Problem Statement:"Evaluating Which model is best-fit for Insurance data

In [97]:
```
import numpy as np,pandas as pd
import matplotlib.pyplot as plt

import seaborn as sns
```

In [98]:
```
df=pd.read_csv(r"C:\Users\Y.Saranya\Downloads\insurance.csv")
df
```

Out[98]:

	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 [99]:
```
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 [100]:
```
df.shape
```

Out[100]:
```
(1338, 7)
```

In [101]:
```
df.describe
```

Out[101]:
```
<bound method NDFrame.describe of       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 x 7 columns]>
```

In [102]:
```
df.isna().any()
```

Out[102]:
```
age         False
sex         False
bmi         False
children    False
smoker      False
region      False
charges     False
dtype: bool
```

In [103]:
```
#To finding null values
df.isnull().sum()
```

Out[103]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [104]:
```
df.head
```

Out[104]:
```
<bound method NDFrame.head of       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 x 7 columns]>
```

In [105]:
```
df.tail
```

Out[105]:
```
<bound method NDFrame.tail of       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 x 7 columns]>
```

In [106]:
```
convert={"sex":{"female":0,"male":1}}
df=df.replace(convert)
df
```

Out[106]:

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

1338 rows × 7 columns

In [107]:
```
convert={'smoker':{"yes":1,"no":0}}
df=df.replace(convert)
df
```

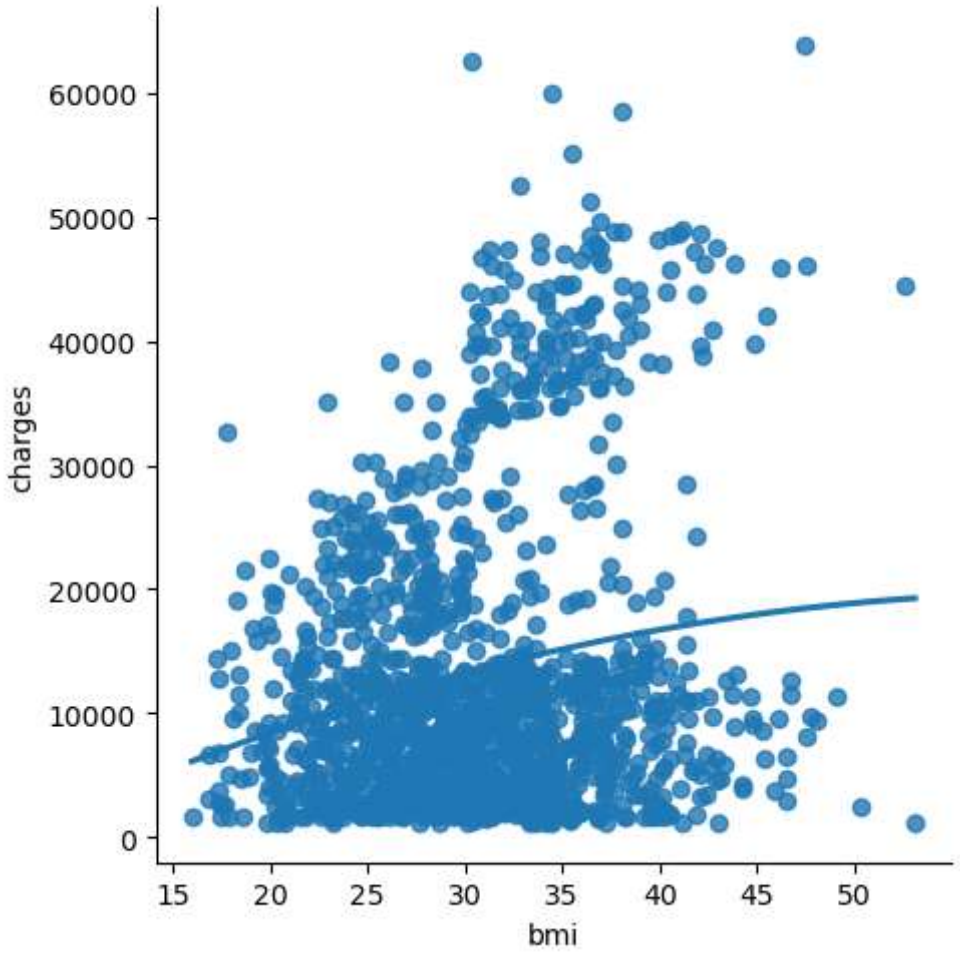
Out[107]:

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

1338 rows × 7 columns

In [108]:
```
sb.lmplot(x="bmi",y="charges",data=df,order=2,ci=None)
```

Out[108]:
```
<seaborn.axisgrid.FacetGrid at 0x1d34af76110>
```



In [109]:
```
df.drop("charges",axis=1)
```

Out[109]:

	age	sex	bmi	children	smoker	region
0	19	0	27.900	0	1	southwest
1	18	1	33.770	1	0	southeast
2	28	1	33.000	3	0	southeast
3	33	1	22.705	0	0	northwest
4	32	1	28.880	0	0	northwest
...	...	...	...	...	...	...
1333	50	1	30.970	3	0	northwest
1334	18	0	31.920	0	0	northeast
1335	18	0	36.850	0	0	southeast
1336	21	0	25.800	0	0	southwest
1337	61	0	29.070	0	1	northwest

1338 rows × 6 columns

In [110]:
```
sex={"sex":{"female":0,"male":1}}
