**Assignment -2**

**Data Visualization and Preprocessing**

|  |  |
| --- | --- |
| Assignment submission | 10 October 2022 |
| Student Name | Uthra v |
| Student Roll Number | 951919CS108 |
| Maximum Marks | 2 Marks |

1. Download the dataset: Dataset
2. Load the dataset.

import pandas as pd

df=pd.read\_csv('Churn\_Modelling.csv')

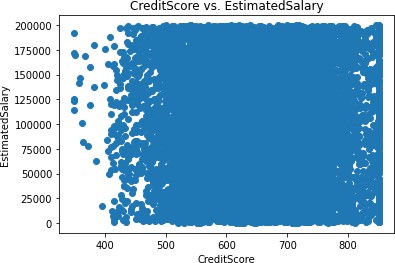
1. Perform Below Visualizations

● Univariate Analysis

import matplotlib.pyplot as plt

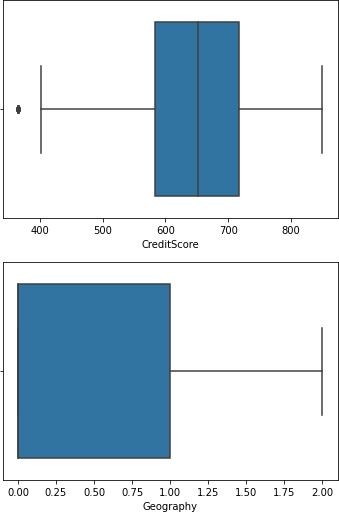
%matplotlib inline

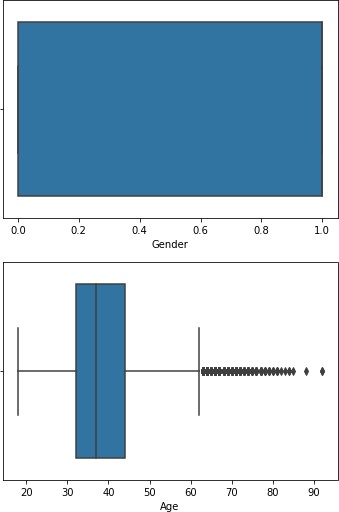
plt.scatter(df.CreditScore,df.EstimatedSalary) plt.title('CreditScore vs. EstimatedSalary') plt.xlabel('CreditScore') plt.ylabel('EstimatedSalary') plt.show()

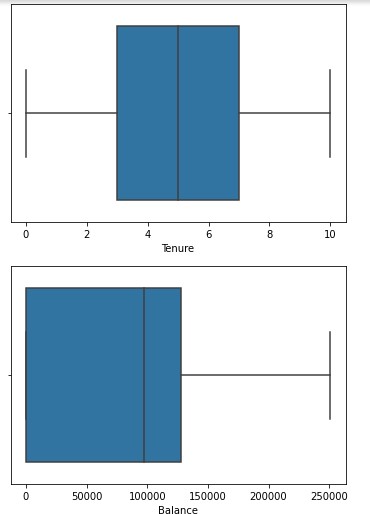


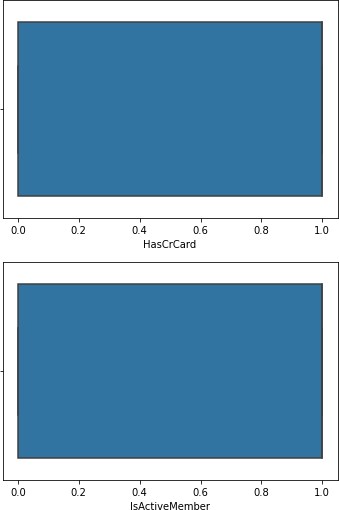
for col in df.columns: if(df.dtypes[col]=='int64' or df.dtypes[col]=='float64' ):

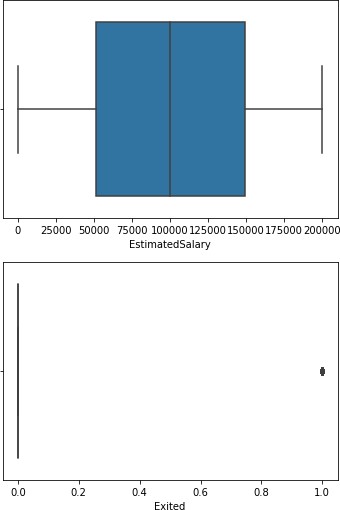
sns.boxplot(x=df[col]).set( xlabel=col) plt.show()







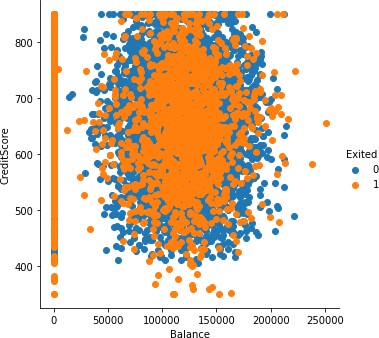




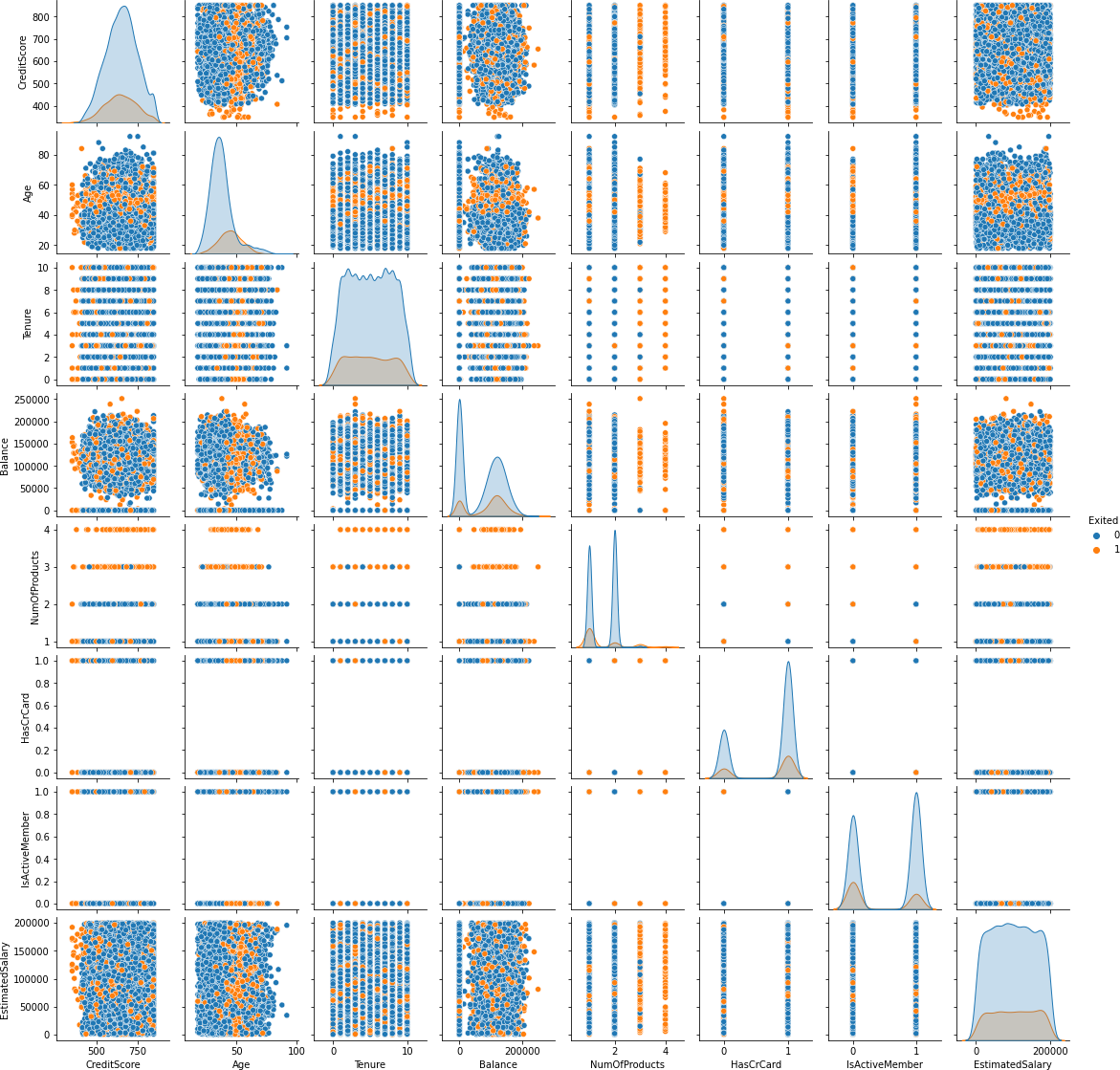
* Bi - Variate Analysis

import seaborn as sns

sns.FacetGrid(df,hue='Exited',height=5).map(plt.scatter,"Balance","CreditScore").add\_legend() plt.show()



* Multi - Variate Analysis sns.pairplot(df, hue='Exited', height=2)



1. Perform descriptive statistics on the dataset.

df.describe()



1. Handle the Missing values. df.isnull().sum()

CreditScore 0

Geography 0

|  |  |
| --- | --- |
| Gender | 0 |
| Age | 0 |
| Tenure | 0 |
| Balance | 0 |
| NumOfProducts | 0 |
| HasCrCard | 0 |
| IsActiveMember | 0 |
| EstimatedSalary | 0 |
| Exited dtype: int64 | 0 |

#there is no missing values

1. Find the outliers and replace the outliers

import numpy as np

#Outliers are found using the univariate

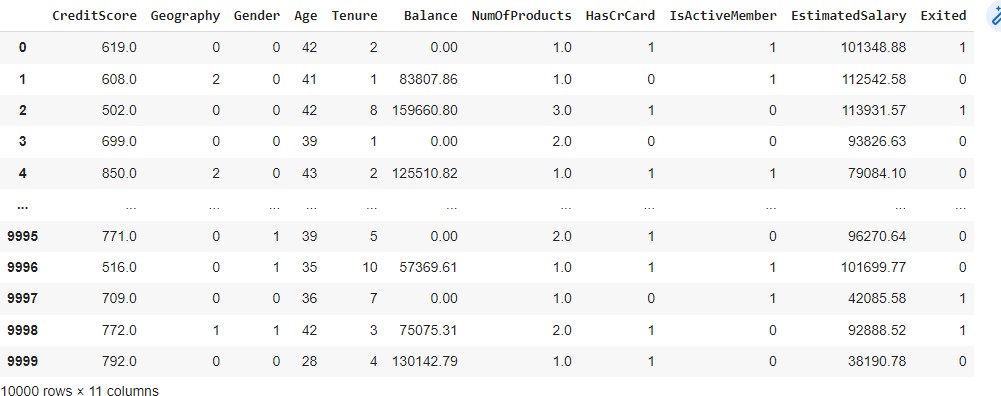
CreditsMedian = df.loc[df['CreditScore']<400, 'CreditScore'].median()

ProdMedian = df.loc[df['NumOfProducts']>=3.5,'NumOfProducts'].median()

df.loc[df.CreditScore < 400, 'CreditScore'] = np.nan df.fillna(CreditsMedian,inplace=True)

df.loc[df.NumOfProducts > 3, 'NumOfProducts'] = np.nan df.fillna(ProdMedian,inplace=True)

df



1. Check for Categorical columns and perform encoding.

df.drop(['RowNumber','CustomerId','Surname'],axis=1,inplace=True)

df.info()

#we have 2 categorial information

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 10000 entries, 0 to 9999 Data columns (total 11 columns):

# Column Non-Null Count Dtype

1. CreditScore 10000 non-null int64
2. Geography 10000 non-null object
3. Gender 10000 non-null object
4. Age 10000 non-null int64 4 Tenure 10000 non-null int64
5. Balance 10000 non-null float64
6. NumOfProducts 10000 non-null int64
7. HasCrCard 10000 non-null int64
8. IsActiveMember 10000 non-null int64
9. EstimatedSalary 10000 non-null float64
10. Exited 10000 non-null int64

dtypes: float64(2), int64(7), object(2) memory usage: 859.5+ KB

from sklearn.preprocessing import LabelEncoder,MinMaxScaler labelencoder = LabelEncoder() df['Geography']= labelencoder.fit\_transform(df['Geography']) df['Gender'] = labelencoder.fit\_transform(df['Gender'])

1. Split the data into dependent and independent variables.

x= df.iloc[:, :-1]

y= df.iloc[:,-1:]

1. Scale the independent variables

from sklearn.preprocessing import MinMaxScaler nm =MinMaxScaler()

X = nm.fit\_transform(x)

10. Split the data into training and testing

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,random\_state=0)