MINI PROJECT

PROBLEM STATEMENT:Ehich model is suitable for insurance dataset

In [1]:
```
#importing packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

Data Collection

Read the data

In [2]:
```
df=pd.read_csv(r"C:\Users\user\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.head()
```

Out[3]:

	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

In [4]:
```
df.tail()
```

Out[4]:

	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 [5]:
```
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 [7]:
```
df.shape
```

Out[7]: (1338, 7)

In [8]:
```
df.describe()
```

Out[8]:

	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 [9]:
```
df.columns
```

Out[9]:
```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtype='object')
```

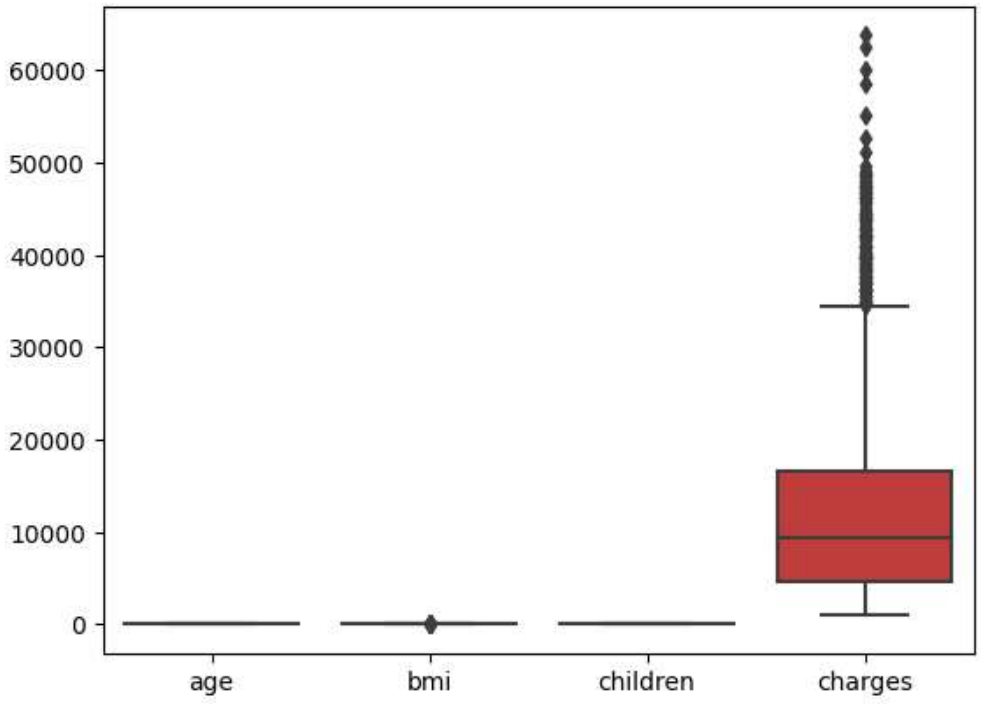
In [10]:
```
df.isnull().sum()
```

Out[10]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

Data Visualization

In [11]:
```
sns.boxplot(df)
```

Out[11]:
```
<Axes: >
```



In [12]:
```
smoker={"smoker":{"yes":0,"no":1}}
df=df.replace(smoker)
df
```

Out[12]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	0	southwest	16884.92400
1	18	male	33.770	1	1	southeast	1725.55230
2	28	male	33.000	3	1	southeast	4449.46200
3	33	male	22.705	0	1	northwest	21984.47061
4	32	male	28.880	0	1	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	1	northwest	10600.54830
1334	18	female	31.920	0	1	northeast	2205.98080
1335	18	female	36.850	0	1	southeast	1629.83350
1336	21	female	25.800	0	1	southwest	2007.94500
1337	61	female	29.070	0	0	northwest	29141.36030

1338 rows × 7 columns

In [13]:
```
sex={"sex":{"male":0,"female":1}}
df=df.replace(sex)
df
```

Out[13]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	0	southwest	16884.92400
1	18	0	33.770	1	1	southeast	1725.55230
2	28	0	33.000	3	1	southeast	4449.46200
3	33	0	22.705	0	1	northwest	21984.47061
4	32	0	28.880	0	1	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	0	30.970	3	1	northwest	10600.54830
1334	18	1	31.920	0	1	northeast	2205.98080
1335	18	1	36.850	0	1	southeast	1629.83350
1336	21	1	25.800	0	1	southwest	2007.94500
1337	61	1	29.070	0	0	northwest	29141.36030

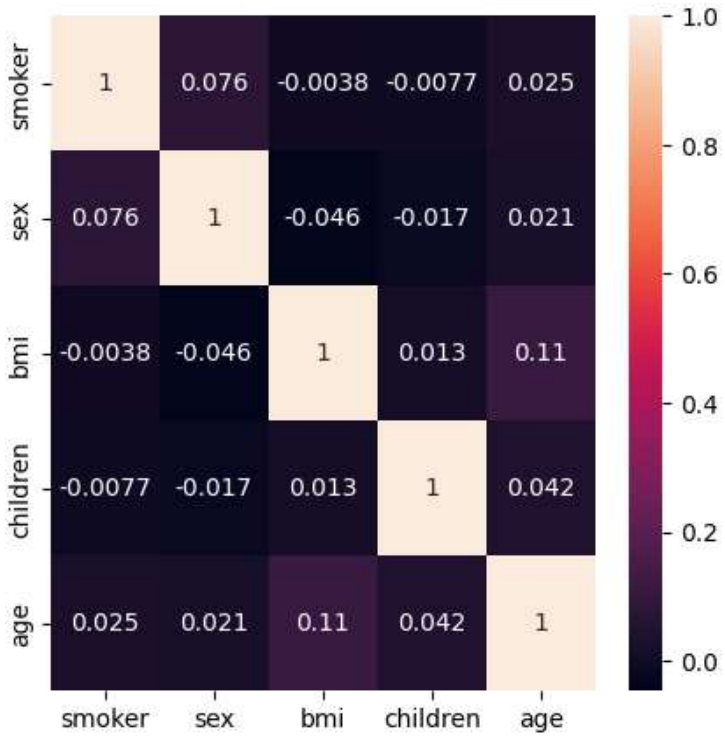
1338 rows × 7 columns

In [14]:
```
df['region'].value_counts()
```

Out[14]:
```
region
southeast    364
southwest    325
northwest    325
northeast    324
Name: count, dtype: int64
```

In [15]:
```
insured=df[['smoker','sex','bmi','children','age']]
plt.figure(figsize=(5,5))
sns.heatmap(insured.corr(),annot=True)
```

Out[15]: <Axes: >



Feature Scalling:To split the data into test and train data

In [16]:
```
x=df[['age','sex','bmi','children','smoker','region']]
y=df['charges']
```

In [17]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=101)
```

In [19]:
```
x=np.array(df['age']).reshape(-1,1)
y=np.array(df['sex']).reshape(-1,1)
```

In [21]:
```
print(regr.intercept_)
```

```
[0.49090098]
```

In [22]:
```
coeff_df=pd.DataFrame(regr.coef_)
coeff_df
```

Out[22]:

	0
0	0.000219

In [23]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
regr=LinearRegression()
regr.fit(x_train,y_train)
print(regr.score(x_test,y_test))
```

