In [1]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

In [2]:
```
from sklearn.ensemble import RandomForestClassifier
import matplotlib.pyplot as plt
from sklearn.model_selection import GridSearchCV
```

In [3]:
```
df=pd.read_csv("C3_bot_detection_data.csv")
df
```

Out[3]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location
0	132131	flong	Station activity person against natural majori...	85	1	2353	False	1	Adkinston
1	289683	hinesstephanie	Authority research natural life material staff...	55	5	9617	True	0	Sanderston
2	779715	roberttran	Manage whose quickly especially foot none to g...	6	2	4363	True	0	Harrisonfurt
3	696168	pmason	Just cover eight opportunity strong policy which.	54	5	2242	True	1	Martinezberg
4	704441	noah87	Animal sign six data good or.	26	3	8438	False	1	Camachoville
...	...	...	...	...	...	...	...	...	...
49995	491196	uberg	Want but put card direction know miss former h...	64	0	9911	True	1	Lake Kimberlyburgh
49996	739297	jessicamunoz	Provide whole maybe agree church respond most ...	18	5	9900	False	1	Greenbury
49997	674475	lynncunningham	Bring different everyone international capital...	43	3	6313	True	1	Deborahfort
49998	167081	richardthompson	Than about single generation itself seek sell ...	45	1	6343	False	0	Stephenside
49999	311204	daniel29	Here morning class various room human true bec...	91	4	4006	False	0	Novakberg

50000 rows × 11 columns

In [4]:
```
df1=df.iloc[:,3:8]
df1
```

0	85	1	2353	False	1
1	55	5	9617	True	0
2	6	2	4363	True	0
3	54	5	2242	True	1
4	26	3	8438	False	1
...	...	...	...	...	...
49995	64	0	9911	True	1
49996	18	5	9900	False	1
49997	43	3	6313	True	1
49998	45	1	6343	False	0
49999	91	4	4006	False	0

50000 rows × 5 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Retweet Count   50000 non-null  int64
 1   Mention Count   50000 non-null  int64
 2   Follower Count  50000 non-null  int64
 3   Verified        50000 non-null  bool
 4   Bot Label       50000 non-null  int64
dtypes: bool(1), int64(4)
memory usage: 1.6 MB
```

In [6]:
```
y=df1["Verified"]
x=df1.drop(["Verified"],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [7]:
```
# lr=LogisticRegression()
# lr.fit(x_train,y_train)
```

In [8]:
```
# lr.predict(x_test)
```

In [9]:
```
# lr.score(x_test,y_test)
```

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

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

In [ ]:
```

```

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

In [12]:
```
grid_search = GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accurac
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]:
```
0.5039714285714285
```

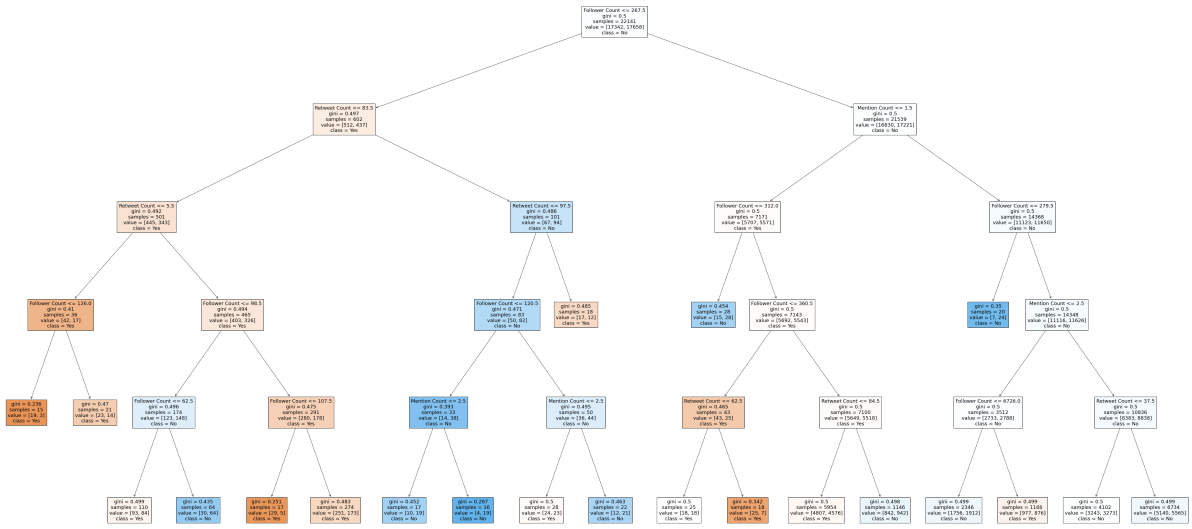
In [14]:
```
rfc_best=grid_search.best_estimator_
```

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'],
```

