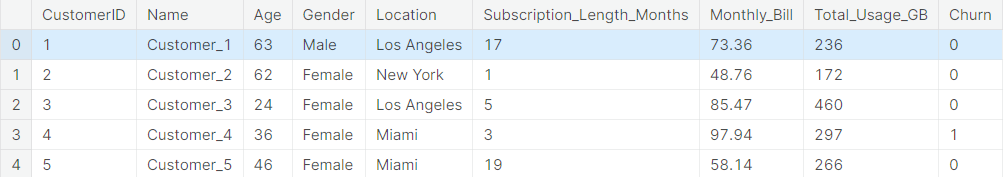
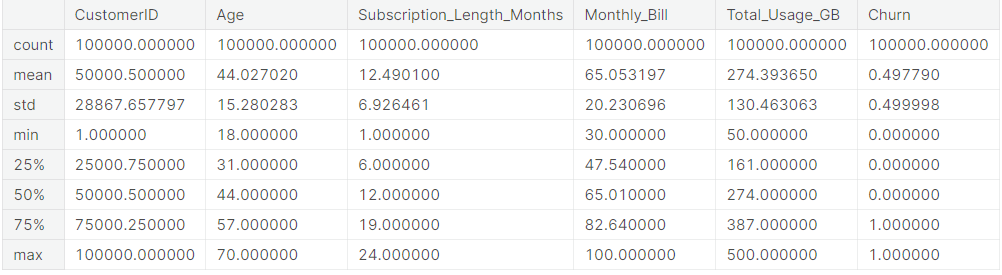
**Data Exploration :**

data.head()

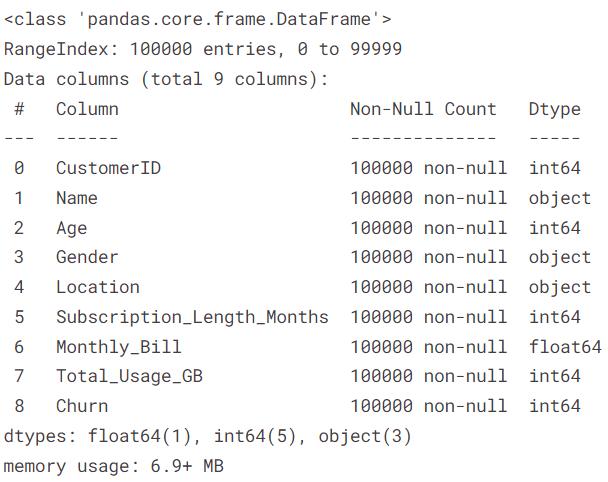
****

*# Simple output of the train dataset to view some of the min/max/varition of the dataset features.*

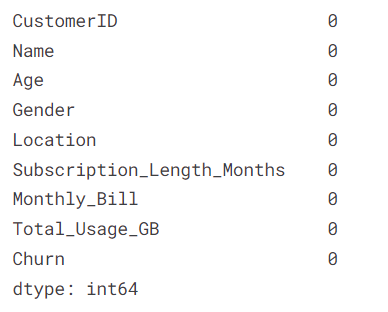
data.describe()



Data-types of columns:

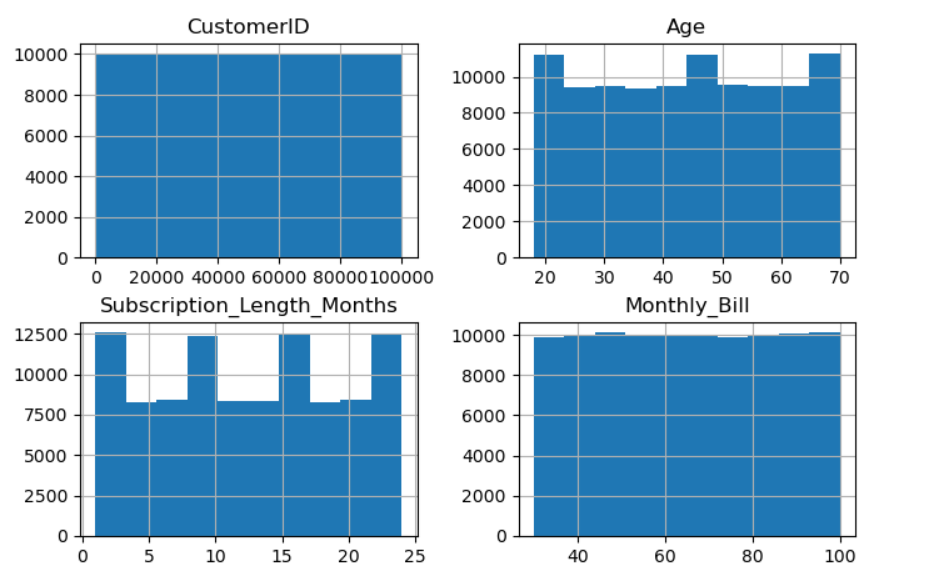


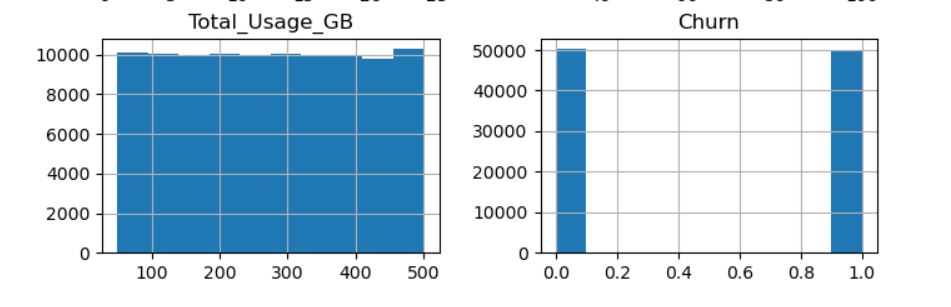
#### CHECKING FOR NULL VALUES: Since there are no missing values, therefore we do not need imputations using mean, mode etc.



Shape of data(Rows,columns):(100000, 9)

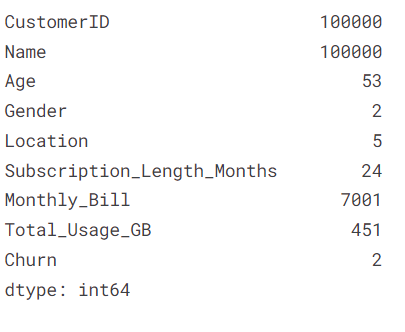
#### Dataset HISTOGRAM PLOTS: From the histogram plots, it seems that there are no outliers, so this question can also be ignored.





**No of unique values in Location column : 5** ['Los Angeles', 'New York', 'Miami', 'Chicago', 'Houston']

*No. of unique values in each column*



## **Data Preparation**

#### CustomerID and Name do not influence the customer churn, so they can be dropped from our dataset

data = data.drop(columns=['CustomerID', 'Name'])

#### After Dropping

#### 

#### Percentage of people churned by location

Percent of People Who Churned from Los Angeles ---> 49.2989371787835 %

Percent of People Who Churned from New York ---> 50.36592136476051 %

Percent of People Who Churned from Miami ---> 50.30203185063152 %

Percent of People Who Churned from Chicago ---> 49.829642248722315 %

Percent of People Who Churned from Houston ---> 49.10949049957831 %

#### Percentage of people churned by gender

Percent of Males Who Churned ---> 49.88550538325566 %

Percent of Females Who Churned ---> 49.67341086506293 %

#### Feature Scaling

#### I have used Standardisation but it did not give any better result.

#### You can see the usage in the code .

#### Gender & Location are categorical data, so we should go for respectively label encoding and Onehot encoding.

#### #Codes are in in the git repo.

#### HeatMap Analysis to check for correlation among the columns

#### 

#### Model Training

#### For train,test split (test\_size = 0.2) go through the code in the repository.

#### I have used several model among them LogisticRegression have given the highest accuracy.

#### That’s why I have trained the model using LogisticRegression.

#### For Model training part Go through the code section.

#### Model Optimization:

#### Used K-fold cross validation .

#### Refered to code section for more details.

#### Containerisation:

#### Altast I have contanerised the application and pushed the image in my DockerHub Repository .

#### Anyone can use the application irrespective of the operating system or dependencies by just going through the process / Can use my public image on dockerhub :

#### docker build -t customer\_churn .

#### docker run -d -p 8500:8501 customer\_churn

#### Acccesible at : http://localhost:8500/

#### My DockerHub Image Repo:

#### https://hub.docker.com/repository/docker/souvik2000/customer\_churn/general

*Everything is detailed in git hub repository :*

https://github.com/Souvik-clr/Churn\_task.git

#### Final App View :

#### 