

Data Collection and Preprocessing Phase

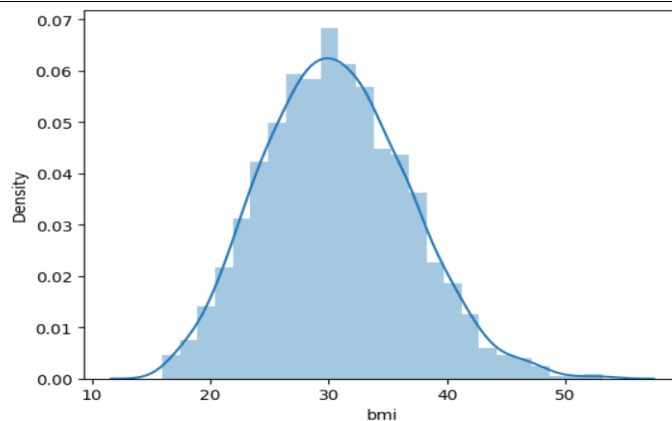
Date	4 July 2024
Team ID	team-739757
Project Title	Medical Cost Prediction
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

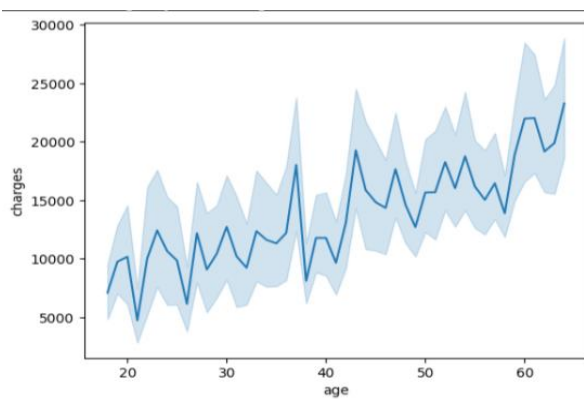
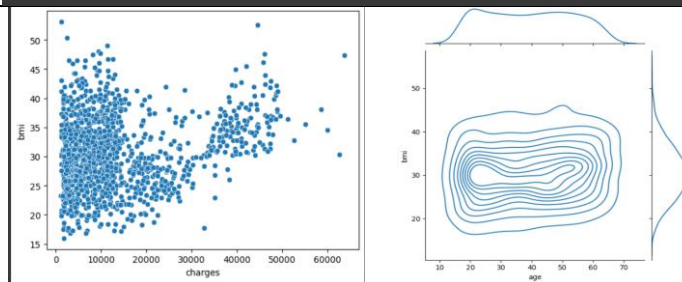
Data Exploration involves several key steps. They are importing libraries, loading the dataset, basic data overview, checking for missing values, visualizing the data distribution, correlation matrix. Data preprocessing involves the following steps they are handling missing values, encoding categorical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

Section	Description
Data Overview	Dimension: 1338 rows x 7 columns
	Descriptive Statistics
	df.describe()

