

Vijay 02-09-2023

In [342]:

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
```

In [382]:

```
1 from sklearn.linear_model import LogisticRegression
2 a=pd.read_csv(r"C:\USERS\user\Downloads\C8_loan-train.csv")
3 a
```

1	LP001003	Male	Yes	1	Graduate	No	4583	150
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	235
4	LP001008	Male	No	0	Graduate	No	6000	
...	
609	LP002978	Female	No	0	Graduate	No	2900	
610	LP002979	Male	Yes	3+	Graduate	No	4106	
611	LP002983	Male	Yes	1	Graduate	No	8072	24
612	LP002984	Male	Yes	2	Graduate	No	7583	
613	LP002990	Female	No	0	Graduate	Yes	4583	

614 rows × 13 columns

In [505]:

1	a=a.head(50)
2	a

Out[505]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
0	LP001002	Male	No	0	Graduate	No	5849	0.0
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0
4	LP001008	Male	No	0	Graduate	No	6000	0.0
5	LP001011	Male	Yes	2	Graduate	Yes	5417	4196.0
6	LP001013	Male	Yes	0	Not Graduate	No	2333	1516.0
7	LP001014	Male	Yes	3+	Graduate	No	3036	2504.0
8	LP001018	Male	Yes	2	Graduate	No	4006	1526.0
9	LP001020	Male	Yes	1	Graduate	No	12841	10968.0
10	LP001024	Male	Yes	2	Graduate	No	3200	700.0
11	LP001027	Male	Yes	2	Graduate	NaN	2500	1840.0
12	LP001028	Male	Yes	2	Graduate	No	3073	8106.0
13	LP001029	Male	No	0	Graduate	No	1853	2840.0
14	LP001030	Male	Yes	2	Graduate	No	1299	1086.0
15	LP001032	Male	No	0	Graduate	No	4950	0.0
16	LP001034	Male	No	1	Not Graduate	No	3596	0.0
17	LP001036	Female	No	0	Graduate	No	3510	0.0
18	LP001038	Male	Yes	0	Not Graduate	No	4887	0.0
19	LP001041	Male	Yes	0	Graduate	NaN	2600	3500.0
20	LP001043	Male	Yes	0	Not Graduate	No	7660	0.0
21	LP001046	Male	Yes	1	Graduate	No	5955	5625.0
22	LP001047	Male	Yes	0	Not Graduate	No	2600	1911.0
23	LP001050	NaN	Yes	2	Not Graduate	No	3365	1917.0
24	LP001052	Male	Yes	1	Graduate	NaN	3717	2925.0
25	LP001066	Male	Yes	0	Graduate	Yes	9560	0.0
26	LP001068	Male	Yes	0	Graduate	No	2799	2253.0
27	LP001073	Male	Yes	2	Not Graduate	No	4226	1040.0
28	LP001086	Male	No	0	Not Graduate	No	1442	0.0
29	LP001087	Female	No	2	Graduate	NaN	3750	2083.0
30	LP001091	Male	Yes	1	Graduate	NaN	4166	3369.0
31	LP001095	Male	No	0	Graduate	No	3167	0.0
32	LP001097	Male	No	1	Graduate	Yes	4692	0.0
33	LP001098	Male	Yes	0	Graduate	No	3500	1667.0

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
34	LP001100	Male	No	3+	Graduate	No	12500	3000.0
35	LP001106	Male	Yes	0	Graduate	No	2275	2067.0
36	LP001109	Male	Yes	0	Graduate	No	1828	1330.0
37	LP001112	Female	Yes	0	Graduate	No	3667	1459.0
38	LP001114	Male	No	0	Graduate	No	4166	7210.0
39	LP001116	Male	No	0	Not Graduate	No	3748	1668.0
40	LP001119	Male	No	0	Graduate	No	3600	0.0
41	LP001120	Male	No	0	Graduate	No	1800	1213.0
42	LP001123	Male	Yes	0	Graduate	No	2400	0.0
43	LP001131	Male	Yes	0	Graduate	No	3941	2336.0
44	LP001136	Male	Yes	0	Not Graduate	Yes	4695	0.0
45	LP001137	Female	No	0	Graduate	No	3410	0.0
46	LP001138	Male	Yes	1	Graduate	No	5649	0.0
47	LP001144	Male	Yes	0	Graduate	No	5821	0.0
48	LP001146	Female	Yes	0	Graduate	No	2645	3440.0
49	LP001151	Female	No	0	Graduate	No	4000	2275.0

In [506]: 1 `from sklearn.linear_model import LogisticRegression`

In [507]: 1 `a.columns`

Out[507]: Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'], dtype='object')