df=df.replace(sex)
df
```

Out[110]:

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

1338 rows × 7 columns

In [111]:
```
smoker={"smoker":{"yes":1,"no":0}}
df=df.replace(smoker)
df
```

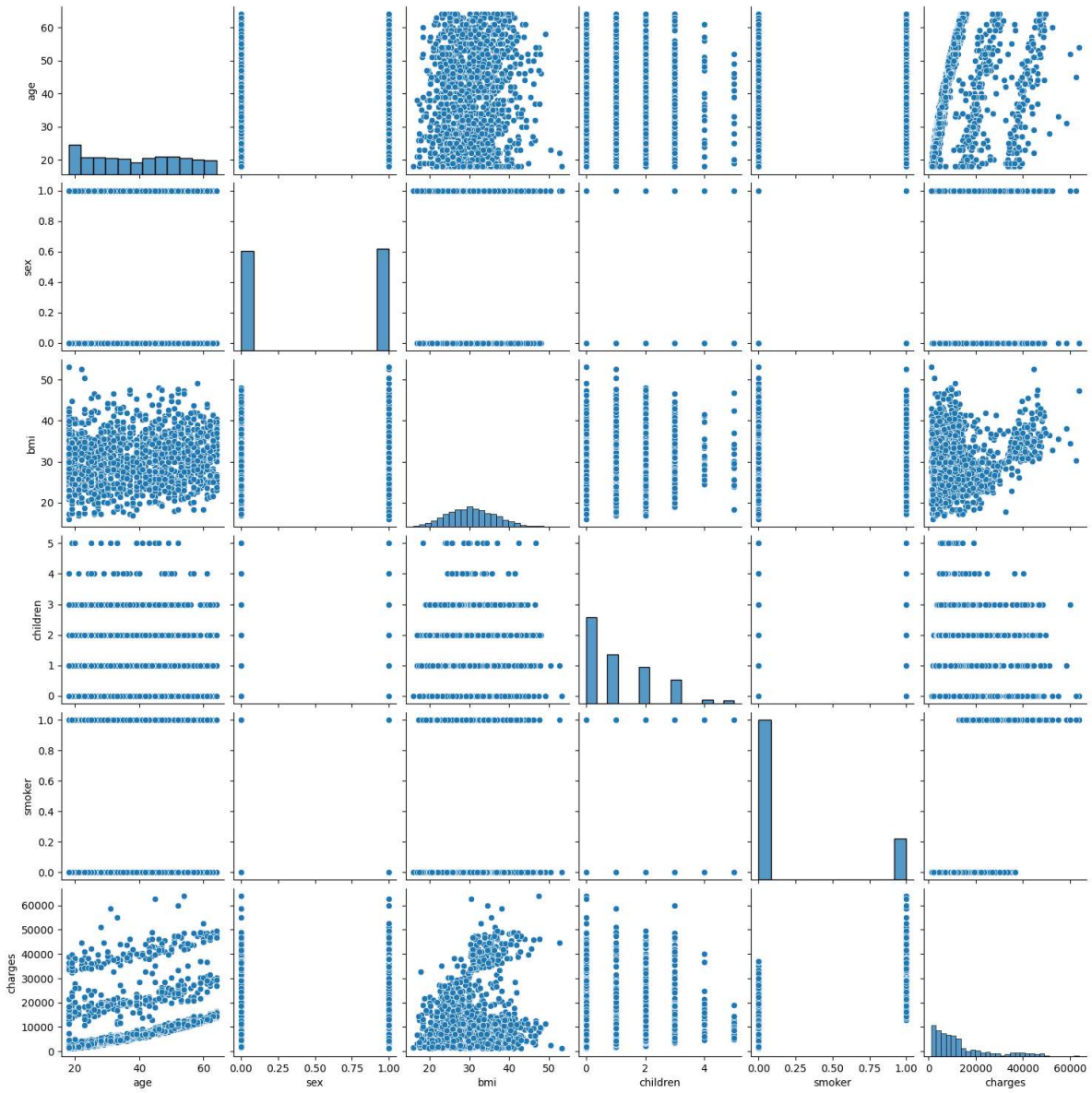
Out[111]:

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

1338 rows × 7 columns

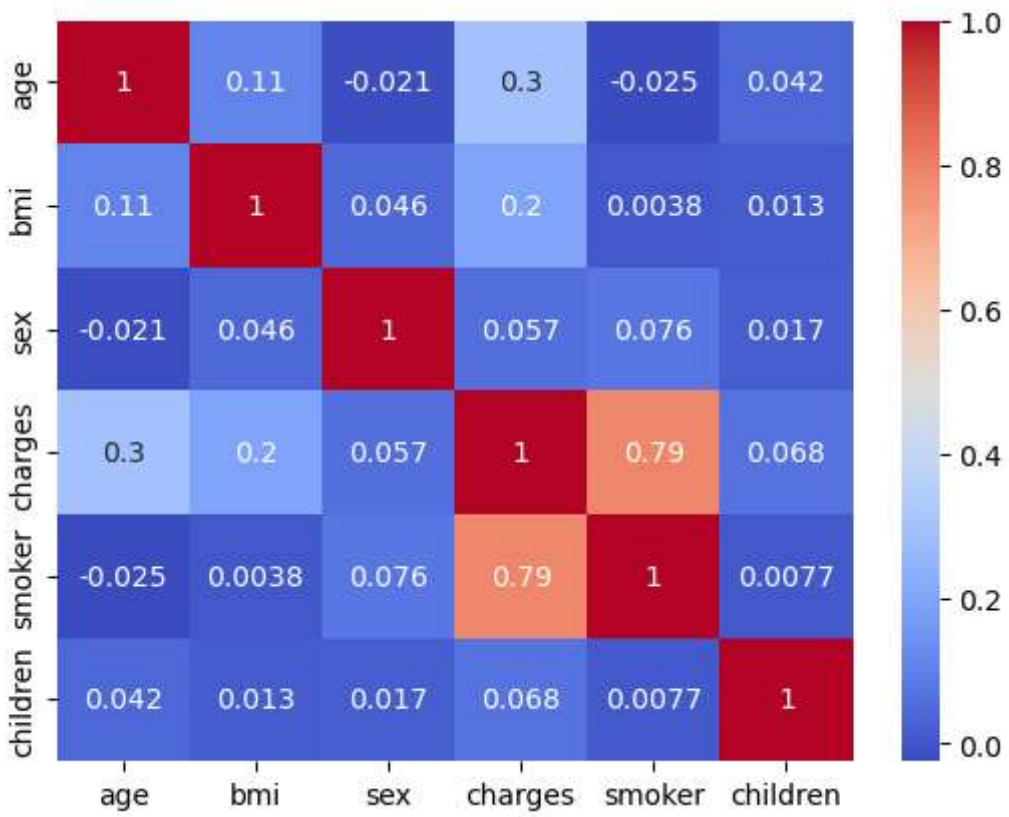
In [112]:
```
#Data visualization
sns.pairplot(df)
```

Out[112]: <seaborn.axisgrid.PairGrid at 0x1d34fa0ff10>



In [113]:
```
#heat map for data
columns=df[['age','bmi','sex','charges','smoker','children']]
subset=columns.corr()
sns.heatmap(subset,annot=True,cmap='coolwarm')
```

Out[113]: <Axes: >



In [114]:
```
#feature scaling or trainning our model
from sklearn.model_selection import train_test_split
X=df[['age','bmi','sex','charges','children']]
y=df['smoker']
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.30,random_state
```

Data modelling

In [115]:
```
#Now we are calculating our data fit for linear regression model
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[115]:
```
▾ LinearRegression
LinearRegression()
```

