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

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

Out[2]:

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

768 rows × 9 columns

In [3]:
```
df.columns
```

Out[3]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [4]:
```
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 [5]:
```
df['Outcome'].value_counts()
```

Out[5]:
```
0    500
1    268
Name: Outcome, dtype: int64
```

In [6]:
```
df['Outcome'].value_counts()
```

Out[6]:
```
0    500
1    268
Name: Outcome, dtype: int64
```

In [7]:
```
x=df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin','Age']]
y=df['Outcome']
```

In [8]:
```
g1={"Outcome":{'1':2,'0':3}}
df=df.replace(g1)
print(df)
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0              6      148             72             35        0  33.6
1              1       85             66             29        0  26.6
2              8      183             64              0        0  23.3
3              1       89             66             23       94  28.1
4              0      137             40             35      168  43.1
..           ...      ...            ...            ...      ...   ...
763           10      101             76             48      180  32.9
764            2      122             70             27        0  36.8
765            5      121             72             23      112  26.2
766            1      126             60              0        0  30.1
767            1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]
```

In [9]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.70)
```

In [10]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[10]:
```
RandomForestClassifier()
```

In [11]:
```
parameters= {
    "max_depth":[1,2,3,4,5],
    "min_samples_leaf":[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

In [12]:
```
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[12]:
```
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')
```

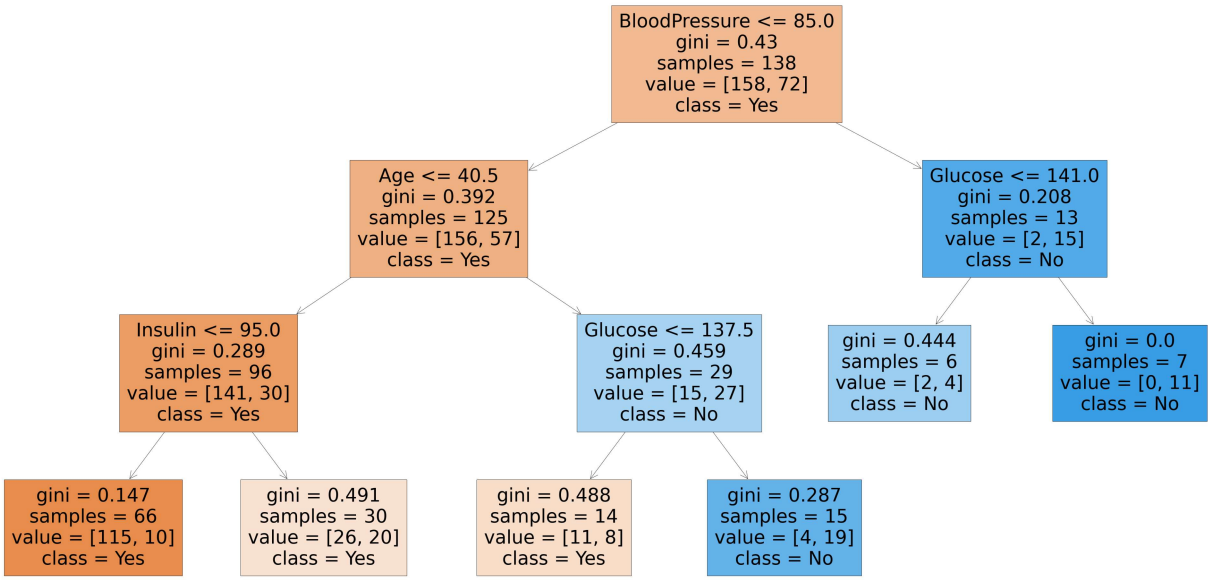
In [13]:
```
grid_search.best_score_
```

Out[13]:
```
0.7652173913043478
```

In [14]:
```
rfc_best=grid_search.best_estimator_
```

In [15]:
```
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'],fill
```

Out[15]:
```
[Text(2637.818181818182, 1902.6000000000001, 'BloodPressure <= 85.0\ngini = 0.43\nsample
s = 138\nvalue = [158, 72]\nclass = Yes'),
 Text(1623.2727272727273, 1359.0, 'Age <= 40.5\ngini = 0.392\nsamples = 125\nvalue = [15
6, 57]\nclass = Yes'),
 Text(811.6363636363636, 815.4000000000001, 'Insulin <= 95.0\ngini = 0.289\nsamples = 96
\nvalue = [141, 30]\nclass = Yes'),
 Text(405.8181818181818, 271.79999999999995, 'gini = 0.147\nsamples = 66\nvalue = [115,
10]\nclass = Yes'),
 Text(1217.4545454545455, 271.79999999999995, 'gini = 0.491\nsamples = 30\nvalue = [26,
20]\nclass = Yes'),
 Text(2434.909090909091, 815.4000000000001, 'Glucose <= 137.5\ngini = 0.459\nsamples = 2
9\nvalue = [15, 27]\nclass = No'),
 Text(2029.090909090909, 271.79999999999995, 'gini = 0.488\nsamples = 14\nvalue = [11,
8]\nclass = Yes'),
 Text(2840.7272727272725, 271.79999999999995, 'gini = 0.287\nsamples = 15\nvalue = [4, 1
9]\nclass = No'),
 Text(3652.3636363636365, 1359.0, 'Glucose <= 141.0\ngini = 0.208\nsamples = 13\nvalue =
[2, 15]\nclass = No'),
 Text(3246.5454545454545, 815.4000000000001, 'gini = 0.444\nsamples = 6\nvalue = [2, 4]
\nclass = No'),
 Text(4058.181818181818, 815.4000000000001, 'gini = 0.0\nsamples = 7\nvalue = [0, 11]\nc
lass = No')]
```


BloodPressure <= 85.0
gini = 0.43
samples = 138
value = [158, 72]
class = Yes
Age <= 40.5
gini = 0.392
samples = 125
value = [156, 57]
class = Yes
Glucose <= 141.0
gini = 0.208
samples = 13
value = [2, 15]
class = No
Insulin <= 95.0
gini = 0.289
samples = 96
value = [141, 30]
class = Yes
Glucose <= 137.5
gini = 0.459
samples = 29
value = [15, 27]
class = No
gini = 0.444
samples = 6
value = [2, 4]
class = No
gini = 0.0
samples = 7
value = [0, 11]
class = No
gini = 0.147
samples = 66
value = [115, 10]
class = Yes
gini = 0.491
samples = 30
value = [26, 20]
class = Yes
gini = 0.488
samples = 14
value = [11, 8]
class = Yes
gini = 0.287
samples = 15
value = [4, 19]
class = No

In [ ]: