

```
In [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.metrics

In [2]: data = pd.read_csv('C:/Users/parab/OneDrive/Desktop/InsuranceCost/insurance (1).csv')

In [3]: data.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]: data.shape

Out[4]: (1338, 7)

In [5]: data.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 [6]: data.isnull().sum

<bound method NDFrame.add_numeric_operations->sum of          age  sex    bmi  children  smoker  region  charges
0      False  False  False      False      False      False      False
1      False  False  False      False      False      False      False
2      False  False  False      False      False      False      False
3      False  False  False      False      False      False      False
4      False  False  False      False      False      False      False
...      ...      ...      ...      ...      ...      ...      ...
1333     False  False  False      False      False      False      False
1334     False  False  False      False      False      False      False
1335     False  False  False      False      False      False      False
1336     False  False  False      False      False      False      False
1337     False  False  False      False      False      False      False
[1338 rows x 7 columns]>

In [7]: data.describe()

Out[7]:
```

	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 [8]: sns.set()
plt.figure(figsize=(6,6))
sns.distplot(data['age'])
plt.title('Age Distribution')
plt.show()

C:\Users\parab\AppData\Local\Temp\ipykernel_13204\454659255.py:3: UserWarning:
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372758bbe5751

sns.distplot(data['age'])

Age Distribution
```



```
In [9]: plt.figure(figsize=(6,6))
sns.countplot(x='sex', data=data)
plt.title('Sex Distribution')
plt.show()

Sex Distribution
```



```
In [10]: plt.figure(figsize=(6,6))
sns.distplot(data['bmi'])
plt.title('BMI Distribution')
plt.show()

C:\Users\parab\AppData\Local\Temp\ipykernel_13204\1680175416.py:2: UserWarning:
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372758bbe5751

sns.distplot(data['bmi'])

BMI Distribution
```



```
In [11]: plt.figure(figsize=(6,6))
sns.countplot(x='children', data=data)
plt.title('children')
plt.show()

children
```



```
In [12]: data['children'].value_counts()

Out[12]:
0    574
1    324
2    240
3    157
4     25
5     18
Name: children, dtype: int64

In [13]: plt.figure(figsize=(6,6))
sns.countplot(x='region', data=data)
plt.title('region')
plt.show()

region
```



```
In [14]: #encoding sex column
data.replace({'sex':{'male':0, 'female':1}}, inplace=True)

#encoding smoker colu
data.replace({'smoker':{'no':0, 'yes':1}}, inplace=True)

#encoding region column
data.replace({'region':{'southeast':0, 'southwest':1, 'northeast':2, 'northwest':3}},inplace=True)

In [15]: X = data.drop(columns='charges', axis=1)
Y = data['charges']

In [16]: print(X)

   age  sex  bmi  children  smoker  region
0    19    1  27.900         0        1        1
1    18    0  33.770         1        0        3
2    28    0  33.000         3        0        0
3    33    0  22.705         0        0        3
4    32    0  28.880         0        0        3
...   ...   ...   ...   ...   ...   ...
1333  50    0  30.970         3        0        3
1334  18    1  31.920         0        0        2
1335  18    1  36.850         0        0        0
1336  21    1  25.800         0        0        1
1337  61    1  29.070         0        1        3
[1338 rows x 6 columns]

In [17]: print(Y)

0      16884.92400
1      1725.55230
2      4449.46200
3      21984.47061
4       3866.85520
...
1333      16690.54830
1334      2295.98080
1335      1629.83350
1336      2997.94500
1337      29141.36030
Name: charges, Length: 1338, dtype: float64

In [18]: X_train, X_test, Y_train, Y_test= train_test_split(X, Y, test_size=0.2, random_state=2)

In [19]: print(X.shape, X_train.shape, X_test.shape)

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

In [20]: #loading the Linear Regression model
regressor=LinearRegression()

In [21]: regressor.fit(X_train, Y_train)

Out[21]: LinearRegression()

In [22]: #prediction on training data
training_data_prediction = regressor.predict(X_train)

In [23]: #R squared value
r2_train= metrics.r2_score(Y_train, training_data_prediction)
print('r squared value:', r2_train)
r squared value: 0.751585643411174

In [24]: #prediction on test data
test_data_prediction = regressor.predict(X_test)

In [25]: #R squared value
r2_test= metrics.r2_score(Y_test, test_data_prediction)
print('R squared value:', r2_test)
R squared value: 0.7447273869684077

In [26]: input_data= (70,0,36,74,1,1)
input_data_as_numpy_array=np.asarray(input_data)
input_data_resshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = regressor.predict(input_data_resshaped)
print(prediction)
print('the insurance cost in USD', prediction[0])

[84911.82342251]
the insurance cost in USD 84911.823422506066
C:\Users\parab\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(

In [27]: regressor=LinearRegression()
regressor.fit(X_train, Y_train)

Out[27]: LinearRegression()

In [28]: import joblib

In [29]: joblib.dump(regressor, 'model_joblib_gr')

Out[29]: ['model_joblib_gr']

In [30]: model = joblib.load('model_joblib_gr')

In [32]: model.predict_data()

In [30]: from tkinter import *

In [31]: import joblib

In [ ]: def show_entry():

    p1 = float(e1.get())
    p2 = float(e2.get())
    p3 = float(e3.get())
    p4 = float(e4.get())
    p5 = float(e5.get())
    p6 = float(e6.get())

    model= joblib.load('model_joblib_gr')
    result = model.predict([[p1,p2,p3,p4,p5,p6]])

    Label(master, text="Insurance cost").grid(row=7)
    Label(master, text=result).grid(row=8)

master=Tk()
master.title("Insurance Cost Prediction")
label = Label(master, text = "Insurance Cost Prediction", bg = "black", fg = "white").grid(row=0,columnspan=2)

Label(master, text = "Enter Your Age").grid(row=1)
Label(master, text = "Male or Female [1/0]").grid(row=2)
Label(master, text = "Enter Your BMI value").grid(row=3)
Label(master, text = "Enter Your No of children").grid(row=4)
Label(master, text = "Smoker yes/no [1/0]").grid(row=5)
Label(master, text = "Region [1-4]").grid(row=6)

e1 = Entry(master)
e2 = Entry(master)
e3 = Entry(master)
e4 = Entry(master)
e5 = Entry(master)
e6 = Entry(master)

e1.grid(row=1, column=1)
e2.grid(row=2, column=1)
e3.grid(row=3, column=1)
e4.grid(row=4, column=1)
e5.grid(row=5, column=1)
e6.grid(row=6, column=1)

Button(master, text="predict", command=show_entry).grid()

master.mainloop()

C:\Users\parab\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```