Out[15]:
```
[Text(2311.714285714286, 1993.2, 'Follower Count <= 267.5\ngini = 0.5\nsamples = 2
2141\nvalue = [17342, 17658]\nclass = No'),
 Text(1307.3142857142857, 1630.8000000000002, 'Retweet Count <= 83.5\ngini = 0.497
\nsamples = 602\nvalue = [512, 437]\nclass = Yes'),
 Text(573.9428571428572, 1268.4, 'Retweet Count <= 5.5\ngini = 0.492\nsamples = 50
1\nvalue = [445, 343]\nclass = Yes'),
 Text(255.0857142857143, 906.0, 'Follower Count <= 126.0\ngini = 0.41\nsamples = 3
6\nvalue = [42, 17]\nclass = Yes'),
 Text(127.54285714285714, 543.5999999999999, 'gini = 0.236\nsamples = 15\nvalue =
[19, 3]\nclass = Yes'),
 Text(382.62857142857143, 543.5999999999999, 'gini = 0.47\nsamples = 21\nvalue =
[23, 14]\nclass = Yes'),
 Text(892.8, 906.0, 'Follower Count <= 98.5\ngini = 0.494\nsamples = 465\nvalue =
[403, 326]\nclass = Yes'),
 Text(637.7142857142858, 543.5999999999999, 'Follower Count <= 62.5\ngini = 0.496
\nsamples = 174\nvalue = [123, 148]\nclass = No'),
 Text(510.1714285714286, 181.19999999999982, 'gini = 0.499\nsamples = 110\nvalue =
[93, 84]\nclass = Yes'),
 Text(765.2571428571429, 181.19999999999982, 'gini = 0.435\nsamples = 64\nvalue =
[30, 64]\nclass = No'),
 Text(1147.8857142857144, 543.5999999999999, 'Follower Count <= 107.5\ngini = 0.47
5\nsamples = 291\nvalue = [280, 178]\nclass = Yes'),
 Text(1020.3428571428572, 181.19999999999982, 'gini = 0.251\nsamples = 17\nvalue =
[29, 5]\nclass = Yes'),
 Text(1275.4285714285716, 181.19999999999982, 'gini = 0.483\nsamples = 274\nvalue
= [251, 173]\nclass = Yes'),
 Text(2040.6857142857143, 1268.4, 'Retweet Count <= 97.5\ngini = 0.486\nsamples =
101\nvalue = [67, 94]\nclass = No'),
 Text(1913.142857142857, 906.0, 'Follower Count <= 120.5\ngini = 0.471\nsamples =
83\nvalue = [50, 82]\nclass = No'),
 Text(1658.057142857143, 543.5999999999999, 'Mention Count <= 2.5\ngini = 0.393\ns
amples = 33\nvalue = [14, 38]\nclass = No'),
 Text(1530.5142857142857, 181.19999999999982, 'gini = 0.452\nsamples = 17\nvalue =
[10, 19]\nclass = No'),
 Text(1785.6, 181.19999999999982, 'gini = 0.287\nsamples = 16\nvalue = [4, 19]\ncl
ass = No'),
 Text(2168.2285714285713, 543.5999999999999, 'Mention Count <= 2.5\ngini = 0.495\n
samples = 50\nvalue = [36, 44]\nclass = No'),
 Text(2040.6857142857143, 181.19999999999982, 'gini = 0.5\nsamples = 28\nvalue =
[24, 23]\nclass = Yes'),
 Text(2295.7714285714287, 181.19999999999982, 'gini = 0.463\nsamples = 22\nvalue =
[12, 21]\nclass = No'),
 Text(2168.2285714285713, 906.0, 'gini = 0.485\nsamples = 18\nvalue = [17, 12]\ncl
ass = Yes'),
 Text(3316.114285714286, 1630.8000000000002, 'Mention Count <= 1.5\ngini = 0.5\nsa
mples = 21539\nvalue = [16830, 17221]\nclass = No'),
 Text(2805.942857142857, 1268.4, 'Follower Count <= 312.0\ngini = 0.5\nsamples = 7
171\nvalue = [5707, 5571]\nclass = Yes'),
 Text(2678.4, 906.0, 'gini = 0.454\nsamples = 28\nvalue = [15, 28]\nclass = No'),
 Text(2933.4857142857145, 906.0, 'Follower Count <= 360.5\ngini = 0.5\nsamples = 7
143\nvalue = [5692, 5543]\nclass = Yes'),
 Text(2678.4, 543.5999999999999, 'Retweet Count <= 62.5\ngini = 0.465\nsamples = 4
3\nvalue = [43, 25]\nclass = Yes'),
 Text(2550.857142857143, 181.19999999999982, 'gini = 0.5\nsamples = 25\nvalue = [1
8, 18]\nclass = Yes'),
 Text(2805.942857142857, 181.19999999999982, 'gini = 0.342\nsamples = 18\nvalue =
[25, 7]\nclass = Yes'),
 Text(3188.5714285714284, 543.5999999999999, 'Retweet Count <= 84.5\ngini = 0.5\ns
amples = 7100\nvalue = [5649, 5518]\nclass = Yes'),
 Text(3061.0285714285715, 181.19999999999982, 'gini = 0.5\nsamples = 5954\nvalue =
```

```
[4807, 4576]\nclass = Yes'),
 Text(3316.114285714286, 181.19999999999982, 'gini = 0.498\nsamples = 1146\nvalue
= [842, 942]\nclass = No'),
 Text(3826.285714285714, 1268.4, 'Follower Count <= 279.5\ngini = 0.5\nsamples = 1
4368\nvalue = [11123, 11650]\nclass = No'),
 Text(3698.7428571428572, 906.0, 'gini = 0.35\nsamples = 20\nvalue = [7, 24]\nclas
s = No'),
 Text(3953.8285714285716, 906.0, 'Mention Count <= 2.5\ngini = 0.5\nsamples = 1434
8\nvalue = [11116, 11626]\nclass = No'),
 Text(3698.7428571428572, 543.5999999999999, 'Follower Count <= 6726.0\ngini = 0.5
\nsamples = 3512\nvalue = [2733, 2788]\nclass = No'),
 Text(3571.2, 181.19999999999982, 'gini = 0.499\nsamples = 2346\nvalue = [1756, 19
12]\nclass = No'),
 Text(3826.285714285714, 181.19999999999982, 'gini = 0.499\nsamples = 1166\nvalue
= [977, 876]\nclass = Yes'),
 Text(4208.914285714286, 543.5999999999999, 'Retweet Count <= 37.5\ngini = 0.5\nsa
mples = 10836\nvalue = [8383, 8838]\nclass = No'),
 Text(4081.3714285714286, 181.19999999999982, 'gini = 0.5\nsamples = 4102\nvalue =
[3243, 3273]\nclass = No'),
 Text(4336.457142857143, 181.19999999999982, 'gini = 0.499\nsamples = 6734\nvalue
= [5140, 5565]\nclass = No')]
```



