```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score,classification_report
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score, KFold, GridSearchCV
from sklearn.feature_selection import SelectKBest, chi2
```

```
#step1 =load model
#step2=missing values handleing
#step3=divide in z and y (indepent and dependent values)
#step4=train-test split
#step5=standardscaling xtrain and xtest
```

```
data=pd.read_csv('heart_disease.csv')
```

```
data
```

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thal
0	1	63	Male	Cleveland	typical angina	145.0	233.0	True	lv hypertrophy	150
1	2	67	Male	Cleveland	asymptomatic	160.0	286.0	False	lv hypertrophy	108
2	3	67	Male	Cleveland	asymptomatic	120.0	229.0	False	lv hypertrophy	129
3	4	37	Male	Cleveland	non-anginal	130.0	250.0	False	normal	187
4	5	41	Female	Cleveland	atypical angina	130.0	204.0	False	lv hypertrophy	172
...	...	...	...	...	...	...	...	...	...	
915	916	54	Female	VA Long Beach	asymptomatic	127.0	333.0	True	st-t abnormality	154
916	917	62	Male	VA Long Beach	typical angina	NaN	139.0	False	st-t abnormality	Na
917	918	55	Male	VA Long Beach	asymptomatic	122.0	223.0	True	st-t abnormality	100
918	919	58	Male	VA Long Beach	asymptomatic	NaN	385.0	True	lv hypertrophy	Na
919	920	62	Male	VA Long Beach	atypical angina	120.0	254.0	False	lv hypertrophy	93

920 rows × 16 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 920 entries, 0 to 919
Data columns (total 16 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   id        920 non-null    int64  
 1   age       920 non-null    int64  
 2   sex       920 non-null    object 
 3   dataset   920 non-null    object 
 4   cp        920 non-null    object 
 5   trestbps  861 non-null    float64
 6   chol      890 non-null    float64
 7   fbs       830 non-null    object 
 8   restecg   918 non-null    object 
 9   thalch    865 non-null    float64
 10  exang     865 non-null    object 
 11  oldpeak   858 non-null    float64
 12  slope     611 non-null    object 
 13  ca        309 non-null    float64
 14  thal      434 non-null    object 
 15  num       920 non-null    int64  
dtypes: float64(5), int64(3), object(8)
memory usage: 115.1+ KB
```

```
data_isnull = data.isnull()
```

```
print(data_isnull.sum())
```

```
id            0
age           0
sex           0
dataset       0
cp            0
trestbps     59
chol         30
fbs          90
restecg       2
thalch       55
exang        55
oldpeak      62
slope       309
ca          611
thal        486
num           0
dtype: int64
```

```
print(data.shape)
```

```
(920, 16)
```

```
data.dropna(inplace=True)
print(data.isnull().sum())
print(data.shape)
```

```
id          0
age         0
sex         0
dataset     0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalch      0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
num         0
dtype: int64
(299, 16)
```

```
data
```

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thal
0	1	63	Male	Cleveland	typical angina	145.0	233.0	True	lv hypertrophy	150
1	2	67	Male	Cleveland	asymptomatic	160.0	286.0	False	lv hypertrophy	108
2	3	67	Male	Cleveland	asymptomatic	120.0	229.0	False	lv hypertrophy	129
3	4	37	Male	Cleveland	non-anginal	130.0	250.0	False	normal	187
4	5	41	Female	Cleveland	atypical angina	130.0	204.0	False	lv hypertrophy	172
...	...	...	...	...	...	...	...	...	...	
299	300	68	Male	Cleveland	asymptomatic	144.0	193.0	True	normal	141
300	301	57	Male	Cleveland	asymptomatic	130.0	131.0	False	normal	115
301	302	57	Female	Cleveland	atypical angina	130.0	236.0	False	lv hypertrophy	174
508	509	47	Male	Hungary	asymptomatic	150.0	226.0	False	normal	98
748	749	56	Male	VA Long Beach	asymptomatic	120.0	100.0	False	normal	120

299 rows × 16 columns

```
convert={"sex":{"Female":0,"Male":1}}
```

```
data=data.replace(convert)
```

```
data
```

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch
0	1	63	1	Cleveland	typical angina	145.0	233.0	True	lv hypertrophy	150.0
1	2	67	1	Cleveland	asymptomatic	160.0	286.0	False	lv hypertrophy	108.0
2	3	67	1	Cleveland	asymptomatic	120.0	229.0	False	lv hypertrophy	129.0
3	4	37	1	Cleveland	non-anginal	130.0	250.0	False	normal	187.0
4	5	41	0	Cleveland	atypical angina	130.0	204.0	False	lv hypertrophy	172.0
...	...	...	...	...	...	...	...	...	...	...
299	300	68	1	Cleveland	asymptomatic	144.0	193.0	True	normal	141.0
300	301	57	1	Cleveland	asymptomatic	130.0	131.0	False	normal	115.0
301	302	57	0	Cleveland	atypical angina	130.0	236.0	False	lv hypertrophy	174.0
508	509	47	1	Hungary	asymptomatic	150.0	226.0	False	normal	98.0
748	749	56	1	VA Long Beach	asymptomatic	120.0	100.0	False	normal	120.0

299 rows × 16 columns

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
```

```
data["sex"]=encoder.fit_transform(data["sex"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
```

```
play_te
```

```
{0: 0, 1: 1}
```

```
data
```

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch
0	1	63	1	Cleveland	typical angina	145.0	233.0	True	lv hypertrophy	150.0
1	2	67	1	Cleveland	asymptomatic	160.0	286.0	False	lv hypertrophy	108.0
2	3	67	1	Cleveland	asymptomatic	120.0	229.0	False	lv hypertrophy	129.0
3	4	37	1	Cleveland	non-anginal	130.0	250.0	False	normal	187.0
4	5	41	0	Cleveland	atypical angina	130.0	204.0	False	lv hypertrophy	172.0
...	...	...	...	...	...	...	...	...	...	...
299	300	68	1	Cleveland	asymptomatic	144.0	193.0	True	normal	141.0
300	301	57	1	Cleveland	asymptomatic	130.0	131.0	False	normal	115.0
301	302	57	0	Cleveland	atypical angina	130.0	236.0	False	lv hypertrophy	174.0
508	509	47	1	Hungary	asymptomatic	150.0	226.0	False	normal	98.0
748	749	56	1	VA Long Beach	asymptomatic	120.0	100.0	False	normal	120.0

299 rows × 16 columns

```
data["dataset"]=encoder.fit_transform(data["dataset"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
{0: 'Cleveland', 1: 'Hungary', 2: 'VA Long Beach'}
```

```
data
```

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch
0	1	63	1	0	typical angina	145.0	233.0	True	lv hypertrophy	150.0
1	2	67	1	0	asymptomatic	160.0	286.0	False	lv hypertrophy	108.0
2	3	67	1	0	asymptomatic	120.0	229.0	False	lv hypertrophy	129.0
3	4	37	1	0	non-anginal	130.0	250.0	False	normal	187.0
4	5	41	0	0	atypical angina	130.0	204.0	False	lv hypertrophy	172.0
...	...	...	...	...	...	...	...	...	...	...
299	300	68	1	0	asymptomatic	144.0	193.0	True	normal	141.0
300	301	57	1	0	asymptomatic	130.0	131.0	False	normal	115.0

```
data["id"]=encoder.fit_transform(data["id"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
{0: 1,
 1: 2,
 2: 3,
 3: 4,
 4: 5,
 5: 6,
 6: 7,
 7: 8,
 8: 9,
 9: 10,
 10: 11,
 11: 12,
 12: 13,
 13: 14,
 14: 15,
 15: 16,
 16: 17,
 17: 18,
 18: 19,
 19: 20,
 20: 21,
 21: 22,
 22: 23,
 23: 24,
 24: 25,
 25: 26,
 26: 27,
 27: 28,
 28: 29,
 29: 30,
 30: 31,
 31: 32,
 32: 33,
 33: 34,
 34: 35,
 35: 36,
 36: 37,
 37: 38,
 38: 39,
 39: 40,
 40: 41,
 41: 42,
 42: 43,
 43: 44,
 44: 45,
 45: 46,
 46: 47,
 47: 48,
 48: 49,
 49: 50,
 50: 51,
 51: 52,
 52: 53,
 53: 54,
 54: 55,
 55: 56,
 56: 57,
```

```
data["cp"]=encoder.fit_transform(data["cp"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
     {0: 'asymptomatic',
      1: 'atypical angina',
      2: 'non-anginal',
      3: 'typical angina'}
```

```
data["trestbps"]=encoder.fit_transform(data["trestbps"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
     {0: 94.0,
      1: 100.0,
      2: 101.0,
      3: 102.0,
      4: 104.0,
      5: 105.0,
      6: 106.0,
      7: 108.0,
      8: 110.0,
      9: 112.0,
      10: 114.0,
      11: 115.0,
      12: 117.0,
      13: 118.0,
      14: 120.0,
      15: 122.0,
      16: 123.0,
      17: 124.0,
      18: 125.0,
      19: 126.0,
      20: 128.0,
      21: 129.0,
      22: 130.0,
      23: 132.0,
      24: 134.0,
      25: 135.0,
      26: 136.0,
      27: 138.0,
      28: 140.0,
      29: 142.0,
      30: 144.0,
      31: 145.0,
      32: 146.0,
      33: 148.0,
      34: 150.0,
      35: 152.0,
      36: 154.0,
      37: 155.0,
      38: 156.0,
      39: 158.0,
      40: 160.0,
      41: 164.0,
      42: 165.0,
      43: 170.0,
      44: 172.0,
      45: 174.0,
      46: 178.0,
      47: 180.0,
      48: 192.0,
      49: 200.0}
```

```
data["chol"]=encoder.fit_transform(data["chol"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
     95: 265.0,
     96: 266.0,
     97: 267.0,
     98: 268.0,
     99: 269.0,
     100: 270.0,
     101: 271.0,
     102: 273.0,
     103: 274.0,
     104: 275.0,
     105: 276.0,
     106: 277.0,
     107: 278.0,
     108: 281.0,
     109: 282.0,
     110: 283.0,
     111: 284.0,
     112: 286.0,
     113: 288.0,
     114: 289.0,
     115: 290.0,
     116: 293.0,
     117: 294.0,
     118: 295.0,
     119: 298.0,
     120: 299.0,
     121: 300.0,
     122: 302.0,
     123: 303.0,
     124: 304.0,
     125: 305.0,
     126: 306.0,
     127: 307.0,
     128: 308.0,
     129: 309.0,
     130: 311.0,
     131: 313.0,
     132: 315.0,
     133: 318.0,
     134: 319.0,
     135: 321.0,
     136: 322.0,
     137: 325.0,
     138: 326.0,
     139: 327.0,
     140: 330.0,
     141: 335.0,
     142: 340.0,
     143: 341.0,
     144: 342.0,
     145: 353.0,
     146: 354.0,
     147: 360.0,
     148: 394.0,
     149: 407.0,
     150: 409.0,
     151: 417.0,
     152: 564.0}
```

```
data["fbs"]=encoder.fit_transform(data["fbs"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
    {0: False, 1: True}
```

```
data["restecg"]=encoder.fit_transform(data["restecg"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
{0: 'lv hypertrophy', 1: 'normal', 2: 'st-t abnormality'}
```

```
data
```

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch	exang	oldpeak
0	0	63	1	0	3	31	65	1	0	150.0	False	2.3
1	1	67	1	0	0	40	112	0	0	108.0	True	1.5
2	2	67	1	0	0	14	61	0	0	129.0	True	2.6
3	3	37	1	0	2	22	81	0	1	187.0	False	3.5
4	4	41	0	0	1	22	36	0	0	172.0	False	1.4
...	...	...	...	...	...	...	...	...	...	...	...	...
299	294	68	1	0	0	30	27	1	1	141.0	False	3.4
300	295	57	1	0	0	22	2	0	1	115.0	True	1.2
301	296	57	0	0	1	22	68	0	0	174.0	False	0.0

```
data["thalch"]=encoder.fit_transform(data["thalch"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
 34: 133.0,
 35: 134.0,
 36: 136.0,
 37: 137.0,
 38: 138.0,
 39: 139.0,
 40: 140.0,
 41: 141.0,
 42: 142.0,
 43: 143.0,
 44: 144.0,
 45: 145.0,
 46: 146.0,
 47: 147.0,
 48: 148.0,
 49: 149.0,
 84: 186.0,
 85: 187.0,
 86: 188.0,
 87: 190.0,
 88: 192.0,
 89: 194.0,
 90: 195.0,
 91: 202.0}
```

```
data["exang"]=encoder.fit_transform(data["exang"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
{0: False, 1: True}
```

```
data["oldpeak"]=encoder.fit_transform(data["oldpeak"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
{0: 0.0,
 1: 0.1,
 2: 0.2,
 3: 0.3,
 4: 0.4,
 5: 0.5,
 6: 0.6,
 7: 0.7,
 8: 0.8,
 9: 0.9,
 10: 1.0,
 11: 1.1,
 12: 1.2,
 13: 1.3,
 14: 1.4,
 15: 1.5,
 16: 1.6,
 17: 1.8,
 18: 1.9,
 19: 2.0,
 20: 2.1,
 21: 2.2,
 22: 2.3,
 23: 2.4,
 24: 2.5,
 25: 2.6,
 26: 2.8,
 27: 2.9,
 28: 3.0,
 29: 3.1,
 30: 3.2,
 31: 3.4,
 32: 3.5,
 33: 3.6,
 34: 3.8,
 35: 4.0,
 36: 4.2,
 37: 4.4,
 38: 5.6,
 39: 6.2}
```

```
data["slope"]=encoder.fit_transform(data["slope"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
{0: 'downsloping', 1: 'flat', 2: 'upsloping'}
```

```
data["ca"]=encoder.fit_transform(data["ca"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
{0: 0.0, 1: 1.0, 2: 2.0, 3: 3.0}
```

```
data
```

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch	exang	oldpeak
0	0	63	1	0	3	31	65	1	0	50	0	22
1	1	67	1	0	0	40	112	0	0	11	1	15
2	2	67	1	0	0	14	61	0	0	30	1	25
3	3	37	1	0	2	22	81	0	1	85	0	32
4	4	41	0	0	1	22	36	0	0	72	0	14
...	...	...	...	...	...	...	...	...	...	...	...	...
299	294	68	1	0	0	30	27	1	1	41	0	31
300	295	57	1	0	0	22	2	0	1	17	1	12
301	296	57	0	0	1	22	68	0	0	74	0	0

```
data["thal"]=encoder.fit_transform(data["thal"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
    {0: 'fixed defect', 1: 'normal', 2: 'reversable defect'}
```

```
data["num"]=encoder.fit_transform(data["num"])
play_te=dict(zip(encoder.transform(encoder.classes_),encoder.classes_))
play_te
```

```
    {0: 0, 1: 1, 2: 2, 3: 3, 4: 4}
```

```
data
```

1 to 100 of 299 entries Filter

index	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch	exang	oldpeak	slope	ca	t
0	0	63	1	0	3	31	65	1	0	50	0	22	0	0	
1	1	67	1	0	0	40	112	0	0	11	1	15	1	3	
2	2	67	1	0	0	14	61	0	0	30	1	25	1	2	
3	3	37	1	0	2	22	81	0	1	85	0	32	0	0	
4	4	41	0	0	1	22	36	0	0	72	0	14	2	0	
5	5	56	1	0	1	14	68	0	1	77	0	8	2	0	
6	6	62	0	0	0	28	98	0	0	60	0	33	0	2	
7	7	57	0	0	0	14	146	0	1	63	1	6	2	0	
8	8	63	1	0	0	22	84	0	0	47	0	14	1	1	
9	9	53	1	0	0	28	35	1	0	55	1	29	0	0	
10	10	57	1	0	0	28	26	0	1	48	0	4	1	0	
11	11	56	0	0	1	28	117	0	0	53	0	13	1	0	
12	12	56	1	0	2	22	86	1	0	42	1	6	1	1	
13	13	44	1	0	1	14	93	0	1	73	0	0	2	0	
14	14	52	1	0	2	44	32	1	1	62	0	5	2	0	
15	15	57	1	0	2	34	10	0	1	74	0	16	2	0	
16	16	48	1	0	1	8	61	0	1	68	0	10	0	0	
17	17	54	1	0	0	28	70	0	1	60	0	12	2	0	
18	18	48	0	0	2	22	104	0	1	39	0	2	2	0	
19	19	49	1	0	1	22	96	0	1	71	0	6	2	0	
20	20	64	1	0	3	8	43	0	0	44	1	17	1	0	
21	21	58	0	0	3	34	110	1	0	62	0	10	2	0	
22	22	58	1	0	1	14	111	0	0	60	0	17	1	0	
23	23	58	1	0	2	23	56	0	0	73	0	30	2	2	
24	24	60	1	0	0	22	38	0	0	33	1	23	1	2	
25	25	50	0	0	2	14	51	0	1	58	0	16	1	0	
26	26	58	0	0	2	14	142	0	1	72	0	0	2	0	
27	27	66	0	0	3	34	58	0	1	16	0	25	0	0	
28	28	43	1	0	0	34	78	0	1	71	0	15	2	0	
29	29	40	1	0	0	8	9	0	0	16	1	19	1	0	
30	30	69	0	0	3	28	70	0	1	51	0	17	2	2	
31	31	60	1	0	0	12	62	1	1	60	1	14	2	2	
32	32	64	1	0	2	28	141	0	1	58	0	0	2	0	
33	33	59	1	0	0	25	66	0	1	61	0	5	1	0	
34	34	44	1	0	2	22	65	0	1	78	1	4	2	0	
35	35	42	1	0	0	28	58	0	1	77	0	0	2	0	
36	36	43	1	0	0	14	16	0	0	21	1	24	1	0	
37	37	57	1	0	0	34	105	0	0	14	1	6	1	1	
38	38	55	1	0	0	23	145	0	1	33	1	12	1	1	
39	39	61	1	0	2	34	74	1	1	37	1	10	1	0	
40	40	65	0	0	0	34	57	0	0	16	0	10	1	3	
41	41	40	1	0	3	28	32	0	1	77	1	14	2	0	
42	42	71	0	0	1	40	122	0	1	62	0	4	2	2	
43	43	59	1	0	2	34	44	1	1	57	0	16	2	0	
44	44	61	0	0	0	22	140	0	0	69	0	0	2	0	
45	45	58	1	0	2	9	62	0	0	65	0	24	1	1	
46	46	51	1	0	2	8	14	0	1	24	0	6	2	0	
47	47	50	1	0	0	34	74	0	0	29	0	25	1	0	
48	48	65	0	0	2	28	151	1	0	57	0	8	2	1	
49	49	53	1	0	2	22	30	1	0	52	0	12	0	0	
50	50	41	0	0	1	5	31	0	1	68	0	0	2	1	
51	51	65	1	0	0	14	16	0	1	40	0	4	2	0	
52	52	44	1	0	0	9	115	0	0	53	0	0	2	1	
53	53	44	1	0	1	22	51	0	0	86	0	0	2	0	
54	54	60	1	0	0	22	83	0	1	44	1	14	2	1	
55	55	54	1	0	0	17	96	0	0	12	1	21	1	1	
56	56	50	1	0	2	28	65	0	1	63	0	6	1	1	
57	57	41	1	0	0	8	12	0	0	58	0	0	2	0	
58	58	54	1	0	2	18	102	0	0	52	0	5	0	1	
59	59	51	1	0	3	18	45	0	0	26	1	14	2	1	
60	60	51	0	0	0	22	125	0	1	42	1	12	1	0	
61	61	46	0	0	2	29	16	0	0	60	1	14	0	0	
62	62	58	1	0	0	20	48	0	0	32	1	21	1	3	
63	63	54	0	0	2	25	124	1	1	70	0	0	2	0	
64	64	54	1	0	0	14	25	0	1	15	0	14	1	1	
65	65	60	1	0	0	31	109	0	0	42	1	26	1	2	
66	66	60	1	0	2	28	22	0	0	55	0	28	1	0	
67	67	54	1	0	2	34	64	0	0	65	0	16	2	0	
68	68	59	1	0	0	43	138	0	0	40	1	31	0	0	
69	69	46	1	0	2	34	63	0	1	47	0	33	1	0	
70	70	65	0	0	2	37	99	0	1	48	0	8	2	0	

70	70	65	0	0	2	37	99	0	1	48	0	8	2	0
71	71	67	1	0	0	18	84	1	1	63	0	2	1	2
72	72	62	1	0	0	14	97	0	1	7	1	17	1	2
73	73	65	1	0	0	8	79	0	0	58	0	6	2	2
74	74	44	1	0	0	8	30	0	0	76	0	0	2	1
75	75	65	0	0	2	40	147	0	0	51	0	8	2	0
76	76	60	1	0	0	18	88	0	0	41	1	26	1	1
77	77	51	0	0	2	28	128	0	0	42	0	15	2	1
78	78	48	1	0	1	22	76	0	0	79	0	2	1	0
79	79	58	1	0	0	34	100	0	0	13	1	8	2	0
80	80	45	1	0	0	4	40	0	0	48	1	28	1	0
81	81	53	0	0	0	22	94	0	0	43	0	4	1	0
82	82	39	1	0	2	28	135	0	0	81	0	0	2	0
83	83	68	1	0	2	47	103	1	0	50	1	16	1	0
84	84	52	1	0	1	14	137	0	1	72	0	2	2	0
85	85	44	1	0	2	28	67	0	0	79	0	0	2	0
86	86	47	1	0	2	27	87	0	0	56	0	0	2	0
88	87	53	0	0	0	27	66	0	0	60	0	0	2	0
89	88	51	0	0	2	22	86	0	0	49	0	5	2	0
90	89	66	1	0	0	14	122	0	0	51	0	4	1	0
91	90	62	0	0	0	40	7	0	0	45	0	39	0	3
92	91	62	1	0	2	22	63	0	1	46	0	17	1	3
93	92	44	0	0	2	7	3	0	1	75	0	6	1	0
94	93	63	0	0	2	25	82	0	0	72	0	0	2	0
95	94	52	1	0	0	20	85	0	1	61	1	0	2	1
96	95	59	1	0	0	8	70	0	0	42	1	12	1	1
97	96	60	0	0	0	34	88	0	0	57	0	25	1	2
98	97	52	1	0	1	24	34	0	1	58	0	8	2	1
99	98	48	1	0	0	15	54	0	0	84	0	0	2	0
100	99	45	1	0	0	11	90	0	0	83	0	0	2	0

Show 100 per page   1 2 3

```
x=data.drop('num',axis=1)
y=data['num']
```

```
model=DecisionTreeClassifier()
```

```
model.fit(x,y)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
# Access  the tree attributes
tree_=model.tree_
root_node =0  # as root node with index will be 0
feature_names = x.columns
# to get the feature indecx of the root node'
root_feature_index = tree_.feature[root_node]
# to get feature name from the original value
root_feature_name=feature_names[root_feature_index]
```

```
print("Index is", root_feature_index)
print("Feature Name is",root_feature_name)
print("Root Node Impurity is ",tree_.impurity[root_node])
```

```
Index is 4
Feature Name is cp
Root Node Impurity is  0.6492768537264684
```

```
# visualize decision trees
from sklearn import tree
import matplotlib.pyplot as plt
plt.figure(figsize=(36,12))
tree.plot_tree(model,feature_names=x.columns,filled=True,rounded=True)
plt.show()
```