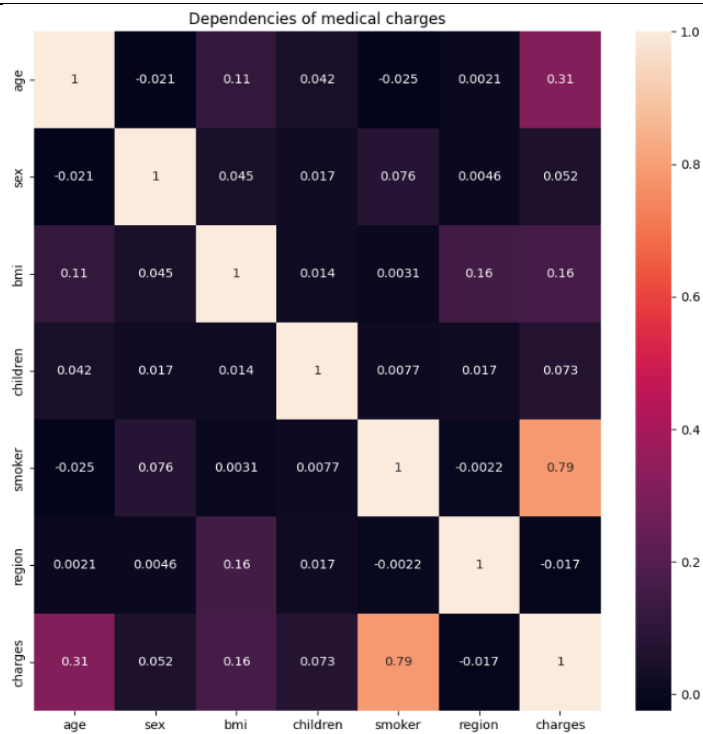
Univariate Analysis



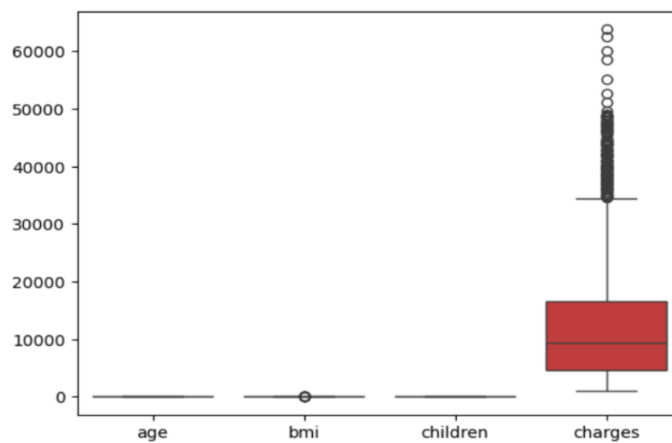
Bivariate Analysis

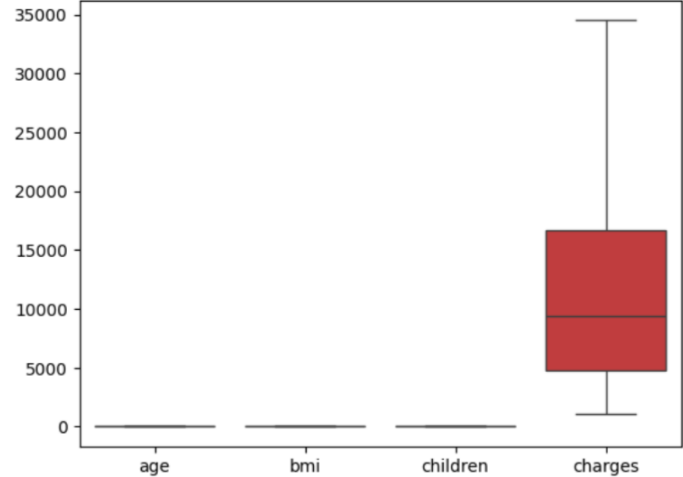


Multivariate Analysis



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>

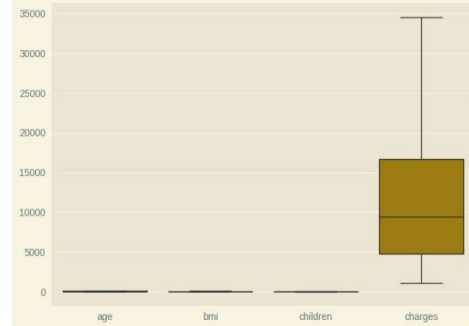
7/7/24, 1:16 PM

```
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'])
```

	<pre>df.head()</pre> <table><thead><tr><th></th><th>age</th><th>sex</th><th>bmi</th><th>children</th><th>smoker</th><th>region</th><th>charges</th></tr></thead><tbody><tr><td>0</td><td>19</td><td>0</td><td>27.900</td><td>0</td><td>1</td><td>3</td><td>16884.92400</td></tr><tr><td>1</td><td>18</td><td>1</td><td>33.770</td><td>1</td><td>0</td><td>2</td><td>1725.55230</td></tr><tr><td>2</td><td>28</td><td>1</td><td>33.000</td><td>3</td><td>0</td><td>2</td><td>4449.46200</td></tr><tr><td>3</td><td>33</td><td>1</td><td>22.705</td><td>0</td><td>0</td><td>1</td><td>21984.47061</td></tr><tr><td>4</td><td>32</td><td>1</td><td>28.880</td><td>0</td><td>0</td><td>1</td><td>3866.85520</td></tr></tbody></table>		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
	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																																										
Save Processed Data	<p>Save as Pickle</p> <p>Pickle is useful for saving and loading data frames in binary format</p> <pre>import pickle import warnings with open("rf.pkl","wb") as f: pickle.dump(rf,f)</pre>																																																

