

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

```
In [124]: data=pd.read_csv(r"C:\Users\user\Desktop\vicky\C8_loan-train.csv")
```

```
In [125]: data.fillna(value=1)
```

Out[125]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coap
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
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	
612	LP002984	Male	Yes	2	Graduate	No	7583	
613	LP002990	Female	No	0	Graduate	Yes	4583	

614 rows × 13 columns

```
In [126]: data.head()
```

Out[126]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coappli
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	

```
In [127]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Loan_ID               614 non-null    object
1   Gender                601 non-null    object
2   Married               611 non-null    object
3   Dependents            599 non-null    object
4   Education             614 non-null    object
5   Self_Employed         582 non-null    object
6   ApplicantIncome       614 non-null    int64
7   CoapplicantIncome     614 non-null    float64
8   LoanAmount            592 non-null    float64
9   Loan_Amount_Term      600 non-null    float64
10  Credit_History         564 non-null    float64
11  Property_Area         614 non-null    object
12  Loan_Status           614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

```
In [128]: data1=data[['male','age','diabetes','TenYearCHD']]
```

```
-----
KeyError                                Traceback (most recent call last)
<ipython-input-128-5c56fd6bad09> in <module>
----> 1 data1=data[['male','age','diabetes','TenYearCHD']]

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
    3028         if is_iterator(key):
    3029             key = list(key)
-> 3030         indexer = self.loc._get_listlike_indexer(key, axis=1, raise_missing=True)[1]
    3031
    3032         # take() does not accept boolean indexers

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexing.py in _get_listlike_indexer(self, key, axis, raise_missing)
    1264         keyarr, indexer, new_indexer = ax._reindex_non_unique(keyarr)
    1265
-> 1266         self._validate_read_indexer(keyarr, indexer, axis, raise_missing=raise_missing)
    1267         return keyarr, indexer
    1268

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexing.py in _validate_read_indexer(self, key, indexer, axis, raise_missing)
    1306         if missing == len(indexer):
    1307             axis_name = self.obj._get_axis_name(axis)
-> 1308             raise KeyError(f"None of [{key}] are in the [{axis_name}]")
    1309
    1310         ax = self.obj._get_axis(axis)

KeyError: "None of [Index(['male', 'age', 'diabetes', 'TenYearCHD'], dtype='object')] are in the [columns]"
```

```
In [129]: data1['TenYearCHD'].value_counts()
```

```
Out[129]: No      3594
          Yes      644
          Name: TenYearCHD, dtype: int64
```

```
In [130]: x=data1.drop('TenYearCHD',axis=1)
          y=data1['TenYearCHD']
```

```
In [ ]:
```

```
In [131]: g1={"TenYearCHD":{1:"Yes",0:"No",}}
data1=data1.replace(g1)
print(data1)
```

	male	age	diabetes	TenYearCHD
0	1	39	0	No
1	0	46	0	No
2	1	48	0	No
3	0	61	0	Yes
4	0	46	0	No
...
4233	1	50	0	Yes
4234	1	51	0	No
4235	0	48	0	No
4236	0	44	0	No
4237	0	52	0	No

[4238 rows x 4 columns]

```
In [132]: from sklearn.model_selection import train_test_split
```

```
In [133]: x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

```
In [134]: from sklearn.ensemble import RandomForestClassifier
```

```
In [135]: rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[135]: RandomForestClassifier()

```
In [136]: parameters = {'max_depth':[1,2,3,4,5],
                        'min_samples_leaf':[5,10,15,20,25],
                        'n_estimators':[10,20,30,40,50]
                        }
```

```
In [137]: from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[137]: 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 [138]: grid_search.best_score_
```

Out[138]: 0.8506405933917734

```
In [139]: from sklearn.tree import plot_tree
```

```
In [140]: rfc_best=grid_search.best_estimator_
```

```
In [141]: plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'])
```

```
Out[141]: [Text(2232.0, 1902.6000000000001, 'male <= 0.5\ngini = 0.255\nsamples = 1877\nvalue = [2522, 444]\nclass = Yes'),
Text(1116.0, 1359.0, 'diabetes <= 0.5\ngini = 0.209\nsamples = 1063\nvalue = [1506, 202]\nclass = Yes'),
Text(558.0, 815.4000000000001, 'age <= 54.5\ngini = 0.201\nsamples = 1045\nvalue = [1486, 190]\nclass = Yes'),
Text(279.0, 271.79999999999995, 'gini = 0.128\nsamples = 708\nvalue = [1072, 79]\nclass = Yes'),
Text(837.0, 271.79999999999995, 'gini = 0.333\nsamples = 337\nvalue = [414, 111]\nclass = Yes'),
Text(1674.0, 815.4000000000001, 'age <= 60.5\ngini = 0.469\nsamples = 18\nvalue = [20, 12]\nclass = Yes'),
Text(1395.0, 271.79999999999995, 'gini = 0.415\nsamples = 10\nvalue = [12, 5]\nclass = Yes'),
Text(1953.0, 271.79999999999995, 'gini = 0.498\nsamples = 8\nvalue = [8, 7]\nclass = Yes'),
Text(3348.0, 1359.0, 'age <= 57.5\ngini = 0.311\nsamples = 814\nvalue = [1016, 242]\nclass = Yes'),
Text(2790.0, 815.4000000000001, 'age <= 41.5\ngini = 0.261\nsamples = 646\nvalue = [837, 153]\nclass = Yes'),
Text(2511.0, 271.79999999999995, 'gini = 0.166\nsamples = 188\nvalue = [259, 26]\nclass = Yes'),
Text(3069.0, 271.79999999999995, 'gini = 0.295\nsamples = 458\nvalue = [578, 127]\nclass = Yes'),
Text(3906.0, 815.4000000000001, 'diabetes <= 0.5\ngini = 0.444\nsamples = 168\nvalue = [179, 89]\nclass = Yes'),
Text(3627.0, 271.79999999999995, 'gini = 0.43\nsamples = 158\nvalue = [173, 79]\nclass = Yes'),
Text(4185.0, 271.79999999999995, 'gini = 0.469\nsamples = 10\nvalue = [6, 10]\nclass = No')]
```

In []:

In []: