Medical Cost Personal Insurance

Problem Statement :

Health insurance is a type of insurance that covers medical expenses that arise due to an illness. These expenses could be related to hospitalisation costs, cost of medicines or doctor consultation fees. The main purpose of medical insurance is to receive the best medical care without any strain on your finances. Health insurance plans offer protection against high medical costs. It covers hospitalization expenses, day care procedures, domiciliary expenses, and ambulance charges, besides many others. Based on certain input features such as age , bmi , no of dependents ,smoker ,region medical insurance is calculated .

Insurance Forecast by using Linear Regression :-

Attribute Information :

- Age : Age of primary beneficiary
- Sex : Insurance contractor gender, female, male
- Bmi : Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9.
- Children : Number of children covered by health insurance / Number of dependents
- Smoker : Smoking
- Region : The beneficiary's residential area in the US, northeast, southeast, southwest, northwest.
- Charges : Individual medical costs billed by health insurance.

Problem Definition –

Health Insurence is a type of insurance which can cover all the medical expenses . could be related to the hospitalization cost .Health Insurance cover Hospitalisation cost , cost of medicines and doctor consultant fees . and Health insurance gives offer on Protection against Medical cost. It covers Hospitalization expenses , Day care procedures , domeciliary expenses . and ambulance charges Etcc...The main purpose of health insurance is to receive best medical care without any financial strain.

In the problem we have get a one dataset and we have to predict the Total charges . behalf of Age , BMI , Childrens , Smoker and Region .

So , Using of Age , BMI , Children , Smoker and Region We have to find the total charges .of any patient . We have to use Python and Supervised Machine Learning Algorithms .

As we can see, we got these features:

1. **`age`: age of the primary beneficiary**
2. **`sex`: insurance contractor gender, female, male**
3. **`bmi`: Body Mass Index, providing an understanding of body weights that are relatively high or low relative to height, objective index of body weight (kg/m^2) using the ratio of height to weight, ideally 18.5 to 24.9**
4. **`children`: number of children covered by health insurance, number of dependents**
5. **`smoker`: smoking or not**
6. **`region`: the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.**
7. **`charges`: individual medical costs billed by health insurance**

Since we are predicting insurance costs, `charges` will be our target feature.

Libraries Used :-

- **Pandas**
- **Numpy**
- **Seaborn**
- **Matplotlib**
- **Scipy**

Medical Cost Personal Insurance Dataset is Supervised Regression Problem . Type of Machine Learning Problem.

- The DataFrame Contain 1338 Rows and 7 Columns.
- The Target Variable is Charges Which is Continious Data Type.
- The Dataset Contain No Missing Values.

Exploring and Data Cleaning :

Data Frame :-

- Duplicates Values – 1
- Missing Values – 0

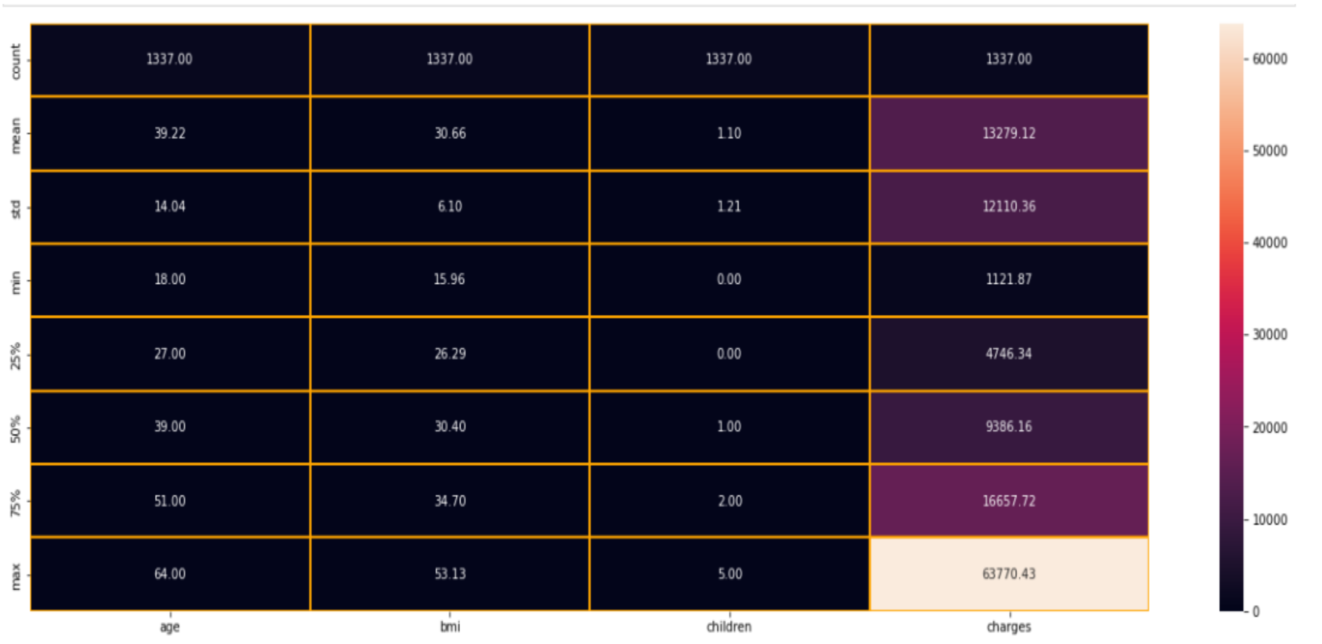
When we Exploring a dataset and its values , There were No Null Values . and 1 Duplicates Values. Found in the Dataset

Further on Exploring the We get Info about Datatype of all the Columns .

- Integer Datatype -> Age and Childer
- Categorical Datatype -> Sex , Smoker Region
- Float Type -> BMI and Charges..

Data Description : -

- Data Description return description of the numerical data . Like Mean , Standard Deviation , Quarter Percentile, Minimum and Maximum Values .



Observation of Data Description : -

- Here we have Outliers in BMI .

Exploratory Data Analysis :-

It is important for us to have insights about data . It help us to find the find information which are hidden in the data .By Presenting the Data Feature on Graph we can observe and Draw Certain Conclusion . For the Graph Sketching we Use Libraries “Seaborn” and “Matplotlib”.

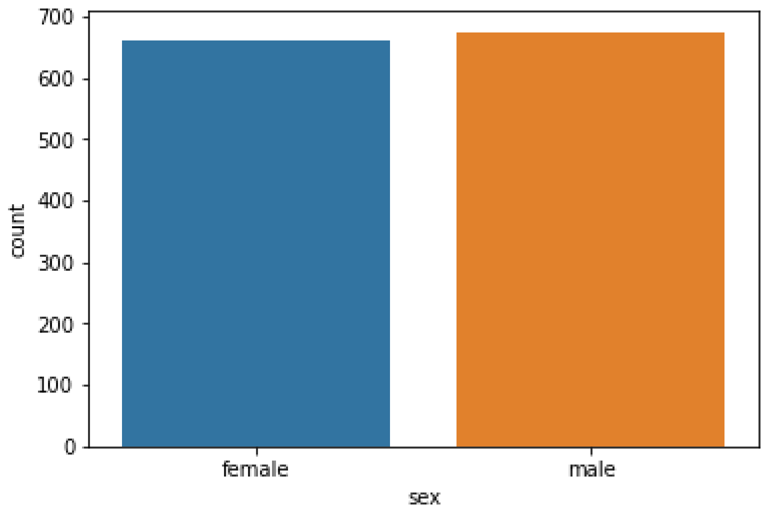
Visualization :

The DataFrame Include both Categorical and Numerical Data:

Univariate Analysis :

- Univariate analysis is a basic kind of analysis technique for statistical data. Here the data contains just one variable and does not have to deal with the relationship of a cause and effect.

- **Lets Check For Sex ..**



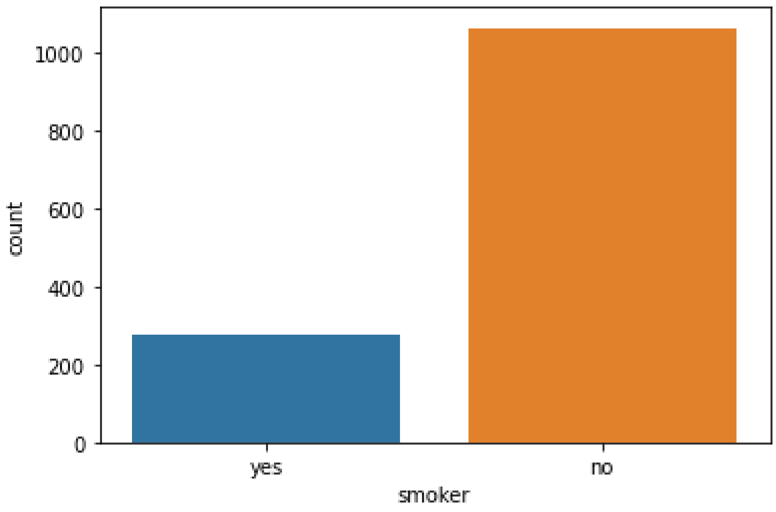
Here we have .

Male -> 675

Female -> 662

So, Here we can see that Male and Female are almost equal but Buying Insurance .

Lets Check For Smoker :



Smoker -> 274

Non Smoker -> 1063 .

So , Here can see that Maximum People are Not Smoker..