In [116]:
```
print(lr.intercept_)
coeff_data=pd.DataFrame(lr.coef_,X.columns,columns=['coefficient'])
coeff_data
```

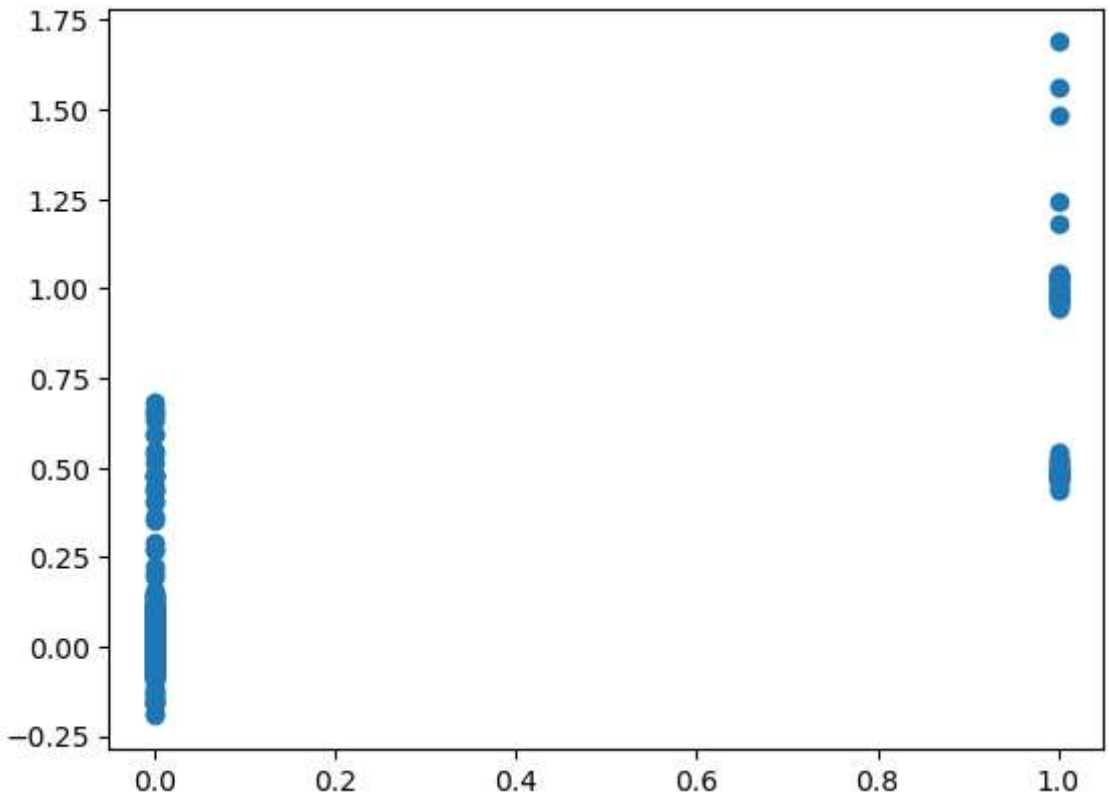
```
0.42484839188702184
```

Out[116]:

	coefficient
age	-0.007772
bmi	-0.010035
sex	0.019906
charges	0.000030
children	-0.017475

In [117]:
```
lr.score(x_test,y_test)
predictions=lr.predict(x_test)
plt.scatter(y_test,predictions)
```

Out[117]:
```
<matplotlib.collections.PathCollection at 0x1d34c9ef310>
```



In [118]:
```
'''In above linear regression model our insurance data is not fitted accuratly
 so now we are on logistic regression model'''
```

Out[118]:
```
'In above linear regression model our insurance data is not fitted accuratl
y.\n so now we are on logistic regression model'
```

