medical-insurance-cost-prediction-model

```
[1]: 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
     from sklearn import metrics
```

```
[2]: medical_dataset = pd.read_csv('insurance.csv')
```

```
[3]: medical_dataset.head()
```

```
[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
```

```
[4]: medical_dataset.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
```

```
[5]: medical_dataset.shape
```

```
[5]:  (1338, 7)
```

```
[6]: #Cheking Missing Values
```

```
[7]: medical_dataset.isnull().sum()
```

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

```
[8]: #Discribing data
```

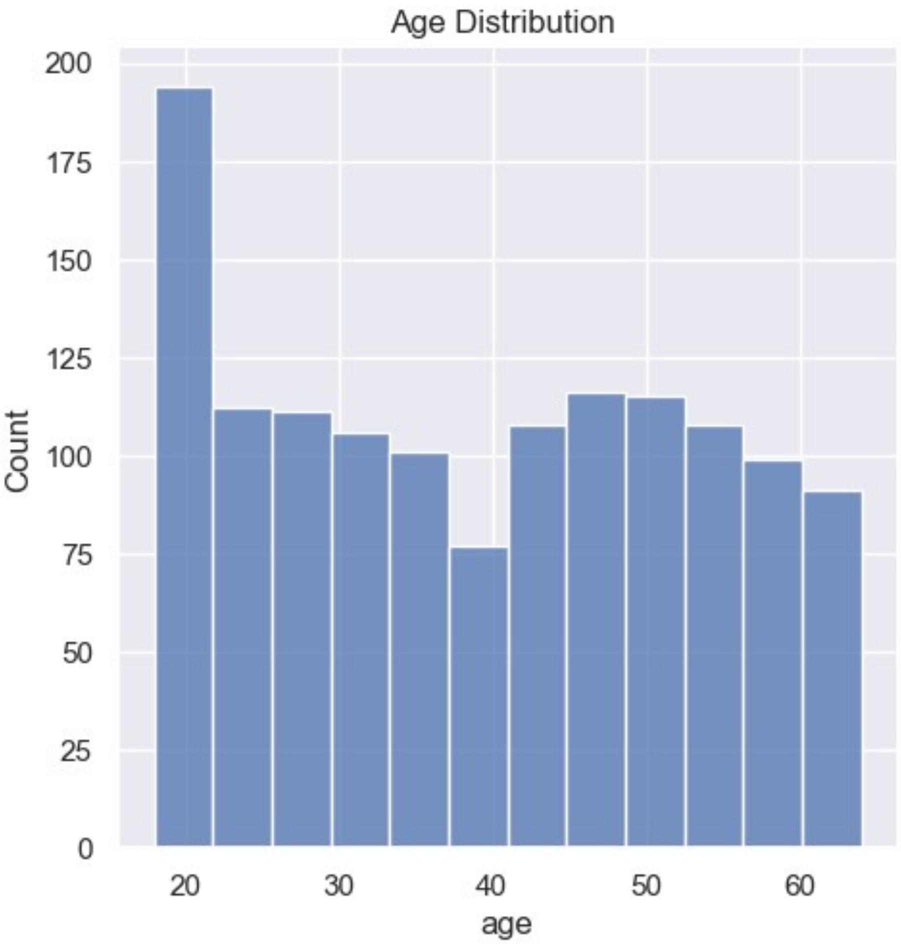
```
[9]: medical_dataset.describe()
```

```
[9]: 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    0.0000001121.873900
     25%      27.000000   26.296250    0.0000004740.287150
     50%      39.000000   30.400000    1.0000009382.033000
     75%      51.000000   34.693750    2.000000
                                       16639.912515
     max      64.000000   53.130000    5.000000  63770.428010
```

```
[10]: #Graphical distribution of Age
```

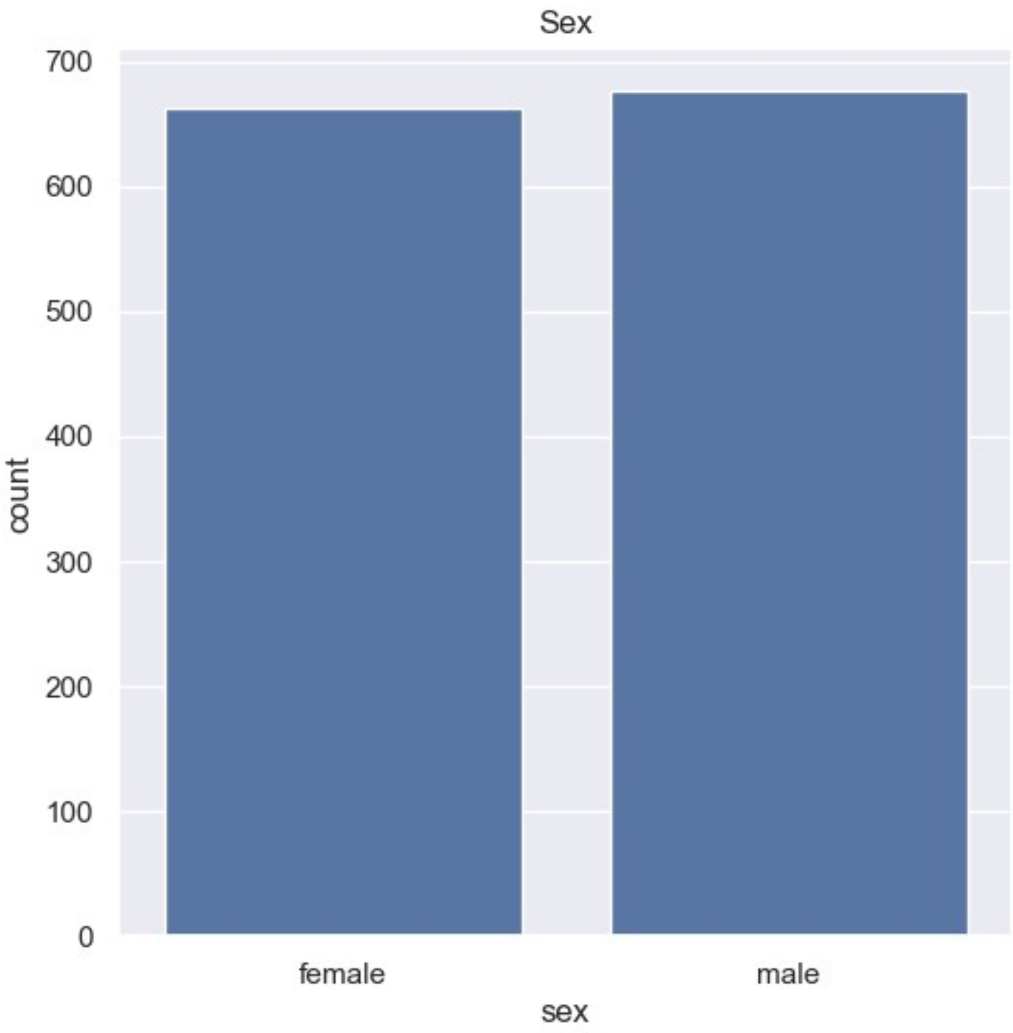
```
[11]: sns.set()
      plt.figure(figsize=(6,6))
      sns.displot(medical_dataset['age'])
      plt.title('Age Distribution ')
      plt.show()
```

```
<Figure size 600x600 with 0 Axes >
```



```
[12]: #Checking the sex
```

```
[13]: sns.set()
      plt.figure(figsize=(6,6))
      sns.countplot(x='sex',data=medical_dataset)
      plt.title('Sex')
      plt.show()
```



```
[14]: #Counting the total male and female
```

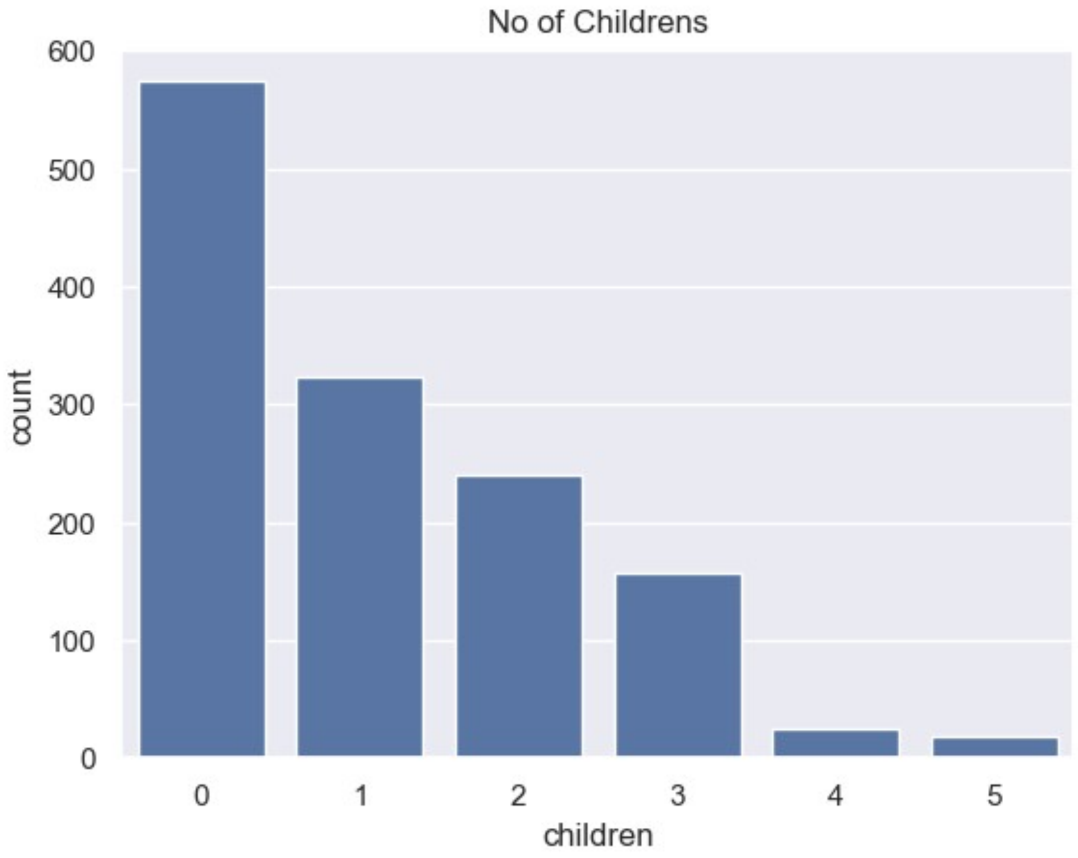
```
[15]: medical_dataset['sex'].value_counts()
```

```
[15]: sex
     male 676 female
     662
```

```
Name: count, dtype: int64
```

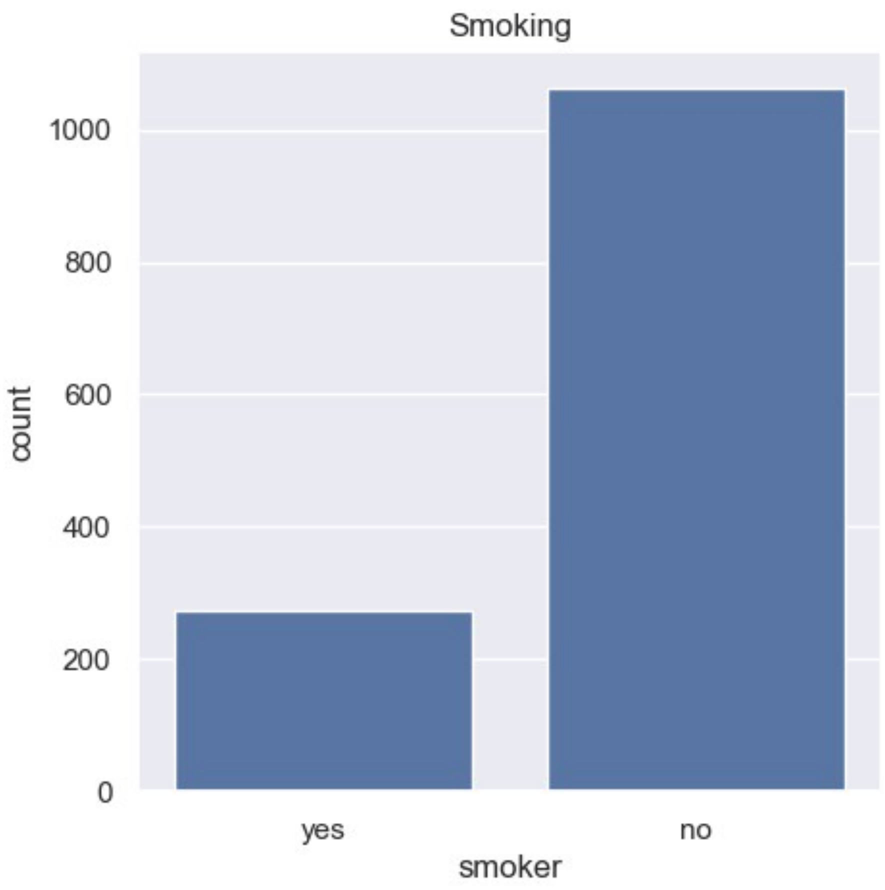
```
[16]: #Total Childer having
```

```
[17]: sns.set()
      sns.countplot(x='children',data=medical_dataset)
      plt.title('No of Childrens')
      plt.show()
```



