# Insurance Dataset

# Linear Regression

Problem Statement: To find how charges are varying based on the selected features.

In [1]:
```
# importing the necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Data Collection-reading the file
df=pd.read_csv(r"C:\Users\yoshitha lakshmi\OneDrive\Desktop\python\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

In [3]:
```
# Data cleaning and Preprocessing
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.describe()
```

Out[5]:

| | 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 [6]:
```
df.shape
```

Out[6]: (1338, 7)

In [7]:
```
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 [8]:
```
df['sex'].value_counts()
```

Out[8]:
```
sex
male      676
female    662
Name: count, dtype: int64
```

In [9]:
```
convert={"sex":{"female":1,"male":2}}
df=df.replace(convert)
df
```

Out[9]:

| | age | sex | bmi | children | smoker | region | charges |
|---|---|---|---|---|---|---|---|
| **0** | 19 | 1 | 27.900 | 0 | yes | southwest | 16884.92400 |
| **1** | 18 | 2 | 33.770 | 1 | no | southeast | 1725.55230 |
| **2** | 28 | 2 | 33.000 | 3 | no | southeast | 4449.46200 |
| **3** | 33 | 2 | 22.705 | 0 | no | northwest | 21984.47061 |
| **4** | 32 | 2 | 28.880 | 0 | no | northwest | 3866.85520 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **1333** | 50 | 2 | 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 × 7 columns

In [10]:
```
convert={'smoker':{'yes':1,'no':2}}
df=df.replace(convert)
df
```

Out[10]:

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

1338 rows × 7 columns

In [11]:
```
df=df.drop('region',axis=1)
```

In [12]:
```
df=df.drop('children',axis=1)
```

In [13]:
```
features=df.columns[0:2]
```

In [14]:
```
target=df.columns[-1]
```

In [15]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [16]:
```
x=np.array(df[features])
y=np.array(df[target])
```

In [17]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.7)
regr=LinearRegression()
regr.fit(x_train,y_train)
print(regr.score(x_train,y_train))
```