Lets Check For Region :-

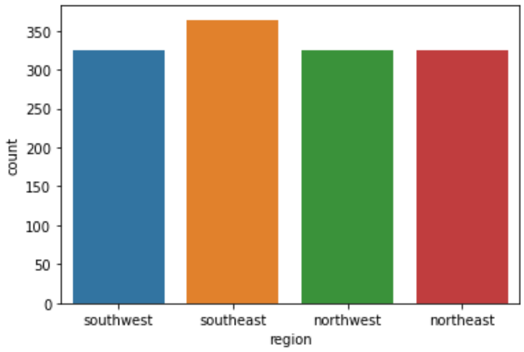
So here we can see that

Beneficiary's residential area in the US from SouthEast is - 364

Beneficiary's residential area in the US from SouthWest is - 325

Beneficiary's residential area in the US from NorthEast is - 324

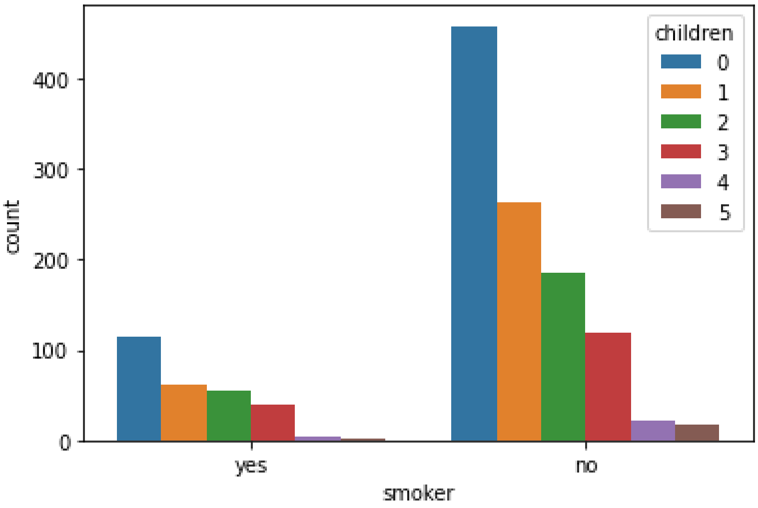
beneficiary's residential area in the US from NorthWest is - 324



Bivariate Analysis :

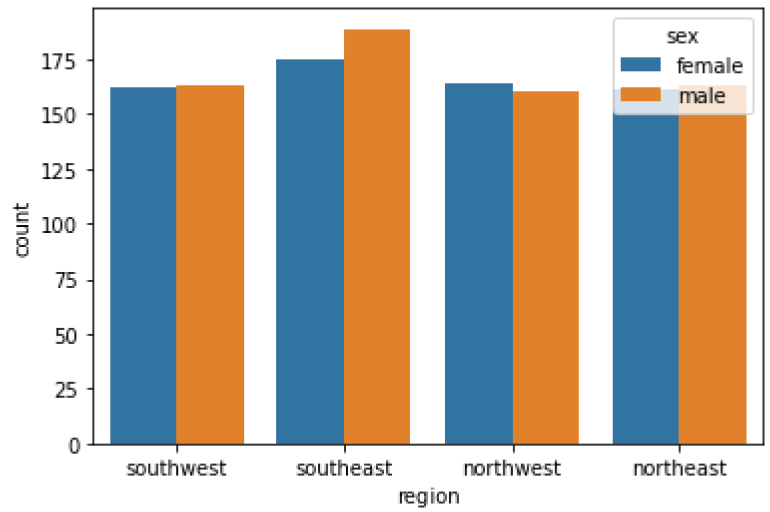
- Bivariate data help you in studying two variables .

- **Lets check how much are smoking who have dependents**



- So who have 0 Dependents - More then 100 peoples are smoking
- So Who have 1 Dependents - More then 60 People are somking
- So Who have 2 Dependents - More then 50 People are smoking
- So Who have 3 Dependents - Almost 50 people are somking
- So Who have 4 Dependents - Almost 4 - 6 People are Smoking
- So who have 5 Dependents - Almost 1-2 People are Smoking

Lets check how much male and how much female from all 4 location



-

So here we can see that from ,

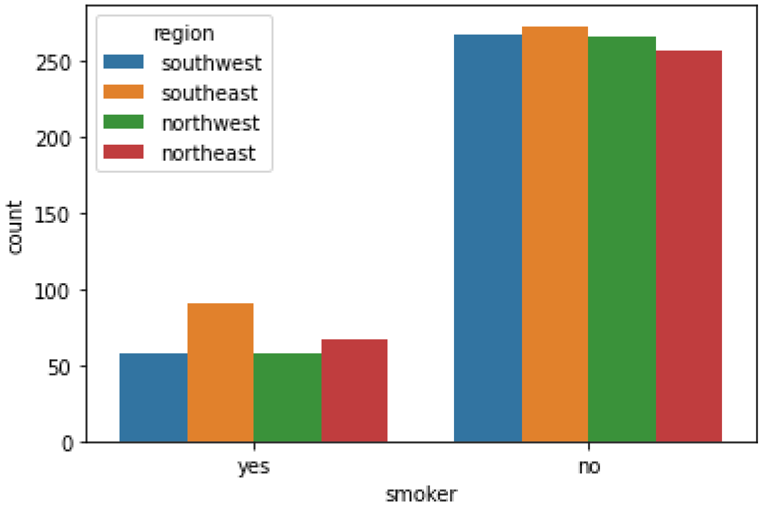
SouthWest - Female is 162 and Male is 165

