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

```
from google.colab import drive
drive.mount('/content/drive')
```

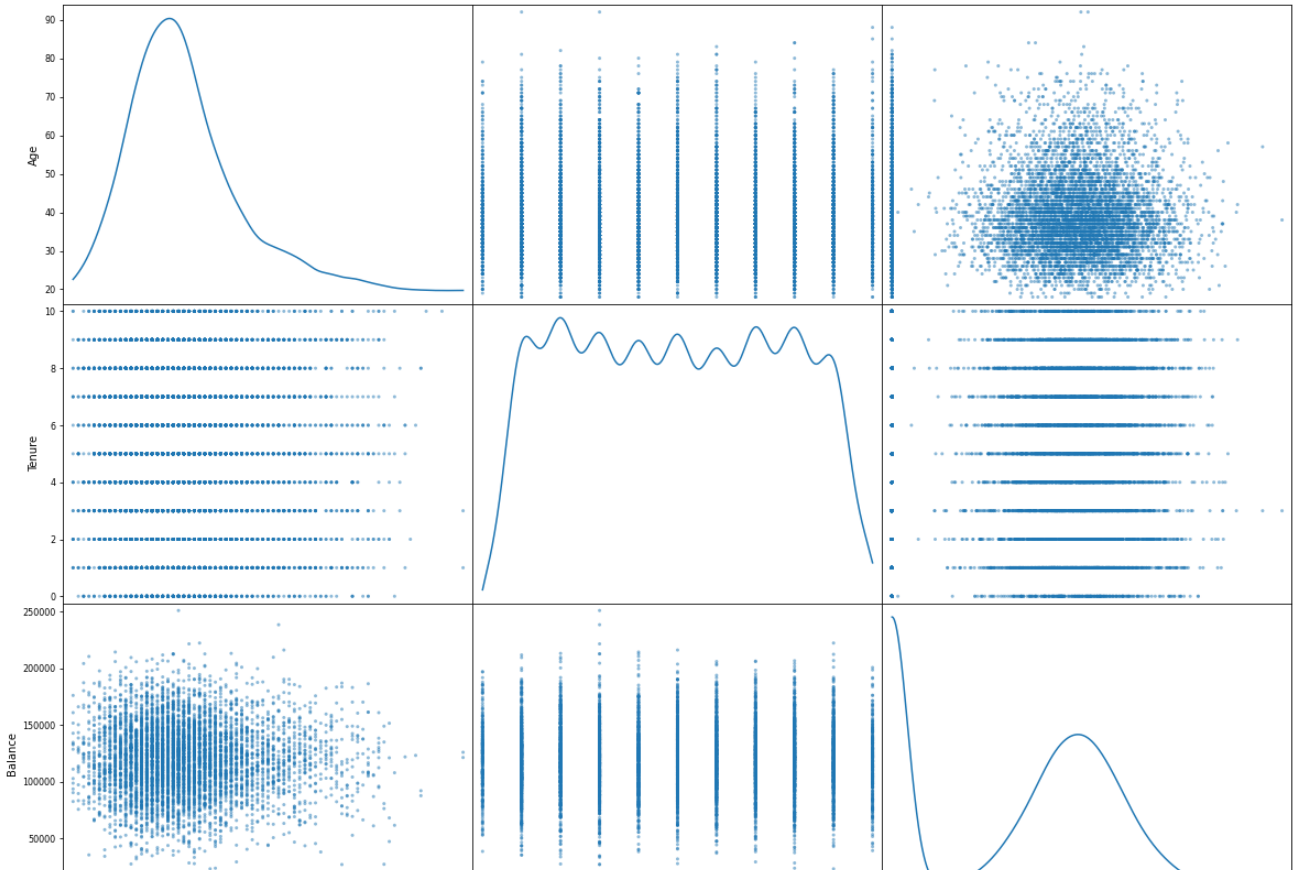
```
Mounted at /content/drive
```

```
data = pd.read_csv("/Churn_Modelling (1).csv")
```

```
data.head() #univariate analysis
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Ba
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	838
2	3	15619304	Onio	502	France	Female	42	8	1596
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	1255

```
pd.plotting.scatter_matrix(data.loc[:, "Age":"Balance"], diagonal="kde",figsize=(20,15))
plt.show() #multivariate analysis
```




```
data.mean() #discriptive analysis
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Droppi
  """Entry point for launching an IPython kernel.
RowNumber          5.000500e+03
CustomerId         1.569094e+07
CreditScore        6.505288e+02
Age                3.892180e+01
Tenure             5.012800e+00
Balance            7.648589e+04
NumOfProducts      1.530200e+00
HasCrCard          7.055000e-01
IsActiveMember     5.151000e-01
EstimatedSalary    1.000902e+05
Exited             2.037000e-01
dtype: float64
```

```
data.isnull().sum() #missing values
```

```
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
```

```
import seaborn as sns
```

```
q = data.quantile(q=[0.25,0.75]) #0utlier detection using upper and lower extreme
q
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts
0.25	2500.75	15628528.25	584.0	32.0	3.0	0.00	1.0
0.75	7500.25	15753233.75	718.0	44.0	7.0	127644.24	2.0

```
IQR = q.loc[0.75]-q.loc[0.25]
IQR
```

```
RowNumber            4999.5000
CustomerId         124705.5000
CreditScore           134.0000
Age                    12.0000
Tenure                  4.0000
Balance            127644.2400
NumOfProducts           1.0000
HasCrCard               1.0000
IsActiveMember          1.0000
EstimatedSalary     98386.1375
Exited                  0.0000
dtype: float64
```

```
upper_ex = q.loc[0.75]+1.5*IQR
upper_ex
```

```
RowNumber          1.499950e+04
CustomerId         1.594029e+07
CreditScore        9.190000e+02
Age                6.200000e+01
Tenure             1.300000e+01
Balance            3.191106e+05
NumOfProducts      3.500000e+00
HasCrCard          2.500000e+00
IsActiveMember     2.500000e+00
EstimatedSalary    2.969675e+05
Exited             0.000000e+00
dtype: float64
```

```
lower_ex = q.loc[0.25]-1.5*IQR
lower_ex
```

```
RowNumber         -4.998500e+03
CustomerId         1.544147e+07
CreditScore        3.830000e+02
Age                1.400000e+01
Tenure            -3.000000e+00
Balance           -1.914664e+05
NumOfProducts     -5.000000e-01
HasCrCard         -1.500000e+00
IsActiveMember    -1.500000e+00
EstimatedSalary   -9.657710e+04
Exited             0.000000e+00
dtype: float64
```

```
data[data['Age']>62]
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Te
58	59	15623944	T'ien	511	Spain	Female	66	
85	86	15805254	Ndukaku	652	Spain	Female	75	
104	105	15804919	Dunbabin	670	Spain	Female	65	
158	159	15589975	Maclean	646	France	Female	73	
181	182	15789669	Hsia	510	France	Male	65	
...	...	...	...	...	...	...	...	
9753	9754	15705174	Chiedozie	656	Germany	Male	68	
9765	9766	15777067	Thomas	445	France	Male	64	
9832	9833	15814690	Chukwujekwu	595	Germany	Female	64	
9894	9895	15704795	Vagin	521	France	Female	77	
9936	9937	15653037	Parks	609	France	Male	77	

359 rows × 14 columns

```
data[data['Age']<14]
```

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal

```
data['Age'] = np.where(data['Age']>62,data['Age'].mean(),data['Age']) #replacing the outli
```

```
data[data['Age']>62]
```

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal

```
data.head() #encoding
```

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

```
pd.get_dummies(data,columns=['Surname']) #encoding method-1
```

	RowNumber	CustomerId	CreditScore	Geography	Gender	Age	Tenure	Balan
0	1	15634602	619	France	Female	42.0	2	0.(
1	2	15647311	608	Spain	Female	41.0	1	83807.8
2	3	15619304	502	France	Female	42.0	8	159660.8
3	4	15701354	699	France	Female	39.0	1	0.(
4	5	15737888	850	Spain	Female	43.0	2	125510.8
...	...	...	...	...	...	...	...	
9995	9996	15606229	771	France	Male	39.0	5	0.(
9996	9997	15569892	516	France	Male	35.0	10	57369.6
9997	9998	15584532	709	France	Female	36.0	7	0.(
9998	9999	15682355	772	Germany	Male	42.0	3	75075.3
9999	10000	15628319	792	France	Female	28.0	4	130142.7

10000 rows × 2945 columns

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()

data['Geography'] = le.fit_transform(data['Geography']) #method-2 (label encoding)
data.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	
0	1	15634602	Hargrave	619	0	Female	42.0	2	
1	2	15647311	Hill	608	2	Female	41.0	1	
2	3	15619304	Onio	502	0	Female	42.0	8	1
3	4	15701354	Boni	699	0	Female	39.0	1	
4	5	15737888	Mitchell	850	2	Female	43.0	2	1

```
#seperating dependent and independent variables
y = data['Exited']
x = data.iloc[:,0:14]


#splitting the data into train and test
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)


x_train.shape
```

```
(8000, 14)
```

```
x_test.shape
```

```
(2000, 14)
```