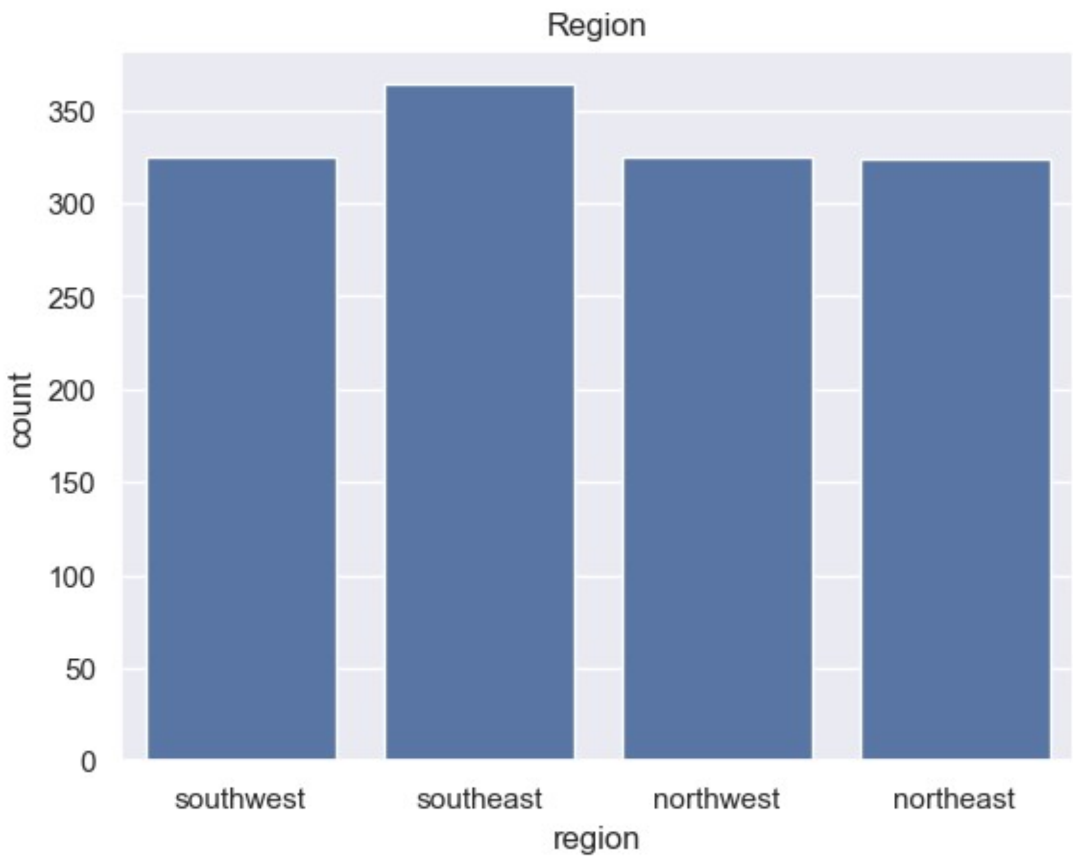
```
[18]: #smokers Graph
```

```
[19]: sns.set()
      plt.figure(figsize=(5,5))
      sns.countplot(x='smoker',data=medical_dataset)
      plt.title('Smoking')
      plt.show()
```



```
[20]: #which Region people are applying for insurence
```

```
[21]: sns.set()
      sns.countplot(x='region',data=medical_dataset)
      plt.title('Region')
      plt.show()
```



```
[22]: medical_dataset['region'].value_counts()
```

```
[22]: region

     southeast    364
     southwest    325
     northwest    325
     northeast    324
```

```
Name: count, dtype: int64
```

[23]:

```
# Data Preprocessing
# Making the Smoker column and Region column into numerical values
```

[84]:

```
#encoding the sex columns
medical_dataset.replace({'sex':{'male':0,'female':1}},inplace=True)

# smoker columsn

medical_dataset.replace({'smoker':{'yes':0,'no': 1}},inplace=True)

# for the region column

medical_dataset.replace({'region':{'southeast':0,'southwest':1,'northeast':
  2,'northwest':3}},inplace=True)
```

[25]:

```
# Splitting the feature and target
```

[26]:

```
X = medical_dataset.drop(columns='charges',axis=1)
Y  = medical_dataset['charges']
```

[27]:

```
print(X)
```

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

[1338 rows x 6 columns]
```

[28]:

```
print(Y)
```

```
0       16884.92400
1       1725.55230
2       4449.46200 3 21984.47061
4      3866.85520
          …
```

```
1333    10600.54830
1334     2205.98080
1335     1629.83350
1336     2007.94500
1337    29141.36030
Name: charges, Length: 1338, dtype: float64
```

[29]:
```
## Splitting the data into train test data
```

[30]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
 random_state=2)
```

[31]:
```
print(X.shape, X_train.shape, X_test.shape)
```

```
(1338, 6) (1070, 6) (268, 6)
```

[32]:
```
## Model Preperation
```

[33]:
```
# loading the Linear Regression model
regressor = LinearRegression()
```

[34]:
```
regressor.fit(X_train, Y_train)
```

[34]:
```
LinearRegression()
```

[35]:
```
# Model Evaluation
```

[36]:
```
training_data_prediction = regressor.predict(X_train)
r2_train = metrics.r2_score(Y_train,training_data_prediction)
print(r2_train)
```

```
0.751505643411174
```

[37]:
```
# prediciting the test data
test_data_predicition = regressor.predict(X_test)
r2_test = metrics.r2_score(Y_test,test_data_predicition)
```

[38]:
```
print(r2_test*100)
```

```
74.47273869684076
```

[39]:
```
# Prediction
# 'male':0,'female':1
## 'yes':0,'no': 1
## 'southeast':0,'southwest':1,'northeast':2,'northwest':3
```

[82]:
```
# Taking user input for each feature age =
float(input("Enter age: ")) sex = int(input("Enter
sex (0 for male, 1 for female): ")) bmi =
float(input("Enter BMI: "))
children = int(input("Enter number of children: ")) smoker =
int(input("Enter smoker status (0 for yes, 1 for no): ")) region
= int(input("Enter region (0 for southeast, 1 for southwest, 2
for␣ ↪northeast, 3 for northwest): "))

# Creating a tuple with the input data
input_data = (age, sex, bmi, children, smoker, region)
#chaning it to numpy array
input_data_as_array = np.asarray(input_data)

#reshapping the data
input_data_reshaped = input_data_as_array.reshape(1,-1)

prediction = regressor.predict(input_data_reshaped)

print("The person will get insurance money:-  ",prediction[0])
```

```
Enter age:  23
Enter sex (0 for male, 1 for female): 0
Enter BMI: 23.845
Enter number of children: 0
Enter smoker status (0 for yes, 1 for no): 1
Enter region (0 for southeast, 1 for southwest, 2 for northeast, 3
for northwest): 2

The person will get insurance money:- 1520.592421607911

C:\Users\marga\anaconda3\Lib\site-packages\sklearn\base.py:493:
UserWarning: X does not have valid feature names, but
LinearRegression was fitted with feature names warnings.warn(
```

```
[ ]:
```