SouthEast - Female is 175 and Male is almost 185

NorthWest -Female is almost 168 and Male is almost 160

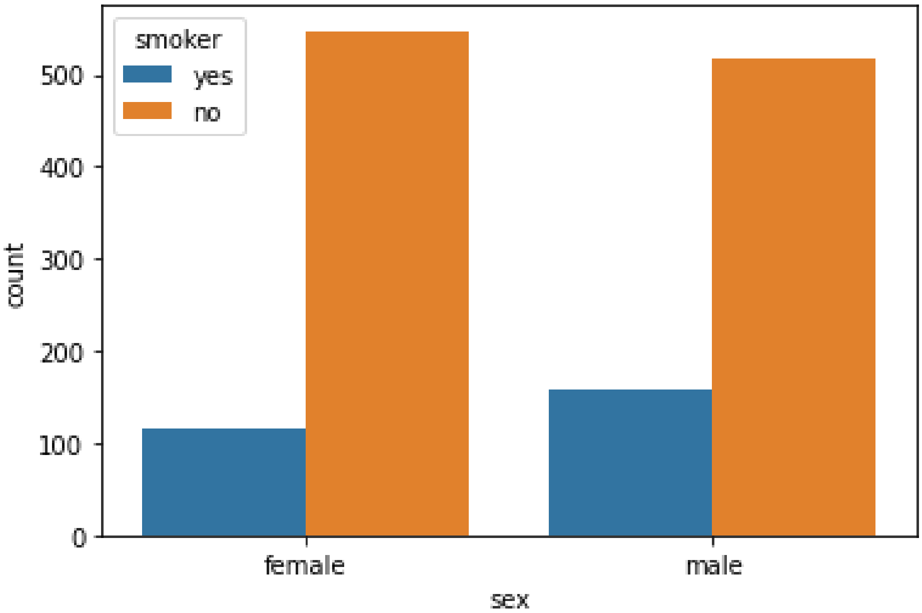
North east - Female is almost almost 160 and Male is almost 168

Lets check how much smoker from which part of USA



- From Southwest from USA only almost 60 person are smoker
- From Southwest from USA more then 250 person are Non smoker
- From SouthEast from USA only almost 90 person are smoker
- From SouthEast from USA more then 250 person are Non smoker
- From NorthWest from USA only almost 60 person are smoker
- From NorthWest from USA more then 250 person are Non smoker
- From NorthEast from USA more then almost 70 person are smoker
- From NorthEast from USA more then 250 person are Non smoker
- So, here we observe that number of people are smoker in 4 region in USA

Lets check how much male and female are smoker

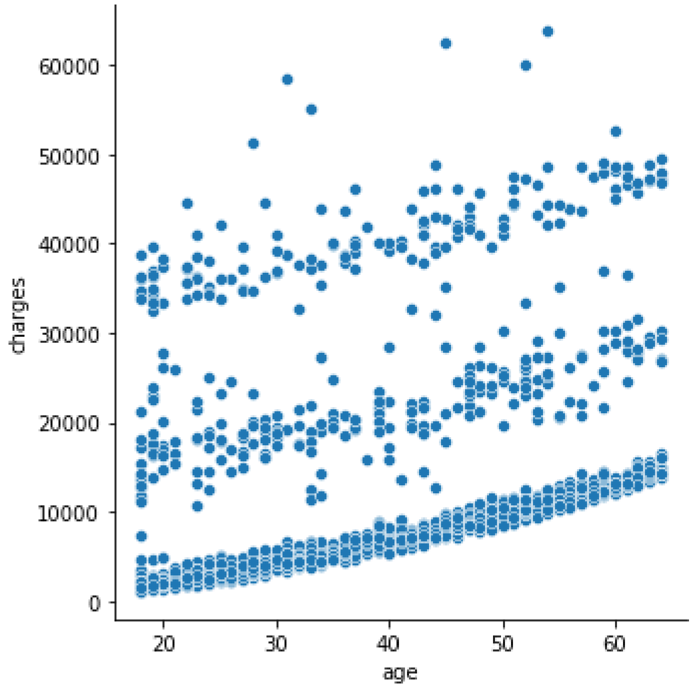


So ,here we can observe that

Female - more 100 peoples are smoker

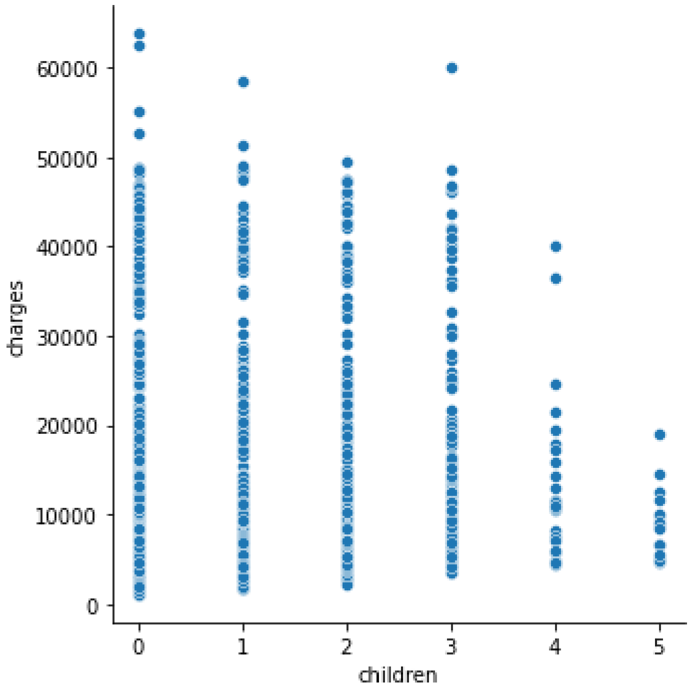
Male - More then peoples are smoker

Lets plot Age and Charges in Scatter Plot



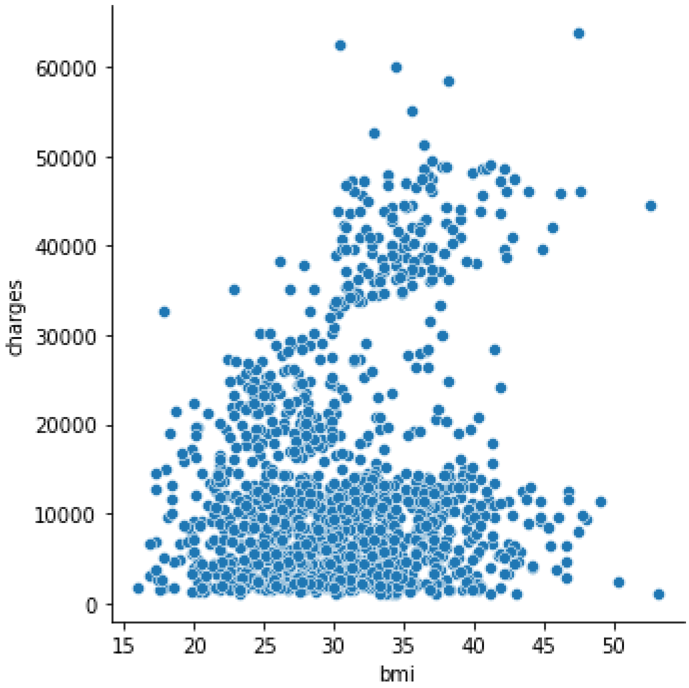
So , Here we can see the . there are some positive relationship.

Lets plot children and Charges in Scatter Plot



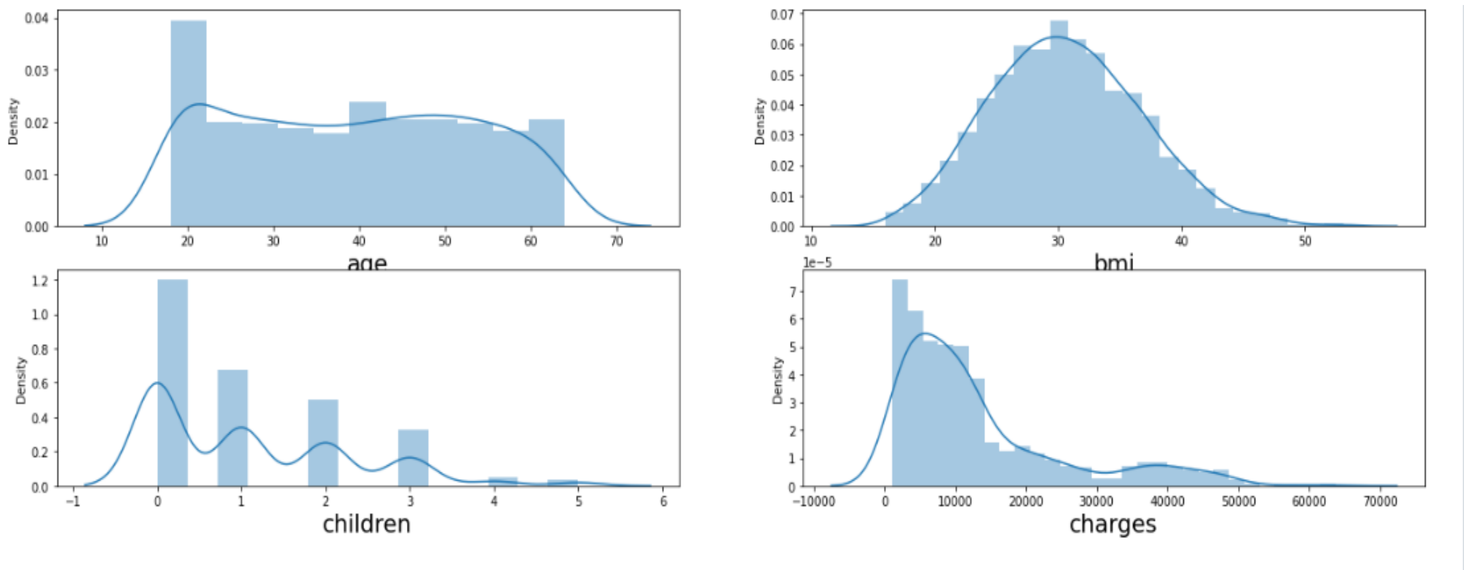
As the number of children increase the charges decrease ,We can clearly see that there is no relationship. in children and charges columns.

Lets plot BMI and Charges in Scatter Plot



We can see that . BMI and Charges has no relationship..

Lets check the distribution of the columns



Observations :

