

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

```
In [2]: df = pd.read_csv(r"C:\Users\user\Downloads\C4_framingham.csv")[0:500]
df
```

Out[2]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
0	1	39	4.0	0	0.0	0.0	0	0
1	0	46	2.0	0	0.0	0.0	0	0
2	1	48	1.0	1	20.0	0.0	0	0
3	0	61	3.0	1	30.0	0.0	0	1
4	0	46	3.0	1	23.0	0.0	0	0
...	...	...	...	...	...	...	...	...
495	1	44	1.0	1	20.0	0.0	0	0
496	1	51	2.0	1	13.0	0.0	0	0
497	1	45	3.0	1	30.0	0.0	0	1
498	1	56	4.0	0	0.0	0.0	0	0
499	0	60	NaN	0	0.0	0.0	0	0

500 rows × 9 columns



In [3]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   male                  500 non-null    int64
1   age                   500 non-null    int64
2   education             486 non-null    float64
3   currentSmoker         500 non-null    int64
4   cigsPerDay            498 non-null    float64
5   BPMeds                493 non-null    float64
6   prevalentStroke       500 non-null    int64
7   prevalentHyp          500 non-null    int64
8   diabetes              500 non-null    int64
9   totChol               496 non-null    float64
10  sysBP                 500 non-null    float64
11  diaBP                 500 non-null    float64
12  BMI                   498 non-null    float64
13  heartRate             500 non-null    float64
14  glucose               452 non-null    float64
15  TenYearCHD            500 non-null    int64
dtypes: float64(9), int64(7)
memory usage: 62.6 KB
```

In [4]: df1 = df.fillna(0)  
df1

Out[4]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
0	1	39	4.0	0	0.0	0.0	0	0
1	0	46	2.0	0	0.0	0.0	0	0
2	1	48	1.0	1	20.0	0.0	0	0
3	0	61	3.0	1	30.0	0.0	0	1
4	0	46	3.0	1	23.0	0.0	0	0
...	...	...	...	...	...	...	...	...
495	1	44	1.0	1	20.0	0.0	0	0
496	1	51	2.0	1	13.0	0.0	0	0
497	1	45	3.0	1	30.0	0.0	0	1
498	1	56	4.0	0	0.0	0.0	0	0
499	0	60	0.0	0	0.0	0.0	0	0

500 rows × 16 columns

```
In [5]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   male                  500 non-null    int64
1   age                   500 non-null    int64
2   education             500 non-null    float64
3   currentSmoker         500 non-null    int64
4   cigsPerDay            500 non-null    float64
5   BPMeds                500 non-null    float64
6   prevalentStroke       500 non-null    int64
7   prevalentHyp          500 non-null    int64
8   diabetes              500 non-null    int64
9   totChol               500 non-null    float64
10  sysBP                 500 non-null    float64
11  diaBP                 500 non-null    float64
12  BMI                   500 non-null    float64
13  heartRate             500 non-null    float64
14  glucose               500 non-null    float64
15  TenYearCHD            500 non-null    int64
dtypes: float64(9), int64(7)
memory usage: 62.6 KB
```

```
In [6]: df1.columns
```

```
Out[6]: Index(['male', 'age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
              'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
              'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD'],
              dtype='object')
```

```
In [7]: x = df1[['age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
                'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
                'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD']]
y = df1['male']
```

```
In [8]: from sklearn.model_selection import train_test_split
```

```
In [9]: x_train,x_test,y_train,y_test = train_test_split(x,y,train_size=0.70)
```

```
In [10]: from sklearn.ensemble import RandomForestClassifier
```

```
In [11]: rfc = RandomForestClassifier()
rfc.fit(x_train,y_train)
```

```
Out[11]: RandomForestClassifier()
```

```
In [12]: parameters = {  
        'max_depth':[11,12,13,14,15],  
        'min_samples_leaf':[15,20,25,30,35],  
        'n_estimators':[10,20,30,40,50]  
        }
```

```
In [13]: from sklearn.model_selection import GridSearchCV
```

```
In [14]: grid_search = GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring='ac  
grid_search.fit(x_train,y_train)
```

```
Out[14]: GridSearchCV(cv=2, estimator=RandomForestClassifier(),  
                      param_grid={'max_depth': [11, 12, 13, 14, 15],  
                                'min_samples_leaf': [15, 20, 25, 30, 35],  
                                'n_estimators': [10, 20, 30, 40, 50]},  
                      scoring='accuracy')
```

```
In [15]: grid_search.best_score_
```

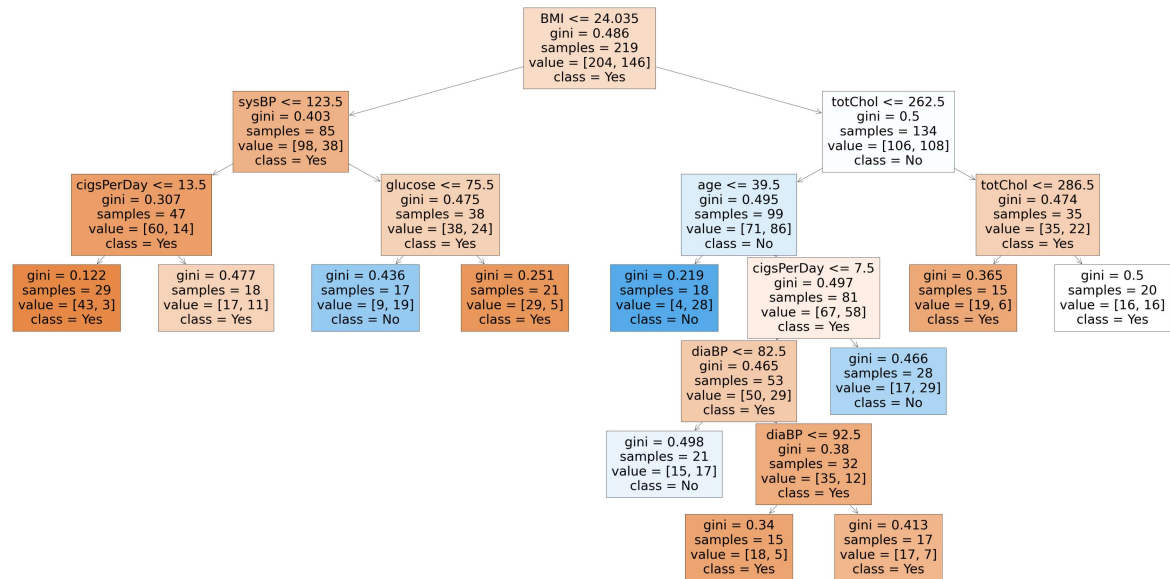
```
Out[15]: 0.6857142857142857
```

```
In [16]: from sklearn.tree import plot_tree
```

```
In [17]: rfc_best= grid_search.best_estimator_
```

```
In [19]: plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=["Yes","No"],
```

```
Out[19]: [Text(2232.0, 2019.0857142857144, 'BMI <= 24.035\ngini = 0.486\nsamples = 219\nvalue = [204, 146]\nclass = Yes'),
Text(1116.0, 1708.457142857143, 'sysBP <= 123.5\ngini = 0.403\nsamples = 85\nvalue = [98, 38]\nclass = Yes'),
Text(558.0, 1397.8285714285716, 'cigsPerDay <= 13.5\ngini = 0.307\nsamples = 47\nvalue = [60, 14]\nclass = Yes'),
Text(279.0, 1087.2, 'gini = 0.122\nsamples = 29\nvalue = [43, 3]\nclass = Yes'),
Text(837.0, 1087.2, 'gini = 0.477\nsamples = 18\nvalue = [17, 11]\nclass = Yes'),
Text(1674.0, 1397.8285714285716, 'glucose <= 75.5\ngini = 0.475\nsamples = 38\nvalue = [38, 24]\nclass = Yes'),
Text(1395.0, 1087.2, 'gini = 0.436\nsamples = 17\nvalue = [9, 19]\nclass = No'),
Text(1953.0, 1087.2, 'gini = 0.251\nsamples = 21\nvalue = [29, 5]\nclass = Yes'),
Text(3348.0, 1708.457142857143, 'totChol <= 262.5\ngini = 0.5\nsamples = 134\nvalue = [106, 108]\nclass = No'),
Text(2790.0, 1397.8285714285716, 'age <= 39.5\ngini = 0.495\nsamples = 99\nvalue = [71, 86]\nclass = No'),
Text(2511.0, 1087.2, 'gini = 0.219\nsamples = 18\nvalue = [4, 28]\nclass = No'),
Text(3069.0, 1087.2, 'cigsPerDay <= 7.5\ngini = 0.497\nsamples = 81\nvalue = [67, 58]\nclass = Yes'),
Text(2790.0, 776.5714285714287, 'diaBP <= 82.5\ngini = 0.465\nsamples = 53\nvalue = [50, 29]\nclass = Yes'),
Text(2511.0, 465.9428571428573, 'gini = 0.498\nsamples = 21\nvalue = [15, 17]\nclass = No'),
Text(3069.0, 465.9428571428573, 'diaBP <= 92.5\ngini = 0.38\nsamples = 32\nvalue = [35, 12]\nclass = Yes'),
Text(2790.0, 155.3142857142857, 'gini = 0.34\nsamples = 15\nvalue = [18, 5]\nclass = Yes'),
Text(3348.0, 155.3142857142857, 'gini = 0.413\nsamples = 17\nvalue = [17, 7]\nclass = Yes'),
Text(3348.0, 776.5714285714287, 'gini = 0.466\nsamples = 28\nvalue = [17, 29]\nclass = No'),
Text(3906.0, 1397.8285714285716, 'totChol <= 286.5\ngini = 0.474\nsamples = 35\nvalue = [35, 22]\nclass = Yes'),
Text(3627.0, 1087.2, 'gini = 0.365\nsamples = 15\nvalue = [19, 6]\nclass = Yes'),
Text(4185.0, 1087.2, 'gini = 0.5\nsamples = 20\nvalue = [16, 16]\nclass = Yes')]
```



In [ ]: