Assignment – 2

Data Visualization and Data Preprocessing

Assignment Date	24 September 2022
Student Name	Mithrha R
Student Roll Number	2019504550
Maximum Marks	2 marks

1. Load The Necessary Libraries:

```
[1]  1  import numpy as np
     2  import pandas as pd
     3  import matplotlib.pyplot as plt
     4  import seaborn as sns
```

2. Load the Dataset

```
[2]  1  df=pd.read_csv(r"Churn_Modelling.csv")
```

3. View the Dataset

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0

10000 rows × 14 columns

```
1  df.head()
2
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Estimated
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	10
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	11
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	11
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	9
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	7

```
1  df.tail()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Estin
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	

4. Data Visualization

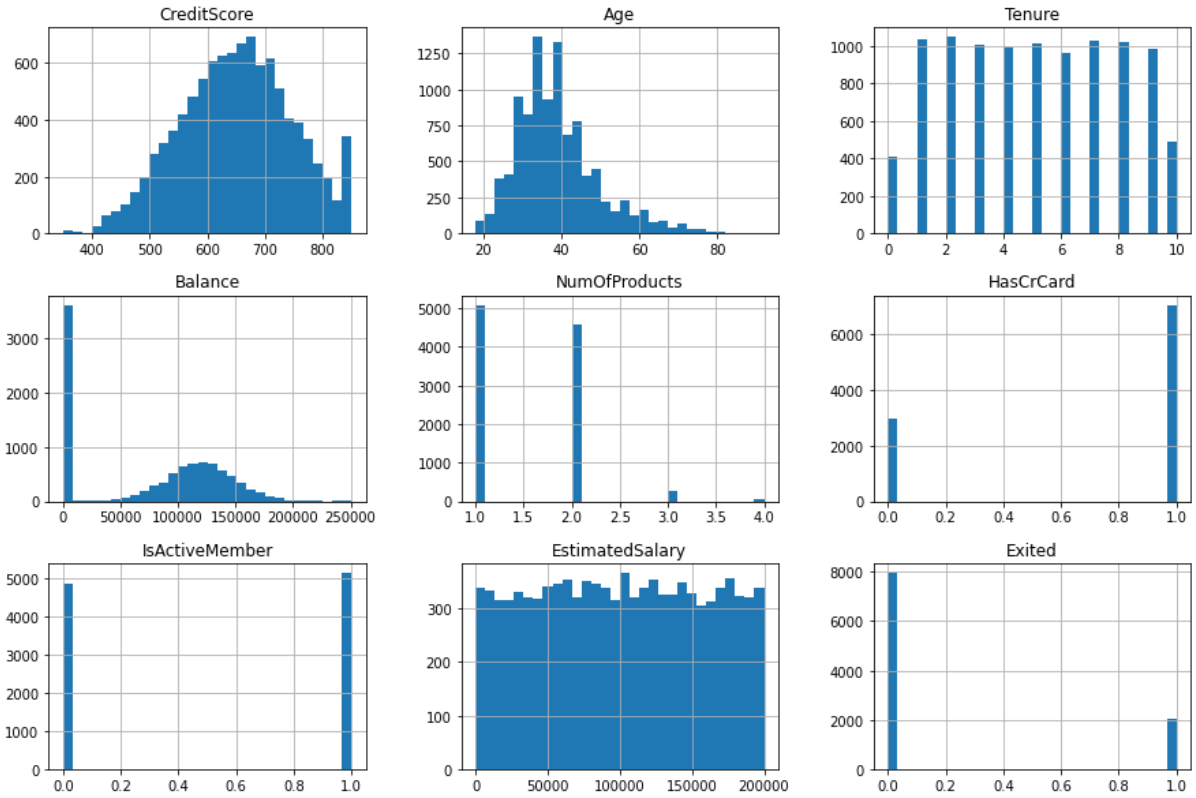
a. Univariate Analysis

```
list(df.columns)
```

```
['RowNumber',
 'CustomerId',
 'Surname',
 'CreditScore',
 'Geography',
 'Gender',
 'Age',
 'Tenure',
 'Balance',
 'NumOfProducts',
 'HasCrCard',
 'IsActiveMember',
 'EstimatedSalary',
 'Exited']
```

