About Dataset :

This paper represents a machine learning-based health insurance prediction system. Recently, many attempts have been made to solve this problem, as after Covid-19 pandemic, health insurance has become one of the most prominent areas of research. We have used the USA's medical cost personal dataset from kaggle, having 1338 entries. Features in the dataset that are used for the prediction of insurance cost include: Age, Gender, BMI, Smoking Habit, number of children etc. We used linear regression and also determined the relation between price and these features. We trained the system using a 70-30 split and achieved an accuracy of 81.3%

Attribute information

- AGE : Age of the person
- SEX : Male|Female
- BMI : Body Mass Index
- CHILDREN : Number of Children
- SMOKER : Yes|No
- REGION : Their Region

Importing the Dependencies

```
import pandas as pd
import numpy as np
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
```

Data Collection & Analysis

```
# Loading of data & Analysis
insurance_dataset=pd.read_csv('/content/archive.zip')
insurance_dataset
```

```
      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
```

```
#number of rows and columns
insurance_dataset.shape
```

```
(1338, 7)
```

```
#information of 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
```

```
# checking missing values
insurance_dataset.isna().sum()
```

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

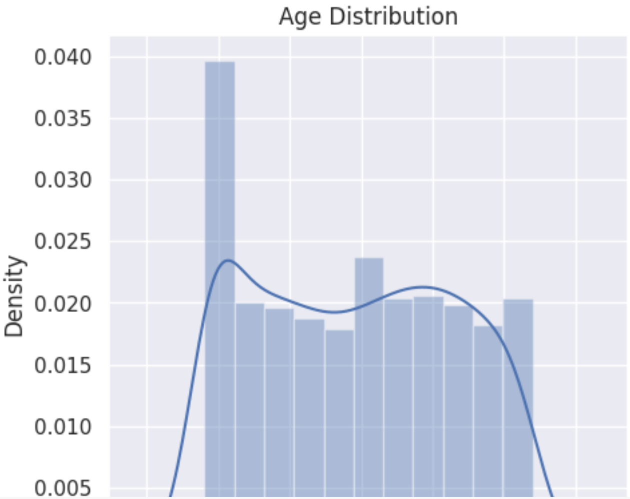
Data Analysis

```
insurance_dataset.describe()
```

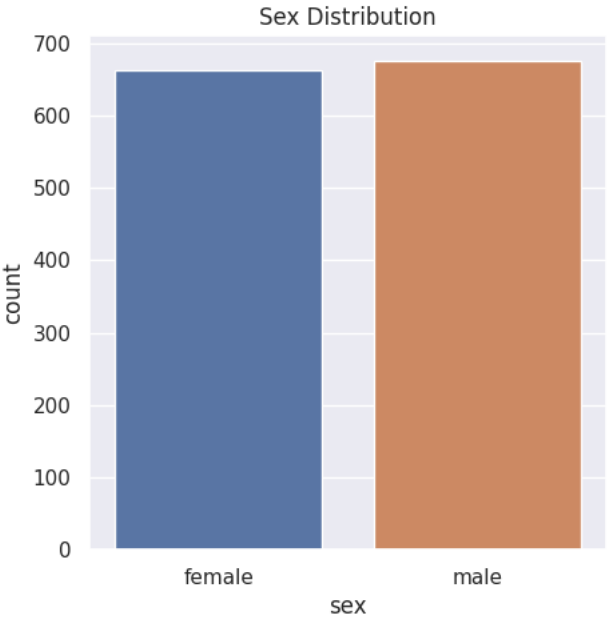
	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000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

```
# distribution of age value
sns.set()
plt.figure(figsize=(5,5))
sns.distplot(insurance_dataset['age'])
plt.title('Age Distribution')
plt.show()
```

```
<ipython-input-233-fd204a27f3e1>:4: 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/de44147ed2974457ad6372750bbe5751

  sns.distplot(insurance_dataset['age'])
```



```
# Gender column
plt.figure(figsize=(5,5))
sns.countplot(x='sex',data=insurance_dataset)
plt.title('Sex Distribution')
plt.show()
```



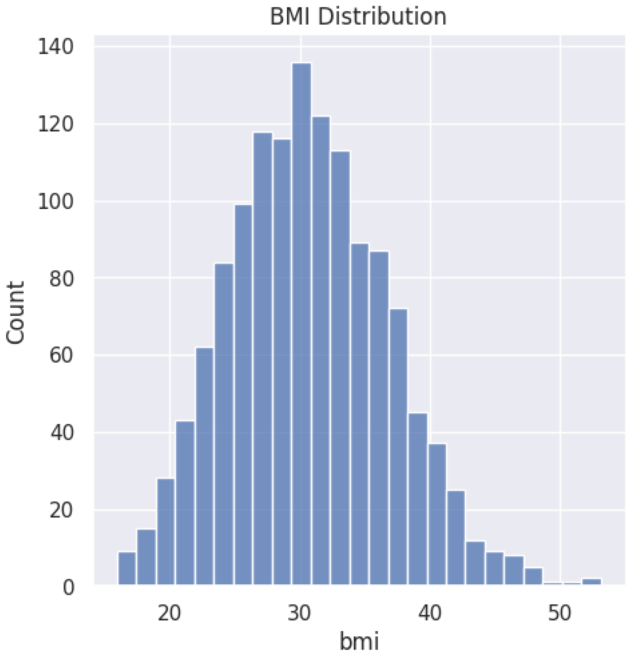
```
insurance_dataset['sex'].value_counts()
```

```
     male      676
     female    662
     Name: sex, dtype: int64
```

```
#bmi distribution
plt.figure(figsize=(5,5))
sns.displot(insurance_dataset['bmi'])
```

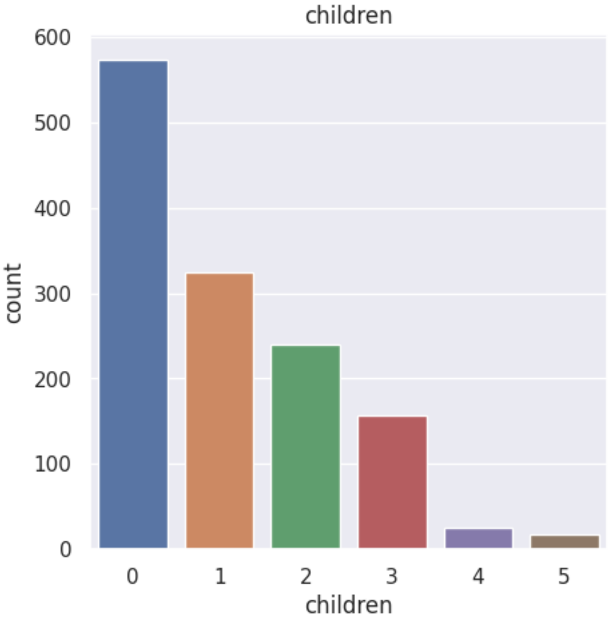
```
plt.title('BMI Distribution')
plt.show()
```

```
<Figure size 500x500 with 0 Axes>
```



```
#childrens column
plt.figure(figsize=(5,5))
sns.countplot(x='children',data=insurance_dataset)
plt.title('children')
```

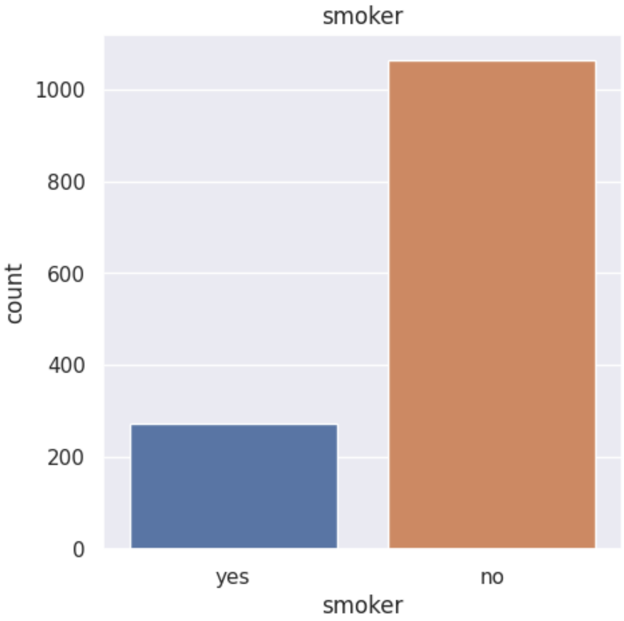
```
Text(0.5, 1.0, 'children')
```



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

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

```
# smoker column
plt.figure(figsize=(5,5))
sns.countplot(x='smoker',data=insurance_dataset)
plt.title('smoker')
plt.show()
```

