Radom Forest

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
from sklearn.linear_model import LogisticRegression
```

In [21]:

```
df=pd.read_csv(r"C:\Users\user\Downloads\C2_train.csv")[0:100]
df
```

Out[21]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
...	...	...	...	...	...	...	...	...	...	
95	96	0	3	Shorney, Mr. Charles Joseph	male	NaN	0	0	374910	8.05
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754	34.65
97	98	1	1	Greenfield, Mr. William Bertram	male	23.0	0	1	PC 17759	63.35
98	99	1	2	Doling, Mrs. John T (Ada Julia Bone)	female	34.0	0	1	231919	23.00
99	100	0	2	Kantor, Mr. Sinai	male	34.0	1	0	244367	26.00

100 rows × 12 columns

In [22]:

```
df.columns
```

Out[22]:

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [23]:

```
df.fillna(value=1)
```

Out[23]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
...	...	...	...	...	...	...	...	...	...	
95	96	0	3	Shorney, Mr. Charles Joseph	male	1.0	0	0	374910	8.05
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754	34.65
97	98	1	1	Greenfield, Mr. William Bertram	male	23.0	0	1	PC 17759	63.35
98	99	1	2	Doling, Mrs. John T (Ada Julia Bone)	female	34.0	0	1	231919	23.00
99	100	0	2	Kantor, Mr. Sinai	male	34.0	1	0	244367	26.00

100 rows × 12 columns

In [24]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  100 non-null    int64  
 1   Survived     100 non-null    int64  
 2   Pclass       100 non-null    int64  
 3   Name         100 non-null    object 
 4   Sex          100 non-null    object 
 5   Age          78 non-null     float64
 6   SibSp        100 non-null    int64  
 7   Parch        100 non-null    int64  
 8   Ticket       100 non-null    object 
 9   Fare         100 non-null    float64
 10  Cabin        20 non-null     object 
 11  Embarked     99 non-null     object 
dtypes: float64(2), int64(5), object(5)
memory usage: 9.5+ KB
```

In [25]:

```
d=df[['PassengerId', 'Pclass', 'SibSp', 'Parch','Sex']][0:50]
```

In [26]:

```
df['Sex'].value_counts()
```

Out[26]:

```
male      61
female    39
Name: Sex, dtype: int64
```

In [27]:

```
x=d.drop('Sex',axis=1)
y=d['Sex']
TenYearCHD1={"Sex":{'male':0,'female':1}}
d=d.replace('Sex')
print(d)
```

```
    PassengerId  Pclass  SibSp  Parch     Sex
0             1       3      1      0    male
1             2       1      1      0  female
2             3       3      0      0  female
3             4       1      1      0  female
4             5       3      0      0    male
5             6       3      0      0    male
6             7       1      0      0    male
7             8       3      3      1    male
8             9       3      0      2  female
9            10       2      1      0  female
10           11       3      1      1  female
11           12       1      0      0  female
12           13       3      0      0    male
13           14       3      1      5    male
14           15       3      0      0  female
15           16       2      0      0  female
16           17       3      4      1    male
17           18       2      0      0    male
18           19       3      1      0  female
19           20       3      0      0  female
20           21       2      0      0    male
21           22       2      0      0    male
22           23       3      0      0  female
23           24       1      0      0    male
24           25       3      3      1  female
25           26       3      1      5  female
26           27       3      0      0    male
27           28       1      3      2    male
28           29       3      0      0  female
29           30       3      0      0    male
30           31       1      0      0    male
31           32       1      1      0  female
32           33       3      0      0  female
33           34       2      0      0    male
34           35       1      1      0    male
35           36       1      1      0    male
36           37       3      0      0    male
37           38       3      0      0    male
38           39       3      2      0  female
39           40       3      1      0  female
40           41       3      1      0  female
41           42       2      1      0  female
42           43       3      0      0    male
43           44       2      1      2  female
44           45       3      0      0  female
45           46       3      0      0    male
46           47       3      1      0    male
47           48       3      0      0  female
48           49       3      2      0    male
49           50       3      1      0  female
```

In [28]:

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

In [29]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [30]:

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[30]:

```
RandomForestClassifier()
```

In [31]:

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

In [32]:

```
from sklearn.model_selection import GridSearchCV
```

In [33]:

```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
```

In [34]:

```
grid_search.fit(x_train,y_train)
```

Out[34]:

```
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 [35]:

```
grid_search.best_score_
```

Out[35]:

```
0.542483660130719
```

In [36]:

```
rfc_best=grid_search.best_estimator_
```

In [37]:

```
from sklearn.tree import plot_tree
```

In [38]:

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

Out[38]:

```
[Text(2232.0, 1630.8000000000002, 'PassengerId <= 20.0\ngini = 0.496\nsamp
les = 22\nvalue = [16, 19]\nclass = Yes'),
 Text(1116.0, 543.5999999999999, 'gini = 0.5\nsamples = 11\nvalue = [9, 9]
\nclass = No'),
 Text(3348.0, 543.5999999999999, 'gini = 0.484\nsamples = 11\nvalue = [7,
10]\nclass = Yes')]
```