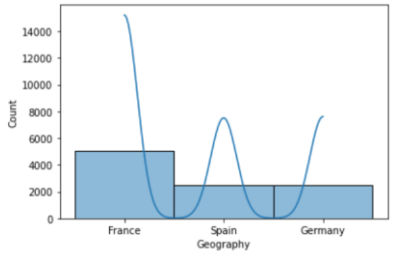
```
df.hist(column=[
 'CreditScore',
  'Age',
 'Tenure',
 'Balance',
 'NumOfProducts',
 'HasCrCard',
 'IsActiveMember',
 'EstimatedSalary',
 'Exited'],bins=30, figsize=(15, 10))
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f1a7a8b3390>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1a7a8879d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1a7a837b10>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f1a7a803610>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1a7a7bac10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1a7a77e250>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f1a7a7338d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1a7a768e10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1a7a768e50>]],
      dtype=object)
```



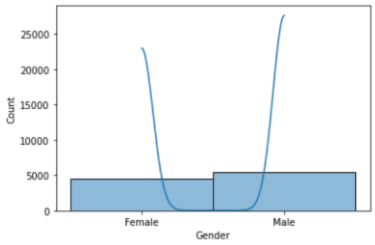

```
1  sns.histplot(df.Geography,kde=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1a7a582d50>
```



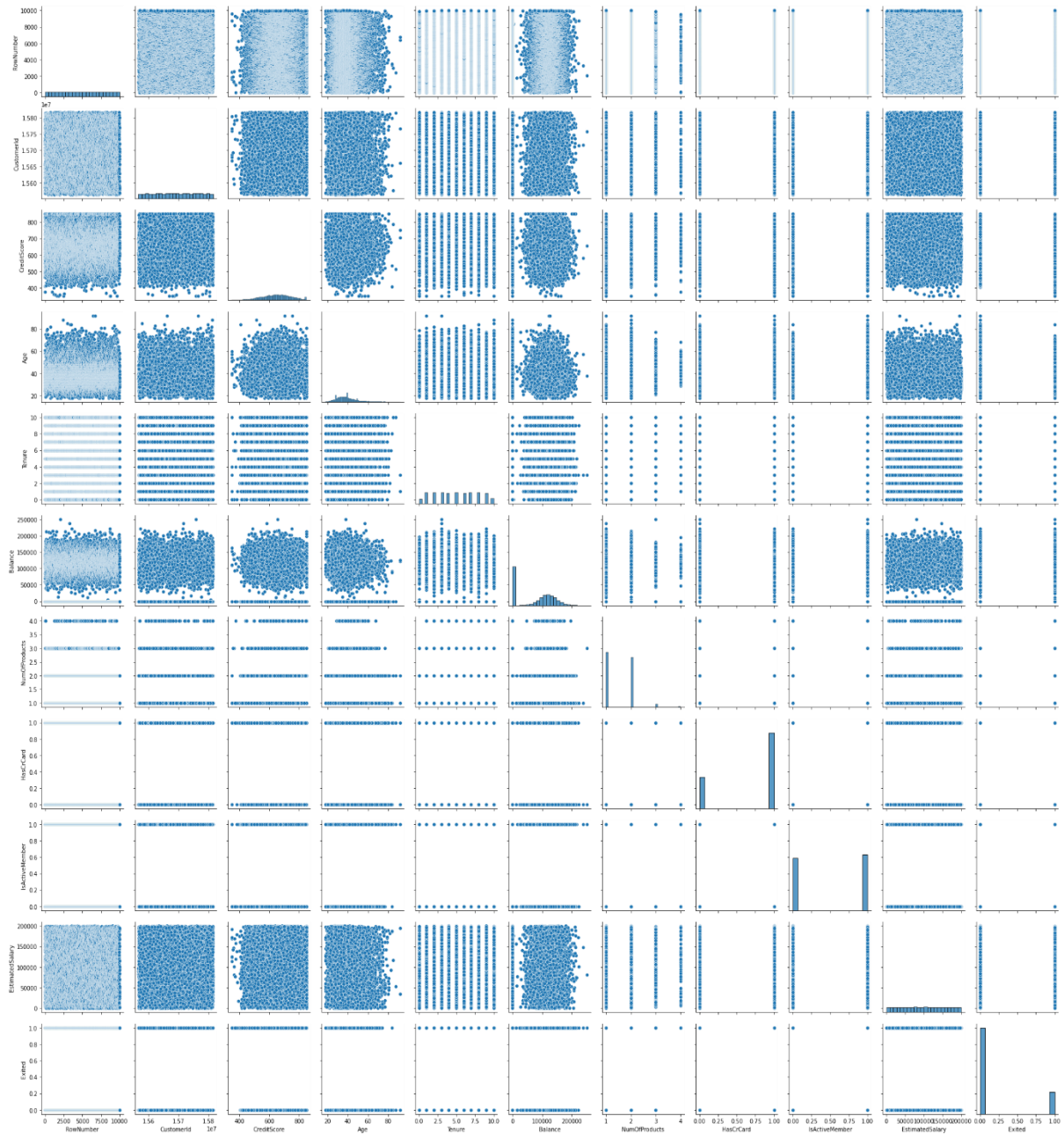

```
1  sns.histplot(df.Gender,kde=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1a79d1c490>
```



b. Bivariate Analysis

```
1  sns.pairplot(df)
```

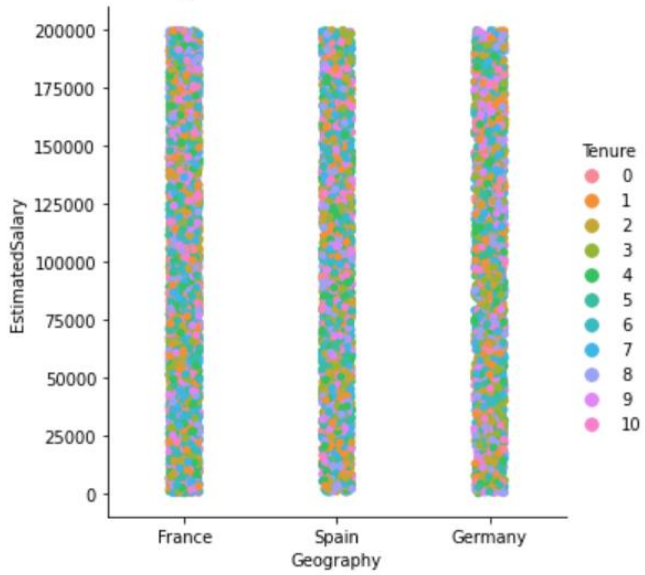


*Diagonal Plots are Univariate Plots, rest are Bivariate

c. Multivariate Analysis

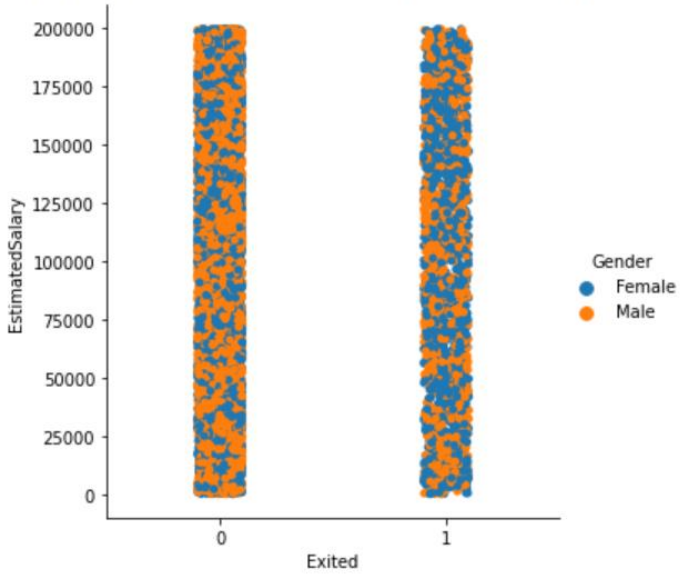
```
[ ]  1  sns.catplot(x='Geography', y='EstimatedSalary', hue='Tenure', data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f6d379add90>
```

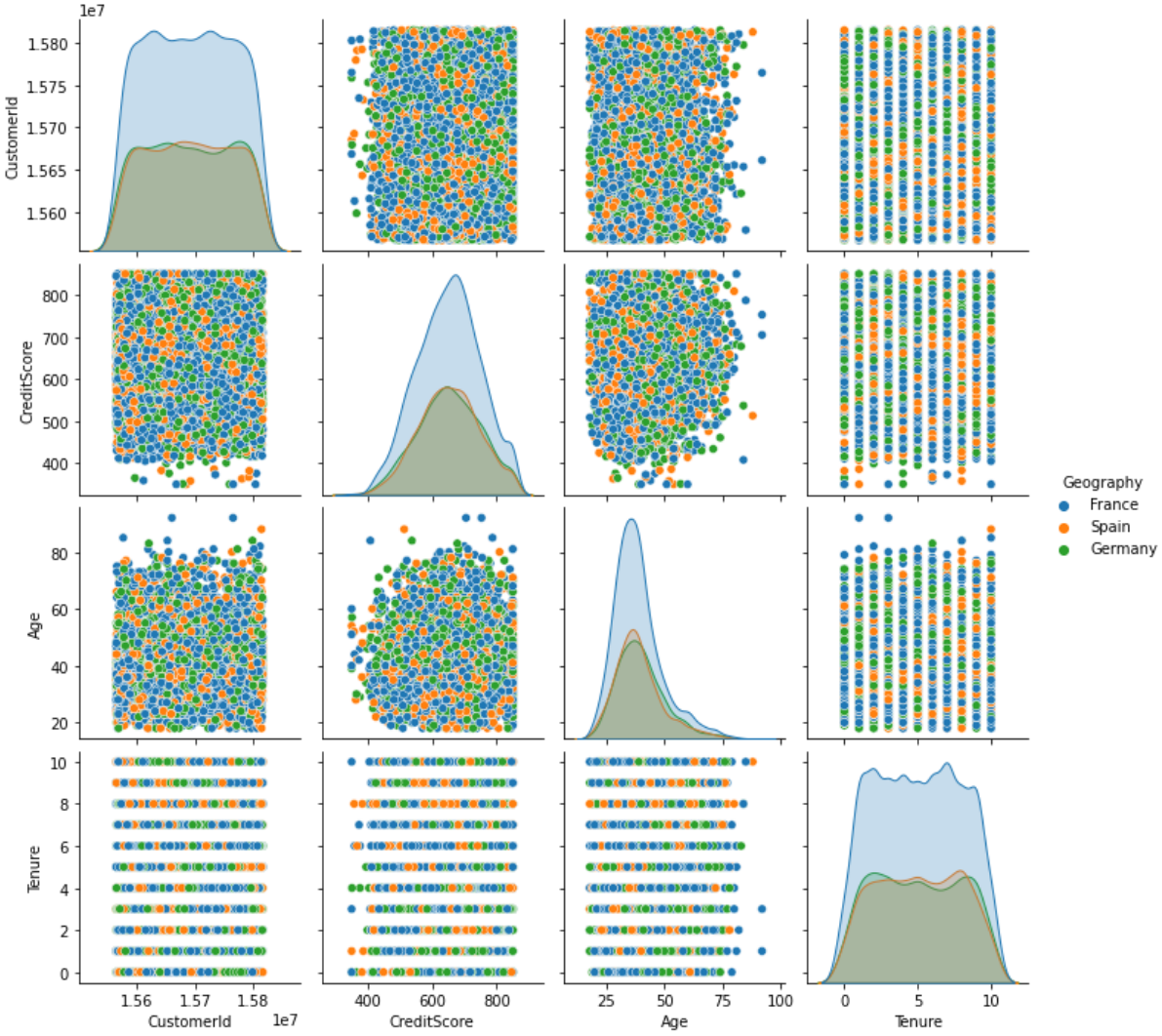



```
[ ]  1  sns.catplot(x='Exited', y='EstimatedSalary', hue='Gender', data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f6d36f87a90>
```




```
[ ]  1  sns.pairplot(df[['CustomerId', 'CreditScore', 'Geography','Gender', 'Age', 'Tenure']],hue='Geography')
```



5. Descriptive Statistics on Dataset

```
[ ]   1  df.describe()
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Estimatec
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.

6. Handling Missing Values

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

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