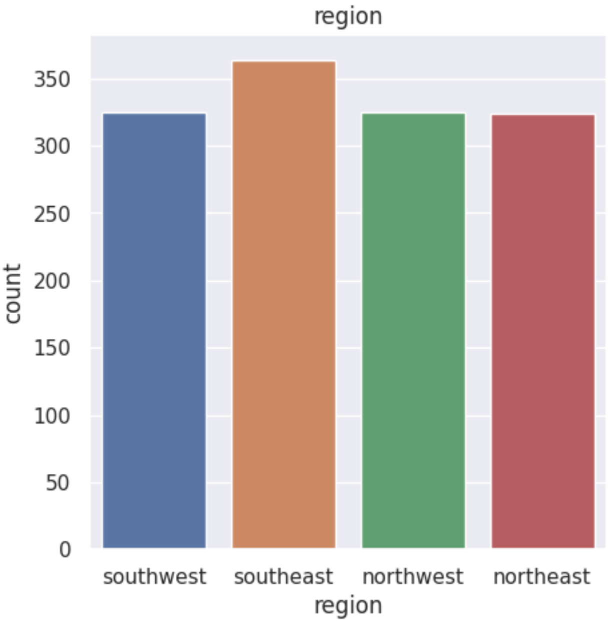


```
insurance_dataset['smoker'].value_counts()
```

```
no     1064
yes     274
```

```
Name: smoker, dtype: int64
```

```
#region column
plt.figure(figsize=(5,5))
sns.countplot(x='region',data=insurance_dataset)
plt.title('region')
plt.show()
```

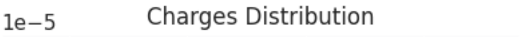


```
insurance_dataset['region'].value_counts()
```

```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

```
#Distribution of chrages value
plt.figure(figsize=(5,5))
sns.distplot(insurance_dataset['charges'])
plt.title('Charges Distribution')
plt.show()
```

```
<ipython-input-243-c3b65175316b>: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/de44147ed2974457ad6372750bbe5751

  sns.distplot(insurance_dataset['charges'])
```



Data Pre-Procrssing

```
# encoding the categorical features




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


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

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

spliting the features and target

```
x=insurance_dataset.iloc[:,:-1]
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 × 6 columns

```
y=insurance_dataset.iloc[:,-1]
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
```

spliting data

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=2)
x_test.shape
```

```
(402, 6)
```

```
x.shape
```

```
(1338, 6)
```

```
x_train.shape
```

```
(936, 6)
```

Model Training

```
# Linear regression model
regressor=LinearRegression()
```

```
regressor.fit(x_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
# model evaluation

#training data

training_data_prediction=regressor.predict(x_train)
```

```
r2_train=metrics.r2_score(y_train,training_data_prediction)
print(' R  squared value :',r2_train)
```

```
 R  squared value : 0.7415730843556845
```

```
# test data
test_data_prediction=regressor.predict(x_test)
```

```
r2_test=metrics.r2_score(y_test,test_data_prediction)
print( 'R square value :',r2_test)
```

```
R square value : 0.7661186068101191
```

Predictive system building

```
data=(39.61,9,27,9,1,2)
# changing data into a numpy array
data_as_numpy_array=np.asarray(data)

#array reshaping
data_reshaped=data_as_numpy_array.reshape(1,-1)
```

```
prediction=regressor.predict(data_reshaped)
print(prediction)

print('the insurance cost is USD',prediction[0])
```

```
[13196.38495713]
the insurance cost is USD 13196.384957126693
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegression
  warnings.warn(
```

✓ 0s completed at 1:50 PM