In [119]:
```
#importing libraries& dropping null values
x=np.array(df['charges']).reshape(-1,1)
y=np.array(df['smoker']).reshape(-1,1)
df.dropna(inplace=True)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=
from sklearn.linear_model import LogisticRegression
lg=LogisticRegression()
```

In [120]:
```
x=df[features].values
y=df[target].values
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[120], line 1
----> 1 x=df[features].values
      2 y=df[target].values

NameError: name 'features' is not defined
```

In [121]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5)
```

In [122]:
```
a=LinearRegression()
a.fit(x_train,y_train)
```

Out[122]:
```
LinearRegression()
```

In [123]:
```
print(a.score(x_test,y_test))
```

```
0.6632708523988279
```

In [124]:
```
a=LinearRegression()
a.fit(x_train,y_train)
train_score_a=a.score(x_train,y_train)
test_score_a=a.score(x_test,y_test)
print("\nLinearModel\n")
print("The train score for lr model is {}".format(train_score_a))
print("The train score for lr model is {}".format(test_score_a))
```

```
LinearModel

The train score for lr model is 0.5683171218495688
The train score for lr model is 0.6632708523988279
```

Decision tree

In [125]:
```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
```

In [126]:
```
df=pd.read_csv(r"C:\Users\Y.Saranya\Downloads\insurance.csv")
df
```

Out[126]:

	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

In [127]:
```
df["region"].value_counts()
```

Out[127]:
```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

In [128]:
```
convert={"sex":{"female":0,"male":1}}
df=df.replace(convert)
df
```

Out[128]:

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

1338 rows × 7 columns

In [129]:
```
convert={"smoker":{"yes":0,"no":1}}
df=df.replace(smoker)
df
```

Out[129]:

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

1338 rows × 7 columns

In [130]:
```
x=["age","sex","children","bmi","charges"]
y=["0","1"]
all_inputs=df[x]
all_classes=df["smoker"]
```

In [131]:
```
x_train,X_test,y_train,y_test=train_test_split(all_inputs,all_classes,test_siz
x_train.shape,x_test.shape
```

Out[131]:
```
((669, 5), (669, 1))
```

In [132]:
```
s=DecisionTreeClassifier(random_state=20)
s.fit(x_train,y_train)
score=s.score(x_test,y_test)
print(score)
```

```
C:\Users\Y.Saranya\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarni
ng: X does not have valid feature names, but DecisionTreeClassifier was fitte
d with feature names
  warnings.warn(
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[132], line 3
      1 s=DecisionTreeClassifier(random_state=20)
      2 s.fit(x_train,y_train)
----> 3 score=s.score(x_test,y_test)
      4 print(score)

File ~\anaconda3\lib\site-packages\sklearn\base.py:649, in ClassifierMixin.sc
ore(self, X, y, sample_weight)
    624 """
    625 Return the mean accuracy on the given test data and labels.
    626
   (...)
    645     Mean accuracy of ``self.predict(X)`` wrt. `y`.
    646 """
    647 from .metrics import accuracy_score
--> 649 return accuracy_score(y, self.predict(X), sample_weight=sample_weigh
t)

File ~\anaconda3\lib\site-packages\sklearn\tree\_classes.py:426, in BaseDecis
ionTree.predict(self, X, check_input)
    403 """Predict class or regression value for X.
    404
    405 For a classification model, the predicted class for each sample in X
is
   (...)
    423     The predicted classes, or the predict values.
    424 """
    425 check_is_fitted(self)
--> 426 X = self._validate_X_predict(X, check_input)
    427 proba = self.tree_.predict(X)
    428 n_samples = X.shape[0]

File ~\anaconda3\lib\site-packages\sklearn\tree\_classes.py:392, in BaseDecis
ionTree._validate_X_predict(self, X, check_input)
    390 """Validate the training data on predict (probabilities)."""
    391 if check_input:
--> 392     X = self._validate_data(X, dtype=DTYPE, accept_sparse="csr", rese
t=False)
    393     if issparse(X) and (
    394         X.indices.dtype != np.intc or X.indptr.dtype != np.intc
    395     ):
    396         raise ValueError("No support for np.int64 index based sparse
matrices")

File ~\anaconda3\lib\site-packages\sklearn\base.py:569, in BaseEstimator._val
idate_data(self, X, y, reset, validate_separately, **check_params)
    566     out = X, y
    568 if not no_val_X and check_params.get("ensure_2d", True):
--> 569     self._check_n_features(X, reset=reset)
    571 return out

File ~\anaconda3\lib\site-packages\sklearn\base.py:370, in BaseEstimator._che
ck_n_features(self, X, reset)
    367     return
    369 if n_features != self.n_features_in_:
```

```
--> 370     raise ValueError(
    371         f"X has {n_features} features, but {self.__class__.__name__}
"
    372         f"is expecting {self.n_features_in_} features as input."
    373     )

ValueError: X has 1 features, but DecisionTreeClassifier is expecting 5 featu
res as input.
```