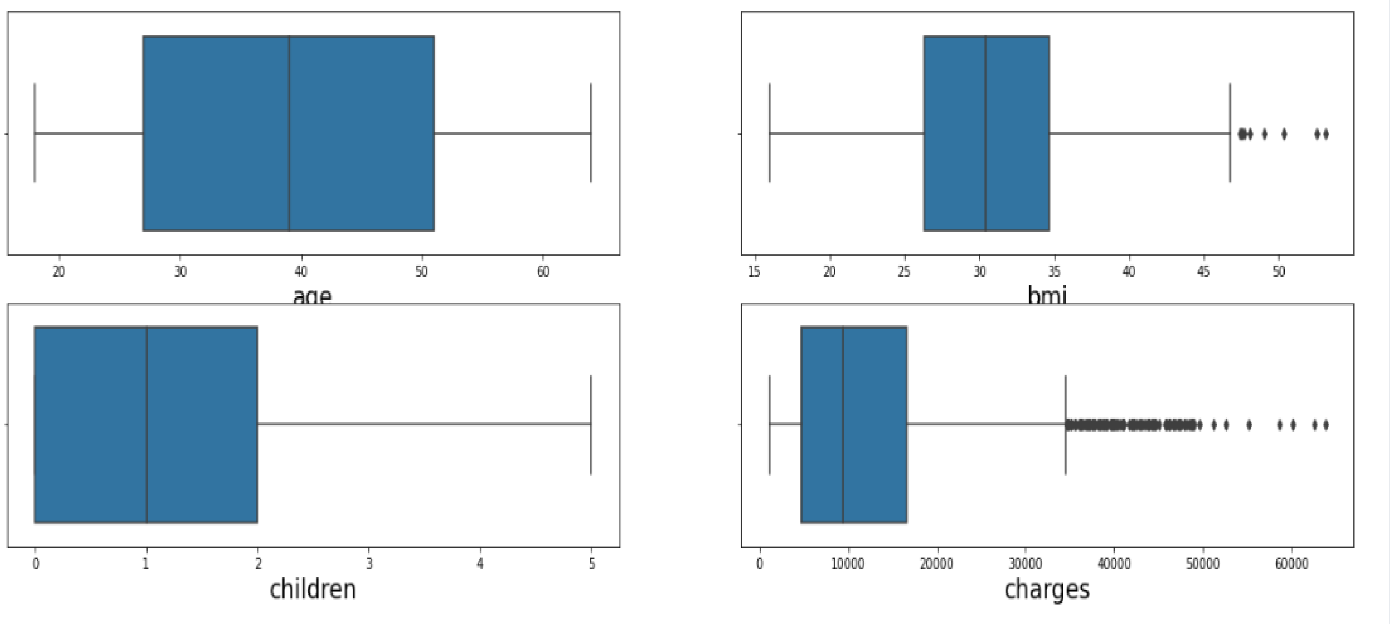
- Age -> We Can See that Mininmum age is 10 and Maximum age is 70. So we can consider is Normal Distribution . We can see in Daily Life there is people who live for More then 70 Years.

- BMI -> Normal BMI Range for Men and Women is 18.5 to 24.9 . and in Distrubition We can see that Mininmum is 10 and Maximum is 50 . Its Meaning is Data is Normally Distributed But There is Some Outliers .

- Childrens -> Normal As per Govt Rule 2 Child is Allowed . But in Normal Life There is Reality More then 5 Childrens are there . So we Consider 6 Children is possible . So we Consider Children Column is Noramally Distributed .

- Charges -> So , The Charges Colums is our Target Column . So we Don’t need to Change anything ..

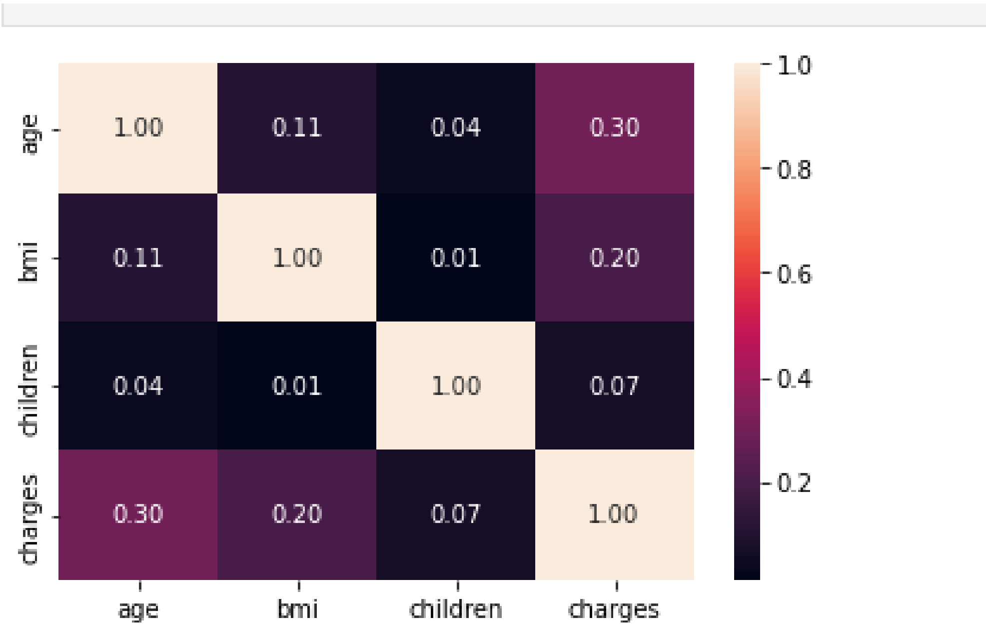
Lets Plot BOX plot and Check Outliers by visualization



Observation :

- So , As we seen in the above Image We can see that BMI and Charges are Some Outliers .But we have to treat only BMI outliers Because Charges Columns are Target So we don’t need to Treat ..

Check correlatiom before losing any data because we have small dataset .



Observations:

- 1 - BMI have almost 20% relationship with charges
- 2- Age have almost 30% relationship with charges
- 2 -Childres have almost 7% relationship with charges

Lets check correlation with target columns

In [35]:
```
#Lets check correlation with target columns
data.corr()['charges']
```

Out[35]:
```
age         0.298308
bmi         0.198401
children    0.067389
charges     1.000000
Name: charges, dtype: float64
```

Observation : -

- Age -> 29 % Relation with Charges
- BMI -> 19 % Relation With Charges
- Children ->06% Realtion With Charges

Data Preprocessing:

In Preprocessing Step we First Select Only Categorical Variable to Encode.

In [65]:
```
#Lets use encoding technique and convert all categorical data to numerical data
#First filter categorical column
numeric=['int8','int16','int32','int64','float','float32','float64']
categorical_column=[]
feature=data.columns.values.tolist()

for col in feature:
    if data[col].dtype in numeric:
        continue
    categorical_column.append(col)
categorical_column
```

Out[65]:
```
['sex', 'smoker', 'region']
```

Then we use pd.get_dummies method to Encode all the categorical variable in Numerical ..

In [66]:
```
df_dummies=pd.get_dummies(data[categorical_column],drop_first=True)
df_dummies.head()
```

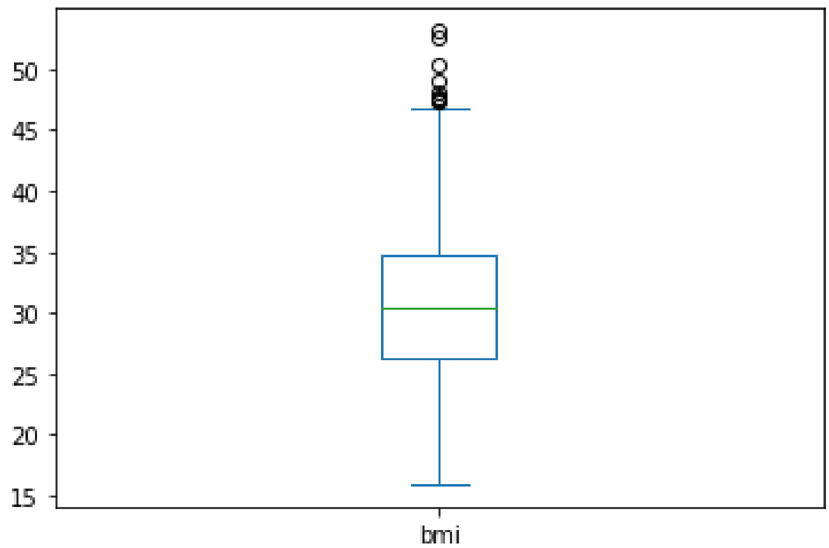
Out[66]:

	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest
0	0	1	0	0	1
1	1	0	0	1	0
2	1	0	0	1	0
3	1	0	1	0	0
4	1	0	1	0	0

So , Here we have to Encode Sex , Smoker and Region. using pd.get_dummies.

Lets Handle the Outlies By using Zscore.

So , In the About Box Plot we can see that BMI contains some outliers .



Lets Remove Outliers By Using Z Score .

After Removing Outliers By using Z score We have The Shape of the dataset is ,

(1333, 9)

After Remoing How Much Data Loss in Percentage : -

```
[83]: data_loss=((1337-1333)/1337)*100
      data_loss
```

```
[83]: 0.2991772625280479
```

So , here we have loss 0.29 % Percent of Our Data .

```
In [148]: from sklearn.preprocessing import PowerTransformer
```

```
In [154]: scaler = StandardScaler()
          x = pd.DataFrame(scaler.fit_transform(x), columns=x.columns)
          x
```

Out[154]:

	age	bmi	children	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest
0	-1.443917	-0.450191	-0.909922	-1.006775	1.970478	-0.566666	-0.608268	1.761119
1	-1.515225	0.527991	-0.080854	0.993271	-0.507491	-0.566666	1.644013	-0.567821
2	-0.802147	0.399678	1.577282	0.993271	-0.507491	-0.566666	1.644013	-0.567821
3	-0.445607	-1.315891	-0.909922	0.993271	-0.507491	1.764709	-0.608268	-0.567821
4	-0.516915	-0.286883	-0.909922	0.993271	-0.507491	1.764709	-0.608268	-0.567821
...	...	...	...	...	...	...	...	...
1328	0.766626	0.061396	1.577282	0.993271	-0.507491	1.764709	-0.608268	-0.567821
1329	-1.515225	0.219705	-0.909922	-1.006775	-0.507491	-0.566666	-0.608268	-0.567821
1330	-1.515225	1.041245	-0.909922	-1.006775	-0.507491	-0.566666	1.644013	-0.567821
1331	-1.301302	-0.800137	-0.909922	-1.006775	-0.507491	-0.566666	-0.608268	1.761119
1332	1.551013	-0.255221	-0.909922	-1.006775	1.970478	1.764709	-0.608268	-0.567821

1333 rows × 8 columns

Dividing data in feature and vectors

```
In [152]: x=data.drop(columns='charges')#Feature
          y=data.charges#Target
```

Check For Skewness : We Find Skewness in the data and we removed skewness using power transformer method.

In [168]:
```
scaler = PowerTransformer()
x = pd.DataFrame(scaler.fit_transform(x), columns=x.columns)
x
```

Out[168]:

	age	bmi	children	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest
0	-1.461680	-0.419157	-1.049423	-1.006775	1.970478	-0.566666	-0.608268	1.761119
1	-1.535995	0.558661	0.209111	0.993271	-0.507491	-0.566666	1.644013	-0.567821
2	-0.797626	0.435826	1.424263	0.993271	-0.507491	-0.566666	1.644013	-0.567821
3	-0.433115	-1.356744	-1.049423	0.993271	-0.507491	1.764709	-0.608268	-0.567821
4	-0.505722	-0.248927	-1.049423	0.993271	-0.507491	1.764709	-0.608268	-0.567821
...	...	...	...	...	...	...	...	...
1328	0.772770	0.104887	1.424263	0.993271	-0.507491	1.764709	-0.608268	-0.567821
1329	-1.535995	0.261159	-1.049423	-1.006775	-0.507491	-0.566666	-0.608268	-0.567821
1330	-1.535995	1.038405	-1.049423	-1.006775	-0.507491	-0.566666	1.644013	-0.567821
1331	-1.313346	-0.791655	-1.049423	-1.006775	-0.507491	-0.566666	-0.608268	1.761119
1332	1.530808	-0.216220	-1.049423	-1.006775	1.970478	1.764709	-0.608268	-0.567821

1333 rows × 8 columns

Lets Standardize the feature data

In [94]:
```
#Lets import standardscaler
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
x_scaled=scaler.fit_transform(x)
x_scaled
```

Out[94]:
```
array([[-1.44391729, -0.45019112, -0.9099223 , ..., -0.5666657 ,
        -0.6082678 ,  1.76111853],
       [-1.51522515,  0.52799105, -0.08085434, ..., -0.5666657 ,
         1.64401271, -0.56782095],
       [-0.80214655,  0.39967754,  1.57728158, ..., -0.5666657 ,
         1.64401271, -0.56782095],
       ...,
       [-1.51522515,  1.04124506, -0.9099223 , ..., -0.5666657 ,
         1.64401271, -0.56782095],
       [-1.30130157, -0.80013704, -0.9099223 , ..., -0.5666657 ,
        -0.6082678 ,  1.76111853],
       [ 1.55101281, -0.25522125, -0.9099223 , ...,  1.76470891,
        -0.6082678 , -0.56782095]])
```

Check For Multicollinearity Problem

Using VIF -> Variance Infaltion Factor.

And we set threshold 10. And all the columns comes in this range . and we decide to move with all the columns without removing any columns.

Lets Build a Model :-

Here we use Machine Learning Algorithms :

1 -> Random Forest :

Cross Validation -> 81 %

:

	MAE	MSE	RMSE	R2-score
Random Forest	2363.828	1.564928e+07	3955.917519	0.897

2 -> Gradient Boosting Regressor :

Cross Validation -> 84%

	MAE	MSE	RMSE	R2-score
Gradient Boost Regressor	2221.579	12617018.64	3552.04429	0.917

3 -> KNeighbors Regressor :

Cross Validation : 84%

	MAE	MSE	RMSE	R2-score
KNN Regressor	3040.9	2.641417e+07	5139.471912	0.841

4 -> XGBRegressor :

Cross Validation : 80%

	MAE	MSE	RMSE	R2-score
XG Boost Regressor	2568.913	1.748359e+07	4181.338259	0.884

Hyperparameter Tuning :

Here we select Gradient Boost Classifier For Tune The parameter

Here For tuning the parameter we use : **'learning_rate': 0.1, 'max_depth': 4, 'min_samples_split': 4, 'n_estimators': 100}**

And get the accuracy same as we get . 91%

In [146]:
```
gbr = XGBRegressor(learning_rate= 0.1, max_depth= 4, min_samples_leaf= 5, n_estimators= 100,min_samples_split=4)

gbr.fit(X_train,y_train)

pred = gbr.predict(X_test)

r2_score(y_test,pred)
```

```
[15:08:09] WARNING: C:/buildkite-agent/builds/buildkite-windows-cpu-autoscaling-group-i-030221e36e1a46bfb-1/xgboost/xgboost-ci-windows/src/learner.cc:767:
Parameters: { "min_samples_leaf", "min_samples_split" } are not used.
```

Out[146]: 0.9179376704479705

Coclusion :

Conclusion

In [263…
```
loaded_model=pickle.load(open('Insurence Foresast','rb'))
result=loaded_model.score(x_test,y_test)
print(result*100)
```

88.2592861936535

In [266…
```
conclusion=pd.DataFrame([loaded_model.predict(x_train)[:],pred_decision[:]],index=['predicted','original'])
conclusion
```

Out[266…

	0	1	2	3	4	5	6	7	8	9	...
predicted	2185.880579	10397.078317	7181.483398	7671.268939	8418.146373	2690.687687	11834.572644	14441.296297	27152.927069	18995.934033	
original	17053.774289	11246.623143	7214.699786	7580.239601	8717.858457	13138.937292	12515.856756	3771.329014	9742.125832	36252.301797	

2 rows × 1066 columns

For a simple model like Linear Regression, feature engineering plays an important role to improve the model. In this article, we apply this technique by making polynomial combinations of features with degree 2. We see that the model improves significantly, with MAE 222.579, MSE 12617018.64,RMSE 3552.04429. However, some assumptions on Linear Regression may break down in the process. Also, smoking is not good for your wallet !!