```
In [508]: 1 b=a[['ApplicantIncome', 'CoapplicantIncome']]  
          2 b
```

Out[508]:

	ApplicantIncome	CoapplicantIncome
0	5849	0.0
1	4583	1508.0
2	3000	0.0
3	2583	2358.0
4	6000	0.0
5	5417	4196.0
6	2333	1516.0
7	3036	2504.0
8	4006	1526.0
9	12841	10968.0
10	3200	700.0
11	2500	1840.0
12	3073	8106.0
13	1853	2840.0
14	1299	1086.0
15	4950	0.0
16	3596	0.0
17	3510	0.0
18	4887	0.0
19	2600	3500.0
20	7660	0.0
21	5955	5625.0
22	2600	1911.0
23	3365	1917.0
24	3717	2925.0
25	9560	0.0
26	2799	2253.0
27	4226	1040.0
28	1442	0.0
29	3750	2083.0
30	4166	3369.0
31	3167	0.0
32	4692	0.0
33	3500	1667.0
34	12500	3000.0
35	2275	2067.0
36	1828	1330.0
37	3667	1459.0
38	4166	7210.0

	ApplicantIncome	CoapplicantIncome
39	3748	1668.0
40	3600	0.0
41	1800	1213.0
42	2400	0.0
43	3941	2336.0
44	4695	0.0
45	3410	0.0
46	5649	0.0
47	5821	0.0
48	2645	3440.0
49	4000	2275.0

```
In [509]: 1 c=b.iloc[:,0:5]
          2 d=b.iloc[:, -1]
```

```
In [510]: 1 c.shape
```

```
Out[510]: (50, 2)
```

```
In [511]: 1 d.shape
```

```
Out[511]: (50,)
```

```
In [512]: 1 from sklearn.preprocessing import StandardScaler
          2 fs=StandardScaler().fit_transform(c)
          3 fs
```

```
Out[512]: array([[ 7.28503730e-01, -8.20253257e-01],
 [ 1.83353167e-01, -1.43855606e-01],
 [-4.98300340e-01, -8.20253257e-01],
 [-6.77864151e-01,  2.37403016e-01],
 [ 7.93525637e-01, -8.20253257e-01],
 [ 5.42480789e-01,  1.06181872e+00],
 [-7.85516316e-01, -1.40267290e-01],
 [-4.82798429e-01,  3.02889791e-01],
 [-6.51080293e-02, -1.35781894e-01],
 [ 3.73931947e+00,  4.09932859e+00],
 [-4.12178609e-01, -5.06275568e-01],
 [-7.13604670e-01,  5.05952619e-03],
 [-4.66865908e-01,  2.81560839e+00],
 [-9.92208473e-01,  4.53599082e-01],
 [-1.23076567e+00, -3.33139299e-01],
 [ 3.41386545e-01, -8.20253257e-01],
 [-2.41657580e-01, -8.20253257e-01],
 [-2.78689924e-01, -8.20253257e-01],
 [ 3.14258200e-01, -8.20253257e-01],
 [-6.70543804e-01,  7.49635189e-01],
 [ 1.50833601e+00, -8.20253257e-01],
 [ 7.74148248e-01,  1.70278175e+00],
 [-6.70543804e-01,  3.69058347e-02],
 [-3.41128180e-01,  3.95970720e-02],
 [-1.89553932e-01,  4.91724945e-01],
 [ 2.32649246e+00, -8.20253257e-01],
 [-5.84852681e-01,  1.90306363e-01],
 [ 2.96258758e-02, -3.53772119e-01],
 [-1.16918863e+00, -8.20253257e-01],
 [-1.75343846e-01,  1.14054638e-01],
 [ 3.78935620e-03,  6.90876508e-01],
 [-4.26388694e-01, -8.20253257e-01],
 [ 2.30289511e-01, -8.20253257e-01],
 [-2.82996011e-01, -7.25378170e-02],
 [ 3.59248192e+00,  5.25365411e-01],
 [-8.10491618e-01,  1.06878005e-01],
 [-1.00297369e+00, -2.23695647e-01],
 [-2.11084365e-01, -1.65834045e-01],
 [ 3.78935620e-03,  2.41371694e+00],
 [-1.76205063e-01, -7.20892775e-02],
 [-2.39935145e-01, -8.20253257e-01],
 [-1.01503073e+00, -2.76174776e-01],
 [-7.56665536e-01, -8.20253257e-01],
 [-9.30975921e-02,  2.27535146e-01],
 [ 2.31581337e-01, -8.20253257e-01],
 [-3.21750790e-01, -8.20253257e-01],
 [ 6.42381998e-01, -8.20253257e-01],
 [ 7.16446687e-01, -8.20253257e-01],
 [-6.51166414e-01,  7.22722816e-01],
 [-6.76916812e-02,  2.00174233e-01]])
```

```
In [513]: 1 logr=LogisticRegression()
          2 logr.fit(fs,d)
```

```
Out[513]: LogisticRegression()
```



```
In [514]: 1 e=[[777,55]]
```

```
In [515]: 1 prediction=logr.predict(e)
          2 prediction
```

```
Out[515]: array([3000.])
```

```
In [516]: 1 logr.classes_
```

```
Out[516]: array([  0.,   700.,  1040.,  1086.,  1213.,  1330.,  1459.,  1508.,
                1516.,  1526.,  1667.,  1668.,  1840.,  1911.,  1917.,  2067.,
                2083.,  2253.,  2275.,  2336.,  2358.,  2504.,  2840.,  2925.,
                3000.,  3369.,  3440.,  3500.,  4196.,  5625.,  7210.,  8106.,
                10968.])
```

```
In [517]: 1 logr.predict_proba(e)[0][0]
```

```
Out[517]: 1.638791572494187e-296
```

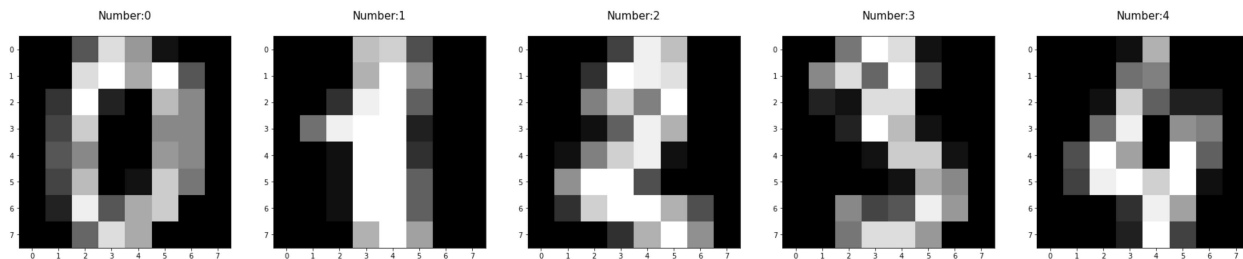
```
In [518]: 1 import re
          2 from sklearn.datasets import load_digits
          3 import numpy as np
          4 import pandas as pd
          5 import matplotlib.pyplot as plt
          6 import seaborn as sns
```

```
In [519]: 1 from sklearn.linear_model import LogisticRegression
          2 from sklearn.model_selection import train_test_split
```

```
In [520]: 1 digits=load_digits()
          2 digits
```

```
digits.data,
['pixel_1_0',
 'pixel_1_4',
 'pixel_1_5',
 'pixel_1_6',
 'pixel_1_7',
 'pixel_2_0',
 'pixel_2_1',
 'pixel_2_2',
 'pixel_2_3',
 'pixel_2_4',
 'pixel_2_5',
 'pixel_2_6',
 'pixel_2_7',
 'pixel_3_0',
 'pixel_3_1',
 'pixel_3_2',
 'pixel_3_3',
 'pixel_3_4',
 'pixel_3_5',
 'pixel_3_6',
 'pixel_3_7']
```

```
In [521]: 1 plt.figure(figsize=(50,25))
2         for index,(image,label) in enumerate(zip(digits.data[0:8],digits.target[0:5])):
3             plt.subplot(1,8,index+1)
4             plt.imshow(np.reshape(image,(8,8)),cmap=plt.cm.gray)
5             plt.title('Number:%i\n'%label,fontsize=15)
```



```
In [522]: 1 x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_size=0
```

```
In [523]: 1 print(x_train.shape)
2 print(x_test.shape)
3 print(y_train.shape)
4 print(y_test.shape)
```

```
(736, 64)
(1061, 64)
(736,)
(1061,)
```

```
In [524]: 1 logre=LogisticRegression(max_iter=10000)
2         logre.fit(x_train,y_train)
3
```

Out[524]: LogisticRegression(max_iter=10000)

```
In [525]: 1 print(logre.predict(x_test))
```

```
[5 8 9 ... 9 3 6]
```

```
In [526]: 1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
```

```
In [527]: 1 a=pd.read_csv(r"C:\USERS\user\Downloads\C8_loan-train.csv")
```

In [529]:

```
1 a=a.head(50)
2 a
```

Out[529]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
0	LP001002	Male	No	0	Graduate	No	5849	0.0
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0
4	LP001008	Male	No	0	Graduate	No	6000	0.0
5	LP001011	Male	Yes	2	Graduate	Yes	5417	4196.0
6	LP001013	Male	Yes	0	Not Graduate	No	2333	1516.0
7	LP001014	Male	Yes	3+	Graduate	No	3036	2504.0
8	LP001018	Male	Yes	2	Graduate	No	4006	1526.0
9	LP001020	Male	Yes	1	Graduate	No	12841	10968.0
10	LP001024	Male	Yes	2	Graduate	No	3200	700.0
11	LP001027	Male	Yes	2	Graduate	NaN	2500	1840.0
12	LP001028	Male	Yes	2	Graduate	No	3073	8106.0
13	LP001029	Male	No	0	Graduate	No	1853	2840.0
14	LP001030	Male	Yes	2	Graduate	No	1299	1086.0
15	LP001032	Male	No	0	Graduate	No	4950	0.0
16	LP001034	Male	No	1	Not Graduate	No	3596	0.0
17	LP001036	Female	No	0	Graduate	No	3510	0.0
18	LP001038	Male	Yes	0	Not Graduate	No	4887	0.0
19	LP001041	Male	Yes	0	Graduate	NaN	2600	3500.0
20	LP001043	Male	Yes	0	Not Graduate	No	7660	0.0
21	LP001046	Male	Yes	1	Graduate	No	5955	5625.0
22	LP001047	Male	Yes	0	Not Graduate	No	2600	1911.0
23	LP001050	NaN	Yes	2	Not Graduate	No	3365	1917.0
24	LP001052	Male	Yes	1	Graduate	NaN	3717	2925.0
25	LP001066	Male	Yes	0	Graduate	Yes	9560	0.0
26	LP001068	Male	Yes	0	Graduate	No	2799	2253.0
27	LP001073	Male	Yes	2	Not Graduate	No	4226	1040.0
28	LP001086	Male	No	0	Not Graduate	No	1442	0.0
29	LP001087	Female	No	2	Graduate	NaN	3750	2083.0
30	LP001091	Male	Yes	1	Graduate	NaN	4166	3369.0
31	LP001095	Male	No	0	Graduate	No	3167	0.0
32	LP001097	Male	No	1	Graduate	Yes	4692	0.0
33	LP001098	Male	Yes	0	Graduate	No	3500	1667.0

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
34	LP001100	Male	No	3+	Graduate	No	12500	3000.0
35	LP001106	Male	Yes	0	Graduate	No	2275	2067.0
36	LP001109	Male	Yes	0	Graduate	No	1828	1330.0
37	LP001112	Female	Yes	0	Graduate	No	3667	1459.0
38	LP001114	Male	No	0	Graduate	No	4166	7210.0
39	LP001116	Male	No	0	Not Graduate	No	3748	1668.0
40	LP001119	Male	No	0	Graduate	No	3600	0.0
41	LP001120	Male	No	0	Graduate	No	1800	1213.0
42	LP001123	Male	Yes	0	Graduate	No	2400	0.0
43	LP001131	Male	Yes	0	Graduate	No	3941	2336.0
44	LP001136	Male	Yes	0	Not Graduate	Yes	4695	0.0
45	LP001137	Female	No	0	Graduate	No	3410	0.0
46	LP001138	Male	Yes	1	Graduate	No	5649	0.0
47	LP001144	Male	Yes	0	Graduate	No	5821	0.0
48	LP001146	Female	Yes	0	Graduate	No	2645	3440.0
49	LP001151	Female	No	0	Graduate	No	4000	2275.0