In [16]:
```
df2=pd.read_csv("C4_framingham.csv")
df2
```

Out[16]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabe
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	
...	...	...	...	...	...	...	...	...	
4233	1	50	1.0	1	1.0	0.0	0	1	
4234	1	51	3.0	1	43.0	0.0	0	0	
4235	0	48	2.0	1	20.0	NaN	0	0	
4236	0	44	1.0	1	15.0	0.0	0	0	
4237	0	52	2.0	0	0.0	0.0	0	0	

4238 rows × 16 columns

In [17]:
```
df2=df2.dropna()
```

In [18]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3656 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             3656 non-null   int64  
 1   age              3656 non-null   int64  
 2   education        3656 non-null   float64
 3   currentSmoker    3656 non-null   int64  
 4   cigsPerDay       3656 non-null   float64
 5   BPMeds           3656 non-null   float64
 6   prevalentStroke  3656 non-null   int64  
 7   prevalentHyp     3656 non-null   int64  
 8   diabetes         3656 non-null   int64  
 9   totChol          3656 non-null   float64
 10  sysBP            3656 non-null   float64
 11  diaBP            3656 non-null   float64
 12  BMI              3656 non-null   float64
 13  heartRate        3656 non-null   float64
 14  glucose          3656 non-null   float64
 15  TenYearCHD       3656 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 485.6 KB
```

In [19]:
```
y=df2["diabetes"]
x=df2.drop(["diabetes"],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [20]:
```
# lr=LogisticRegression()
# lr.fit(x_train,y_train)
```

In [21]:
```
# val=[[1,34,5,1,4,1,0,1,123,108,89,29,84,70,1]]
# lr.predict(val)
```

In [22]:
```
# lr.score(x_test,y_test)
```

In [23]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[23]:
```
RandomForestClassifier()
```

In [24]:
```
parameter={'max_depth':[1,2,3,4,5],
          "min_samples_leaf":[5,10,15,20,25],
          "n_estimators":[10,20,30,40,50]}
```

In [25]:
```
from sklearn.model_selection import GridSearchCV

grid_search = GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accurac
grid_search.fit(x_train,y_train)
```

Out[25]:
```
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 [26]:
```
grid_search.best_score_
```

Out[26]:
```
0.980851739640344
```

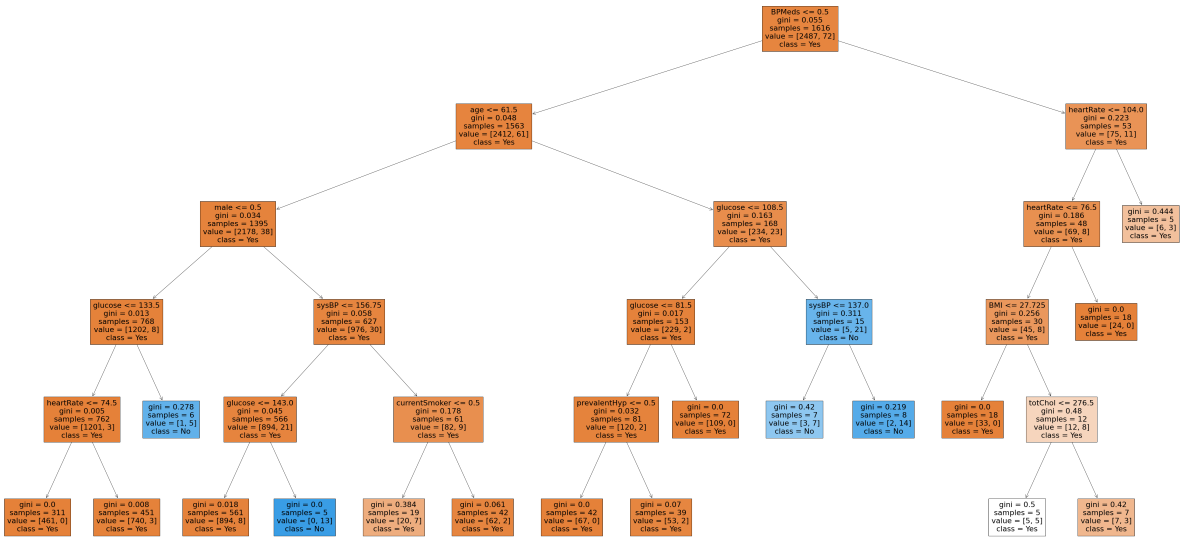
In [27]:
```
rfc_best=grid_search.best_estimator_
```

In [28]:
```
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'],
```

Out[28]:
```
[Text(2996.666666666667, 1993.2, 'BPMeds <= 0.5\ngini = 0.055\nsamples = 1616\nvalue = [2487, 72]\nclass = Yes'),
 Text(1860.0, 1630.8000000000002, 'age <= 61.5\ngini = 0.048\nsamples = 1563\nvalue = [2412, 61]\nclass = Yes'),
 Text(909.3333333333334, 1268.4, 'male <= 0.5\ngini = 0.034\nsamples = 1395\nvalue = [2178, 38]\nclass = Yes'),
 Text(496.0, 906.0, 'glucose <= 133.5\ngini = 0.013\nsamples = 768\nvalue = [1202, 8]\nclass = Yes'),
 Text(330.6666666666667, 543.5999999999999, 'heartRate <= 74.5\ngini = 0.005\nsamples = 762\nvalue = [1201, 3]\nclass = Yes'),
 Text(165.33333333333334, 181.19999999999982, 'gini = 0.0\nsamples = 311\nvalue = [461, 0]\nclass = Yes'),
 Text(496.0, 181.19999999999982, 'gini = 0.008\nsamples = 451\nvalue = [740, 3]\nclass = Yes'),
 Text(661.3333333333334, 543.5999999999999, 'gini = 0.278\nsamples = 6\nvalue = [1, 5]\nclass = No'),
 Text(1322.6666666666667, 906.0, 'sysBP <= 156.75\ngini = 0.058\nsamples = 627\nvalue = [976, 30]\nclass = Yes'),
 Text(992.0, 543.5999999999999, 'glucose <= 143.0\ngini = 0.045\nsamples = 566\nvalue = [894, 21]\nclass = Yes'),
 Text(826.6666666666667, 181.19999999999982, 'gini = 0.018\nsamples = 561\nvalue = [894, 8]\nclass = Yes'),
 Text(1157.3333333333335, 181.19999999999982, 'gini = 0.0\nsamples = 5\nvalue = [0, 13]\nclass = No'),
 Text(1653.3333333333335, 543.5999999999999, 'currentSmoker <= 0.5\ngini = 0.178\nsamples = 61\nvalue = [82, 9]\nclass = Yes'),
 Text(1488.0, 181.19999999999982, 'gini = 0.384\nsamples = 19\nvalue = [20, 7]\nclass = Yes'),
 Text(1818.6666666666667, 181.19999999999982, 'gini = 0.061\nsamples = 42\nvalue = [62, 2]\nclass = Yes'),
 Text(2810.666666666667, 1268.4, 'glucose <= 108.5\ngini = 0.163\nsamples = 168\nvalue = [234, 23]\nclass = Yes'),
 Text(2480.0, 906.0, 'glucose <= 81.5\ngini = 0.017\nsamples = 153\nvalue = [229, 2]\nclass = Yes'),
 Text(2314.666666666667, 543.5999999999999, 'prevalentHyp <= 0.5\ngini = 0.032\nsamples = 81\nvalue = [120, 2]\nclass = Yes'),
 Text(2149.3333333333335, 181.19999999999982, 'gini = 0.0\nsamples = 42\nvalue = [67, 0]\nclass = Yes'),
 Text(2480.0, 181.19999999999982, 'gini = 0.07\nsamples = 39\nvalue = [53, 2]\nclass = Yes'),
 Text(2645.3333333333335, 543.5999999999999, 'gini = 0.0\nsamples = 72\nvalue = [109, 0]\nclass = Yes'),
 Text(3141.3333333333335, 906.0, 'sysBP <= 137.0\ngini = 0.311\nsamples = 15\nvalue = [5, 21]\nclass = No'),
 Text(2976.0, 543.5999999999999, 'gini = 0.42\nsamples = 7\nvalue = [3, 7]\nclass = No'),
 Text(3306.666666666667, 543.5999999999999, 'gini = 0.219\nsamples = 8\nvalue = [2, 14]\nclass = No'),
 Text(4133.333333333334, 1630.8000000000002, 'heartRate <= 104.0\ngini = 0.223\nsamples = 53\nvalue = [75, 11]\nclass = Yes'),
 Text(3968.0, 1268.4, 'heartRate <= 76.5\ngini = 0.186\nsamples = 48\nvalue = [69, 8]\nclass = Yes'),
 Text(3802.666666666667, 906.0, 'BMI <= 27.725\ngini = 0.256\nsamples = 30\nvalue = [45, 8]\nclass = Yes'),
 Text(3637.3333333333335, 543.5999999999999, 'gini = 0.0\nsamples = 18\nvalue = [33, 0]\nclass = Yes'),
 Text(3968.0, 543.5999999999999, 'totChol <= 276.5\ngini = 0.48\nsamples = 12\nvalue = [12, 8]\nclass = Yes'),
 Text(3802.666666666667, 181.19999999999982, 'gini = 0.5\nsamples = 5\nvalue = [5, 5]\nclass = Yes'),
```

```
 Text(4133.333333333334, 181.19999999999982, 'gini = 0.42\nsamples = 7\nvalue =
[7, 3]\nclass = Yes'),
 Text(4133.333333333334, 906.0, 'gini = 0.0\nsamples = 18\nvalue = [24, 0]\nclass
= Yes'),
 Text(4298.666666666667, 1268.4, 'gini = 0.444\nsamples = 5\nvalue = [6, 3]\nclass
= Yes')]
```



In [29]:
```
df3=pd.read_csv("C5_health care diabetes.csv")
df3
```

Out[29]:

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

768 rows × 9 columns

In [41]:
```
df3.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 [42]:
```
y1=df3["Outcome"]
x1=df3.drop(["Outcome"],axis=1)
x1_train,x1_test,y1_train,y1_test=train_test_split(x1,y1,test_size=0.3)
```

In [32]:
```
# lr=LogisticRegression()
# lr.fit(x_train,y_train)
```

In [33]:
```
# lr.predict(x_test)
```

In [34]:
```
# val1=[[1,34,5,1,4,1,123,10]]
# lr.predict(val1)
```

In [43]:
```
rfc=RandomForestClassifier()
rfc.fit(x1_train,y1_train)
```

Out[43]:
```
RandomForestClassifier()
```

In [44]:
```
parameter={'max_depth':[1,2,3,4,5],
           "min_samples_leaf":[5,10,15,20,25],
           "n_estimators":[10,20,30,40,50]}
```

In [45]:
```
grid_search = GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accurac
grid_search.fit(x1_train,y1_train)
```

Out[45]:
```
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 [38]:
```
grid_search.best_score_
```

Out[38]:
```
0.7635243855074072
```

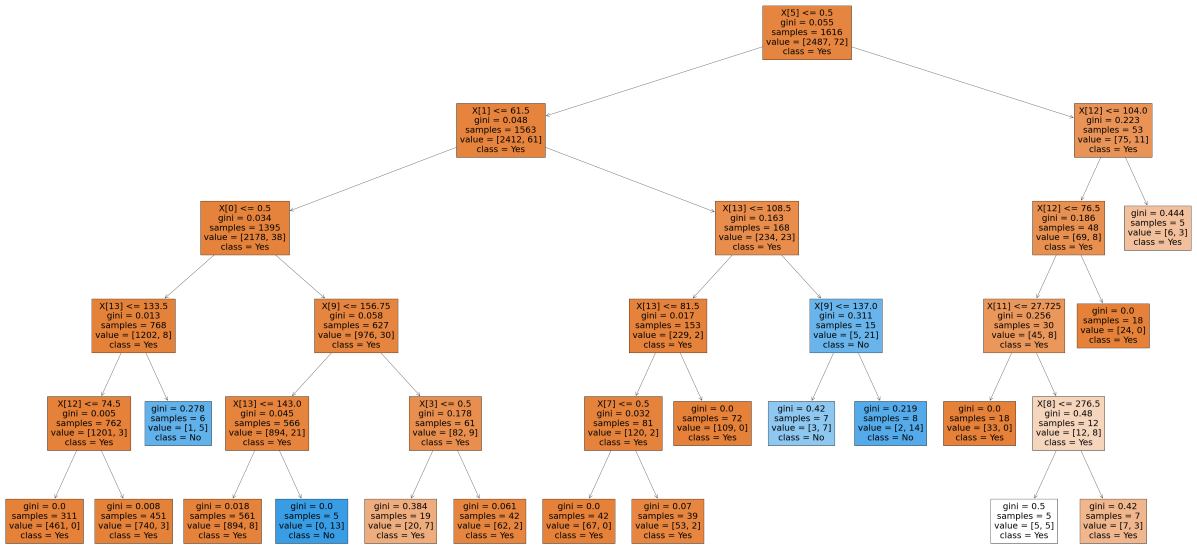
In [54]:
```
len(x1.columns)
```

Out[54]: 8

In [56]:
```
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],class_names=['Yes','No'],filled=True)
```

Out[56]:
```
[Text(2996.666666666667, 1993.2, 'X[5] <= 0.5\ngini = 0.055\nsamples = 1616\nvalue
= [2487, 72]\nclass = Yes'),
 Text(1860.0, 1630.8000000000002, 'X[1] <= 61.5\ngini = 0.048\nsamples = 1563\nval
ue = [2412, 61]\nclass = Yes'),
 Text(909.3333333333334, 1268.4, 'X[0] <= 0.5\ngini = 0.034\nsamples = 1395\nvalue
= [2178, 38]\nclass = Yes'),
 Text(496.0, 906.0, 'X[13] <= 133.5\ngini = 0.013\nsamples = 768\nvalue = [1202,
8]\nclass = Yes'),
 Text(330.6666666666667, 543.5999999999999, 'X[12] <= 74.5\ngini = 0.005\nsamples
= 762\nvalue = [1201, 3]\nclass = Yes'),
 Text(165.33333333333334, 181.19999999999982, 'gini = 0.0\nsamples = 311\nvalue =
[461, 0]\nclass = Yes'),
 Text(496.0, 181.19999999999982, 'gini = 0.008\nsamples = 451\nvalue = [740, 3]\nc
lass = Yes'),
 Text(661.3333333333334, 543.5999999999999, 'gini = 0.278\nsamples = 6\nvalue =
[1, 5]\nclass = No'),
 Text(1322.6666666666667, 906.0, 'X[9] <= 156.75\ngini = 0.058\nsamples = 627\nval
ue = [976, 30]\nclass = Yes'),
 Text(992.0, 543.5999999999999, 'X[13] <= 143.0\ngini = 0.045\nsamples = 566\nvalu
e = [894, 21]\nclass = Yes'),
 Text(826.6666666666667, 181.19999999999982, 'gini = 0.018\nsamples = 561\nvalue =
[894, 8]\nclass = Yes'),
 Text(1157.3333333333335, 181.19999999999982, 'gini = 0.0\nsamples = 5\nvalue =
[0, 13]\nclass = No'),
 Text(1653.3333333333335, 543.5999999999999, 'X[3] <= 0.5\ngini = 0.178\nsamples =
61\nvalue = [82, 9]\nclass = Yes'),
 Text(1488.0, 181.19999999999982, 'gini = 0.384\nsamples = 19\nvalue = [20, 7]\ncl
ass = Yes'),
 Text(1818.6666666666667, 181.19999999999982, 'gini = 0.061\nsamples = 42\nvalue =
[62, 2]\nclass = Yes'),
 Text(2810.666666666667, 1268.4, 'X[13] <= 108.5\ngini = 0.163\nsamples = 168\nval
ue = [234, 23]\nclass = Yes'),
 Text(2480.0, 906.0, 'X[13] <= 81.5\ngini = 0.017\nsamples = 153\nvalue = [229, 2]
\nclass = Yes'),
 Text(2314.666666666667, 543.5999999999999, 'X[7] <= 0.5\ngini = 0.032\nsamples =
81\nvalue = [120, 2]\nclass = Yes'),
 Text(2149.3333333333335, 181.19999999999982, 'gini = 0.0\nsamples = 42\nvalue =
[67, 0]\nclass = Yes'),
 Text(2480.0, 181.19999999999982, 'gini = 0.07\nsamples = 39\nvalue = [53, 2]\ncla
ss = Yes'),
 Text(2645.3333333333335, 543.5999999999999, 'gini = 0.0\nsamples = 72\nvalue = [1
09, 0]\nclass = Yes'),
 Text(3141.3333333333335, 906.0, 'X[9] <= 137.0\ngini = 0.311\nsamples = 15\nvalue
= [5, 21]\nclass = No'),
 Text(2976.0, 543.5999999999999, 'gini = 0.42\nsamples = 7\nvalue = [3, 7]\nclass
= No'),
 Text(3306.666666666667, 543.5999999999999, 'gini = 0.219\nsamples = 8\nvalue =
[2, 14]\nclass = No'),
 Text(4133.333333333334, 1630.8000000000002, 'X[12] <= 104.0\ngini = 0.223\nsample
s = 53\nvalue = [75, 11]\nclass = Yes'),
 Text(3968.0, 1268.4, 'X[12] <= 76.5\ngini = 0.186\nsamples = 48\nvalue = [69, 8]
\nclass = Yes'),
 Text(3802.666666666667, 906.0, 'X[11] <= 27.725\ngini = 0.256\nsamples = 30\nvalu
e = [45, 8]\nclass = Yes'),
 Text(3637.3333333333335, 543.5999999999999, 'gini = 0.0\nsamples = 18\nvalue = [3
3, 0]\nclass = Yes'),
 Text(3968.0, 543.5999999999999, 'X[8] <= 276.5\ngini = 0.48\nsamples = 12\nvalue
= [12, 8]\nclass = Yes'),
 Text(3802.666666666667, 181.19999999999982, 'gini = 0.5\nsamples = 5\nvalue = [5,
5]\nclass = Yes'),
```

```
 Text(4133.333333333334, 181.19999999999982, 'gini = 0.42\nsamples = 7\nvalue =
[7, 3]\nclass = Yes'),
 Text(4133.333333333334, 906.0, 'gini = 0.0\nsamples = 18\nvalue = [24, 0]\nclass
= Yes'),
 Text(4298.666666666667, 1268.4, 'gini = 0.444\nsamples = 5\nvalue = [6, 3]\nclass
= Yes')]
```



In [ ]: