In [24]:
```
import pandas as pd
import pickle
import warnings
warnings.filterwarnings("ignore")
```

In [25]:
```
a =pd.read_csv("Health_insurance (1).csv")
```

In [26]:
```
a
```

Out[26]:

	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 [27]:
```
a.describe()
```

Out[27]:

	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 [28]:
```
a.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 [29]:
```
a.shape
```

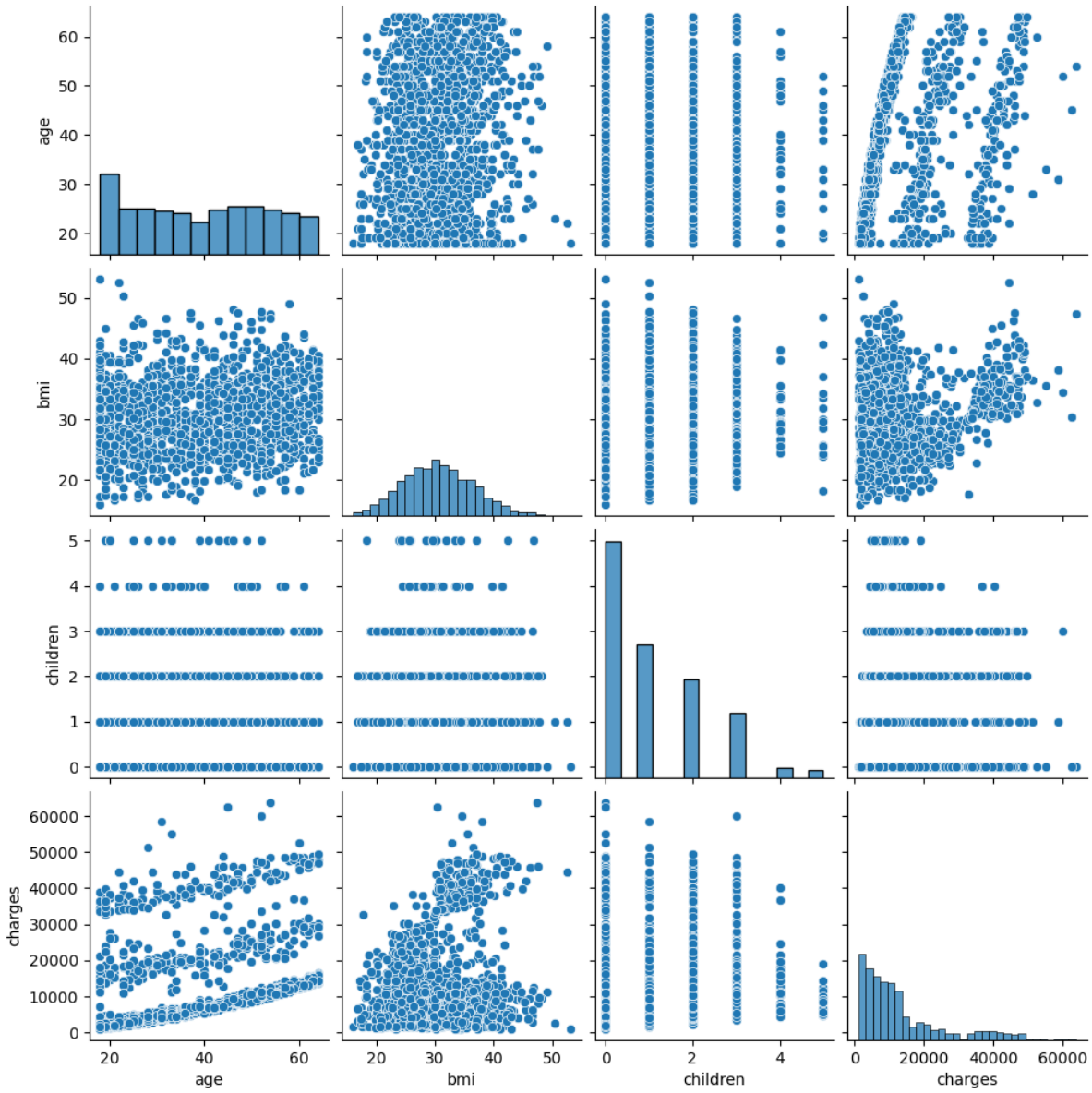
Out[29]:
```
(1338, 7)
```

In [30]:
```
a.isnull().sum()
```

Out[30]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [31]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

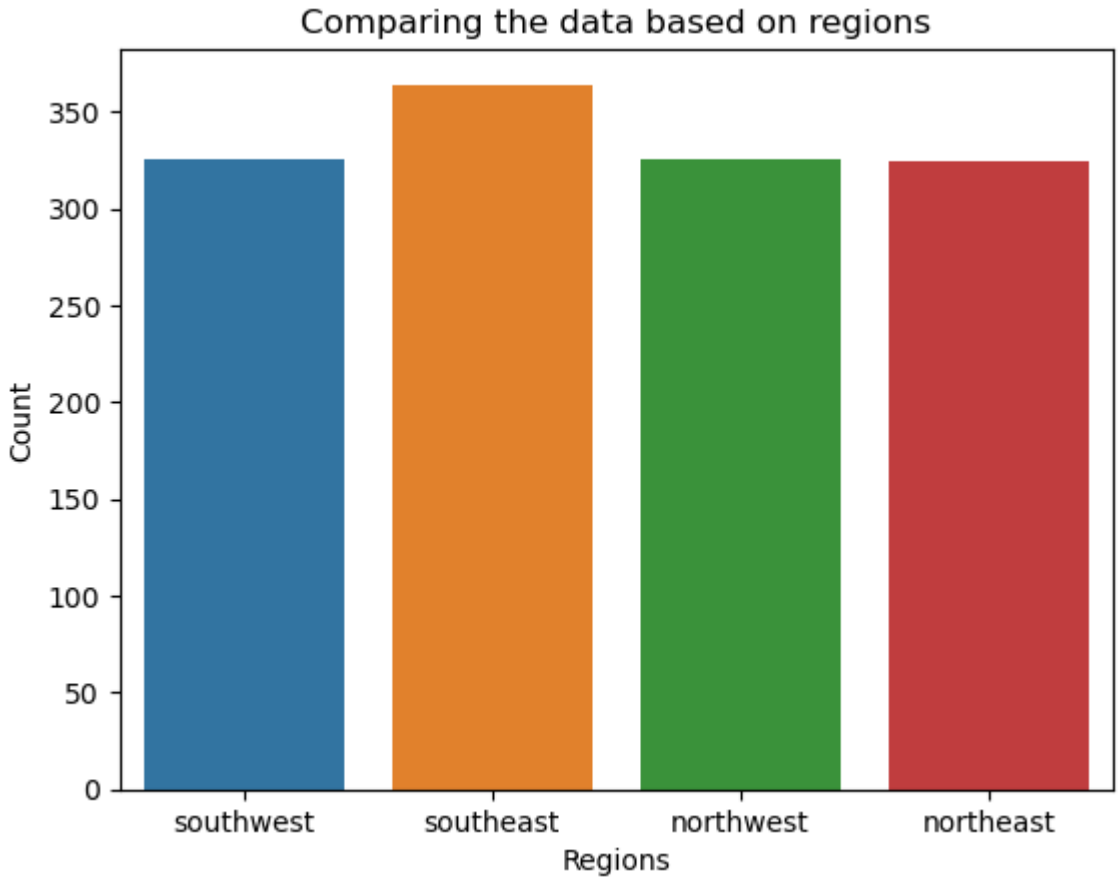
In [59]:
```
sns.pairplot(a)
plt.show()
```



In [61]:
```
a["region"].value_counts()
```

Out[61]:
```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