```
In [530]: 1 b=a[[ 'ApplicantIncome', 'CoapplicantIncome', 'Loan_Status']]  
          2 b
```

Out[530]:

	ApplicantIncome	CoapplicantIncome	Loan_Status
0	5849	0.0	Y
1	4583	1508.0	N
2	3000	0.0	Y
3	2583	2358.0	Y
4	6000	0.0	Y
5	5417	4196.0	Y
6	2333	1516.0	Y
7	3036	2504.0	N
8	4006	1526.0	Y
9	12841	10968.0	N
10	3200	700.0	Y
11	2500	1840.0	Y
12	3073	8106.0	Y
13	1853	2840.0	N
14	1299	1086.0	Y
15	4950	0.0	Y
16	3596	0.0	Y
17	3510	0.0	N
18	4887	0.0	N
19	2600	3500.0	Y
20	7660	0.0	N
21	5955	5625.0	Y
22	2600	1911.0	N
23	3365	1917.0	N
24	3717	2925.0	N
25	9560	0.0	Y
26	2799	2253.0	Y
27	4226	1040.0	Y
28	1442	0.0	N
29	3750	2083.0	Y
30	4166	3369.0	N
31	3167	0.0	N
32	4692	0.0	N
33	3500	1667.0	Y
34	12500	3000.0	N
35	2275	2067.0	Y
36	1828	1330.0	N
37	3667	1459.0	Y
38	4166	7210.0	Y

	ApplicantIncome	CoapplicantIncome	Loan_Status
39	3748	1668.0	Y
40	3600	0.0	N
41	1800	1213.0	Y
42	2400	0.0	Y
43	3941	2336.0	Y
44	4695	0.0	Y
45	3410	0.0	Y
46	5649	0.0	Y
47	5821	0.0	Y
48	2645	3440.0	N
49	4000	2275.0	Y

```
In [531]: 1 b['Loan_Status'].value_counts()
```

Out[531]: Y 32
N 18
Name: Loan_Status, dtype: int64


