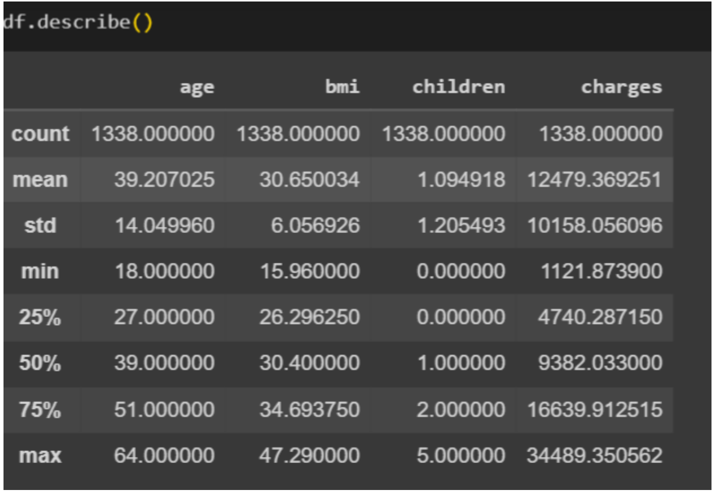
Data Collection and Preprocessing Phase

Date	4 July 2024
Team ID	740102
Project Title	Medical Cost Prediction
Maximum Marks	6 Marks

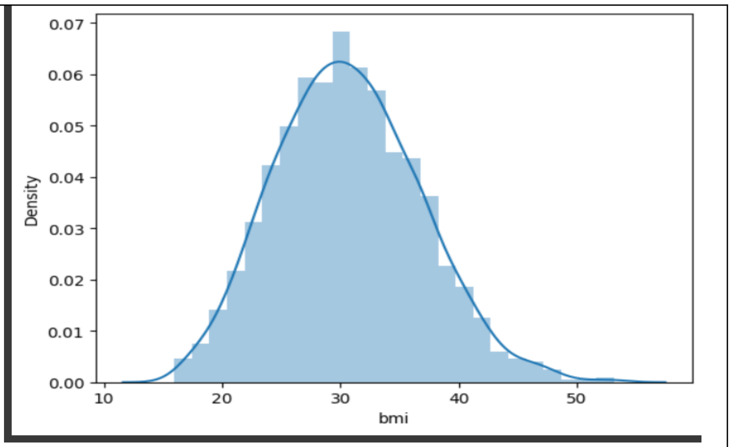
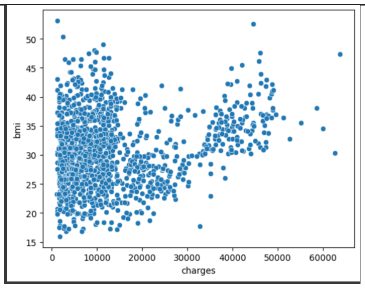
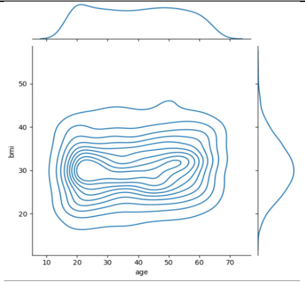
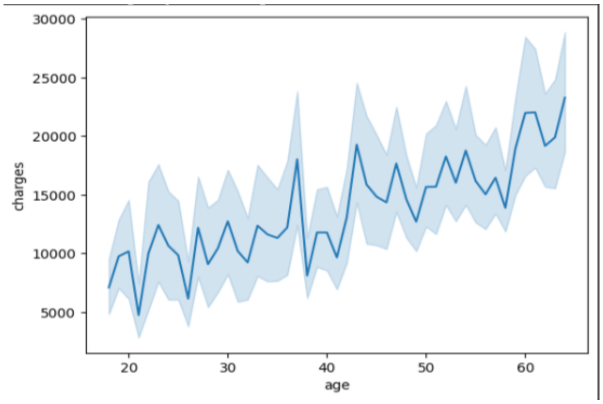
Data Exploration and Preprocessing Template

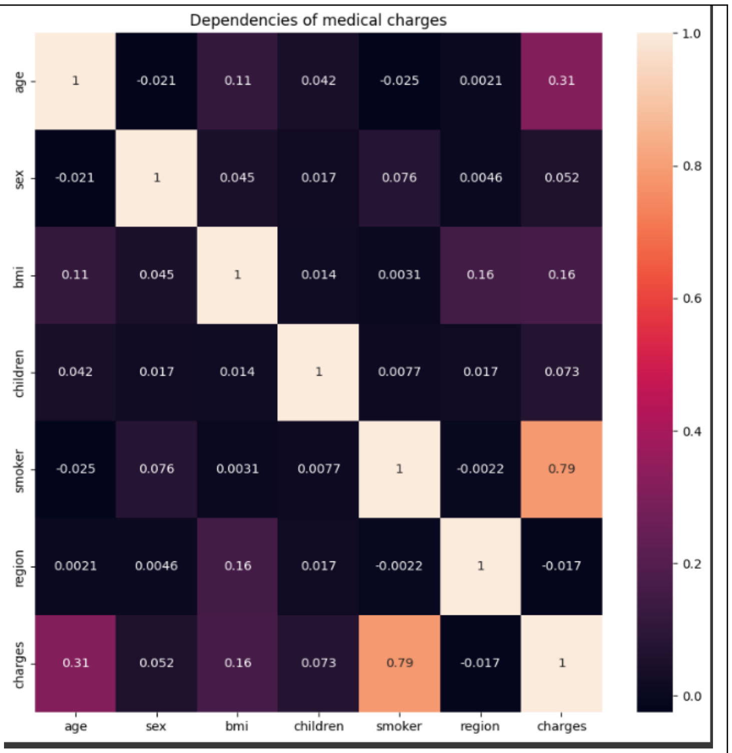
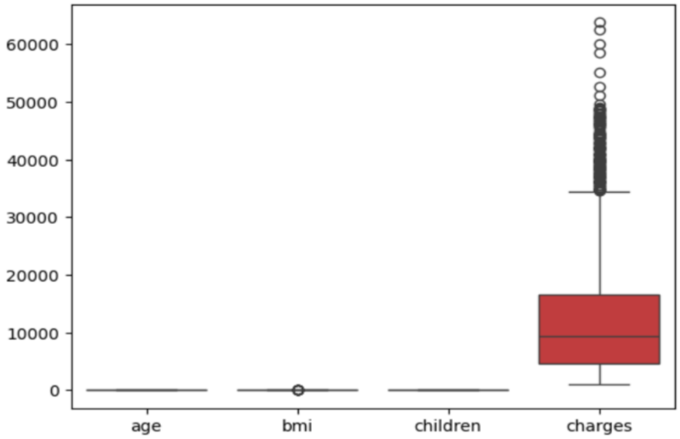
Data Exploration involves several key steps. They are importing libraries, loading the dataset, basic data overview, checking for missing values, visualizing the data distribution, correlation matrix. Data preprocessing involves the following steps they are handling missing values, encoding categorical variables, feature scaling, splitting the dataset.

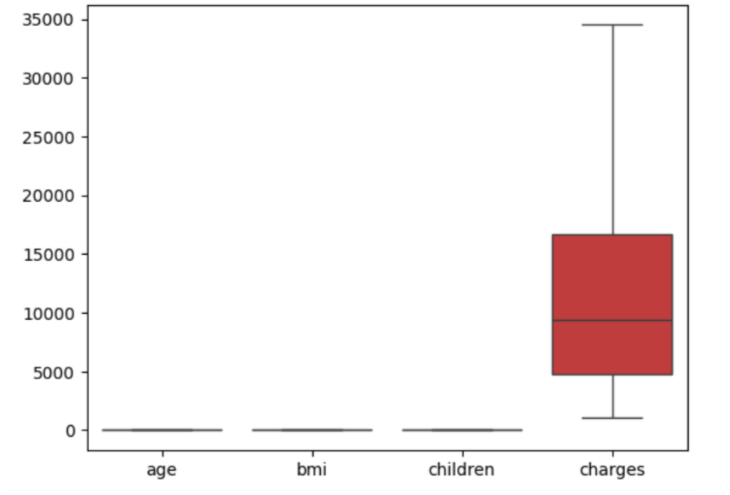
Section	Description
Data Overview	Dimension: 1338 rows x 7 columns Descriptive Statistics 

```
df.describe()
```

```
              age          bmi     children       charges
count  1338.000000  1338.000000  1338.000000   1338.000000
mean     39.207025    30.650034     1.094918  12479.369251
std      14.049960     6.056926     1.205493  10158.056096
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    47.290000     5.000000  34489.350562
```

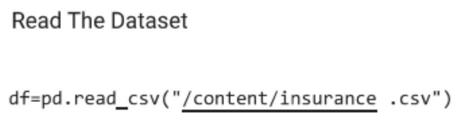
Univariate Analysis	
Bivariate Analysis	  

Multivariate Analysis	
Outliers and Anomalies	



Data Preprocessing Code Screenshots

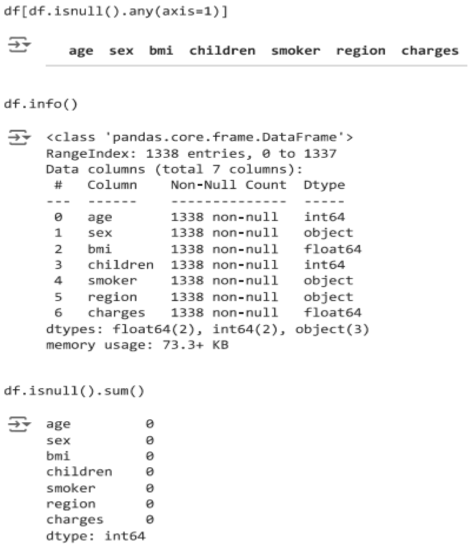
Loading Data



Read The Dataset

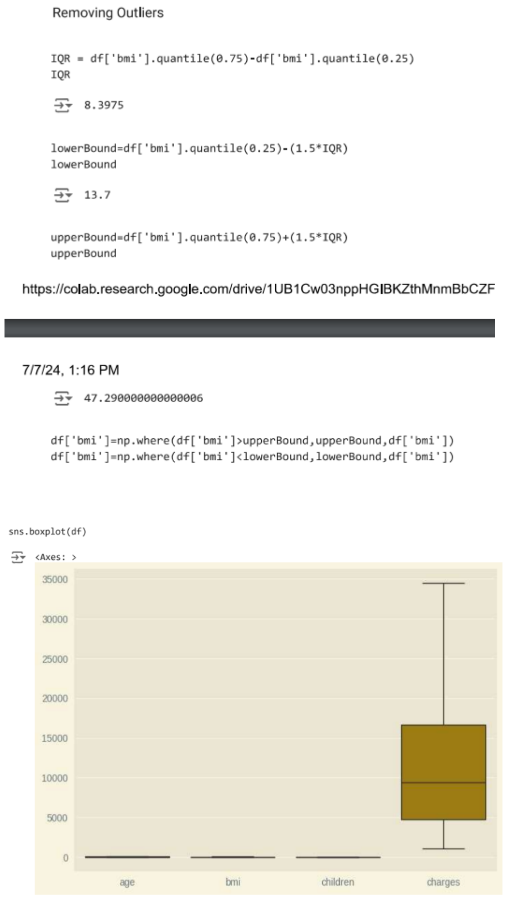
```
df=pd.read_csv("/content/insurance .csv")
```

Handling Missing Data



```
df[df.isnull().any(axis=1)]

     age  sex  bmi  children  smoker  region  charges

df.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

df.isnull().sum()

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

Denoising(Removing Outliers)



Removing Outliers

```
IQR = df['bmi'].quantile(0.75)-df['bmi'].quantile(0.25)
IQR
```

```
8.3975
```

```
lowerBound=df['bmi'].quantile(0.25)-(1.5*IQR)
lowerBound
```

```
13.7
```

```
upperBound=df['bmi'].quantile(0.75)+(1.5*IQR)
upperBound
```

https://colab.research.google.com/drive/1UB1Cw03nppHGIBKZthMnmBbCZF

7/7/24, 1:16 PM

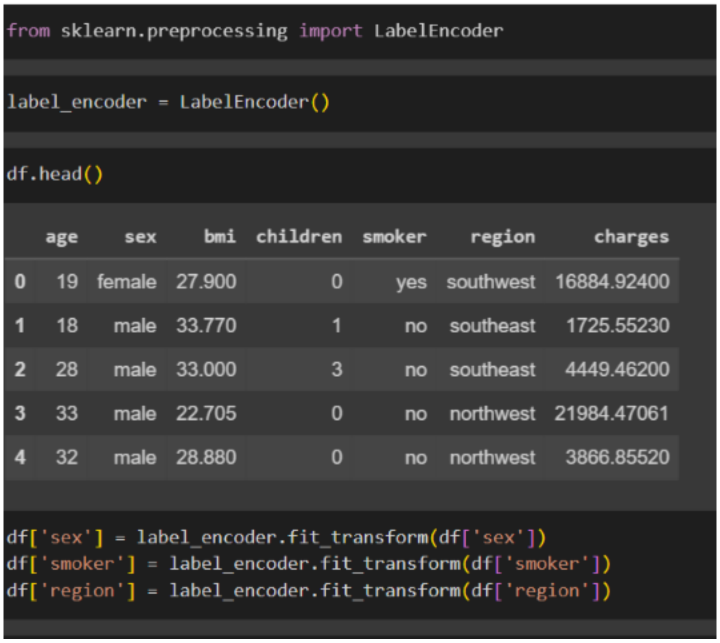
```
47.290000000000006
```

```
df['bmi']=np.where(df['bmi']>upperBound,upperBound,df['bmi'])
df['bmi']=np.where(df['bmi']<lowerBound,lowerBound,df['bmi'])
```

```
sns.boxplot(df)
```

```
<Axes: >
```

Data Transformation



```
from sklearn.preprocessing import LabelEncoder
```

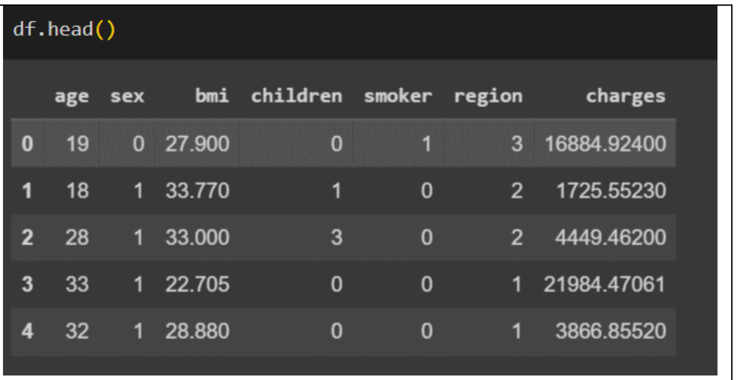
```
label_encoder = LabelEncoder()
```

```
df.head()
```

	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

```
df['sex'] = label_encoder.fit_transform(df['sex'])
df['smoker'] = label_encoder.fit_transform(df['smoker'])
df['region'] = label_encoder.fit_transform(df['region'])
```

<table>
<tr>
<td></td>
<td>

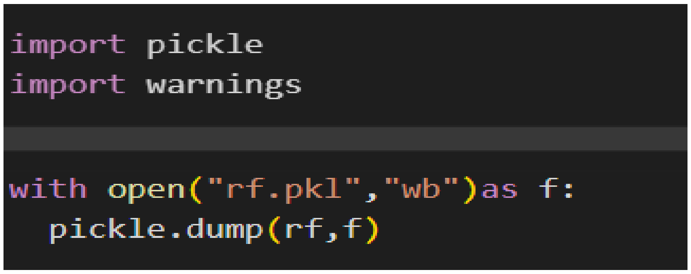


```
df.head()
```

```
   age  sex     bmi  children  smoker  region      charges
0   19    0  27.900         0       1       3  16884.92400
1   18    1  33.770         1       0       2   1725.55230
2   28    1  33.000         3       0       2   4449.46200
3   33    1  22.705         0       0       1  21984.47061
4   32    1  28.880         0       0       1   3866.85520
```

</td>
</tr>
<tr>
<td>Save Processed Data</td>
<td>

Save as Pickle
 Pickle is useful for saving and loading data frames in binary format



```
import pickle
import warnings
```

```
with open("rf.pkl","wb")as f:
  pickle.dump(rf,f)
```

</td>
</tr>
</table>