RANDOM FOREST - 5

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

In [2]:
```
df=pd.read_csv(r"C:\Users\BHOOMISH\Downloads\C5_health care diabetes.csv")
df
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [4]:
```
df=df.dropna()
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [6]:
```
df.describe()
```

Out[6]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outco
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000(
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000(
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000(
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000(
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000(
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000(

In [7]:
```
df.columns
```

Out[7]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [8]:
```
df['Outcome'].value_counts()
```

Out[8]:
```
0    500
1    268
Name: Outcome, dtype: int64
```

In [9]:
```
df1=df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']]
```

In [10]:
```
x=df1.drop('Outcome',axis=1)
y=df1['Outcome']
```

In [11]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [12]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[12]:
```
RandomForestClassifier
RandomForestClassifier()
```

In [13]:
```
parameters={'max_depth':[1,2,3,4,5],
            'min_samples_leaf':[5,10,15,20,25],
            'n_estimators':[10,20,30,40,50]}
```

In [14]:
```
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[14]:
```
GridSearchCV
estimator: RandomForestClassifier
RandomForestClassifier
```

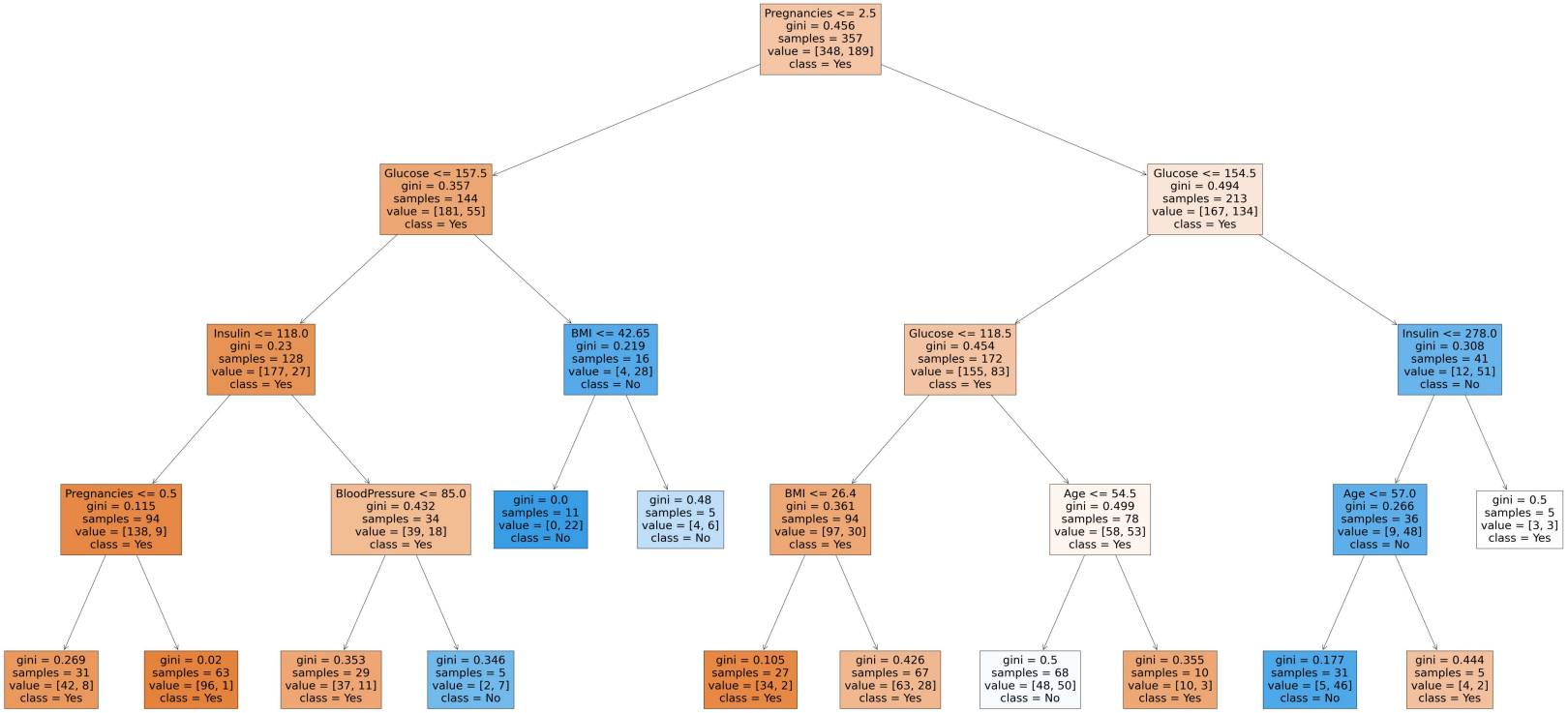
In [15]:
```
grid_search.best_score_
```

Out[15]:
```
0.7765355379237641
```

In [16]:
```
rfc_best=grid_search.best_estimator_
```

In [18]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```

Out[18]:
```
[Text(0.5217391304347826, 0.9, 'Pregnancies <= 2.5\ngini = 0.456\nsamples = 357\nvalue = [348, 189]\nclass =
Yes'),
 Text(0.2826086956521739, 0.7, 'Glucose <= 157.5\ngini = 0.357\nsamples = 144\nvalue = [181, 55]\nclass = Ye
s'),
 Text(0.17391304347826086, 0.5, 'Insulin <= 118.0\ngini = 0.23\nsamples = 128\nvalue = [177, 27]\nclass = Ye
s'),
 Text(0.08695652173913043, 0.3, 'Pregnancies <= 0.5\ngini = 0.115\nsamples = 94\nvalue = [138, 9]\nclass = Y
es'),
 Text(0.043478260869565216, 0.1, 'gini = 0.269\nsamples = 31\nvalue = [42, 8]\nclass = Yes'),
 Text(0.13043478260869565, 0.1, 'gini = 0.02\nsamples = 63\nvalue = [96, 1]\nclass = Yes'),
 Text(0.2608695652173913, 0.3, 'BloodPressure <= 85.0\ngini = 0.432\nsamples = 34\nvalue = [39, 18]\nclass =
Yes'),
 Text(0.21739130434782608, 0.1, 'gini = 0.353\nsamples = 29\nvalue = [37, 11]\nclass = Yes'),
 Text(0.30434782608695654, 0.1, 'gini = 0.346\nsamples = 5\nvalue = [2, 7]\nclass = No'),
 Text(0.391304347826087, 0.5, 'BMI <= 42.65\ngini = 0.219\nsamples = 16\nvalue = [4, 28]\nclass = No'),
 Text(0.34782608695652173, 0.3, 'gini = 0.0\nsamples = 11\nvalue = [0, 22]\nclass = No'),