```
In [532]: 1 x=b.drop('Loan_Status',axis=1)
          2 y=b['Loan_Status']
          3 print(b)
```

	ApplicantIncome	CoapplicantIncome	Loan_Status
0	5849	0.0	Y
1	4583	1508.0	N
2	3000	0.0	Y
3	2583	2358.0	Y
4	6000	0.0	Y
5	5417	4196.0	Y
6	2333	1516.0	Y
7	3036	2504.0	N
8	4006	1526.0	Y
9	12841	10968.0	N
10	3200	700.0	Y
11	2500	1840.0	Y
12	3073	8106.0	Y
13	1853	2840.0	N
14	1299	1086.0	Y
15	4950	0.0	Y
16	3596	0.0	Y
17	3510	0.0	N
18	4887	0.0	N
19	2600	3500.0	Y
20	7660	0.0	N
21	5955	5625.0	Y
22	2600	1911.0	N
23	3365	1917.0	N
24	3717	2925.0	N
25	9560	0.0	Y
26	2799	2253.0	Y
27	4226	1040.0	Y
28	1442	0.0	N
29	3750	2083.0	Y
30	4166	3369.0	N
31	3167	0.0	N
32	4692	0.0	N
33	3500	1667.0	Y
34	12500	3000.0	N
35	2275	2067.0	Y
36	1828	1330.0	N
37	3667	1459.0	Y
38	4166	7210.0	Y
39	3748	1668.0	Y
40	3600	0.0	N
41	1800	1213.0	Y
42	2400	0.0	Y
43	3941	2336.0	Y
44	4695	0.0	Y
45	3410	0.0	Y
46	5649	0.0	Y
47	5821	0.0	Y
48	2645	3440.0	N
49	4000	2275.0	Y

```
In [533]: 1 g1={"Loan_Status":{'g1':1}}
          2 a=a.replace(g1)
          3 print(a)
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	\
0	LP001002	Male	No	0	Graduate	No	
1	LP001003	Male	Yes	1	Graduate	No	
2	LP001005	Male	Yes	0	Graduate	Yes	
3	LP001006	Male	Yes	0	Not Graduate	No	
4	LP001008	Male	No	0	Graduate	No	
5	LP001011	Male	Yes	2	Graduate	Yes	
6	LP001013	Male	Yes	0	Not Graduate	No	
7	LP001014	Male	Yes	3+	Graduate	No	
8	LP001018	Male	Yes	2	Graduate	No	
9	LP001020	Male	Yes	1	Graduate	No	
10	LP001024	Male	Yes	2	Graduate	No	
11	LP001027	Male	Yes	2	Graduate	NaN	
12	LP001028	Male	Yes	2	Graduate	No	
13	LP001029	Male	No	0	Graduate	No	
14	LP001030	Male	Yes	2	Graduate	No	
15	LP001032	Male	No	0	Graduate	No	
16	LP001034	Male	No	1	Not Graduate	No	
17	LP001036	Female	No	0	Graduate	No	

```
In [534]: 1 from sklearn.model_selection import train_test_split
          2 x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

```
In [535]: 1 from sklearn.ensemble import RandomForestClassifier
```

```
In [536]: 1 rfc=RandomForestClassifier()
          2 rfc.fit(x_train,y_train)
```

Out[536]: RandomForestClassifier()

```
In [537]: 1 parameters={'max_depth':[1,2,3,4,5],
          2               'min_samples_leaf':[5,10,15,20,25],
          3               'n_estimators':[10,20,30,40,50]}
```

```
In [538]: 1 from sklearn.model_selection import GridSearchCV
```

```
In [539]: 1 grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
          2 grid_search.fit(x_train,y_train)
```

Out[539]: GridSearchCV(cv=2, estimator=RandomForestClassifier(),
 param_grid={'max_depth': [1, 2, 3, 4, 5],
 'min_samples_leaf': [5, 10, 15, 20, 25],
 'n_estimators': [10, 20, 30, 40, 50]},
 scoring='accuracy')

```
In [540]: 1 grid_search.best_score_
```

Out[540]: 0.6290849673202614

```
In [541]: 1 rfc_best=grid_search.best_estimator_
```

```
In [542]: 1 from sklearn.tree import plot_tree
```

```
In [543]: 1 plt.figure(figsize=(20,10))  
2 plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],f  
3
```

```
Out[543]: [Text(558.0, 407.70000000000005, 'CoapplicantIncome <= 1997.0\ngini = 0.467\nsamples =  
23\nvalue = [13, 22]\nnclass = No'),  
Text(279.0, 135.89999999999998, 'gini = 0.498\nsamples = 13\nvalue = [8, 9]\nnclass = N  
o'),  
Text(837.0, 135.89999999999998, 'gini = 0.401\nsamples = 10\nvalue = [5, 13]\nnclass =  
No')]
```

