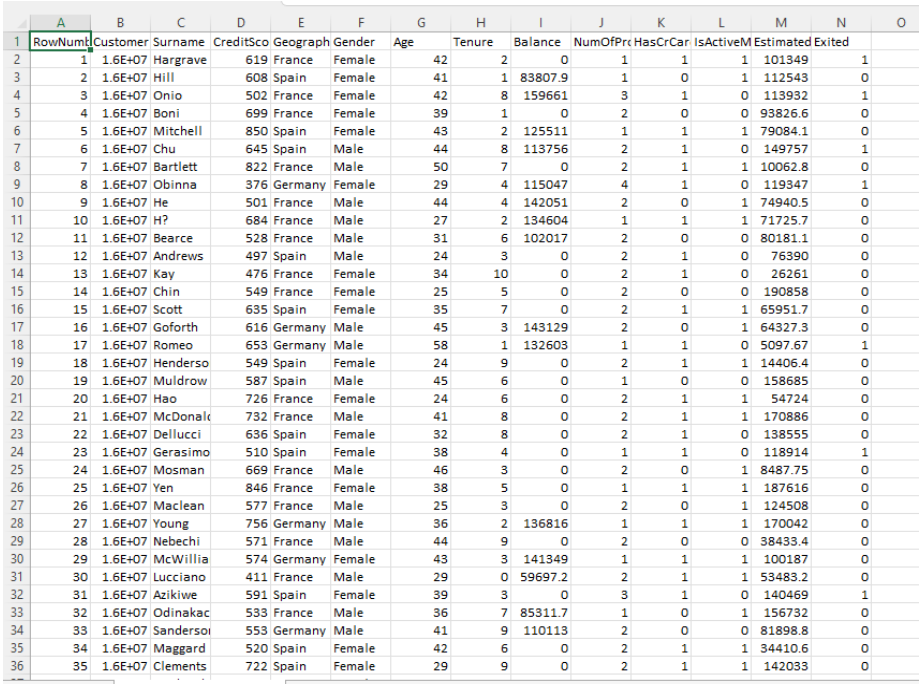
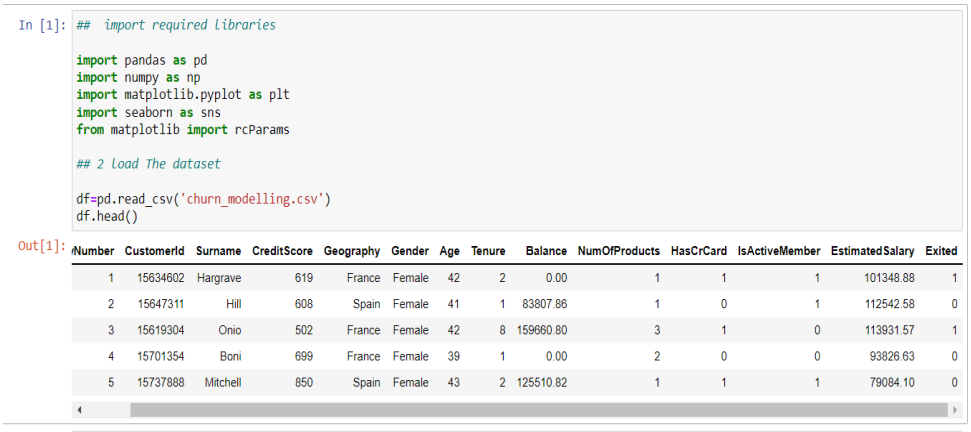
Date	26 September 2022
Team ID	PNT2022TMID38667
Project Name	Project – Early Detection of Chronic Kidney Disease using Machine Learning
Name	KEERTHANA V