```
1  df.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
```

```
1  df.isnull()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	E
0	False	False	False	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	
9995	False	False	False	False	False	False	False	False	False	False	False	False	
9996	False	False	False	False	False	False	False	False	False	False	False	False	
9997	False	False	False	False	False	False	False	False	False	False	False	False	
9998	False	False	False	False	False	False	False	False	False	False	False	False	
9999	False	False	False	False	False	False	False	False	False	False	False	False	

10000 rows × 14 columns

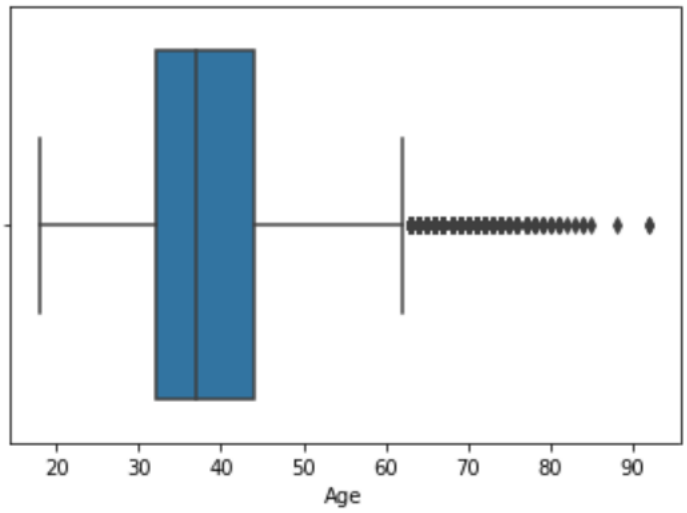
7. Handling Outliers

```
1  df.skew()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions
  """Entry point for launching an IPython kernel.
RowNumber          0.000000
CustomerId         0.001149
CreditScore       -0.071607
Age                1.011320
Tenure             0.010991
Balance           -0.141109
NumOfProducts      0.745568
HasCrCard         -0.901812
IsActiveMember    -0.060437
EstimatedSalary    0.002085
Exited             1.471611
dtype: float64
```

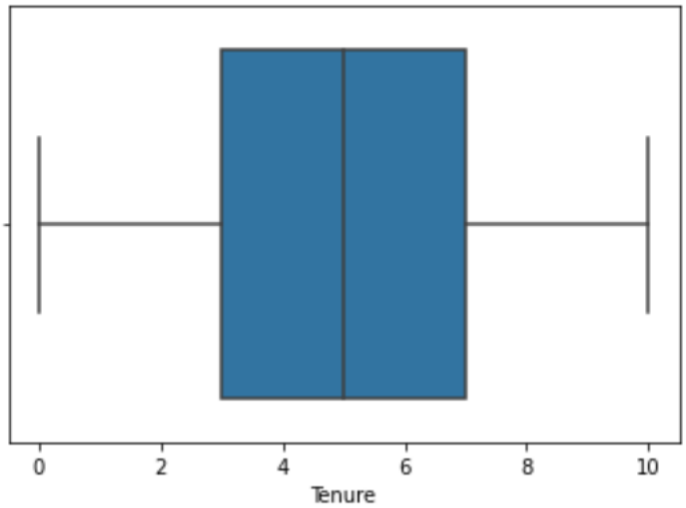
```
1  sns.boxplot(x=df['Age'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6d34ae7250>
```



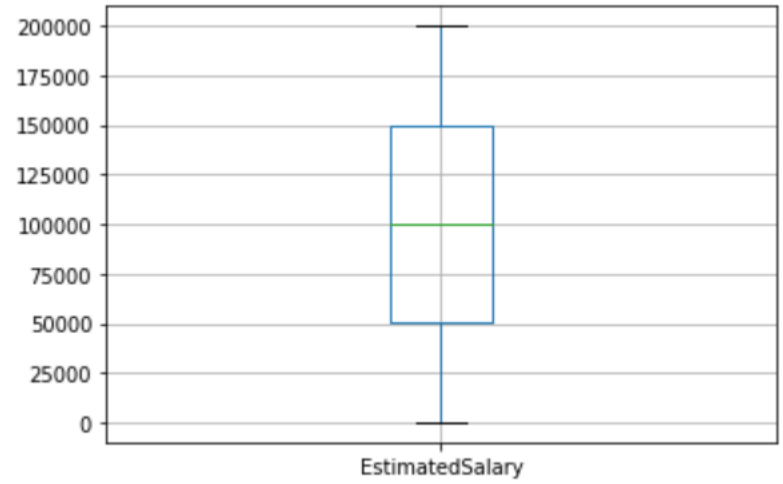

```
1  sns.boxplot(x=df['Tenure'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6d34a6e950>
```



```
df.boxplot(column="EstimatedSalary")
```

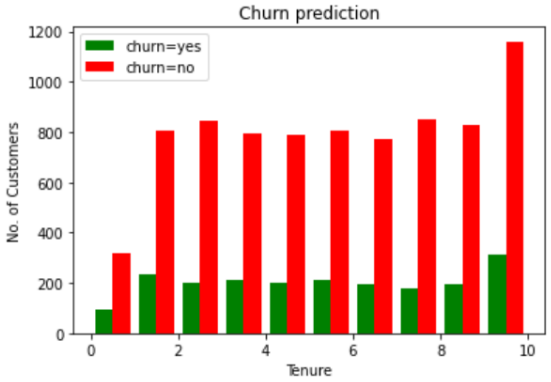
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6d34a566d0>
```




```
churn_yes=df[df.Exited==1].Tenure
churn_no=df[df.Exited==0].Tenure
```

```
plt.xlabel("Tenure")
plt.ylabel("No. of Customers")
plt.title('Churn prediction')
plt.hist([churn_yes,churn_no],color=["green","red"],label=["churn=yes","churn=no"])
plt.legend()
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/numpy/core/fromnumeric.py:3208: VisibleDeprecationWarning: Cre
  return asarray(a).size
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecationWarning:
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
```

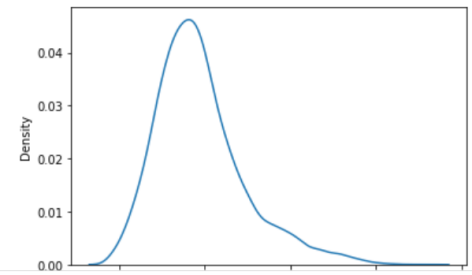



```
df.describe()
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Estimatec
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.

```
sns.kdeplot(df["Age"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6d34984750>
```




```
q1 = df["Age"].describe()["25%"]
q1
```

```
32.0
```

```
q3 = df["Age"].describe()["75%"]
q3
```

```
44.0
```

```
iqr = q3-q1
iqr
```

```
12.0
```

```
l_b = q1 -(1.5*iqr)
u_b = q3 + (1.5*iqr)
```

```
l_b
```

```
14.0
```

```
u_b
```

```
62.0
```

```
df[df["Age"]<l_b]
```

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited

```
df[df["Age"]>u_b]
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
58	59	15623944	T'ien	511	Spain	Female	66	4	0.00	1	1	0	1643.11	1
85	86	15805254	Ndukaku	652	Spain	Female	75	10	0.00	2	1	1	114675.75	0
104	105	15804919	Dunbabin	670	Spain	Female	65	1	0.00	1	1	1	177655.68	1
158	159	15589975	Maclean	646	France	Female	73	6	97259.25	1	0	1	104719.66	0
181	182	15789669	Hsia	510	France	Male	65	2	0.00	2	1	1	48071.61	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9753	9754	15705174	Chiedozie	656	Germany	Male	68	7	153545.11	1	1	1	186574.68	0
9765	9766	15777067	Thomas	445	France	Male	64	2	136770.67	1	0	1	43678.06	0
9832	9833	15814690	Chukwujekwu	595	Germany	Female	64	2	105736.32	1	1	1	89935.73	1
9894	9895	15704795	Vagin	521	France	Female	77	6	0.00	2	1	1	49054.10	0
9936	9937	15653037	Parks	609	France	Male	77	1	0.00	1	0	1	18708.76	0

359 rows × 14 columns

```
outlier_list = list(df[df["Age"] > u_b]["Age"])
```

```
outlier_list = list(df[df["Age"] > u_b]["Age"])
```

```
outlier_list
```

```
71,
66,
70,
63,
64,
65,
63,
67,
71,
67,
65,
66,
63,
73,
66,
64,
72,
71,
69,
67,
64,
81,
73,
63,
```

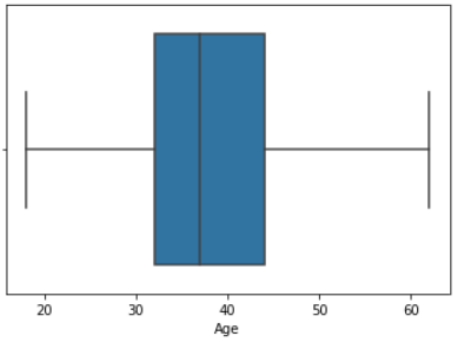
```
outlier_dict = {}.fromkeys(outlier_list,u_b)
outlier_dict
```

```
{66: 62.0,
 75: 62.0,
 65: 62.0,
 73: 62.0,
 72: 62.0,
 67: 62.0,
 79: 62.0,
 80: 62.0,
 68: 62.0,
 70: 62.0,
 63: 62.0,
 64: 62.0,
 82: 62.0,
 69: 62.0,
 74: 62.0,
 71: 62.0,
 76: 62.0,
 77: 62.0,
 88: 62.0,
 85: 62.0,
 84: 62.0,
 78: 62.0,
 81: 62.0,
 92: 62.0,
 83: 62.0}
```

```
df["Age"] = df["Age"].replace(outlier_dict)
```

```
sns.boxplot(df["Age"])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarn
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f1a6f384810>
```




```
df[df["Age"]>u_b]
```

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure

8.Check for Categorical columns and perform encoding

```
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	1	15634602	Hargrave	619	France	Female	42.0	2	0.00	1	1	1
1	2	15647311	Hill	608	Spain	Female	41.0	1	83807.86	1	0	1
2	3	15619304	Onio	502	France	Female	42.0	8	159660.80	3	1	0
3	4	15701354	Boni	699	France	Female	39.0	1	0.00	2	0	0
4	5	15737888	Mitchell	850	Spain	Female	43.0	2	125510.82	1	1	1

```
[54]  df["Geography"].unique()
```

```
array(['France', 'Spain', 'Germany'], dtype=object)
```

```
[55]  df["Gender"].unique()
```

```
array(['Female', 'Male'], dtype=object)
```

```
[56]  from sklearn.compose import ColumnTransformer
```

```
[57]  from sklearn.preprocessing import OneHotEncoder
```

```
[58]  ct=ColumnTransformer([("oh",OneHotEncoder(),[1,2])],remainder="passthrough")
```

9. Split the data into dependent and independent variables

```
x=df.iloc[:,3:13].values
```

```
x.shape
```

```
(10000, 10)
```

```
y=df.iloc[:,13:14].values
```

```
y.shape
```

```
(10000, 1)
```

```
x=ct.fit_transform(x)
```

```
x.shape
```

```
(10000, 13)
```

```
x
```

```
array([[1.0, 0.0, 0.0, ..., 1, 1, 101348.88],
       [0.0, 0.0, 1.0, ..., 0, 1, 112542.58],
       [1.0, 0.0, 0.0, ..., 1, 0, 113931.57],
       ...,
       [1.0, 0.0, 0.0, ..., 0, 1, 42085.58],
       [0.0, 1.0, 0.0, ..., 1, 0, 92888.52],
       [1.0, 0.0, 0.0, ..., 1, 0, 38190.78]], dtype=object)
```

```
import joblib
joblib.dump(ct,"churnct.pkl")
```

```
['churnct.pkl']
```

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)
```

+ Code + Text

```
x_train.shape
```

```
(8000, 13)
```

```
x_test.shape
```

```
(2000, 13)
```

11.Scale the independent variables

```
from sklearn.preprocessing import StandardScaler
```

+ Code + Text

```
sc=StandardScaler()
```

+ Code + Text

```
x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)
```

```
joblib.dump(sc,"churnsc.pkl")
```

```
['churnsc.pkl']
```