In [135]:
```
# random forest
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt,seaborn as sb
```

In [136]:
```
df=pd.read_csv(r"C:\Users\Y.Saranya\Downloads\insurance.csv")
```

In [137]:
```
df.describe
```

Out[137]:
```
<bound method NDFrame.describe of       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 x 7 columns]>
```

In [138]:
```
df.info
```

Out[138]:
```
<bound method DataFrame.info of       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 x 7 columns]>
```

In [139]:
```
df.isna().any()
```

Out[139]:
```
age         False
sex         False
bmi         False
children    False
smoker      False
region      False
charges     False
dtype: bool
```

In [140]:
```
df.shape
```

Out[140]:
```
(1338, 7)
```

In [141]:
```
convert={"sex":{"female":0,"male":1}}
df=df.replace(convert)
df
```

Out[141]:

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

1338 rows × 7 columns

In [142]:
```
df["region"].value_counts()
```

Out[142]:
```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

In [143]:
```
r={"region":{"southeast":0,"southwest":1,"northeast":2,"northwest":3}}
df=df.replace(r)
df
```

Out[143]:

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

1338 rows × 7 columns

In [144]:
```
x=df.drop("smoker",axis=1)
y=df["smoker"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5)
from sklearn.ensemble import RandomForestClassifier
```

In [145]:
```
#importing libraries& fittingdata
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[145]:
```
▾ RandomForestClassifier

RandomForestClassifier()
```

In [146]:
```
params={"max_depth":[1,23,4,56,85],"min_samples_leaf":[4,6,8,10,12],"n_estimat
```

In [147]:
```
#for finding optimal parameter values we are importing GridSearchCv
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=params,cv=2,scoring="accurac
```

In [148]:
```
grid_search.fit(x_train,y_train)
```

Out[148]:
```
GridSearchCV
estimator: RandomForestClassifier
RandomForestClassifier
```

In [149]:
```
grid_search.best_score_
rf_best=grid_search.best_estimator_
rf_best
```

Out[149]:
```
RandomForestClassifier
RandomForestClassifier(max_depth=23, min_samples_leaf=4, n_estimators=42)
```

In [150]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[4],feature_names=X.columns,class_names=['1','0']
```

```
---------------------------------------------------------------------------
-
IndexError                                Traceback (most recent call las
t)
Cell In[150], line 3
      1 from sklearn.tree import plot_tree
      2 plt.figure(figsize=(80,40))
----> 3 plot_tree(rf_best.estimators_[4],feature_names=X.columns,class_nam
es=['1','0'],filled=True)

File ~\anaconda3\lib\site-packages\sklearn\tree\_export.py:194, in plot_tr
ee(decision_tree, max_depth, feature_names, class_names, label, filled, im
purity, node_ids, proportion, rounded, precision, ax, fontsize)
    179 check_is_fitted(decision_tree)
    181 exporter = _MPLTreeExporter(
    182     max_depth=max_depth,
    183     feature_names=feature_names,
   (...)
    192     fontsize=fontsize,
```

In [151]:
```
#accurate score for random forest
score=rfc.score(x_test,y_test)
print(score)
```

```
0.9611360239162929
```

In [152]:
```
"""In above all three models we more accuracy in LINEAR REGRESSION
 with respcet to other two models"""
```

Out[152]:
```
'In above all three models we more accuracy in LINEAR REGRESSION\n with respc
et to other two models'
```