```
0.109345856853253382
```

In [18]:
```
print(regr.intercept_)
```

```
317.7620065058163
```

In [19]:
```
coeff_df=pd.DataFrame(regr.coef_)
coeff_df
```

Out[19]:

| | 0 |
|---|---|
| 0 | 255.553361 |
| 1 | 1178.226540 |

# Conclusion

The accuracy for this Dataset is very low while using LinearRegression. Accuracy = 0.0835

# Ridge Regression

In [20]:
```
from sklearn.linear_model import Ridge,Lasso,RidgeCV
```

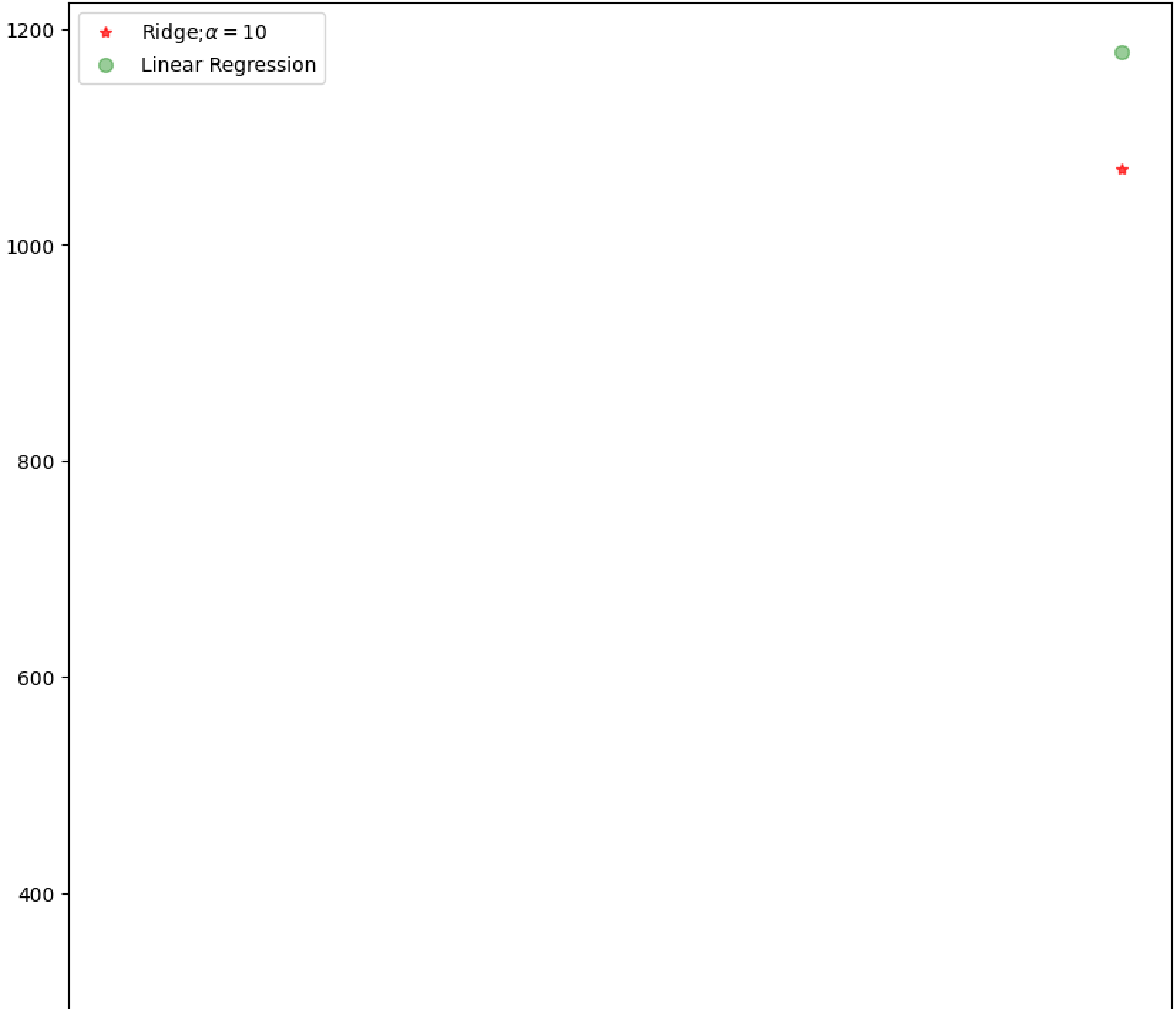
In [21]:
```
ridgeReg = Ridge(alpha=10)
ridgeReg.fit(x_train,y_train)
train_score_ridge = ridgeReg.score(x_train,y_train)
test_score_ridge = ridgeReg.score(x_test,y_test)

print('\nRidge Model\n')
print('Train score for ridge model is {}'.format(train_score_ridge))
print('Test score for ridge model is{}'.format(test_score_ridge))
```

```
Ridge Model

Train score for ridge model is 0.10932328710163086
Test score for ridge model is0.06876009932822547
```

In [22]:
```
plt.figure(figsize = (10,10))
plt.plot(features,ridgeReg.coef_,alpha=0.7,linestyle='none',marker='*',markersize=6,color='red',label=r'Ridge;$\
plt.plot(features,regr.coef_,alpha=0.4,linestyle='none',marker='o',markersize=7,color='green',label='Linear Regr
plt.xticks(rotation=90)
plt.legend()
plt.show()
```


Ridge;$\alpha = 10$
Linear Regression
1200
1000
800
600
400


# Conclusion

For Ridge Regression also the Accuracy value is very low.

Train score for ridge model is 0.08393600452093364

Test score for ridge model is0.09723547364784435

# Logostic Regression

Problem Statement: To find smokers count based on the features - sex, age.

In [23]:
```
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 LogisticRegression
```

In [24]:
```
df=pd.read_csv(r"C:\Users\yoshitha lakshmi\OneDrive\Desktop\python\insurance.csv")
df
```

Out[24]:

| | 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 [25]:
```
features=df.columns[0:3]
```

In [26]:
```
target=df.columns[-3]
```

In [27]:
```
from sklearn.datasets import load_digits
digits=load_digits()
```

In [28]:
```
print(digits.data.shape)
print(digits.target.shape)
```

```
(1797, 64)
(1797,)
```

In [29]:
```
x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_size=0.7,random_state=2)
lor=LogisticRegression(max_iter=10000)
lor.fit(x_train,y_train)
```

Out[29]:
```
LogisticRegression
LogisticRegression(max_iter=10000)
```

In [30]:
```
score=lor.score(x_test,y_test)
print(score)
```

```
0.9523052464228935
```

# Conclusion

Using Logistic Regression score is little high compared to Linear Regression. So, further process continued in Logistic Regression. Accuracy = 0.9523052464228935

# Decision Tree

In [31]:

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

In [32]:

```
df=pd.read_csv(r"C:\Users\yoshitha lakshmi\OneDrive\Desktop\python\insurance.csv")
df
```

Out[32]:

| | 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 [33]:
```
df['sex'].value_counts()
```

Out[33]:
```
sex
male      676
female    662
Name: count, dtype: int64
```

In [34]:
```
df['bmi'].value_counts()
```

Out[34]:
```
bmi
32.300    13
28.310     9
30.495     8
30.875     8
31.350     8
          ..
46.200     1
23.800     1
44.770     1
32.120     1
30.970     1
Name: count, Length: 548, dtype: int64
```

In [35]:
```
convert={'sex':{'female':0,'male':1}}
df=df.replace(convert)
df
```

Out[35]:

| | 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 [36]:
```
X=['age','sex']
y=['yes','no']
all_inputs=df[X]
all_classes=df['smoker']
```

In [37]:
```
X_train,x_test,y_train,y_test=train_test_split(all_inputs,all_classes,test_size=0.7)
```

In [38]:
```
clf=DecisionTreeClassifier(random_state=0)
```

In [39]:
```
clf.fit(X_train,y_train)
```

Out[39]:
```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```

In [40]:
```
score=clf.score(X_train,y_train)
print(score)
```

```
0.7930174563591023
```

# Random Forest

In [41]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt,seaborn as sns
from sklearn.model_selection import train_test_split
```

In [42]:
```
df=pd.read_csv(r"C:\Users\yoshitha lakshmi\OneDrive\Desktop\python\insurance.csv")
df
```

Out[42]:

| | 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 [43]:
```
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 [44]:
```
x=df.drop('smoker',axis=1)
y=df['smoker']
```

In [45]:
```
convert={'sex':{'female':0,'male':1}}
df=df.replace(convert)
df
```

Out[45]:

| | 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 [46]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(X_train,y_train)
```

Out[46]:
```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [47]:
```
score=rfc.score(x_test,y_test)
print(score)
```

```
0.7385272145144077
```

In [48]:
```
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 [49]:
```
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[49]:
```
▸ GridSearchCV
▸ estimator: RandomForestClassifier
  ▸ RandomForestClassifier
```

In [50]:
```
grid_search.best_score_
```

Out[50]:
```
0.7780597014925373
```

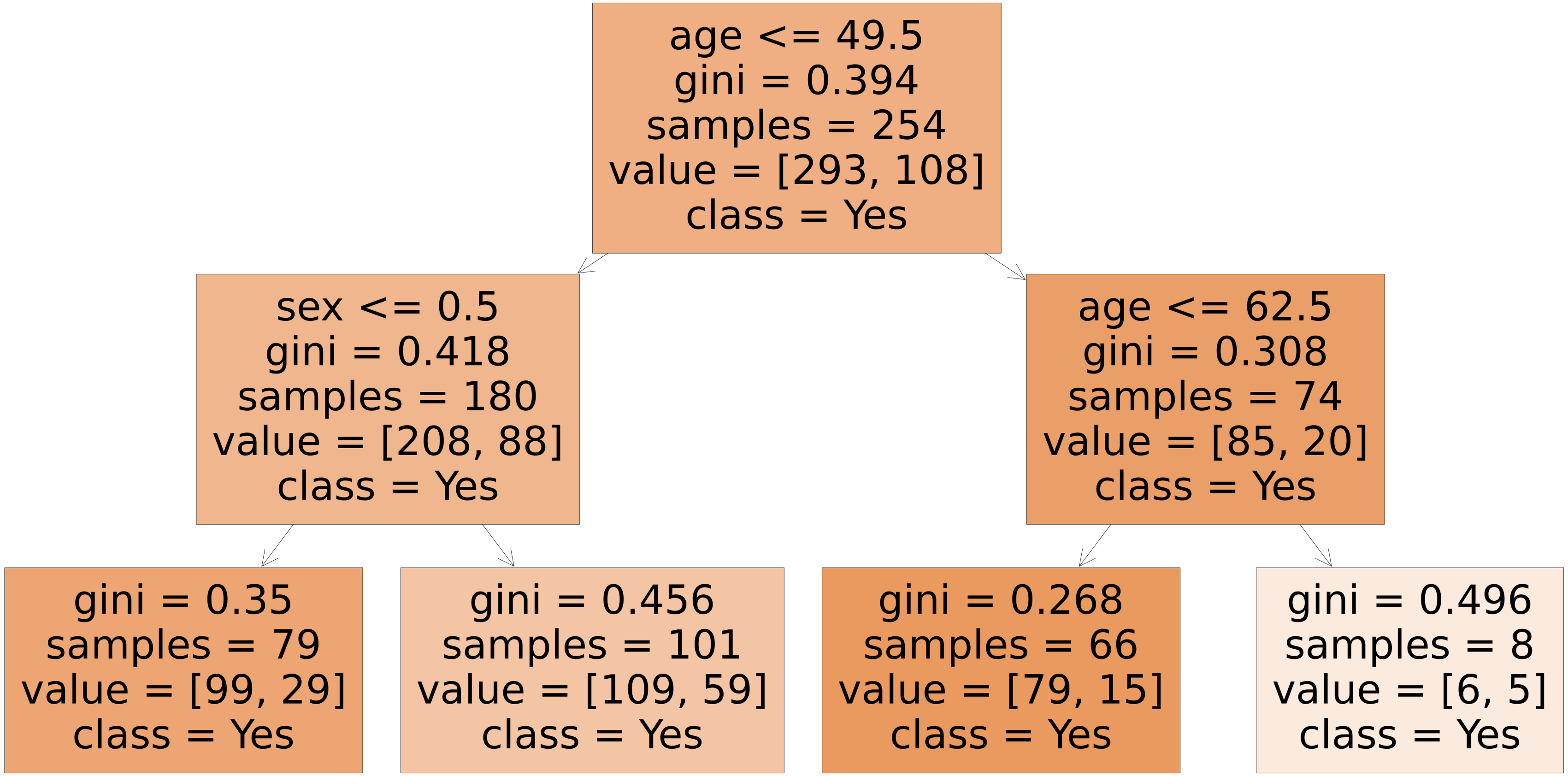
In [51]:
```
rf_best=grid_search.best_estimator_
```

In [52]:

```python
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)
```

Out[52]:

```
[Text(0.5, 0.8333333333333334, 'age <= 49.5\ngini = 0.394\nsamples = 254\nvalue = [293, 108]\nclass = Yes'),
 Text(0.25, 0.5, 'sex <= 0.5\ngini = 0.418\nsamples = 180\nvalue = [208, 88]\nclass = Yes'),
 Text(0.125, 0.16666666666666666, 'gini = 0.35\nsamples = 79\nvalue = [99, 29]\nclass = Yes'),
 Text(0.375, 0.16666666666666666, 'gini = 0.456\nsamples = 101\nvalue = [109, 59]\nclass = Yes'),
 Text(0.75, 0.5, 'age <= 62.5\ngini = 0.308\nsamples = 74\nvalue = [85, 20]\nclass = Yes'),
 Text(0.625, 0.16666666666666666, 'gini = 0.268\nsamples = 66\nvalue = [79, 15]\nclass = Yes'),
 Text(0.875, 0.16666666666666666, 'gini = 0.496\nsamples = 8\nvalue = [6, 5]\nclass = Yes')]
```



In [53]:
```
imp_df=pd.DataFrame({"varname":X_train.columns,"Imp":rf_best.feature_importances_})
imp_df.sort_values(by="Imp",ascending=False)
```

Out[53]:

| | varname | Imp |
|---|---|---|
| **0** | age | 0.798067 |
| **1** | sex | 0.201933 |

# Coclusion

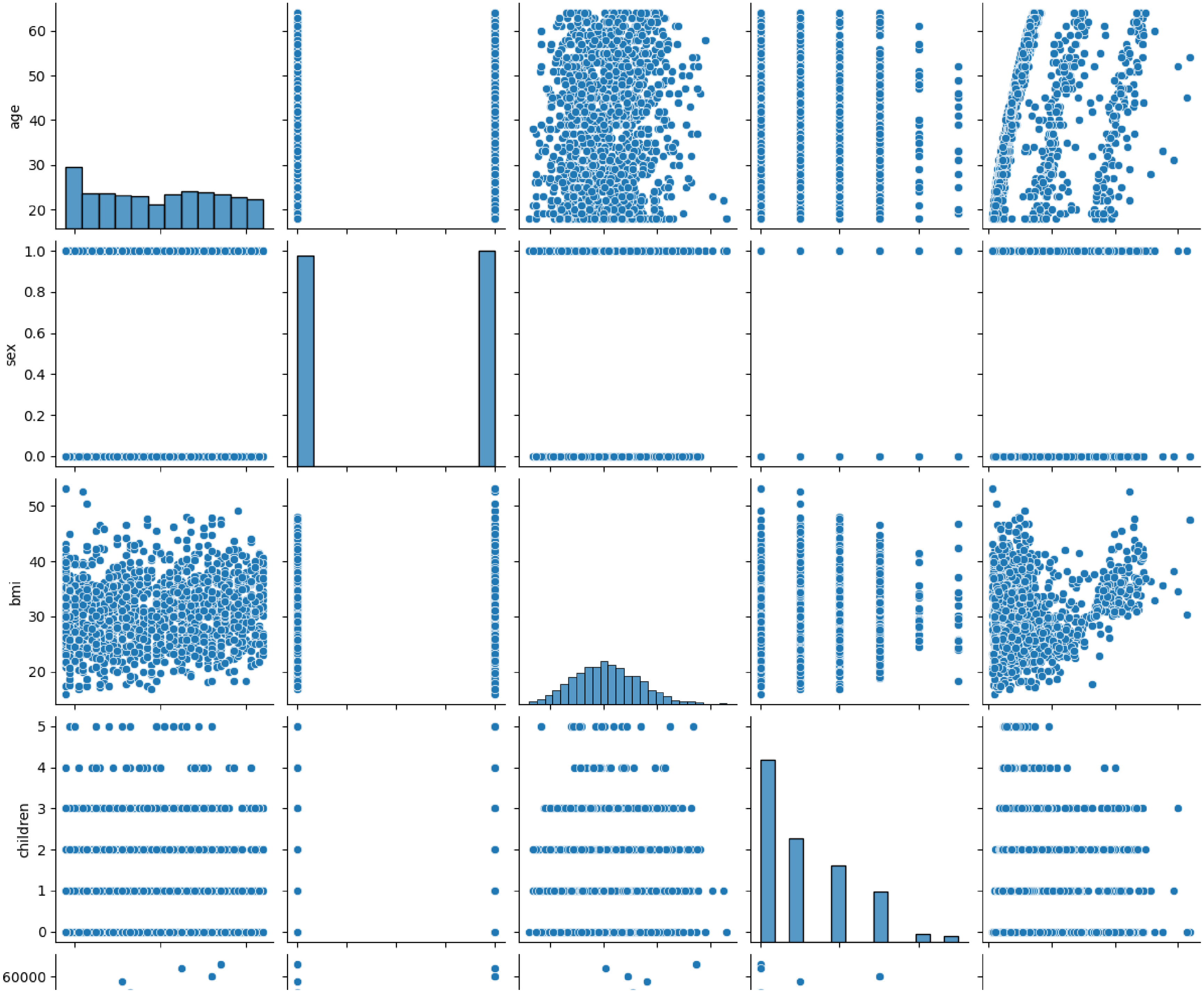
For both Decision Tree and Random Forest

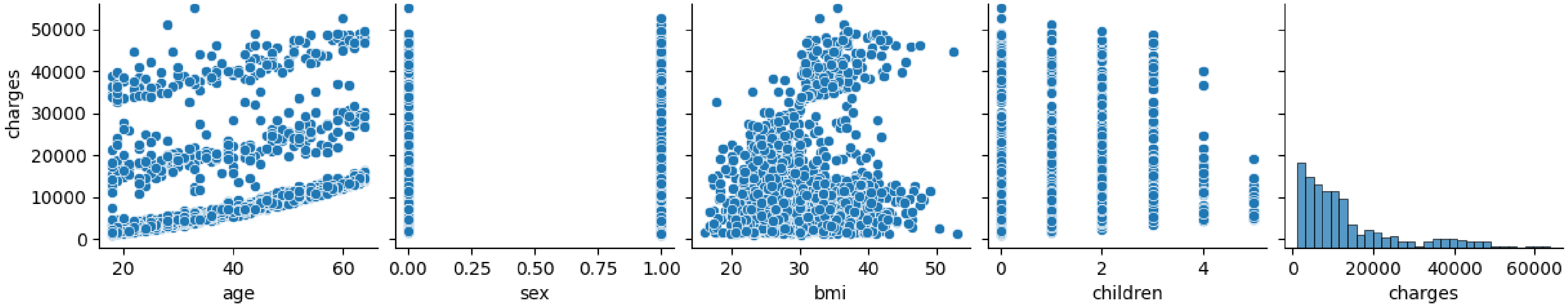
Based on our Problem Statement we classified data and build using Random Forest.

# Exploratory Data Analysis

In [54]:
```
sns.pairplot(df)
```

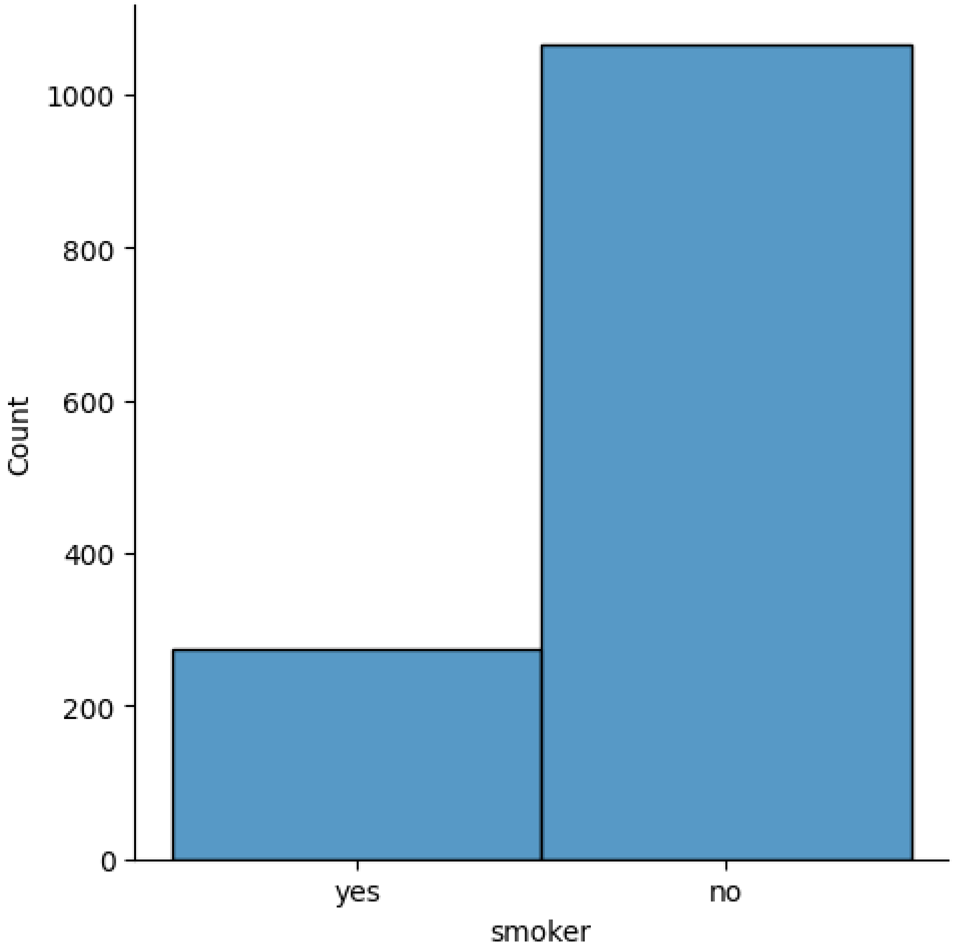
Out[54]: <seaborn.axisgrid.PairGrid at 0x1caa19f5420>


age
sex
bmi
children
60
50
40
30
20
1.0
0.8
0.6
0.4
0.2
0.0
50
40
30
20
5
4
3
2
1
0
60000


charges
age
sex
bmi
children
charges

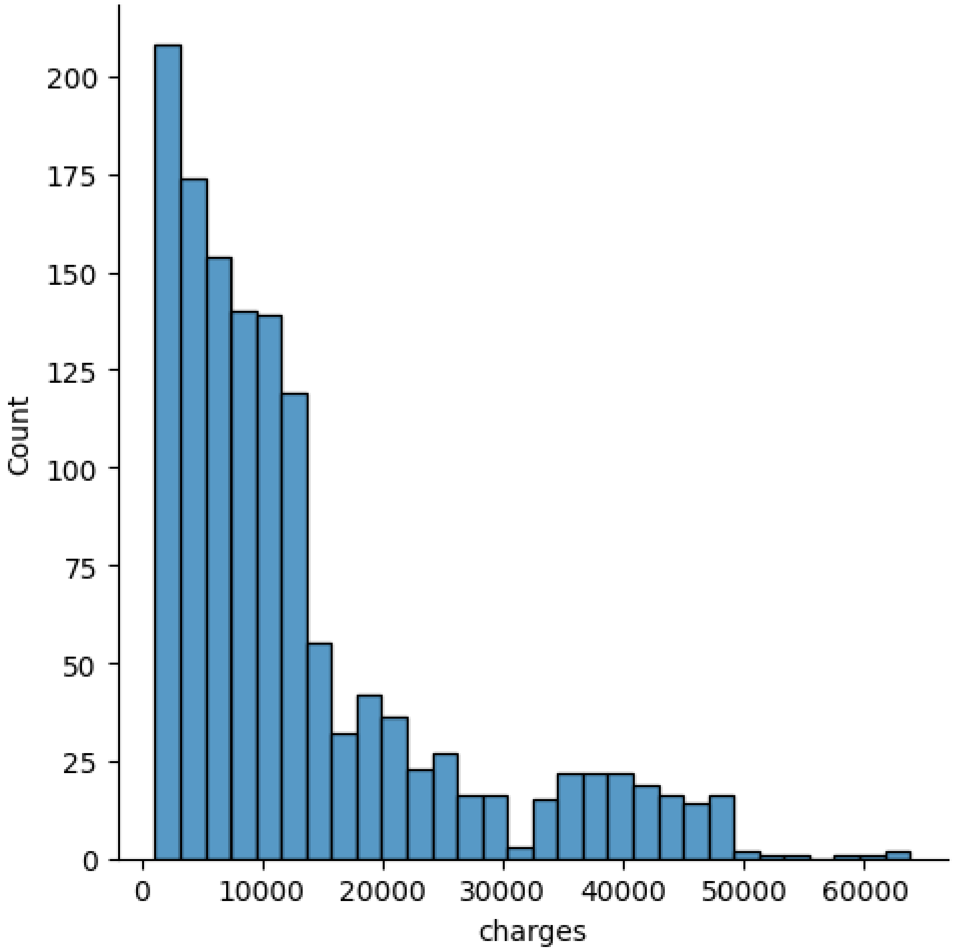
In [55]:
```
sns.displot(df['smoker'])
```

Out[55]: <seaborn.axisgrid.FacetGrid at 0x1cabaa07c40>



In [56]:
```
sns.displot(df['charges'])
```

Out[56]: <seaborn.axisgrid.FacetGrid at 0x1cabaa953c0>




# Conclusion

Using Exploratory Analysis, the relation between features has discovered.

In [57]:
```
import pickle
```

In [58]:
```
df="prediction"
pickle.dump(rfc,open(df,'wb'))
```

In [ ]:
```

```