In [62]:
```
sns.countplot(x = a["region"])
plt.title("Comparing the data based on regions")
plt.xlabel("Regions")
plt.ylabel("Count")
plt.show()
```



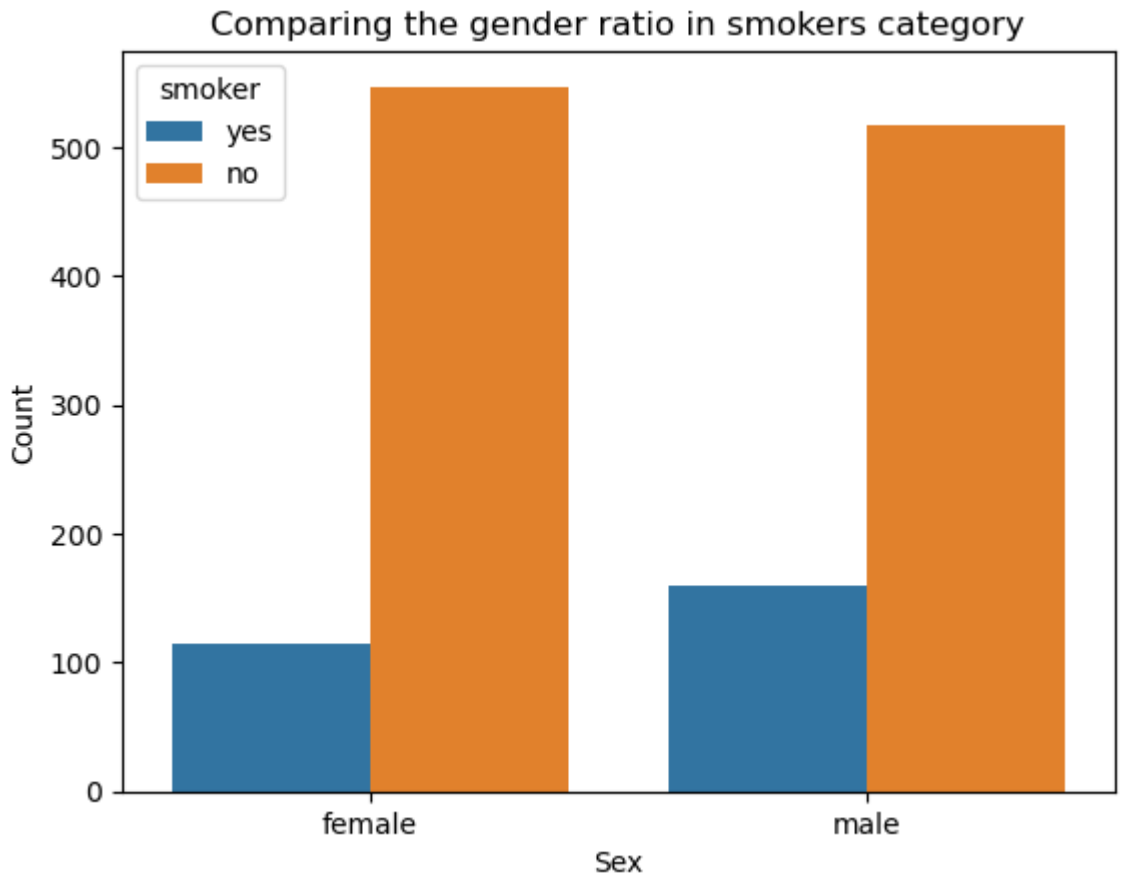
In [63]:
```
a["sex"].value_counts()
```

Out[63]:
```
male      676
female    662
Name: sex, dtype: int64
```

In [65]:
```
a["smoker"].value_counts()
```

Out[65]:
```
no     1064
yes     274
Name: smoker, dtype: int64
```

In [67]:
```
sns.countplot(data=a, x="sex", hue="smoker")
plt.title("Comparing the gender ratio in smokers category")
plt.xlabel("Sex")
plt.ylabel("Count")
plt.show()
```



In [32]:
```
data=pd.get_dummies(a,dtype=int)
```

In [33]:
```
data.shape
```

Out[33]:
```
(1338, 12)
```

In [34]:
```
data.head(10)
```

Out[34]:

	age	bmi	children	charges	sex_female	sex_male	smoker_no	smoker_yes	region_north
0	19	27.900	0	16884.92400	1	0	0	1	
1	18	33.770	1	1725.55230	0	1	1	0	
2	28	33.000	3	4449.46200	0	1	1	0	
3	33	22.705	0	21984.47061	0	1	1	0	
4	32	28.880	0	3866.85520	0	1	1	0	
5	31	25.740	0	3756.62160	1	0	1	0	
6	46	33.440	1	8240.58960	1	0	1	0	
7	37	27.740	3	7281.50560	1	0	1	0	
8	37	29.830	2	6406.41070	0	1	1	0	
9	60	25.840	0	28923.13692	1	0	1	0	

In [35]:
```
y=data['charges']
x=data.drop('charges',axis=1)
```

In [47]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=11)
```

In [48]:
```
x_test.head(5)
```

Out[48]:

	age	bmi	children	sex_female	sex_male	smoker_no	smoker_yes	region_northeast	regi
1313	19	34.700	2	1	0	0	1	0	
1254	34	27.720	0	1	0	1	0	0	
372	42	33.155	1	1	0	1	0	1	
937	39	24.225	5	1	0	1	0	0	
484	48	34.300	3	0	1	1	0	0	

In [49]:
```
from sklearn.model_selection import GridSearchCV #GridSearchCV is for parameter tur
from sklearn.ensemble import RandomForestRegressor
reg=RandomForestRegressor()
n_estimators=[25,50,75,100,125,150,175,200] #number of decision trees in the forest
criterion=['mse'] #criteria for choosing nodes default = 'gini'
max_depth=[3,5,10] #maximum number of nodes in a tree default = None (it will go ti
parameters={'n_estimators': n_estimators,'criterion':criterion,'max_depth':max_dept
RFC_reg = GridSearchCV(reg, parameters)
RFC_reg.fit(x_train,y_train)
```

Out[49]:
```
GridSearchCV(estimator=RandomForestRegressor(),
             param_grid={'criterion': ['mse'], 'max_depth': [3, 5, 10],
                         'n_estimators': [25, 50, 75, 100, 125, 150, 175, 200]})
```

In [50]:
```
RFC_reg.best_params_
```

Out[50]:
```
{'criterion': 'mse', 'max_depth': 5, 'n_estimators': 175}
```

In [51]:
```
reg=RandomForestRegressor(n_estimators=125,criterion='mse',max_depth=5)
```

In [52]:
```
reg.fit(x_train,y_train)
```

Out[52]:
```
RandomForestRegressor(criterion='mse', max_depth=5, n_estimators=125)
```

In [53]:
```
y_pred=reg.predict(x_test)
```

In [54]:
```
y_pred
```

Out[54]:
```
array([36313.49832168,  6051.51233791,  7900.41727041,  7937.67247513,
       10412.3678673 , 12403.6283807 ,  6277.64195999, 14343.30322443,
       14882.33840172, 44249.81917802,  4062.77278782, 13343.73081329,
       12041.84455988,  2834.12461468,  7188.28390092, 25427.98836945,
        4006.13893588,  2557.09887707,  7021.9161639 , 14106.49905248,
        9365.56219324,  6956.19687256,  2500.35866915,  4308.2448154 ,
        3699.66037698,  9807.96475179,  6563.13933865,  7254.44562817,
       18699.83738138,  9616.43020131, 40362.21595493, 21291.20679336,
        2664.59462964,  2631.64384729, 21826.95848098, 12335.74248548,
        6013.82681346, 16836.40668246, 10862.09855014,  6569.68611999,
        6941.44268142, 12627.07818441, 35122.45467305,  2393.603264  ,
        9669.9189261 , 11921.23517353, 21303.61754544,  2500.35866915,
       10048.01664037, 35458.47065285, 12729.08685913, 38506.68985557,
       17324.17466636, 46436.35062487, 10472.89902437,  2355.73613423,
        6055.05784912, 14370.01156659, 12463.8219875 ,  8491.33303052,
        7198.14714857, 13512.28659585, 12856.17994674, 48735.5470598 ,
        8356.98657032,  7616.54453884, 14957.46307138, 35043.52854693,
        9678.7539448 ,  4480.71448355, 39885.12979704,  7609.31303965,
        6951.33418237,  7109.10803502, 10177.16933216,  8960.19490078,
       13539.99710094, 10648.13644681,  5912.52432664,  8539.55635905,
        3662.22075512, 18736.7938785 ,  5848.16302611, 41239.06282292,
        8068.64998466,  8475.02989634,  7482.59603125,  3908.20801452,
       23420.85705297, 13727.93010504, 14191.05343656,  4368.79741178,
        7559.70699243, 14322.10917806,  7872.53208335, 13958.54192606,
        4359.36222611, 17866.0959465 , 11981.71261561, 10752.12656594,
       12046.84449792, 14291.35132816, 13104.60756802,  2526.91411074,
        6067.51735657,  9635.70602386,  7583.56686625,  7712.34876029,
        7208.50409621,  4400.34774558,  6830.06902848, 11479.02344809,
       12321.16337157,  5788.98687665, 42395.37701945,  7306.00501997,
        6037.10888844, 43010.11621789,  7144.92769436,  8228.22818152,
       10061.04056519,  2363.41849767, 35439.37452351, 10543.15217227,
       39643.82270666,  6954.70705226, 20275.24573511, 44381.61604454,
        2819.40568323,  2673.5143155 , 35305.67310557,  9105.11012605,
        6740.56333892,  8378.97944397, 14024.38931533,  7122.03549581,
        4229.80677525,  7469.86316975, 17563.33810842,  2670.62900161,
        7493.83327319, 13092.87466267, 11981.71261561, 12435.77394438,
        8695.09231729,  6689.05282984, 10226.75584409, 46494.25914675,
       13714.39271181, 44253.56976325, 21291.20679336,  6403.05666349,
        2652.45514599,  6941.44268142, 14179.72637045, 43207.12525487,
        7288.70964222, 14675.26248191,  4152.06444018, 12447.93393894,
       23350.88219663,  2804.79949364,  7150.4505041 , 18451.68131695,
        2673.5143155 ,  7573.76732059, 18125.02918937, 27245.76513473,
        9065.0062455 ,  5851.96862414,  2646.25003687, 36298.71713636,
        6489.27212974,  6901.16941935, 18801.42475425, 10913.4639419 ,
        7171.83879692,  7495.73637003, 12677.50867809, 12142.28242813,
        6157.15946239,  6695.85697842, 26329.02716395, 10506.8470944 ,
       10445.94843506, 10659.43087603,  8885.84014538,  9661.10462036,
       10513.10479488,  6600.07166598, 47182.47812368,  9218.04739214,
        7648.93065515,  6572.51909658,  7664.17002724, 10912.03363079,
       35817.78274345,  7129.44591912, 11476.27740821, 42692.11032902,
       47055.78913112, 18155.58210003, 44935.78520674, 11872.08379126,
        2834.12461468,  3742.00031943,  2716.58686298, 26331.02282402,
        6436.49842205,  4282.95861034, 13743.89434131, 16145.87108556,
        6942.52928658,  9097.94666386,  9431.03779621, 11944.83377469,
        3696.03148827, 12039.14241618,  5854.90454739,  5426.51872528,
       10885.11587288, 35997.63095791, 44080.64017615, 11304.93933293,
        7117.74028414, 35458.47065285, 12714.29698755,  6836.74074822,
       13321.72748318, 26759.98384881,  4686.60930148,  7697.97629726,
       13878.41436193, 11216.40202106, 12352.43956111, 17857.48762498,
       11158.44229825,  4326.50831005, 10415.5573113 ,  4025.86925492,
        6509.96850124,  6437.39082645,  7609.0323122 , 46861.30648317,
       41083.18229876,  4143.30402881, 11892.68589197,  2518.53252477,
        6263.83087449, 12550.75987885, 20047.76218608,  9988.89955438,
       27368.24133547, 14240.90584694,  6883.94933902,  3974.1451684 ,
```

```
 7114.46424286, 39883.58859567,  3997.43885783,  6865.03206507,
 7261.3050877 ,  7283.84091537,  5548.38163647,  2804.79949364,
 6536.32702383,  2528.15143316,  8606.94701796, 11211.13882994,
10471.27125583,  3874.48657635, 10990.98515772, 14718.30393437,
 6297.5920192 ,  4070.67463158,  5744.08882908, 14153.71501863,
12532.22919217, 48811.06744707, 14848.56231146, 11421.98614211,
 9654.42071912,  2393.603264  ,  9792.03723751,  6777.02790228,
12392.42583847,  6653.95105999, 37396.27786352,  5511.18225555,
 6732.15864627, 18392.39615544, 15654.45079018,  7000.59659045,
12465.45502876,  3943.41779229, 12091.23443857,  8066.84792806,
 5914.86876479,  3742.00031943,  3803.47827797,  4377.84573543,
45114.22451736, 10386.27412988, 13261.37814423, 17834.05673859,
 8611.12758971, 26288.7142562 ,  2393.603264  ,  9158.62802602,
14183.01241607, 44129.15506875, 19322.15387849,  4112.64005241,
 4290.25391287, 16152.64080824,  8418.88530117,  2468.26588413,
15694.50014103, 44960.0890115 , 22120.3149577 ,  7212.40589157,
13865.17291704,  2631.64384729, 14100.40276341, 39931.95451331,
12080.36901226,  6736.18388538, 18360.20774276, 12732.89203948,
 8475.61238728, 49835.47506513,  7637.76989474, 14329.22359134,
 7722.89415992,  7060.96093608,  7067.67460515])
```

In [55]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

Out[55]:
```
0.8804199439174997
```

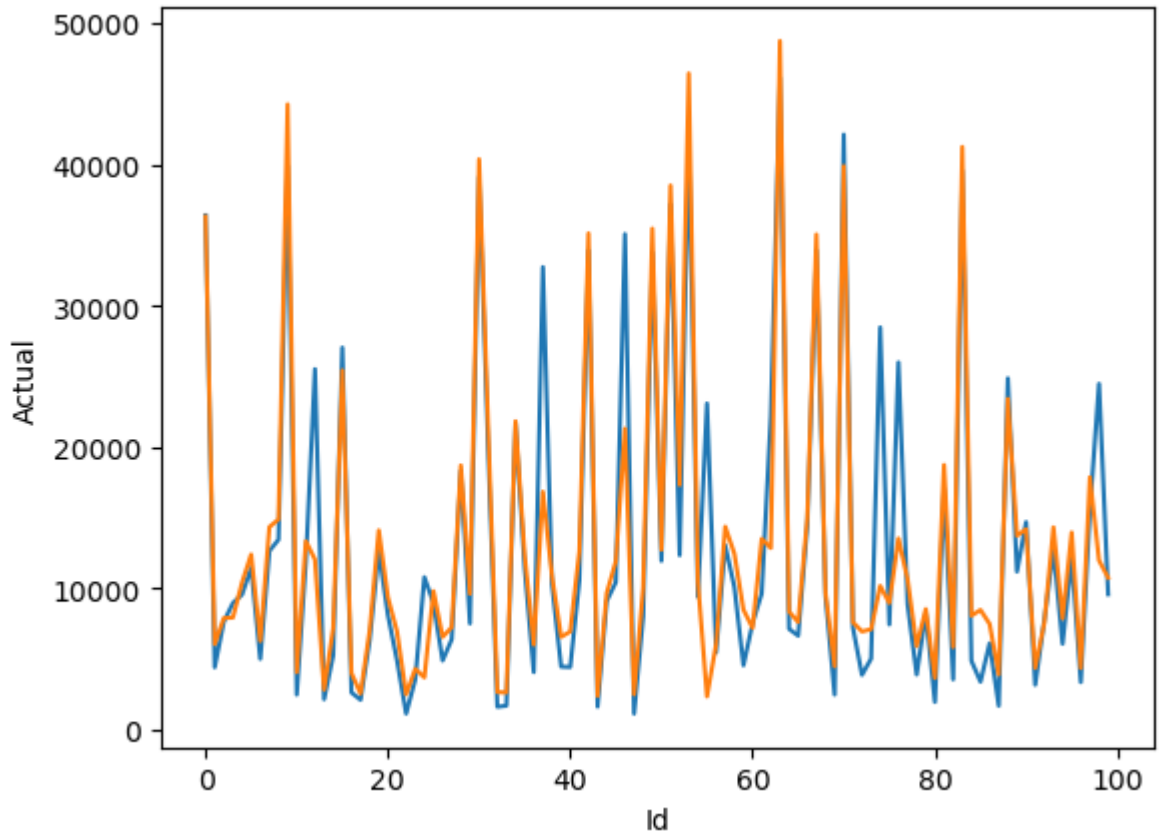
In [56]:
```
res= pd.DataFrame(columns=['Actual','Predicted'])
res['Actual']=y_test
res['Predicted']=y_pred
#Results['km']=X_test['km']
res=res.reset_index()
res['Id']=res.index
res.head(10)
```

Out[56]:

	index	Actual	Predicted	Id
0	1313	36397.57600	36313.498322	0
1	1254	4415.15880	6051.512338	1
2	372	7639.41745	7900.417270	2
3	937	8965.79575	7937.672475	3
4	484	9563.02900	10412.367867	4
5	447	11454.02150	12403.628381	5
6	220	5012.47100	6277.641960	6
7	246	12648.70340	14343.303224	7
8	920	13451.12200	14882.338402	8
9	1070	39871.70430	44249.819178	9

In [57]:
```
sns.lineplot(x='Id',y='Actual',data=res.head(100))
sns.lineplot(x='Id',y='Predicted',data=res.head(100))
plt.plot()
```

Out[57]:
```
[]
```



In [ ]: