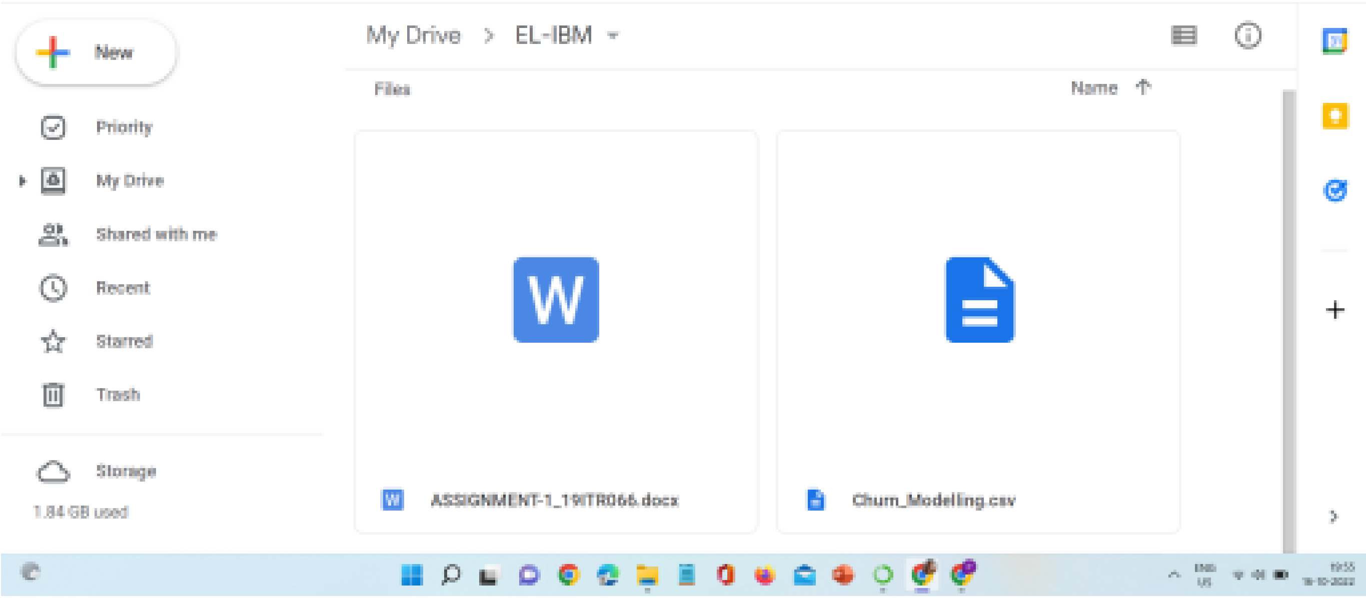
Priyanka V
19ITR065

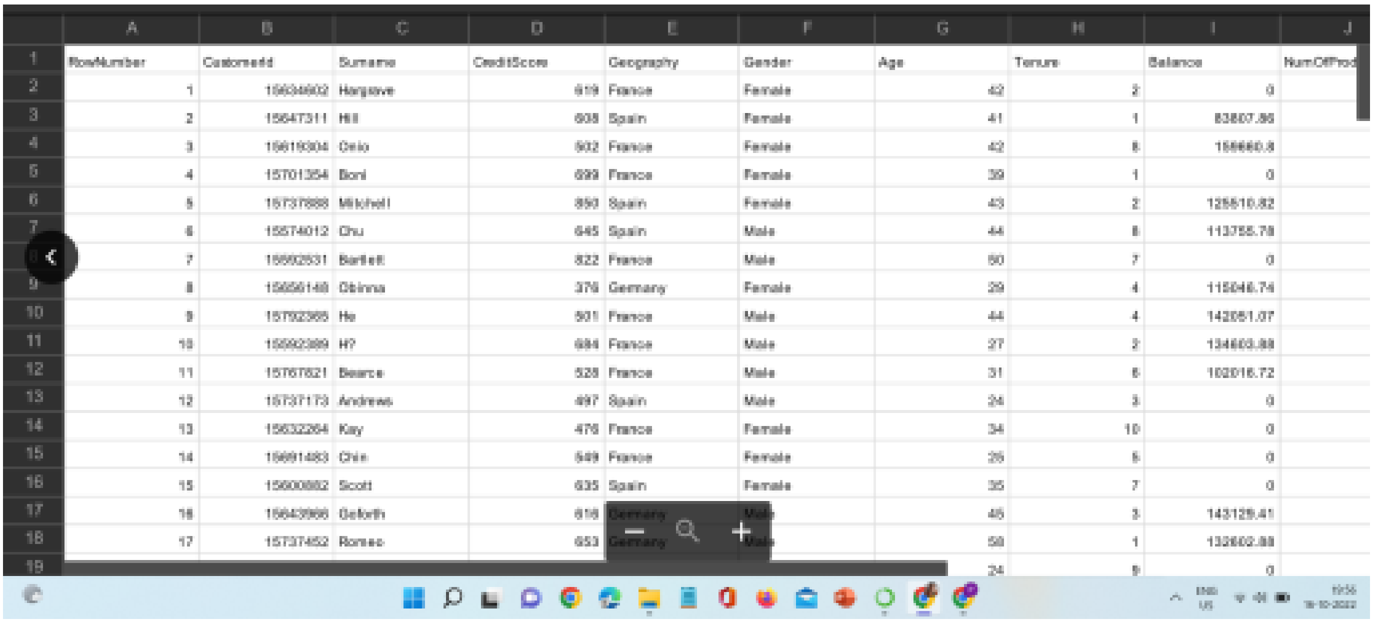
Data Visualization and Pre-processing:

Perform Below Tasks to complete the assignment:

Tasks:

1. Download the dataset: Dataset






```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

2. Load the dataset.

```
data = pd.read_csv("/content/drive/MyDrive/EL-IBM/Churn_Modelling.csv")
```

```
data
```

RowNumber CustomerId Surname CreditScore Geography Gender Age Tenure 0 1 15634602 Hargrave 619 France Female 42 2 **1** 2 15647311 Hill 608 Spain Female 41 1 **2** 3 15619304 Onio 502 France Female 42 8 **3** 4 15701354 Boni 699 France Female 39 1 **4** 5 15737888 Mitchell 850 Spain Female 43 2 **...** ... ...

9995 9996 15606229 Obijiaku 771 France Male 39 5 **9996** 9997 15569892 Johnstone 516 France Male 35 10 **9997** 9998 15584532 Liu 709 France Female 36 7 **9998** 9999 15682355 Sabbatini 772 Germany Male 42 3 **9999** 10000 15628319 Walker 792 France Female 28 4 10000 rows × 14 columns

```
data.tail()
```

RowNumber CustomerId Surname CreditScore Geography Gender Age Tenure 9995 9996 15606229 Obijiaku 771 France Male 39 5 **9996** 9997 15569892 Johnstone 516 France Male 35 10 **9997** 9998 15584532 Liu 709 France Female 36 7 **9998** 9999 15682355 Sabbatini 772 Germany Male 42 3 **9999** 10000 15628319 Walker 792 France Female 28 4

3. Perform Below Visualizations.

Univariate Analysis

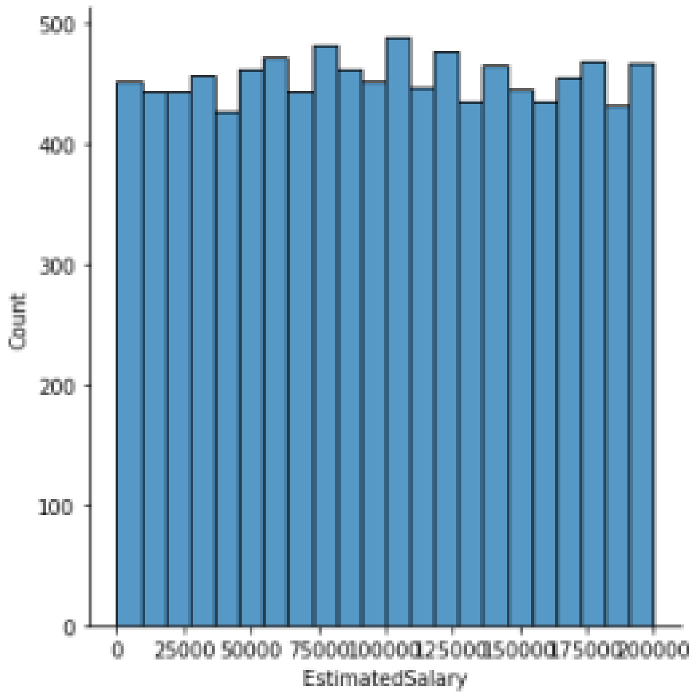
Bi - Variate Analysis

Multi - Variate Analysis

Univariate Analysis

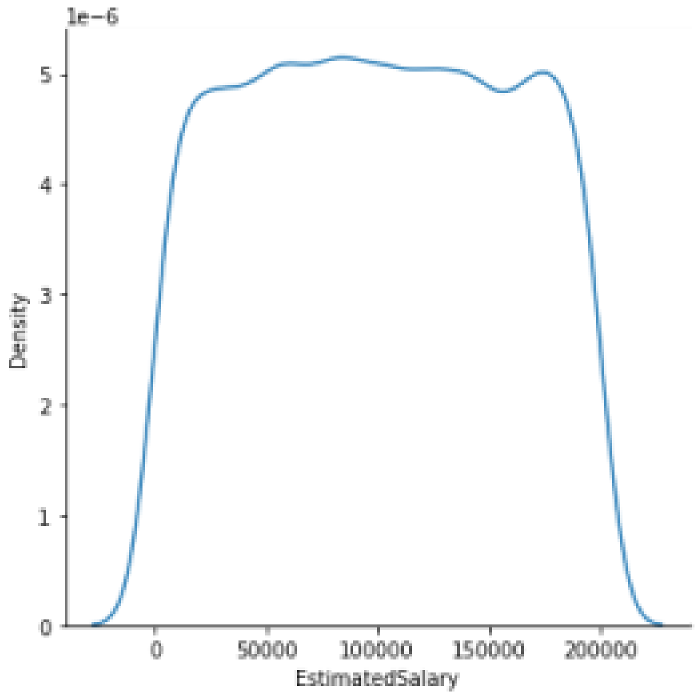
```
sns.displot(data.EstimatedSalary)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f8303fbbb90>
```



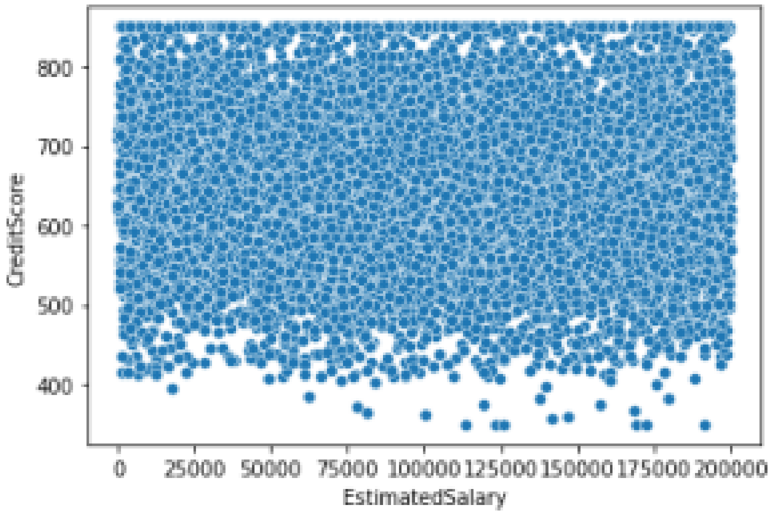

```
sns.displot(data.EstimatedSalary,kind="kde")
```

<seaborn.axisgrid.FacetGrid at 0x7f82e8ae4d10>



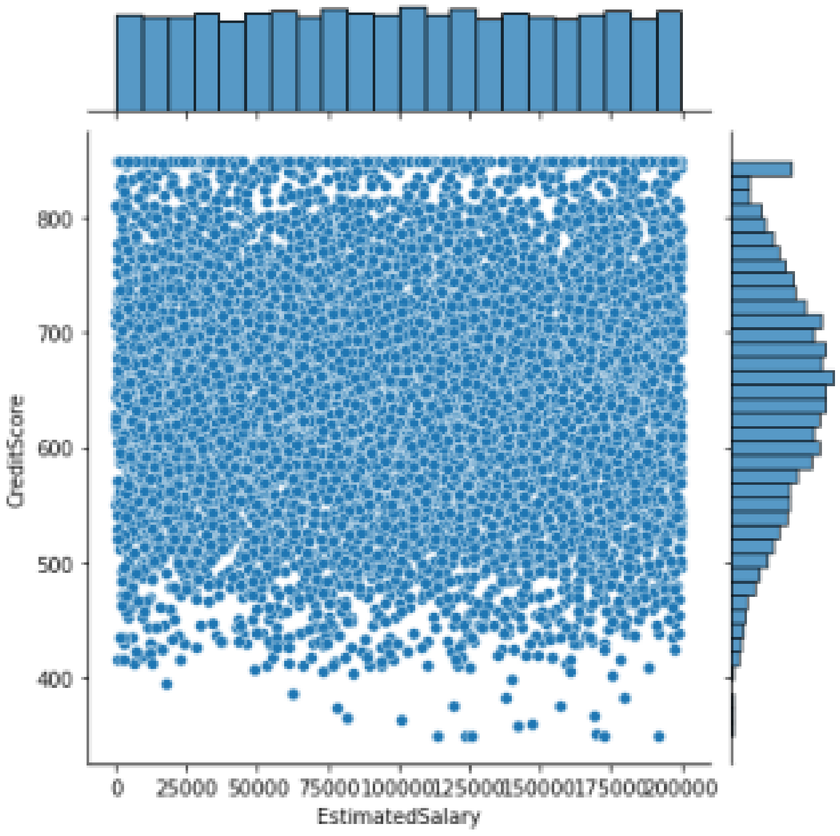

```
sns.scatterplot(data.EstimatedSalary,data.CreditScore
                                )
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f82e864a250>



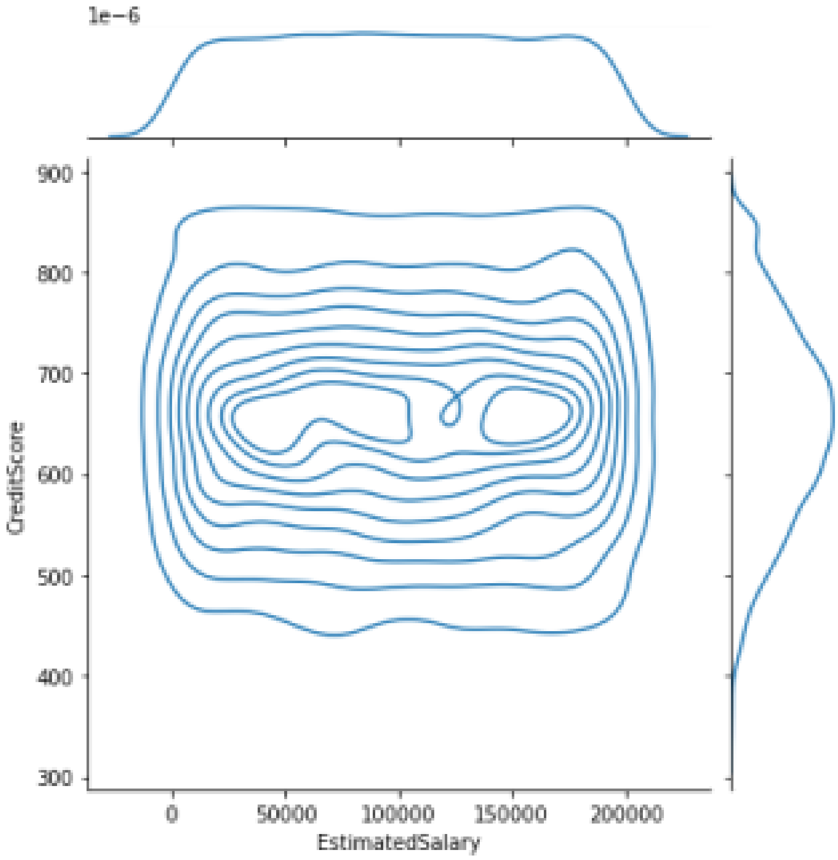

```
sns.jointplot(data.EstimatedSalary,data.CreditScore)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
FutureWarning
<seaborn.axisgrid.JointGrid at 0x7f82e8584410>
```



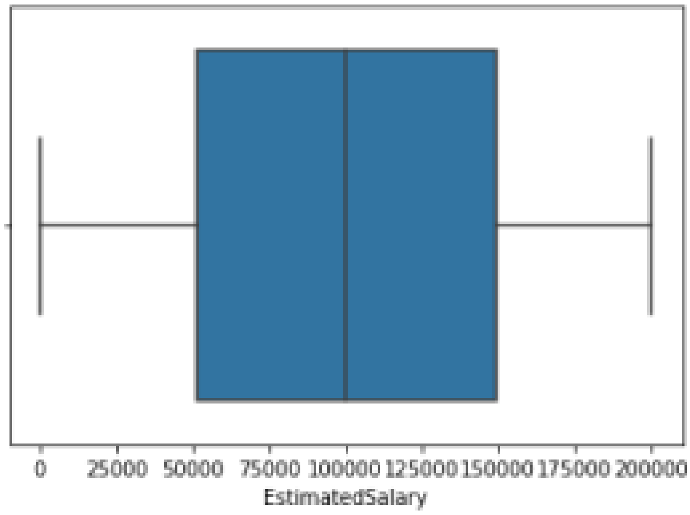

```
sns.jointplot(data.EstimatedSalary,data.CreditScore,kind="kde")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
FutureWarning
<seaborn.axisgrid.JointGrid at 0x7f82e6ba0ad0>
```



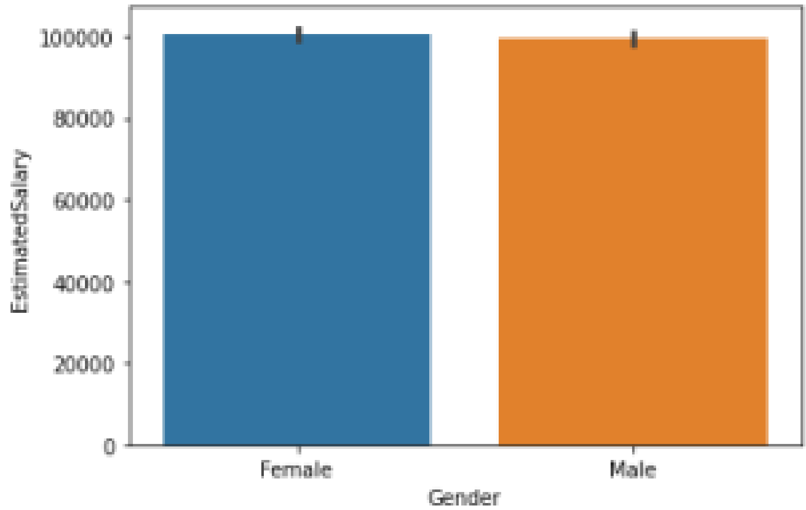

```
sns.boxplot(data.EstimatedSalary)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f82e681f290>
```




```
sns.barplot(y = data.EstimatedSalary,x = data.Gender)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f82e67d1cd0>
```



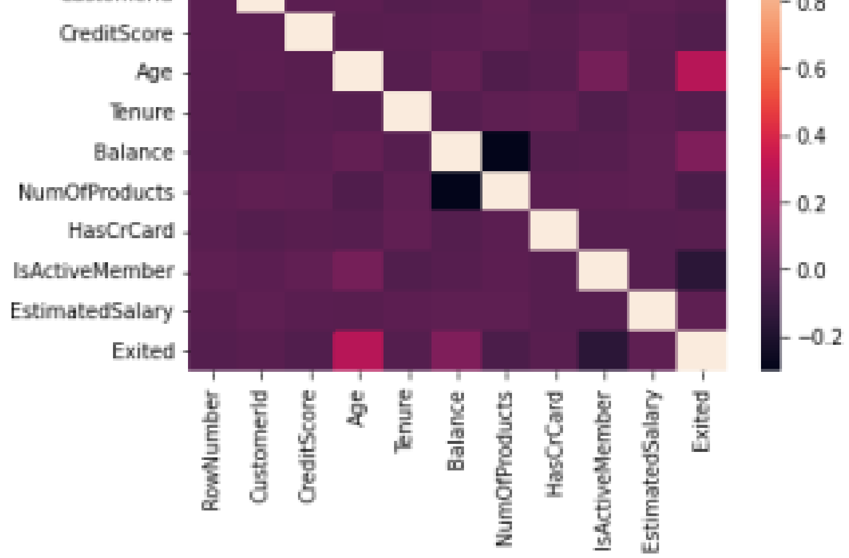

```
data.corr()
```

RowNumber CustomerId CreditScore Age Tenure Balance Nu RowNumber 1.000000 0.004202 0.005840 0.000783 -0.006495 -0.009067 **CustomerId** 0.004202 1.000000 0.005308 0.009497 -0.014883 -0.012419 **CreditScore** 0.005840 0.005308 1.000000 -0.003965 0.000842 0.006268

Age 0.000783 0.009497 -0.003965 1.000000 -0.009997 0.028308 **Tenure** -0.006495 -0.014883 0.000842 -0.009997 1.000000 -0.012254 **Balance** -0.009067 -0.012419 0.006268 0.028308 -0.012254 1.000000 **NumOfProducts** 0.007246 0.016972 0.012238 -0.030680 0.013444 -0.304180 **HasCrCard** 0.000599 -0.014025 -0.005458 -0.011721 0.022583 -0.014858 **IsActiveMember** 0.012044 0.001665 0.025651 0.085472 -0.028362 -0.010084 **EstimatedSalary** -0.005988 0.015271 -0.001384 -0.007201 0.007784 0.012797 **Exited** -0.016571 -0.006248 -0.027094 0.285323 -0.014001 0.118533

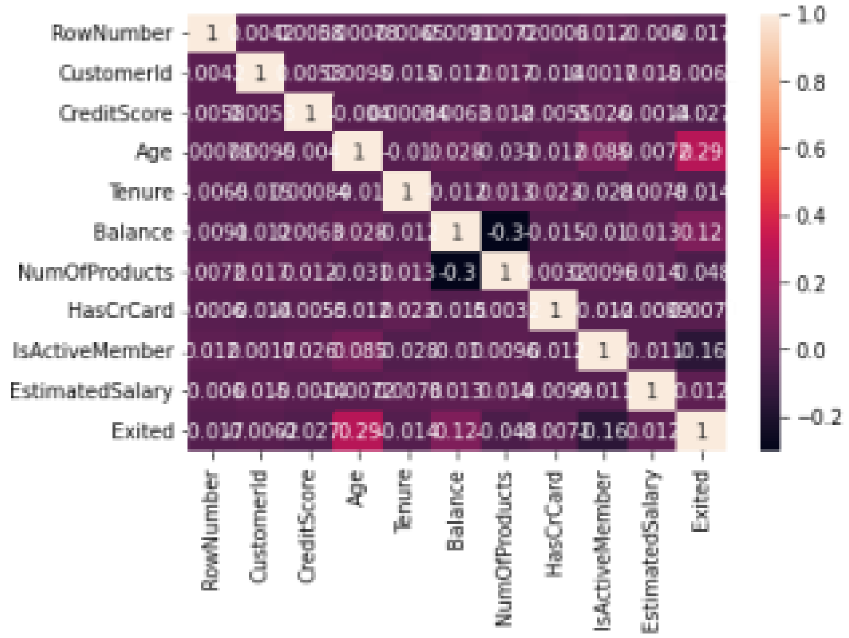
```
sns.heatmap(data.corr())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f82e6b81150>
```



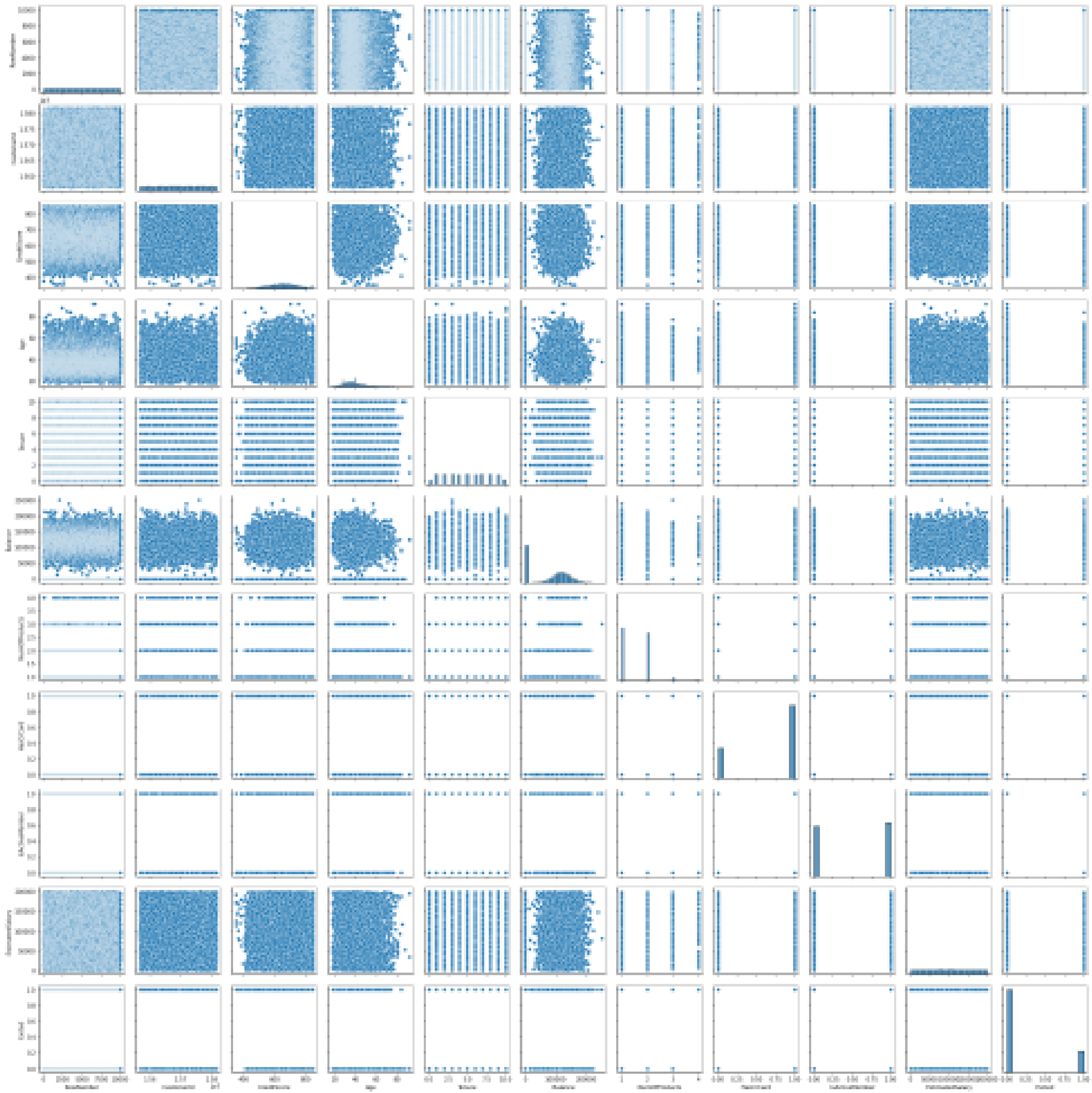

```
sns.heatmap(data.corr(),annot=True)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f82e6654f50>




```
sns.pairplot(data)
```

<seaborn.axisgrid.PairGrid at 0x7f82e649b110>



4. Perform descriptive statistics on the dataset.

```
data.sum(1)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Droppi
"""Entry point for launching an IPython kernel.
0 15736618.88
1 15844315.44
2 15893456.37
3 15795925.63
```

```
4 15943385.92
 ...
9995 15713313.64
9996 15739522.38
9997 15637370.58
9998 15861138.83
9999 15807478.57
Length: 10000, dtype: float64
```

```
data.std()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Droppi
"""Entry point for launching an IPython kernel.
RowNumber 2886.895680
CustomerId 71936.186123
CreditScore 96.653299
Age 10.487806
Tenure 2.892174
Balance 62397.405202
NumOfProducts 0.581654
HasCrCard 0.455840
IsActiveMember 0.499797
EstimatedSalary 57510.492818
Exited 0.402769
dtype: float64
```

```
data.describe()
```

RowNumber CustomerId CreditScore Age Tenure Balan count 10000.00000 1.000000e+04 10000.000000 10000.000000 10000.000000 10000.0000 **mean** 5000.50000 1.569094e+07 650.528800 38.921800 5.012800 76485.8892

std 2886.89568 7.193619e+04 96.653299 10.487806 2.892174 62397.4052 **min** 1.00000 1.556570e+07 350.000000 18.000000 0.000000 0.0000 **25%** 2500.75000 1.562853e+07 584.000000 32.000000 3.000000 0.0000 **50%** 5000.50000 1.569074e+07 652.000000 37.000000 5.000000 97198.5400 **75%** 7500.25000 1.575323e+07 718.000000 44.000000 7.000000 127644.2400 **max** 10000.00000 1.581569e+07 850.000000 92.000000 10.000000 250898.0900

5. Handle the Missing values.

```
data.isnull()
```

RowNumber CustomerId Surname CreditScore Geography Gender Age Tenure 0 False False False False False False False False **1** False False False False False False False False **2** False False False False False False False False **3** False False False False False False False False

4 False False False False False False False False **...** ... ...

9995 False False False False False False False False **9996** False False False False False False False False **9997** False False False False False False False False **9998** False False False False False False False False **9999** False False False False False False False False 10000 rows × 14 columns

```
data[pd.isnull(data)]
```

RowNumber CustomerId Surname CreditScore Geography Gender Age Tenure B **0** NaN NaN NaN NaN NaN NaN NaN NaN **1** NaN NaN NaN NaN NaN NaN NaN NaN **2** NaN NaN NaN NaN NaN NaN NaN NaN **3** NaN NaN NaN NaN NaN NaN NaN NaN **4** NaN NaN NaN NaN NaN NaN NaN NaN

... ...

9995 NaN NaN NaN NaN NaN NaN NaN NaN **9996** NaN NaN NaN NaN NaN NaN NaN NaN **9997** NaN NaN NaN NaN NaN NaN NaN NaN **9998** NaN NaN NaN NaN NaN NaN NaN NaN **9999** NaN NaN NaN NaN NaN NaN NaN NaN 10000 rows × 14 columns

```
data.isnull().sum()
```

```
RowNumber 0
CustomerId 0
Surname 0
CreditScore 0
Geography 0
Gender 0
Age 0
Tenure 0
Balance 0
NumOfProducts 0
HasCrCard 0
IsActiveMember 0
EstimatedSalary 0
Exited 0
dtype: int64
```

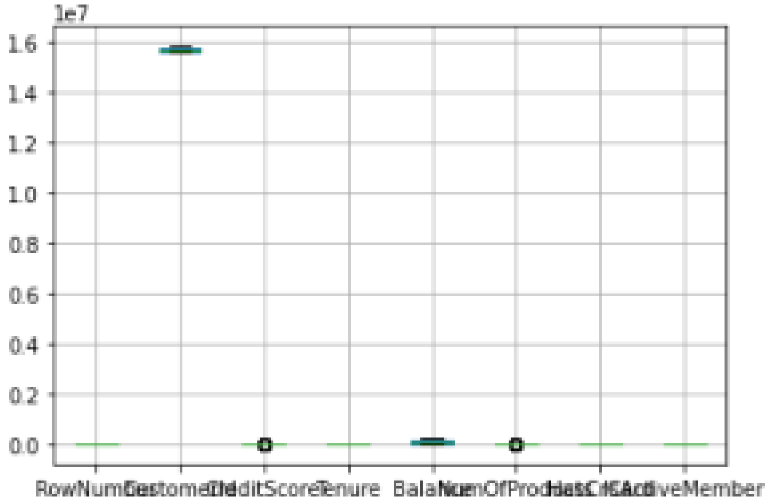
```
data["Gender"].fillna("No Gender", inplace = True)
```

6. Find the outliers and replace the outliers.

```
numeric_col = ['RowNumber', 'CustomerId', 'CreditScore', 'Tenure', 'Balance', 'NumOfProd
```

```
data.boxplot(numeric_col)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f82e3434e50>
```




```
for x in ['CreditScore']:
    q75,q25 = np.percentile(data.loc[:,x],[75,25])
    intr_qr = q75-q25

    max = q75+(1.5*intr_qr)
    min = q25-(1.5*intr_qr)

    data.loc[data[x] < min,x] = np.nan
    data.loc[data[x] > max,x] = np.nan

data.isnull().sum()
```

```
RowNumber 0
CustomerId 0
Surname 0
CreditScore 15
Geography 0
Gender 0
Age 0
Tenure 0
Balance 0
NumOfProducts 0
HasCrCard 0
IsActiveMember 0
EstimatedSalary 0
Exited 0
dtype: int64
```

```
data = data.dropna(axis=0)
data
```

RowNumber CustomerId Surname CreditScore Geography Gender Age Tenure 0 1

15634602 Hargrave 619.0 France Female 42 2 **1** 2 15647311 Hill 608.0 Spain Female 41 1 **2** 3

15619304 Onio 502.0 France Female 42 8 **3** 4 15701354 Boni 699.0 France Female 39 1 **4** 5

15737888 Mitchell 850.0 Spain Female 43 2 **...** ... ...

9995 9996 15606229 Obijiaku 771.0 France Male 39 5 **9996** 9997 15569892 Johnstone 516.0 France

Male 35 10 **9997** 9998 15584532 Liu 709.0 France Female 36 7 **9998** 9999 15682355 Sabbatini 772.0

Germany Male 42 3 **9999** 10000 15628319 Walker 792.0 France Female 28 4 9985 rows × 14 columns

```
data.isnull().sum()
```

```
RowNumber 0
CustomerId 0
Surname 0
CreditScore 0
Geography 0
Gender 0
Age 0
Tenure 0
Balance 0
NumOfProducts 0

HasCrCard 0
IsActiveMember 0
EstimatedSalary 0
Exited 0
dtype: int64
```

7. Check for Categorical columns and perform encoding.

```
data.dtypes
```

```
RowNumber int64
CustomerId int64
Surname object
CreditScore float64
Geography object
Gender object
Age int64
Tenure int64
Balance float64
NumOfProducts int64
HasCrCard int64
IsActiveMember int64
EstimatedSalary float64
Exited int64
dtype: object
```

```
obj = data.select_dtypes(include=['object']).copy()
obj.head()
```

	Surname	Geography	Gender
0	Hargrave	France	Female
1	Hill	Spain	Female
2	Onio	France	Female
3	Boni	France	Female
4	Mitchell	Spain	Female

```
obj[obj.isnull().any(axis=1)].sum()
```

```
Surname 0.0
Geography 0.0
Gender 0.0
dtype: float64
```

```
pd.get_dummies(obj, columns=["Geography"]).head()
```

Surname Gender Geography_France Geography_Germany Geography_Spain 0

Hargrave Female 1 0 0 **1** Hill Female 0 0 1 **2** Onio Female 1 0 0

3 Boni Female 1 0 0 `pd.get_dummies(obj, columns=["Geography", "Gender"], prefix=["Geo","Gen"]).head()`

4 Mitchell Female 0 0 1 **Surname Geo_France Geo_Germany Geo_Spain Gen_Female Gen_Male**

0 Hargrave 1 0 0 1 0 **1** Hill 0 0 1 1 0 **2** Onio 1 0 0 1 0 **3** Boni 1 0 0 1 0 **4** Mitchell 0 0 1

1 0

```
data["CreditScore"].min()
```

```
383.0
```

```
data["CreditScore"].max()
```

```
850.0
```

```
data["CreditScore"].mean()
```

```
650.963244867301
```

```
data.count(0)
```

```
RowNumber 9985
CustomerId 9985
Surname 9985
CreditScore 9985
Geography 9985
Gender 9985
Age 9985
Tenure 9985
Balance 9985
NumOfProducts 9985
HasCrCard 9985
IsActiveMember 9985
EstimatedSalary 9985
Exited 9985
dtype: int64
```

```
data.shape
```

```
(9985, 14)
```

```
data.size
```

```
139790
```

```
data.iloc[:, :-1].values
```

```
array([[1, 15634602, 'Hargrave', ..., 1, 1, 101348.88],
 [2, 15647311, 'Hill', ..., 0, 1, 112542.58],
 [3, 15619304, 'Onio', ..., 1, 0, 113931.57],
 ...,
 [9998, 15584532, 'Liu', ..., 0, 1, 42085.58],
 [9999, 15682355, 'Sabbatini', ..., 1, 0, 92888.52],
 [10000, 15628319, 'Walker', ..., 1, 0, 38190.78]], dtype=object)
```

```
data.iloc[:, -1].values
```

```
array([1, 0, 1, ..., 1, 1, 0])
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit_transform(data[numeric_col])
```

```
array([[-1.73298629, -0.78261344, -0.3327168 , ..., -0.91274609,
 0.64646813, 0.96951794],
 [-1.7326397 , -0.6059255 , -0.44721972, ..., -0.91274609,
 -1.54686666, 0.96951794],
 [-1.73229312, -0.99529517, -1.55061149, ..., 2.53031008,
 0.64646813, -1.03144043],
 ...,
 [ 1.73179791, -1.47871581, 0.60412526, ..., -0.91274609,
 -1.54686666, 0.96951794],
 [ 1.73214449, -0.11872336, 1.25991471, ..., 0.808782 ,
 0.64646813, -1.03144043],
 [ 1.73249107, -0.86996338, 1.46810183, ..., -0.91274609,
 0.64646813, -1.03144043]])
```

```
from sklearn.model_selection import train_test_split
X = data.loc[:, numeric_col]
categoric_col=['Surname','Geography','Gender']
y = data.loc[:, categoric_col]
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, train_size = .75
X_train
```

RowNumber CustomerId CreditScore Tenure Balance NumOfProducts HasCrCard 8158

8159 15744127 641.0 2 0.00 2 1 **2469** 2470 15630617 727.0 6 140418.81 1 1 **6455** 6456 15701522

711.0 9 0.00 2 0 **2763** 2764 15654495 706.0 6 120621.89 1 1 **8243** 8244 15572174 825.0 3 148874.01 2

0

... ... **9238** 9239 15639133 773.0 4 0.00 2 1 **4868** 4869 15661330 754.0 6 0.00 1 1

3273 3274 15646091 560.0 4 95140.44 2 1 X_test

9860 9861 15716431 775.0 10 191091.74 2 1 **RowNumber CustomerId CreditScore Tenure Balance NumOfProducts HasCrCard** 2741 2742 15687738 535.0 8 0.00 2 1 **335** 336 15697441 485.0 7 182123.79 1 1 7488 rows × 8 columns

62[redacted] 2 0 **5807** 5808 15607395 679.0 9 112528.65 2 1 **6041** 6042

15749472 775.0 8 0.00 1 1 **8506** 8507 15605215 767.0 9 0.00 2 0

... ... **5108** 5109 15777772 650.0 9 119618.42 1 1 **3052** 3053 15605327 607.0 2 0.00 2 1

2337 2338 15660688 701.0 9 0.00 2 0 **6866** 6867 15664506 675.0 8 197436.82 1 1

641 642 15580684 706.0 5 112564.62 1 1 2497 rows × 8 columns

y_train

Surname Geography Gender

8158 Kosovich France Female **2469** Lo Duca

Germany Male **6455** Yermolayeva France

Female **2763** Potter Germany Female **8243**

Mazzi France Male

y_test **...** ... ...

9238 Ku France Female **Surname Geography Gender**

4868 Gilbert France Male **335** Hsueh France Male **3273** Frankland Spain Female **6245** Ch'ang Spain Male **9860** Brookes France Female **5807** Holt France Female **2741** Nwagugheuzo France Female **6041** Lucciano France Male 7488 rows × 3 columns

8506 Stevenson France Male **...** ... ...

5108 Whittaker Spain Male **3052** Namatjira

France Male **2337** King Spain Female

6866 Goodwin Spain Male **641** Feng

France Female 2497 rows × 3 columns