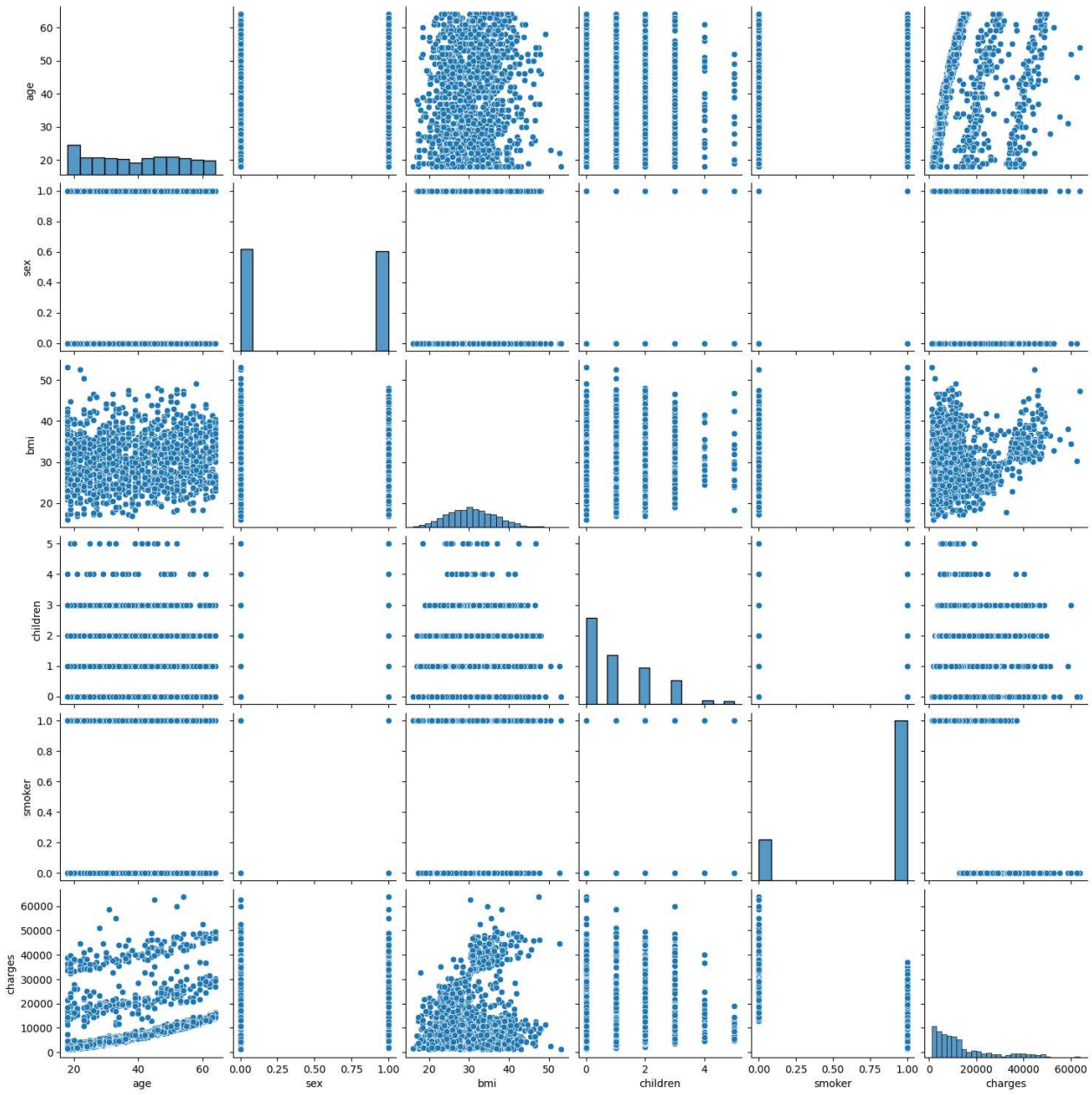
```
-0.009515626704726055
```

Logistic Regression

In [24]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

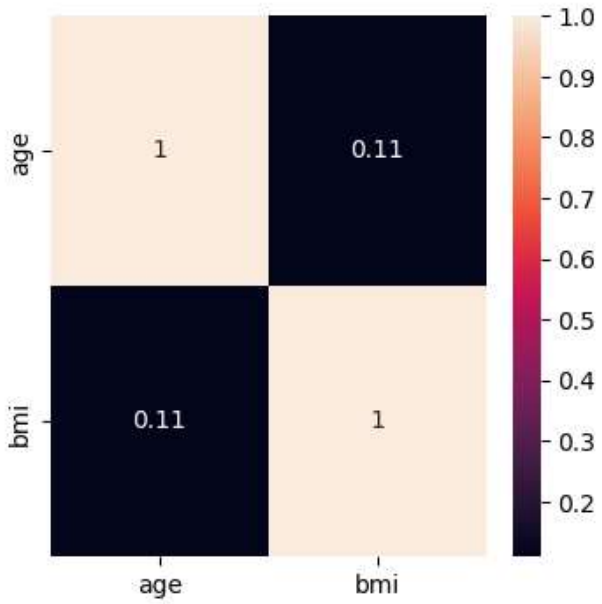
In [25]:
```
sns.pairplot(df)
```

Out[25]: <seaborn.axisgrid.PairGrid at 0x20216e33150>



In [26]:
```
Insuranced=df[['age','bmi']]
plt.figure(figsize=(4,4))
sns.heatmap(Insuranced.corr(),annot=True)
```

Out[26]: <Axes: >



In [28]:
```
x = df.iloc[:,:-1].values
y = df.iloc[:,1].values
```

In [29]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.2)
```

In [30]:
```
ml = LogisticRegression()
```

In [31]:
```
x=np.array(df['smoker']).reshape(-1,1)
x=np.array(df['age']).reshape(-1,1)
df.dropna(inplace=True)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=1)
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression(max_iter=10000)
```

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

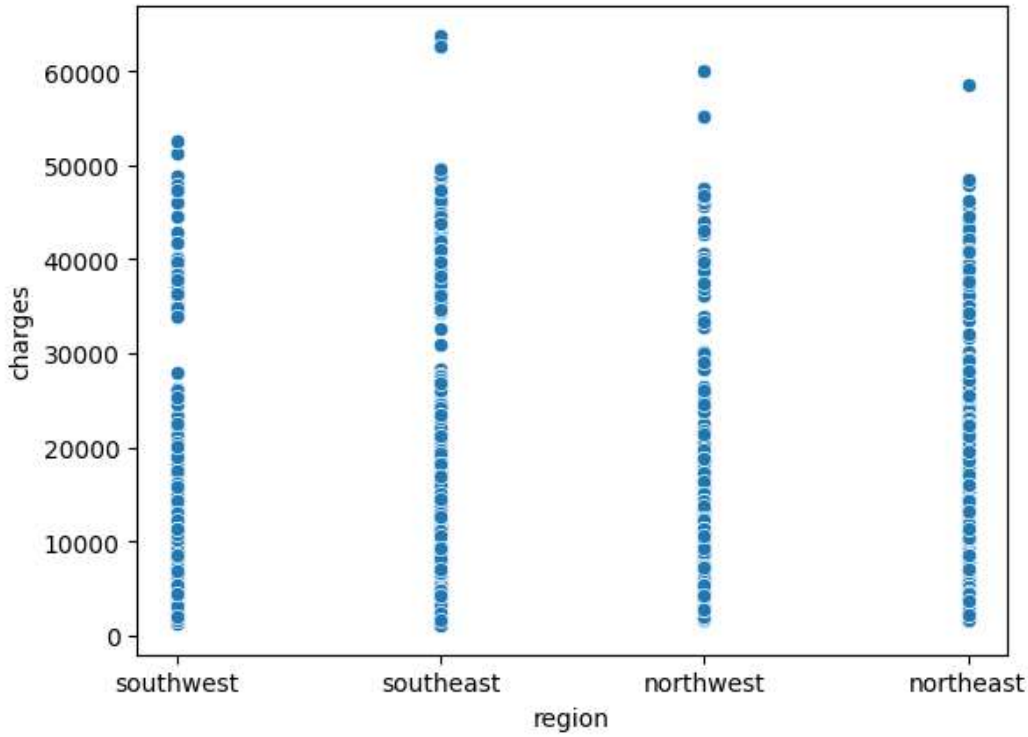
Out[32]:
```
LogisticRegression
LogisticRegression(max_iter=10000)
```

In [33]:
```
score=lr.score(x_test,y_test)
print(score)
```

```
0.48059701492537316
```

In [35]:
```
sns.scatterplot(data=df,x='region',y='charges')
```

Out[35]:
```
<Axes: xlabel='region', ylabel='charges'>
```



Desion tree

In [36]:
```
from sklearn.tree import DecisionTreeClassifier
clf=DecisionTreeClassifier()
clf.fit(x_train,y_train)
```

Out[36]:
```
DecisionTreeClassifier
DecisionTreeClassifier()
```

In [37]:
```
convert={'sex':{'female':0,'male':1}}
df=df.replace(convert)
df
```

Out[37]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	0	southwest	16884.92400
1	18	0	33.770	1	1	southeast	1725.55230
2	28	0	33.000	3	1	southeast	4449.46200
3	33	0	22.705	0	1	northwest	21984.47061
4	32	0	28.880	0	1	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	0	30.970	3	1	northwest	10600.54830
1334	18	1	31.920	0	1	northeast	2205.98080
1335	18	1	36.850	0	1	southeast	1629.83350
1336	21	1	25.800	0	1	southwest	2007.94500
1337	61	1	29.070	0	0	northwest	29141.36030

1338 rows × 7 columns

In [38]:
```
X=['age','sex']
y=['yes','no']
all_inputs=df[X]
all_classes=df['smoker']
```

In [39]:
```
X_train,x_test,y_train,y_test=train_test_split(all_inputs,all_classes,test_size=0.7)
```

In [40]:
```
clf=DecisionTreeClassifier(random_state=0)
```

In [41]:
```
clf.fit(X_train,y_train)
```

Out[41]:
```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```

In [44]:
```
score=clf.score(X_train,y_train)
print(score)
```

```
0.85785536159601
```

Random Forest

In [47]:
```
import matplotlib.pyplot as plt,seaborn as sns
from sklearn.model_selection import train_test_split
```

In [48]:
```
x=df.drop('smoker',axis=1)
y=df['smoker']
```

In [49]:
```
convert={'sex':{'female':0,'male':1}}
df=df.replace(convert)
df
```

Out[49]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	0	southwest	16884.92400
1	18	0	33.770	1	1	southeast	1725.55230
2	28	0	33.000	3	1	southeast	4449.46200
3	33	0	22.705	0	1	northwest	21984.47061
4	32	0	28.880	0	1	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	0	30.970	3	1	northwest	10600.54830
1334	18	1	31.920	0	1	northeast	2205.98080
1335	18	1	36.850	0	1	southeast	1629.83350
1336	21	1	25.800	0	1	southwest	2007.94500
1337	61	1	29.070	0	0	northwest	29141.36030

1338 rows × 7 columns

In [50]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(X_train,y_train)
```

Out[50]:
```
RandomForestClassifier
RandomForestClassifier()
```

In [51]:
```
score=rfc.score(x_test,y_test)
print(score)
```

```
0.7342582710779082
```

In [52]:
```
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 [53]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=params,cv=2,scoring="accuracy")
grid_search.fit(X_train,y_train)
```

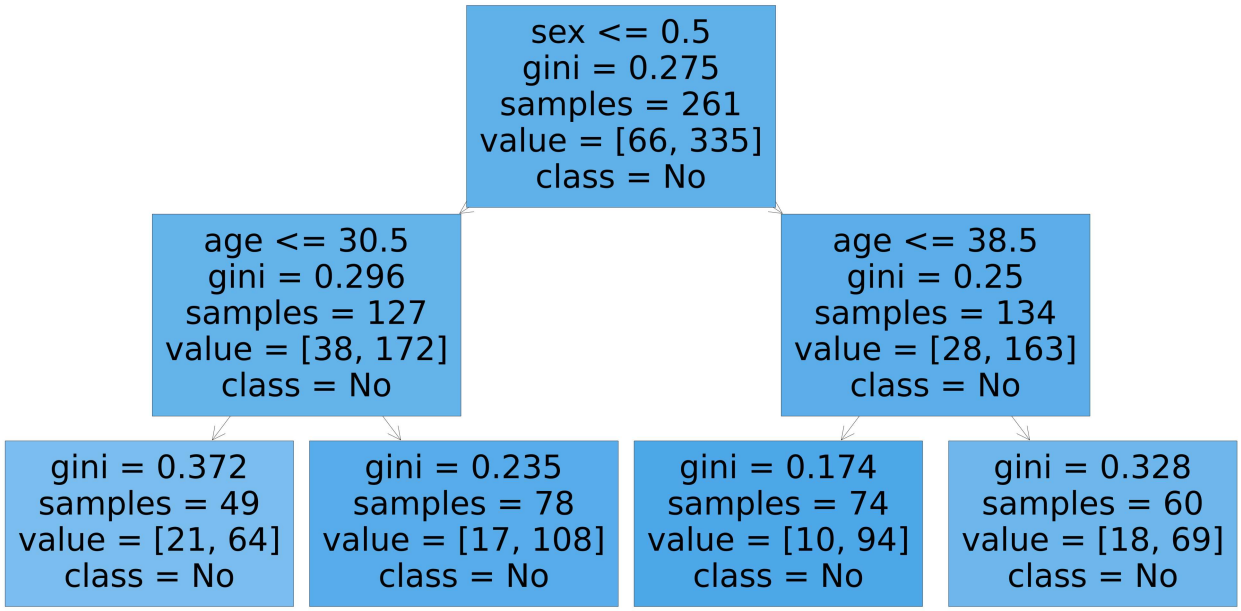
Out[53]:
```
GridSearchCV
estimator: RandomForestClassifier
RandomForestClassifier
```

In [55]:
```
grid_search.best_score_
```

Out[55]:
```
0.8453855721393035
```

In [56]:
```
rf_best=grid_search.best_estimator_
```

In [58]:
```
from sklearn.tree import plot_tree
from sklearn.tree import DecisionTreeClassifier
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```



In [59]:
```
rf_best.feature_importances_
```

Out[59]:
```
array([0.8925801, 0.1074199])
```

In [61]:
```
imp_df=pd.DataFrame({"varname":X_train.columns,"Imp":rf_best.feature_importances_})
imp_df.sort_values(by="Imp",ascending=False)
```

Out[61]:

	varname	Imp
0	age	0.89258
1	sex	0.10742

Conclusion:The above implemented models "Decision Tree" is high accuracy score.So it is the best model

In [ ]:
```
```