PROBLEM STATEMENT :

TO PREDICT AND ANALYZE WHICH AGE HAS A HIGH CHANCE TO SMOKE...

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import preprocessing
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

In [2]:

```
df=pd.read_csv(r"C:\Users\sravya\Downloads\insurance.csv")
df
```

Out[2]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

DATA CLEANING AND PREPROCESSING

In [3]:

```
df.describe()
```

Out[3]:

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

In [5]:

```
df.columns
```

Out[5]:

```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtype='object')
```

In [6]:

```
df.tail()
```

Out[6]:

	age	sex	bmi	children	smoker	region	charges
1333	50	male	30.97	3	no	northwest	10600.5483
1334	18	female	31.92	0	no	northeast	2205.9808
1335	18	female	36.85	0	no	southeast	1629.8335
1336	21	female	25.80	0	no	southwest	2007.9450
1337	61	female	29.07	0	yes	northwest	29141.3603

In [7]:

```
df.shape
```

Out[7]:

```
(1338, 7)
```

TO FIND MISSING VALUES

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

IN THESE DATA SET I AM USING LOGISTIC REGRESSION BECAUSE ACCURACY VALUES IS VERY LESS...

In [9]:

```
convert={"sex":{"female":1,"male":0}}
df=df.replace(convert)
print(df)
```

```
      age  sex     bmi  children smoker     region      charges
0      19    1  27.900         0    yes  southwest  16884.92400
1      18    0  33.770         1     no  southeast   1725.55230
2      28    0  33.000         3     no  southeast   4449.46200
3      33    0  22.705         0     no  northwest  21984.47061
4      32    0  28.880         0     no  northwest   3866.85520
...   ...  ...     ...       ...    ...        ...          ...
1333   50    0  30.970         3     no  northwest  10600.54830
1334   18    1  31.920         0     no  northeast   2205.98080
1335   18    1  36.850         0     no  southeast   1629.83350
1336   21    1  25.800         0     no  southwest   2007.94500
1337   61    1  29.070         0    yes  northwest  29141.36030

[1338 rows x 7 columns]
```

DECISION TREE CLASSIFIER

In [10]:

```
convert={"region":{"southwest":1,"southeast":2,"northeast":3,"northwest":4}}
df=df.replace(convert)
print(df)
```

```
      age  sex     bmi  children smoker  region      charges
0      19    1  27.900         0    yes       1  16884.92400
1      18    0  33.770         1     no       2   1725.55230
2      28    0  33.000         3     no       2   4449.46200
3      33    0  22.705         0     no       4  21984.47061
4      32    0  28.880         0     no       4   3866.85520
...   ...  ...     ...       ...    ...     ...          ...
1333   50    0  30.970         3     no       4  10600.54830
1334   18    1  31.920         0     no       3   2205.98080
1335   18    1  36.850         0     no       2   1629.83350
1336   21    1  25.800         0     no       1   2007.94500
1337   61    1  29.070         0    yes       4  29141.36030

[1338 rows x 7 columns]
```

In [11]:

```
x=["age","sex","bmi","children","charges","region"]
y=["Yes","No"]
all_inputs=df[x]
all_classes=df["smoker"]
```

In [12]:

```
x_train,x_test,y_train,y_test=train_test_split(all_inputs,all_classes,test_size=0.25)
```

In [13]:

```
clt=DecisionTreeClassifier(random_state=0)
```

In [14]:

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

Out[14]:

```
▾        DecisionTreeClassifier

DecisionTreeClassifier(random_state=0)
```

In [15]:

```
score=clt.score(x_test,y_test)
print(score)
```

```
0.9552238805970149
```

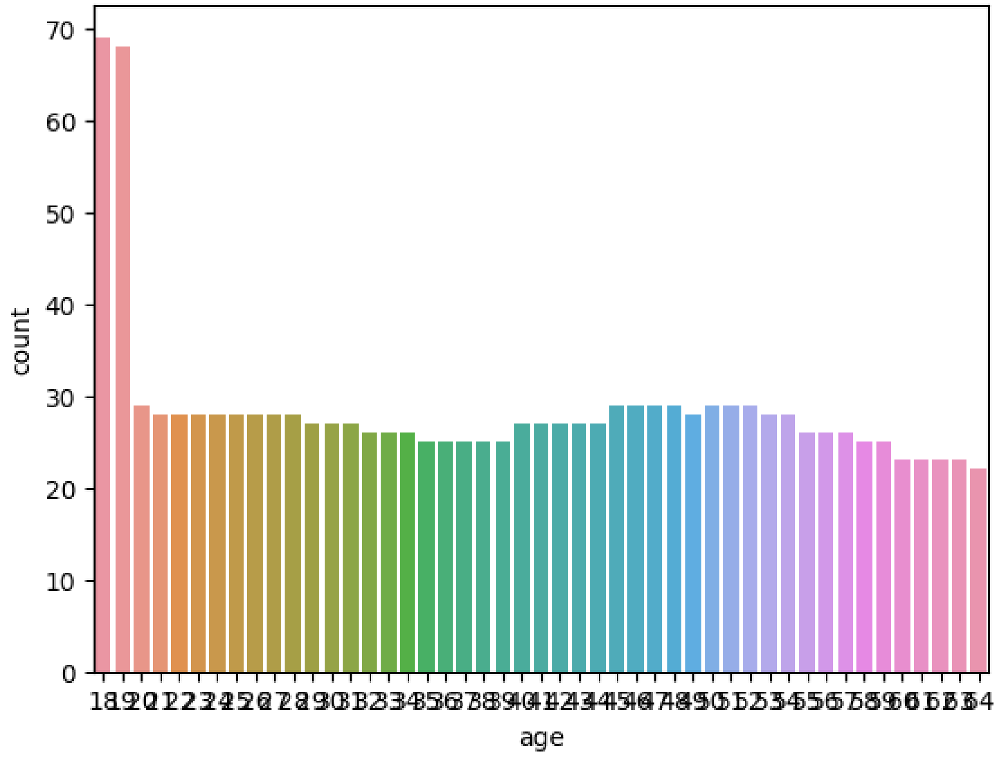
DATA VISUALIZATION

In [16]:

```
sns.countplot(x="age",data=df)
```

Out[16]:

```
<Axes: xlabel='age', ylabel='count'>
```

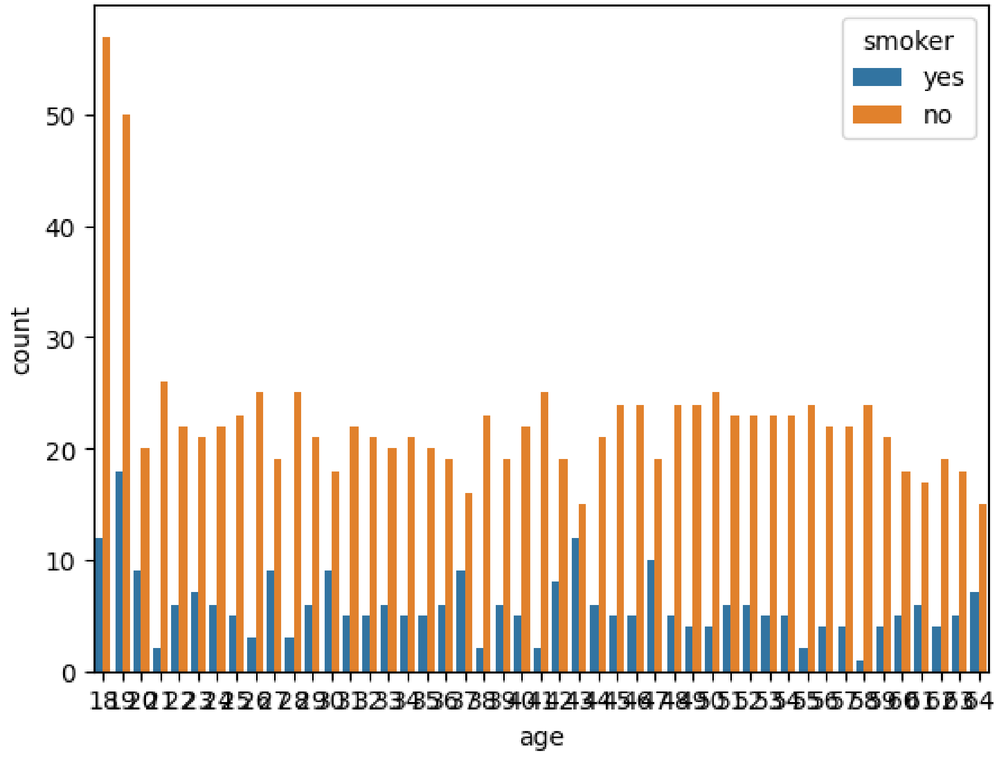


In [17]:

```
sns.countplot(x="age",hue="smoker",data=df)
```

Out[17]:

```
<Axes: xlabel='age', ylabel='count'>
```



RANDOM FOREST

In [18]:

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

Out[18]:

```
▾ RandomForestClassifier

RandomForestClassifier()
```

In [19]:

```
rf=RandomForestClassifier()
```

In [22]:

```
params={'max_depth':[2,3,5,10,20],
 'min_samples_leaf':[5,10,20,50,100,200],
  'n_estimators':[10,25,30,50,100,200]}
```

In [23]:

```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rf,param_grid=params,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[23]:

```
▸            GridSearchCV
▸ estimator: RandomForestClassifier
      ▸ RandomForestClassifier
```

In [24]:

```
grid_search.best_score_
```

Out[24]:

```
0.96410167712384
```

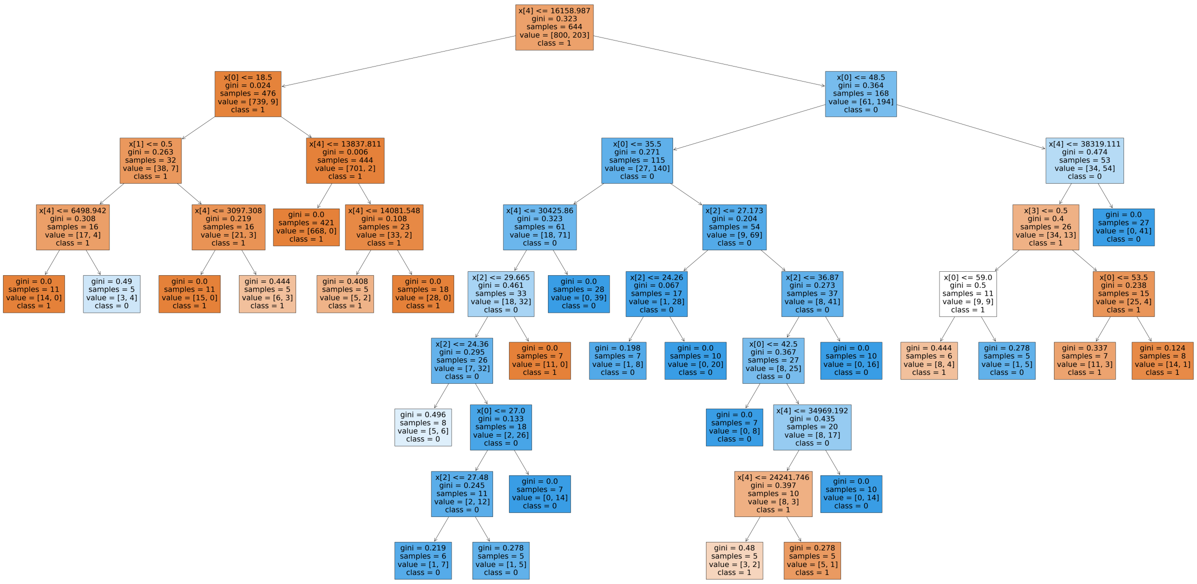
In [25]:

```
rf_best=grid_search.best_estimator_
print(rf_best)
```

```
RandomForestClassifier(max_depth=20, min_samples_leaf=5, n_estimators=30)
```

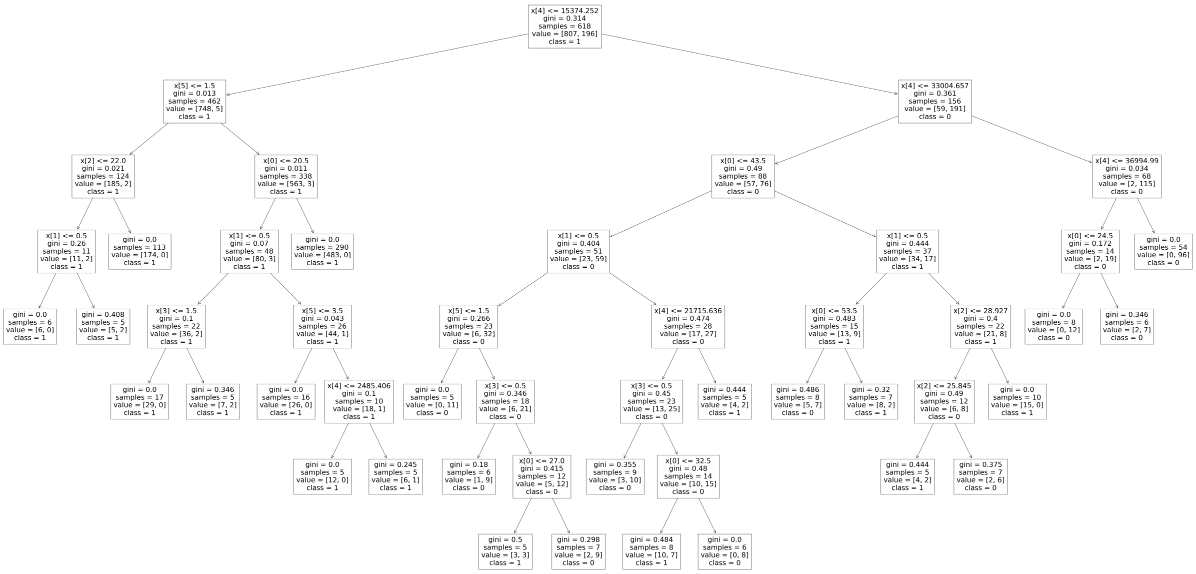
In [26]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[4],class_names=['1','0'],filled=True);
```



In [27]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[6],class_names=['1','0'],filled=False);
```



In [28]:

```
rf_best.feature_importances_
```

Out[28]:

```
array([0.044777  , 0.00845872, 0.07230586, 0.00993018, 0.85373523,
       0.01079302])
```

In [29]:

```
imp_df=pd.DataFrame({"varname":x_train.columns,"Imp":rf_best.feature_importances_})
imp_df.sort_values(by="Imp",ascending=False)
```

Out[29]:

	varname	Imp
4	charges	0.853735
2	bmi	0.072306
0	age	0.044777
5	region	0.010793
3	children	0.009930
1	sex	0.008459

# CONCLUSION

TO PREDICT AND ANALYZE THE DATA IN THE 20TH AGE HAS HIGH CHANCE TO SMOKE....