 Text(0.43478260869565216, 0.3, 'gini = 0.48\nsamples = 5\nvalue = [4, 6]\nclass = No'),
 Text(0.7608695652173914, 0.7, 'Glucose <= 154.5\ngini = 0.494\nsamples = 213\nvalue = [167, 134]\nclass = Y
es'),
 Text(0.6086956521739131, 0.5, 'Glucose <= 118.5\ngini = 0.454\nsamples = 172\nvalue = [155, 83]\nclass = Ye
s'),
 Text(0.5217391304347826, 0.3, 'BMI <= 26.4\ngini = 0.361\nsamples = 94\nvalue = [97, 30]\nclass = Yes'),
 Text(0.4782608695652174, 0.1, 'gini = 0.105\nsamples = 27\nvalue = [34, 2]\nclass = Yes'),
 Text(0.5652173913043478, 0.1, 'gini = 0.426\nsamples = 67\nvalue = [63, 28]\nclass = Yes'),
 Text(0.6956521739130435, 0.3, 'Age <= 54.5\ngini = 0.499\nsamples = 78\nvalue = [58, 53]\nclass = Yes'),
 Text(0.6521739130434783, 0.1, 'gini = 0.5\nsamples = 68\nvalue = [48, 50]\nclass = No'),
 Text(0.7391304347826086, 0.1, 'gini = 0.355\nsamples = 10\nvalue = [10, 3]\nclass = Yes'),
 Text(0.9130434782608695, 0.5, 'Insulin <= 278.0\ngini = 0.308\nsamples = 41\nvalue = [12, 51]\nclass = N
o'),
 Text(0.8695652173913043, 0.3, 'Age <= 57.0\ngini = 0.266\nsamples = 36\nvalue = [9, 48]\nclass = No'),
 Text(0.8260869565217391, 0.1, 'gini = 0.177\nsamples = 31\nvalue = [5, 46]\nclass = No'),
 Text(0.9130434782608695, 0.1, 'gini = 0.444\nsamples = 5\nvalue = [4, 2]\nclass = Yes'),
 Text(0.9565217391304348, 0.3, 'gini = 0.5\nsamples = 5\nvalue = [3, 3]\nclass = Yes')]
```


Pregnancies <= 2.5
gini = 0.456
samples = 357
value = [348, 189]
class = Yes
Glucose <= 157.5
gini = 0.357
samples = 144
value = [181, 55]
class = Yes
Glucose <= 154.5
gini = 0.494
samples = 213
value = [167, 134]
class = Yes
Insulin <= 118.0
gini = 0.23
samples = 128
value = [177, 27]
class = Yes
BMI <= 42.65
gini = 0.219
samples = 16
value = [4, 28]
class = No
Glucose <= 118.5
gini = 0.454
samples = 172
value = [155, 83]
class = Yes
Insulin <= 278.0
gini = 0.308
samples = 41
value = [12, 51]
class = No
Pregnancies <= 0.5
gini = 0.115
samples = 94
value = [138, 9]
class = Yes
BloodPressure <= 85.0
gini = 0.432
samples = 34
value = [39, 18]
class = Yes
gini = 0.0
samples = 11
value = [0, 22]
class = No
gini = 0.48
samples = 5
value = [4, 6]
class = No
BMI <= 26.4
gini = 0.361
samples = 94
value = [97, 30]
class = Yes
Age <= 54.5
gini = 0.499
samples = 78
value = [58, 53]
class = Yes
Age <= 57.0
gini = 0.266
samples = 36
value = [9, 48]
class = No
gini = 0.5
samples = 5
value = [3, 3]
class = Yes
gini = 0.269
samples = 31
value = [42, 8]
class = Yes
gini = 0.02
samples = 63
value = [96, 1]
class = Yes
gini = 0.353
samples = 29
value = [37, 11]
class = Yes
gini = 0.346
samples = 5
value = [2, 7]
class = No
gini = 0.105
samples = 27
value = [34, 2]
class = Yes
gini = 0.426
samples = 67
value = [63, 28]
class = Yes
gini = 0.5
samples = 68
value = [48, 50]
class = No
gini = 0.355
samples = 10
value = [10, 3]
class = Yes
gini = 0.177
samples = 31
value = [5, 46]
class = No
gini = 0.444
samples = 5
value = [4, 2]
class = Yes

In [ ]: