71	4.50	Male	No	Sun	4
110	14.00	3.00	Male	No	Sat	2
225	16.27	2.50	Female	Yes	Fri	2
...	...	...	...	...	...	...
137	14.15	2.00	Female	No	Thur	2
72	26.86	3.14	Female	Yes	Sat	2
140	17.47	3.50	Female	No	Thur	2
235	10.07	1.25	Male	No	Sat	2
37	16.93	3.07	Female	No	Sat	3

195 rows × 6 columns

In [74]:

```
y_train
```

Out[74]:

```
0      0
154    0
167    0
110    0
225    1
      ..
137    1
72     0
140    1
235    0
37     0
Name: time, Length: 195, dtype: int32
```

In [75]:
```
x_train.head()
```

Out[75]:

	total_bill	tip	sex	smoker	day	size
0	16.99	1.01	Female	No	Sun	2
154	19.77	2.00	Male	No	Sun	4
167	31.71	4.50	Male	No	Sun	4
110	14.00	3.00	Male	No	Sat	2
225	16.27	2.50	Female	Yes	Fri	2

In [76]:
```
# handling the missing value
#data encoding
#feature scaling
from sklearn.impute import SimpleImputer  #foe missing value
from sklearn.preprocessing import OneHotEncoder #fro encoding
from sklearn.preprocessing import StandardScaler # for scaling


from sklearn.pipeline import Pipeline  #A squential of data transformer
from sklearn.compose import ColumnTransformer # group the above the step for specific columns
```

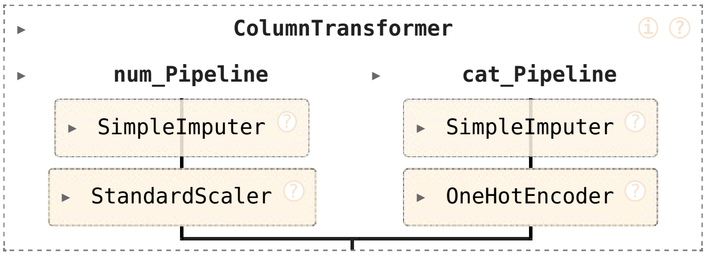
In [77]:
```
cat_cols = ["sex","smoker","day"]
num_cols = ["total_bill","tip","size"]
```

In [78]:
```
num_Pipeline = Pipeline(steps= [('imputation',SimpleImputer(strategy="median")),('scaling',StandardScaler())])
cat_Pipeline = Pipeline(steps = [('imputation',SimpleImputer(strategy="most_frequent")),('encoding', OneHotEnco
```

In [79]:
```
preprocessor = ColumnTransformer([("num_Pipeline",num_Pipeline,num_cols),("cat_Pipeline",cat_Pipeline,cat_cols)
```

In [80]:
```
preprocessor
```

Out[80]:



In [81]:
```
x_train = preprocessor.fit_transform(x_train)
x_test = preprocessor.transform(x_test)
```

In [82]:
```
x_train
```

Out[82]:
```
array([[-0.28611937, -1.47443803, -0.57766863, ...,  0.        ,
         1.        ,  0.        ],
       [ 0.02695905, -0.71612531,  1.47042924, ...,  0.        ,
         1.        ,  0.        ],
       [ 1.3716196 ,  1.19880579,  1.47042924, ...,  0.        ,
         1.        ,  0.        ],
       ...,
       [-0.23206267,  0.43283335, -0.57766863, ...,  0.        ,
         0.        ,  1.        ],
       [-1.06543688, -1.29060464, -0.57766863, ...,  1.        ,
         0.        ,  0.        ],
       [-0.29287646,  0.1034652 ,  0.44638031, ...,  1.        ,
         0.        ,  0.        ]])
```

In [83]:
```
x_test
```

Out[83]:
```
array([[-1.85376383, -1.48209775, -1.60171757,  1.        ,  0.        ,
         0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
         0.        ],
       [-0.08453291,  0.04984713, -0.57766863,  1.        ,  0.        ,
         1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
         1.        ],
       [ 0.79501474,  0.36389583,  0.44638031,  0.        ,  1.        ,
         0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
         0.        ],
       [-0.59356688, -0.33313909, -0.57766863,  0.        ,  1.        ,
         1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
         1.        ],
       [ 0.18349826,  0.04984713, -0.57766863,  0.        ,  1.        ,
         1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
         1.        ],
       [-1.32783714, -1.14506988, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [-0.93367366, -0.8999587 , -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 0.15421754,  0.43283335,  0.44638031,  1.        ,  0.        ,
          0.        ,  1.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [-0.56879089, -0.71612531, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [ 0.56977847,  0.51709032,  1.47042924,  1.        ,  0.        ,
          1.        ,  0.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [-0.83006497, -0.56293082, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  1.        ,  0.        ,  0.        ,
          0.        ],
        [-1.06318451, -0.71612531, -0.57766863,  1.        ,  0.        ,
          0.        ,  1.        ,  1.        ,  0.        ,  0.        ,
          0.        ],
        [ 1.17115932,  1.63541008,  1.47042924,  0.        ,  1.        ,
          1.        ,  0.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [ 0.63960171,  1.03795158, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 3.52262601,  5.41165421,  0.44638031,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 0.35355165, -0.71612531, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 0.42787962,  2.07967409, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [-1.04066089, -0.25654185, -0.57766863,  1.        ,  0.        ,
          1.        ,  0.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [-0.77037736, -0.71612531, -0.57766863,  0.        ,  1.        ,
          1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 0.86145944, -0.71612531, -0.57766863,  1.        ,  0.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 1.19480913,  0.11878465,  1.47042924,  1.        ,  0.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [-0.19715104,  0.25665969, -0.57766863,  0.        ,  1.        ,
          1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [-0.36157352,  1.0456113 , -0.57766863,  1.        ,  0.        ,
          0.        ,  1.        ,  1.        ,  0.        ,  0.        ,
          0.        ],
        [ 0.40986071,  2.73075067,  1.47042924,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [-0.79965807, -0.8693198 , -0.57766863,  1.        ,  0.        ,
          1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
          1.        ],
        [-1.00011836, -1.09911153, -0.57766863,  1.        ,  0.        ,
          1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
          1.        ],
        [ 0.10804411,  0.86177792, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 0.64748498,  1.58179201, -0.57766863,  1.        ,  0.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [-0.09466854, -0.47867385,  0.44638031,  0.        ,  1.        ,
          1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 1.18579968, -0.71612531,  0.44638031,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [-0.73546574, -0.71612531, -0.57766863,  1.        ,  0.        ,
          0.        ,  1.        ,  0.        ,  0.        ,  0.        ,
          1.        ],
        [ 1.22408984,  2.04137547,  1.47042924,  0.        ,  1.        ,
          1.        ,  0.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [-0.95844965, -0.73144476, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
       [ 1.83560633,  1.3290211 ,  0.44638031,  1.        ,  0.        ,
         1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
         0.        ],
       [ 1.72298819,  1.71200732,  1.47042924,  1.        ,  0.        ,
         1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
         1.        ],
       [-0.68816612, -1.03783373, -0.57766863,  0.        ,  1.        ,
         0.        ,  1.        ,  1.        ,  0.        ,  0.        ,
         0.        ],
       [ 1.32206762,  1.58179201,  0.44638031,  0.        ,  1.        ,
         1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
         0.        ],
       [-0.1329587 , -0.33313909,  1.47042924,  0.        ,  1.        ,
         1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
         0.        ],
       [-0.28837173,  0.43283335,  0.44638031,  1.        ,  0.        ,
         1.        ,  0.        ,  0.        ,  0.        ,  1.        ,
         0.        ],
       [-0.94155693, -1.09911153, -0.57766863,  1.        ,  0.        ,
         1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
         1.        ],
       [ 0.51234322, -0.01143067,  1.47042924,  1.        ,  0.        ,
         1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
         1.        ],
       [ 0.12606301,  0.31793748,  0.44638031,  0.        ,  1.        ,
         1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
         0.        ],
       [ 0.33328038,  0.43283335, -0.57766863,  0.        ,  1.        ,
         1.        ,  0.        ,  1.        ,  0.        ,  0.        ,
         0.        ],
       [-0.50910328, -0.64718779, -0.57766863,  0.        ,  1.        ,
         0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
         0.        ],
       [ 0.13056774, -0.37143771,  1.47042924,  1.        ,  0.        ,
         1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
         0.        ],
       [ 1.06304591,  0.04984713, -0.57766863,  0.        ,  1.        ,
         0.        ,  1.        ,  1.        ,  0.        ,  0.        ,
         0.        ],
       [ 0.51009086,  0.50943059,  0.44638031,  0.        ,  1.        ,
         1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
         0.        ],
       [ 1.77141399,  1.58179201,  1.47042924,  1.        ,  0.        ,
         1.        ,  0.        ,  0.        ,  0.        ,  1.        ,
         0.        ],
       [ 0.01344487,  0.04984713, -0.57766863,  1.        ,  0.        ,
         1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
         0.        ]])
```

In [84]:

```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
```

In [85]:

```python
models ={"support vector classifier":SVC(),
        "DT classifier":DecisionTreeClassifier()}
```

In [86]:

```python
from sklearn.metrics import accuracy_score
def model_train_eval(x_train,y_train,x_test,y_test,models):
    evaluation= {}
    for i in range(len(models)):
        model=list(models.values())[i]
        model.fit(x_train,y_train)
        y_pred = model.predict(x_test)
        model_score = accuracy_score(y_test,y_pred)
        evaluation[list(models.keys())[i]]=model_score
    return evaluation
```

In [87]:

```python
 model_train_eval(x_train,y_train,x_test,y_test,models)
```

Out[87]:

```
{'support vector classifier': 0.9183673469387755,
 'DT classifier': 0.9387755102040817}
```

In [88]:

```python
#Random forest
```

In [89]:

```python
from sklearn.ensemble import RandomForestClassifier
```

In [90]:

```python
rf =RandomForestClassifier()
```

In [91]:

```python
rf
```

Out[91]:

```
▾ RandomForestClassifier ⓘ ?

RandomForestClassifier()
```

In [92]:

```
x_train,x_test
```

Out[92]:

```
(array([[-0.28611937, -1.47443803, -0.57766863, ...,  0.        ,
          1.        ,  0.        ],
        [ 0.02695905, -0.71612531,  1.47042924, ...,  0.        ,
          1.        ,  0.        ],
        [ 1.3716196 ,  1.19880579,  1.47042924, ...,  0.        ,
          1.        ,  0.        ],
        ...,
        [-0.23206267,  0.43283335, -0.57766863, ...,  0.        ,
          0.        ,  1.        ],
        [-1.06543688, -1.29060464, -0.57766863, ...,  1.        ,
          0.        ,  0.        ],
        [-0.29287646,  0.1034652 ,  0.44638031, ...,  1.        ,
          0.        ,  0.        ]]),
 array([[-1.85376383, -1.48209775, -1.60171757,  1.        ,  0.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [-0.08453291,  0.04984713, -0.57766863,  1.        ,  0.        ,
          1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
          1.        ],
        [ 0.79501474,  0.36389583,  0.44638031,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [-0.59356688, -0.33313909, -0.57766863,  0.        ,  1.        ,
          1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
          1.        ],
        [ 0.18349826,  0.04984713, -0.57766863,  0.        ,  1.        ,
          1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
          1.        ],
        [-1.32783714, -1.14506988, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [-0.93367366, -0.8999587 , -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 0.15421754,  0.43283335,  0.44638031,  1.        ,  0.        ,
          0.        ,  1.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [-0.56879089, -0.71612531, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [ 0.56977847,  0.51709032,  1.47042924,  1.        ,  0.        ,
          1.        ,  0.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [-0.83006497, -0.56293082, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  1.        ,  0.        ,  0.        ,
          0.        ],
        [-1.06318451, -0.71612531, -0.57766863,  1.        ,  0.        ,
          0.        ,  1.        ,  1.        ,  0.        ,  0.        ,
          0.        ],
        [ 1.17115932,  1.63541008,  1.47042924,  0.        ,  1.        ,
          1.        ,  0.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [ 0.63960171,  1.03795158, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 3.52262601,  5.41165421,  0.44638031,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 0.35355165, -0.71612531, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 0.42787962,  2.07967409, -0.57766863,  0.        ,  1.        ,
          0.        ,  1.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [-1.04066089, -0.25654185, -0.57766863,  1.        ,  0.        ,
          1.        ,  0.        ,  0.        ,  0.        ,  1.        ,
          0.        ],
        [-0.77037736, -0.71612531, -0.57766863,  0.        ,  1.        ,
          1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 0.86145944, -0.71612531, -0.57766863,  1.        ,  0.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
        [ 1.19480913,  0.11878465,  1.47042924,  1.        ,  0.        ,
          0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
          0.        ],
[-0.19715104,  0.25665969, -0.57766863,  0.        ,  1.        ,
  1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[-0.36157352,  1.0456113 , -0.57766863,  1.        ,  0.        ,
  0.        ,  1.        ,  1.        ,  0.        ,  0.        ,
  0.        ],
[ 0.40986071,  2.73075067,  1.47042924,  0.        ,  1.        ,
  0.        ,  1.        ,  0.        ,  0.        ,  1.        ,
  0.        ],
[-0.79965807, -0.8693198 , -0.57766863,  1.        ,  0.        ,
  1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
  1.        ],
[-1.00011836, -1.09911153, -0.57766863,  1.        ,  0.        ,
  1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
  1.        ],
[ 0.10804411,  0.86177792, -0.57766863,  0.        ,  1.        ,
  0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[ 0.64748498,  1.58179201, -0.57766863,  1.        ,  0.        ,
  0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[-0.09466854, -0.47867385,  0.44638031,  0.        ,  1.        ,
  1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[ 1.18579968, -0.71612531,  0.44638031,  0.        ,  1.        ,
  0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[-0.73546574, -0.71612531, -0.57766863,  1.        ,  0.        ,
  0.        ,  1.        ,  0.        ,  0.        ,  0.        ,
  1.        ],
[ 1.22408984,  2.04137547,  1.47042924,  0.        ,  1.        ,
  1.        ,  0.        ,  0.        ,  0.        ,  1.        ,
  0.        ],
[-0.95844965, -0.73144476, -0.57766863,  0.        ,  1.        ,
  0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[ 1.83560633,  1.3290211 ,  0.44638031,  1.        ,  0.        ,
  1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[ 1.72298819,  1.71200732,  1.47042924,  1.        ,  0.        ,
  1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
  1.        ],
[-0.68816612, -1.03783373, -0.57766863,  0.        ,  1.        ,
  0.        ,  1.        ,  1.        ,  0.        ,  0.        ,
  0.        ],
[ 1.32206762,  1.58179201,  0.44638031,  0.        ,  1.        ,
  1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[-0.1329587 , -0.33313909,  1.47042924,  0.        ,  1.        ,
  1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[-0.28837173,  0.43283335,  0.44638031,  1.        ,  0.        ,
  1.        ,  0.        ,  0.        ,  0.        ,  1.        ,
  0.        ],
[-0.94155693, -1.09911153, -0.57766863,  1.        ,  0.        ,
  1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
  1.        ],
[ 0.51234322, -0.01143067,  1.47042924,  1.        ,  0.        ,
  1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
  1.        ],
[ 0.12606301,  0.31793748,  0.44638031,  0.        ,  1.        ,
  1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[ 0.33328038,  0.43283335, -0.57766863,  0.        ,  1.        ,
  1.        ,  0.        ,  1.        ,  0.        ,  0.        ,
  0.        ],
[-0.50910328, -0.64718779, -0.57766863,  0.        ,  1.        ,
  0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[ 0.13056774, -0.37143771,  1.47042924,  1.        ,  0.        ,
  1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[ 1.06304591,  0.04984713, -0.57766863,  0.        ,  1.        ,
  0.        ,  1.        ,  1.        ,  0.        ,  0.        ,
  0.        ],
[ 0.51009086,  0.50943059,  0.44638031,  0.        ,  1.        ,
  1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
  0.        ],
[ 1.77141399,  1.58179201,  1.47042924,  1.        ,  0.        ,
  1.        ,  0.        ,  0.        ,  0.        ,  1.        ,
  0.        ],
[ 0.01344487,  0.04984713, -0.57766863,  1.        ,  0.        ,
  1.        ,  0.        ,  0.        ,  1.        ,  0.        ,
          0.        ]]))
```

In [93]:
```
from sklearn.model_selection import RandomizedSearchCV
params = {"max_depth":[1,2,3,5,10,None,],
          "n_estimators":[50,100,200,300],
           "criterion":["ginni","entropy"]}
```

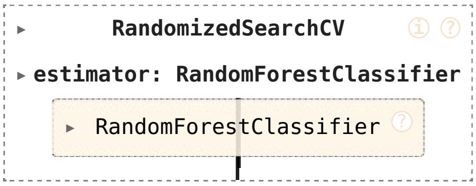
In [94]:
```
params
```

Out[94]:
```
{'max_depth': [1, 2, 3, 5, 10, None],
 'n_estimators': [50, 100, 200, 300],
 'criterion': ['ginni', 'entropy']}
```

In [95]:
```
clf = RandomizedSearchCV(rf,param_distributions=params,cv= 5,verbose=5,scoring='accuracy')
```

In [56]:
```
clf
```

Out[56]:



▸ RandomizedSearchCV

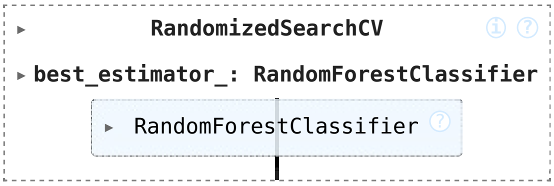
▸ estimator: RandomForestClassifier

▸ RandomForestClassifier

In [57]:
```
clf.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
[CV 1/5] END criterion=ginni, max_depth=3, n_estimators=50;, score=nan total time=   0.0s
[CV 2/5] END criterion=ginni, max_depth=3, n_estimators=50;, score=nan total time=   0.0s
[CV 3/5] END criterion=ginni, max_depth=3, n_estimators=50;, score=nan total time=   0.0s
[CV 4/5] END criterion=ginni, max_depth=3, n_estimators=50;, score=nan total time=   0.0s
[CV 5/5] END criterion=ginni, max_depth=3, n_estimators=50;, score=nan total time=   0.0s
[CV 1/5] END criterion=entropy, max_depth=5, n_estimators=200;, score=0.923 total time=   0.1s
[CV 2/5] END criterion=entropy, max_depth=5, n_estimators=200;, score=0.974 total time=   0.1s
[CV 3/5] END criterion=entropy, max_depth=5, n_estimators=200;, score=1.000 total time=   0.1s
[CV 4/5] END criterion=entropy, max_depth=5, n_estimators=200;, score=1.000 total time=   0.1s
[CV 5/5] END criterion=entropy, max_depth=5, n_estimators=200;, score=0.974 total time=   0.1s
[CV 1/5] END criterion=entropy, max_depth=3, n_estimators=200;, score=0.923 total time=   0.2s
[CV 2/5] END criterion=entropy, max_depth=3, n_estimators=200;, score=0.974 total time=   0.1s
[CV 3/5] END criterion=entropy, max_depth=3, n_estimators=200;, score=1.000 total time=   0.1s
[CV 4/5] END criterion=entropy, max_depth=3, n_estimators=200;, score=1.000 total time=   0.1s
[CV 5/5] END criterion=entropy, max_depth=3, n_estimators=200;, score=1.000 total time=   0.1s
[CV 1/5] END criterion=entropy, max_depth=5, n_estimators=300;, score=0.923 total time=   0.2s
[CV 2/5] END criterion=entropy, max_depth=5, n_estimators=300;, score=0.974 total time=   0.2s
[CV 3/5] END criterion=entropy, max_depth=5, n_estimators=300;, score=1.000 total time=   0.1s
[CV 4/5] END criterion=entropy, max_depth=5, n_estimators=300;, score=1.000 total time=   0.1s
[CV 5/5] END criterion=entropy, max_depth=5, n_estimators=300;, score=0.974 total time=   0.1s
[CV 1/5] END criterion=ginni, max_depth=5, n_estimators=200;, score=nan total time=   0.0s
[CV 2/5] END criterion=ginni, max_depth=5, n_estimators=200;, score=nan total time=   0.0s
[CV 3/5] END criterion=ginni, max_depth=5, n_estimators=200;, score=nan total time=   0.0s
[CV 4/5] END criterion=ginni, max_depth=5, n_estimators=200;, score=nan total time=   0.0s
[CV 5/5] END criterion=ginni, max_depth=5, n_estimators=200;, score=nan total time=   0.0s
[CV 1/5] END criterion=ginni, max_depth=1, n_estimators=50;, score=nan total time=   0.0s
[CV 2/5] END criterion=ginni, max_depth=1, n_estimators=50;, score=nan total time=   0.0s
[CV 3/5] END criterion=ginni, max_depth=1, n_estimators=50;, score=nan total time=   0.0s
[CV 4/5] END criterion=ginni, max_depth=1, n_estimators=50;, score=nan total time=   0.0s
[CV 5/5] END criterion=ginni, max_depth=1, n_estimators=50;, score=nan total time=   0.0s
[CV 1/5] END criterion=entropy, max_depth=3, n_estimators=100;, score=0.923 total time=   0.0s
[CV 2/5] END criterion=entropy, max_depth=3, n_estimators=100;, score=0.974 total time=   0.0s
[CV 3/5] END criterion=entropy, max_depth=3, n_estimators=100;, score=1.000 total time=   0.0s
[CV 4/5] END criterion=entropy, max_depth=3, n_estimators=100;, score=1.000 total time=   0.0s
[CV 5/5] END criterion=entropy, max_depth=3, n_estimators=100;, score=1.000 total time=   0.0s
[CV 1/5] END criterion=ginni, max_depth=3, n_estimators=300;, score=nan total time=   0.0s
[CV 2/5] END criterion=ginni, max_depth=3, n_estimators=300;, score=nan total time=   0.0s
[CV 3/5] END criterion=ginni, max_depth=3, n_estimators=300;, score=nan total time=   0.0s
[CV 4/5] END criterion=ginni, max_depth=3, n_estimators=300;, score=nan total time=   0.0s
[CV 5/5] END criterion=ginni, max_depth=3, n_estimators=300;, score=nan total time=   0.0s
[CV 1/5] END criterion=entropy, max_depth=None, n_estimators=200;, score=0.923 total time=   0.1s
[CV 2/5] END criterion=entropy, max_depth=None, n_estimators=200;, score=0.974 total time=   0.0s
[CV 3/5] END criterion=entropy, max_depth=None, n_estimators=200;, score=1.000 total time=   0.0s
[CV 4/5] END criterion=entropy, max_depth=None, n_estimators=200;, score=0.974 total time=   0.0s
[CV 5/5] END criterion=entropy, max_depth=None, n_estimators=200;, score=0.974 total time=   0.0s
[CV 1/5] END criterion=ginni, max_depth=2, n_estimators=300;, score=nan total time=   0.0s
[CV 2/5] END criterion=ginni, max_depth=2, n_estimators=300;, score=nan total time=   0.0s
[CV 3/5] END criterion=ginni, max_depth=2, n_estimators=300;, score=nan total time=   0.0s
[CV 4/5] END criterion=ginni, max_depth=2, n_estimators=300;, score=nan total time=   0.0s
[CV 5/5] END criterion=ginni, max_depth=2, n_estimators=300;, score=nan total time=   0.0s
```

Out[57]:



▸ RandomizedSearchCV

▸ best_estimator_: RandomForestClassifier

▸ RandomForestClassifier

In [58]:
```
clf.best_params_
```

Out[58]:
```
{'n_estimators': 200, 'max_depth': 3, 'criterion': 'entropy'}
```

In [59]:
```
clf.best_score_
```

Out[59]:
```
0.9794871794871796
```

In [60]:
```
models ={"support vector classifier":SVC(),
        "DT classifier":DecisionTreeClassifier(),
        "random_forest":RandomForestClassifier()}
```

In [61]:
```
from sklearn.metrics import accuracy_score
def model_train_eval(x_train,y_train,x_test,y_test,models):
    evaluation= {}
    for i in range(len(models)):
        model=list(models.values())[i]
        model.fit(x_train,y_train)
        y_pred = model.predict(x_test)
        model_score = accuracy_score(y_test,y_pred)
        evaluation[list(models.keys())[i]]=model_score
    return evaluation
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: