MEDICAL INSURANCE COST PREDICTIOR USING LINEAR REGRESSION

MADE BY: SHUBHAM SHARMA

DATASET LINK: https://www.kaggle.com/datasets/mirichoi0218/insurance

Importing the Libraries

In [3]:
```
import numpy as np # numpy is used to work with arrays.
import pandas as pd # pandas is used to create dataframes in table format.
import matplotlib.pyplot as plt # pyplot is used in plotting the dataset.
import seaborn as sns # seaborn is also used to visualize the data.
from sklearn.model_selection import train_test_split # It is used to split data int
from sklearn.linear_model import LinearRegression # It is used to create and implem
from sklearn import metrics # Metrics is used for evaluating the performance of the
```

Data Collection and Analysis

In [5]:
```
# Loading the data from a csv file to a panda dataframe.
insurance_dataset = pd.read_csv('insurance.csv')
```

In [6]:
```
# First 10 rows of the dataset.
insurance_dataset.head(10)
```

Out[6]:

	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
5	31	female	25.740	0	no	southeast	3756.62160
6	46	female	33.440	1	no	southeast	8240.58960
7	37	female	27.740	3	no	northwest	7281.50560
8	37	male	29.830	2	no	northeast	6406.41070
9	60	female	25.840	0	no	northwest	28923.13692

In [7]:
```
# Identifying number of rows and columns in this dataset.
insurance_dataset.shape
```

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

In [8]:
```
# Getting some information about dataset.
insurance_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
```

Categorical Features:

- Sex

- Smoker

- Region

In [10]:
```
# Checking for any missing values in the dataset.
insurance_dataset.isnull().sum()
```

Out[10]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

Data Analysis

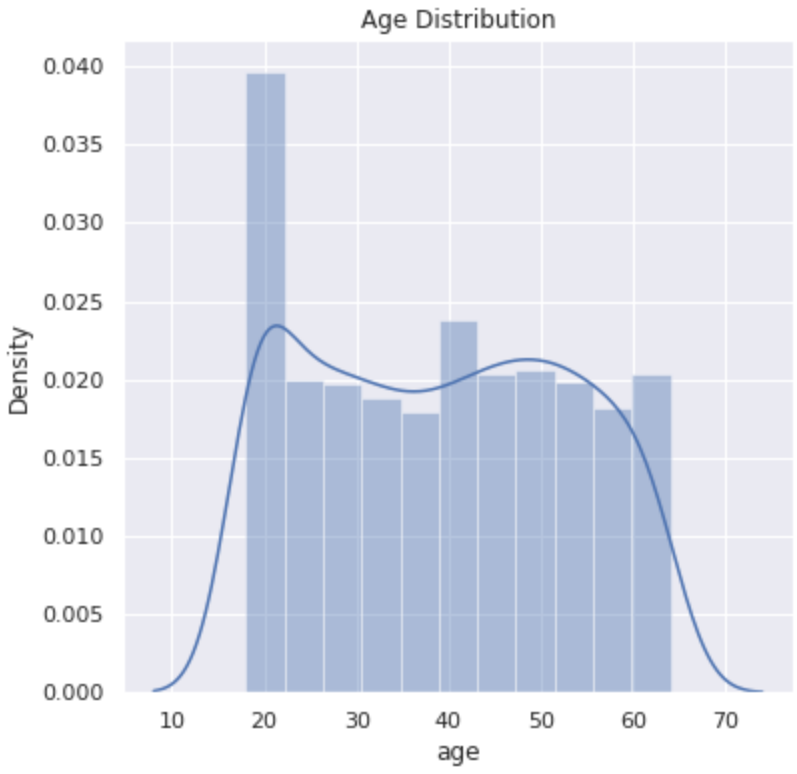
In [12]:
```
# Statistical Measures of the dataset.
insurance_dataset.describe()
```

Out[12]:

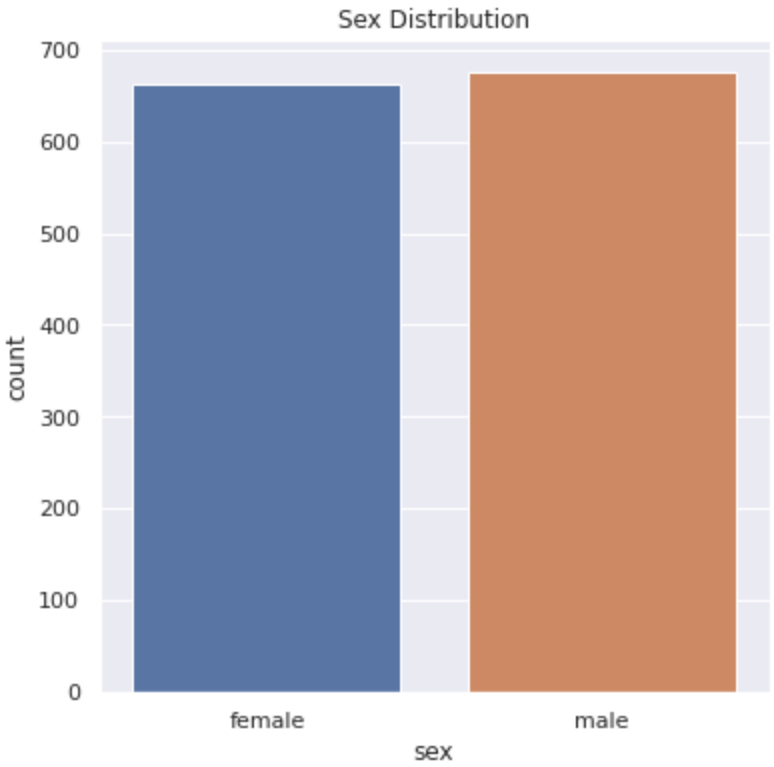
	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

In [13]:
```
# Distribution of Age value.
sns.set()
plt.figure(figsize=(6,6))
sns.distplot(insurance_dataset['age'])
plt.title('Age Distribution')
plt.show()
```

```
/opt/conda/envs/anaconda-2022.05-py39/lib/python3.9/site-packages/seaborn/distributi
ons.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for histogr
ams).
  warnings.warn(msg, FutureWarning)
```



In [14]:
```
# Distribution of Sex column.
plt.figure(figsize=(6,6))
sns.countplot(x='sex', data=insurance_dataset)
plt.title('Sex Distribution')
plt.show()
```

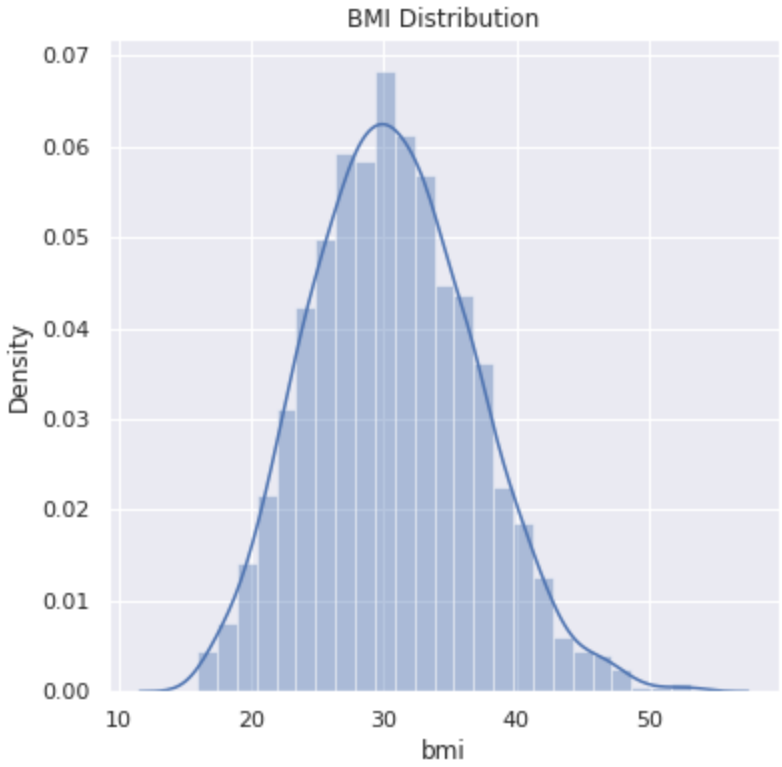


In [15]:
```
# Alternative for the above.
insurance_dataset['sex'].value_counts()
```

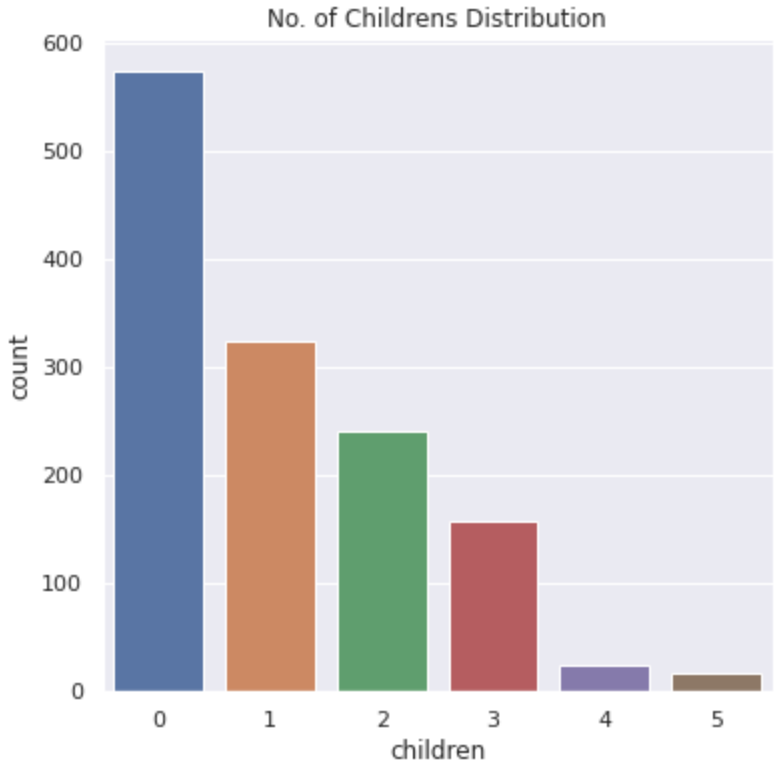
Out[15]:
```
male      676
female    662
Name: sex, dtype: int64
```

In [16]:
```
# Distribution of BMI column.
plt.figure(figsize=(6,6))
sns.distplot(insurance_dataset['bmi'])
plt.title('BMI Distribution')
plt.show()
```

```
/opt/conda/envs/anaconda-2022.05-py39/lib/python3.9/site-packages/seaborn/distributi
ons.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for histogr
ams).
  warnings.warn(msg, FutureWarning)
```



In [17]:
```
# Distribution of Children column.
plt.figure(figsize=(6,6))
sns.countplot(x='children', data=insurance_dataset)
plt.title('No. of Childrens Distribution')
plt.show()
```

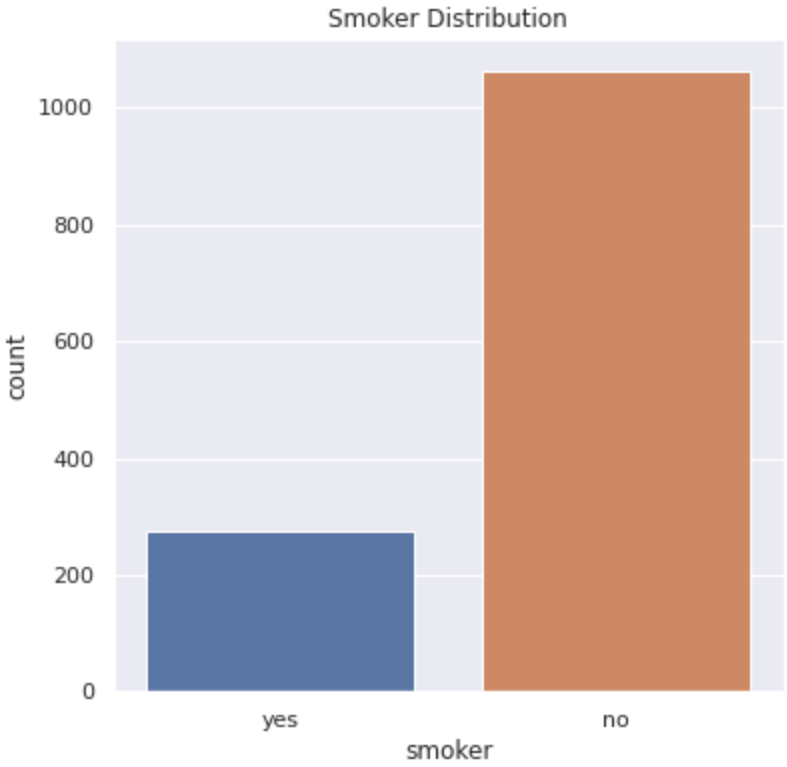


In [18]:
```
# Alternative to the above.
```

```
insurance_dataset['children'].value_counts()
```

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

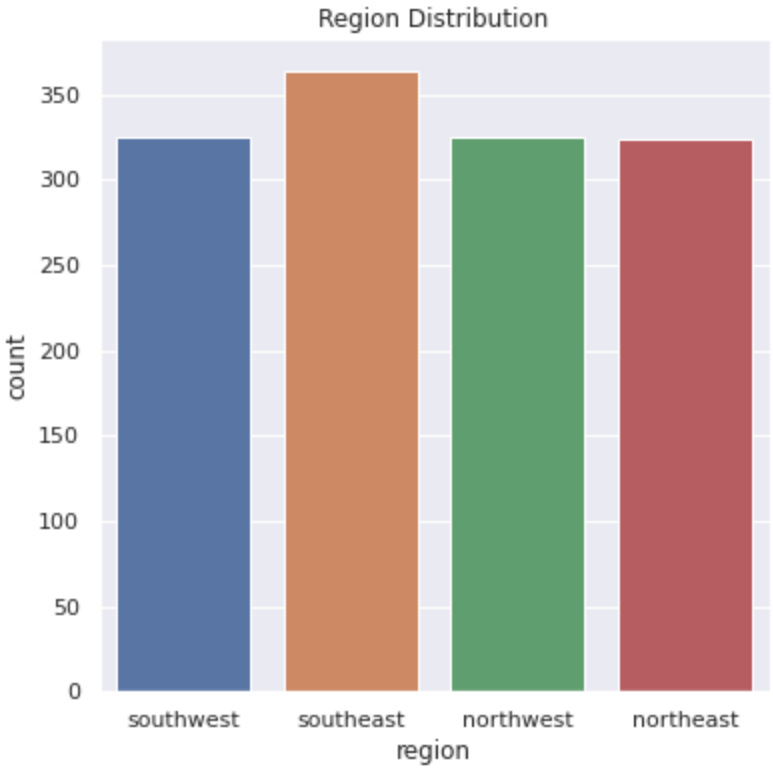
In [19]:
```
# Distribution of Smoker column.
plt.figure(figsize=(6,6))
sns.countplot(x='smoker', data=insurance_dataset)
plt.title('Smoker Distribution')
plt.show()
```



In [20]:
```
insurance_dataset['smoker'].value_counts()
```

Out[20]:
```
no     1064
yes     274
Name: smoker, dtype: int64
```

In [21]:
```
# Distribution of Region column.
plt.figure(figsize=(6,6))
sns.countplot(x='region', data=insurance_dataset)
plt.title('Region Distribution')
plt.show()
```

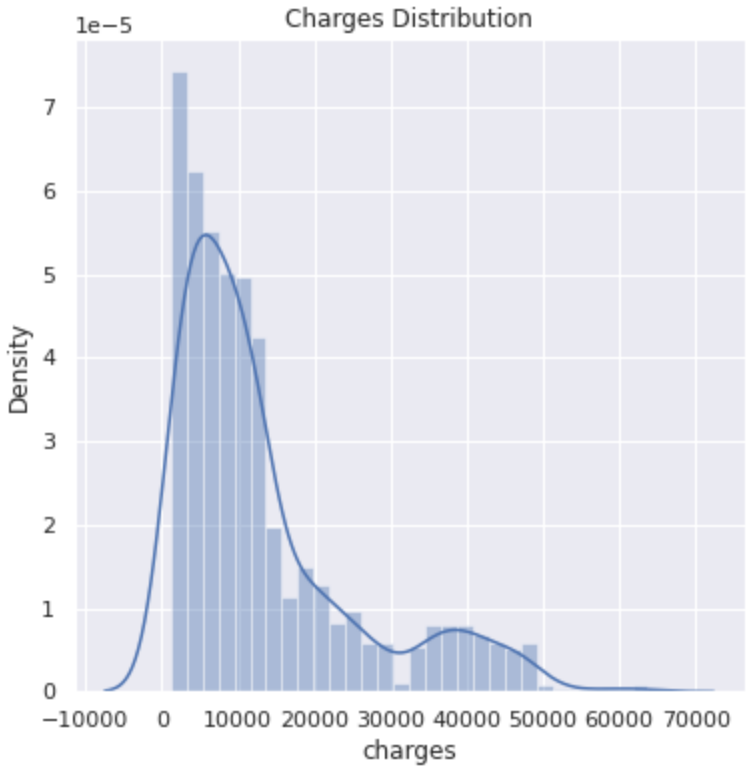


In [22]:
```
insurance_dataset['region'].value_counts()
```

Out[22]:
```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

In [23]:
```
# Distribution of Charges column.
plt.figure(figsize=(6,6))
sns.distplot(insurance_dataset['charges'])
plt.title('Charges Distribution')
plt.show()
```

```
/opt/conda/envs/anaconda-2022.05-py39/lib/python3.9/site-packages/seaborn/distributi
ons.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for histogr
ams).
  warnings.warn(msg, FutureWarning)
```



Data PreProcessing

Encoding the Categorical Features

In [26]:
```
# Encoding sex column.
insurance_dataset.replace({'sex':{'male':0,'female':1}}, inplace=True)
```

In [27]:
```
# Encoding smoker column.
insurance_dataset.replace({'smoker':{'yes':0,'no':1}}, inplace=True)
```

In [28]:
```
# Encoding region column.
insurance_dataset.replace({'region':{'southeast':0,'southwest':1,'northeast':2,'nor
```

Splitting the Features and Target variables.

In [30]:
```
x= insurance_dataset.drop(columns='charges',axis=1)
y=insurance_dataset['charges']
```

In [31]:
```
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]
```

In [32]:
```
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
```

Splitting the dataset into Training data and Testing data.

In [34]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_stat
```

In [35]:
```
print(x.shape, x_train.shape, x_test.shape)
```

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

In [36]:
```
print(y.shape, y_train.shape, y_test.shape)
```

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

Model Training

Linear Regression

In [39]:
```
# Loading the Linear Regression Model.
regressor = LinearRegression()
```

In [40]:
```
regressor.fit(x_train, y_train)
```

Out[40]:
```
LinearRegression()
```

Model Evaluation

In [42]:
```
# Prediction on the Training data.
training_data_prediction = regressor.predict(x_train)
```

In [43]:
```
# R squared value
r2_train = metrics.r2_score(y_train, training_data_prediction)
```

In [44]:
```
print('R Squared value:',r2_train)
```

```
R Squared value: 0.751505643411174
```

In [45]:
```
# Prediction on the Testing data.
testing_data_prediction = regressor.predict(x_test)

# R squared value
r2_test = metrics.r2_score(y_test, testing_data_prediction)

print('R Squared Value:', r2_test)
```

```
R Squared Value: 0.7447273869684077
```

Building a Predictive System

In [47]:
```
input_data = (31,1,25.74,0,1,0)

# Changing input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# Reshape the array
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = regressor.predict(input_data_reshaped)

print('The Insurance Cost is USD:', prediction[0]) # Actual value is USD 3756.6216
```

```
The Insurance Cost is USD: 3760.0805764960496
/opt/conda/envs/anaconda-2022.05-py39/lib/python3.9/site-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

We can clearly observe the Predicted value is very much close to the actual value. Thus we can say that our Predictor Model is Ready to Use.

THANK YOU