1. Download the dataset: Dataset



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	RowNumb	Customer	Surname	CreditSco	Geograph	Gender	Age	Tenure	Balance	NumOfPro	HasCrCar	IsActiveM	Estimated	Exited	
2	1	1.6E+07	Hargrave	619	France	Female	42	2	0	1	1	1	101349	1	
3	2	1.6E+07	Hill	608	Spain	Female	41	1	83807.9	1	0	1	112543	0	
4	3	1.6E+07	Onio	502	France	Female	42	8	159661	3	1	0	113932	1	
5	4	1.6E+07	Boni	699	France	Female	39	1	0	2	0	0	93826.6	0	
6	5	1.6E+07	Mitchell	850	Spain	Female	43	2	125511	1	1	1	79084.1	0	
7	6	1.6E+07	Chu	645	Spain	Male	44	8	113756	2	1	0	149757	1	
8	7	1.6E+07	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0	
9	8	1.6E+07	Obinna	376	Germany	Female	29	4	115047	4	1	0	119347	1	
10	9	1.6E+07	He	501	France	Male	44	4	142051	2	0	1	74940.5	0	
11	10	1.6E+07	H?	684	France	Male	27	2	134604	1	1	1	71725.7	0	
12	11	1.6E+07	Bearce	528	France	Male	31	6	102017	2	0	0	80181.1	0	
13	12	1.6E+07	Andrews	497	Spain	Male	24	3	0	2	1	0	76390	0	
14	13	1.6E+07	Kay	476	France	Female	34	10	0	2	1	0	26261	0	
15	14	1.6E+07	Chin	549	France	Female	25	5	0	2	0	0	190858	0	
16	15	1.6E+07	Scott	635	Spain	Female	35	7	0	2	1	1	65951.7	0	
17	16	1.6E+07	Goforth	616	Germany	Male	45	3	143129	2	0	1	64327.3	0	
18	17	1.6E+07	Romeo	653	Germany	Male	58	1	132603	1	1	0	5097.67	1	
19	18	1.6E+07	Henderso	549	Spain	Female	24	9	0	2	1	1	14406.4	0	
20	19	1.6E+07	Muldrow	587	Spain	Male	45	6	0	1	0	0	158685	0	
21	20	1.6E+07	Hao	726	France	Female	24	6	0	2	1	1	54724	0	
22	21	1.6E+07	McDonal	732	France	Male	41	8	0	2	1	1	170886	0	
23	22	1.6E+07	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555	0	
24	23	1.6E+07	Gerasimo	510	Spain	Female	38	4	0	1	1	0	118914	1	
25	24	1.6E+07	Mosman	669	France	Male	46	3	0	2	0	1	8487.75	0	
26	25	1.6E+07	Yen	846	France	Female	38	5	0	1	1	1	187616	0	
27	26	1.6E+07	Maclean	577	France	Male	25	3	0	2	0	1	124508	0	
28	27	1.6E+07	Young	756	Germany	Male	36	2	136816	1	1	1	170042	0	
29	28	1.6E+07	Nebechi	571	France	Male	44	9	0	2	0	0	38433.4	0	
30	29	1.6E+07	McWillia	574	Germany	Female	43	3	141349	1	1	1	100187	0	
31	30	1.6E+07	Lucciano	411	France	Male	29	0	59697.2	2	1	1	53483.2	0	
32	31	1.6E+07	Azikiwe	591	Spain	Female	39	3	0	3	1	0	140469	1	
33	32	1.6E+07	Odinakac	533	France	Male	36	7	85311.7	1	0	1	156732	0	
34	33	1.6E+07	Sanderso	553	Germany	Male	41	9	110113	2	0	0	81898.8	0	
35	34	1.6E+07	Maggard	520	Spain	Female	42	6	0	2	1	1	34410.6	0	
36	35	1.6E+07	Clements	722	Spain	Female	29	9	0	2	1	1	142033	0	

2. Load the dataset



In [1]:
```
##  import required libraries

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

## 2 load The dataset

df=pd.read_csv('churn_modelling.csv')
df.head()
```

Out[1]:

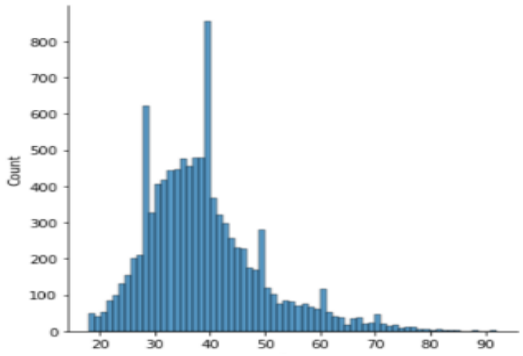
	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
1	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
2	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
3	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
4	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
5	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

3. Perform Below Visualizations.

- Univariate Analysis

```
In [3]: ##  import required libraries

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

        ## 3 univariate analysis

        df=pd.read_csv('churn_modelling.csv')
        df.head()
        sns.displot(df.Age)
```

```
Out[3]: <seaborn.axisgrid.FacetGrid at 0x131c31d2f10>
```

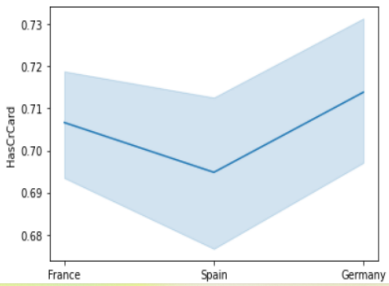


- Bi - Variate Analysis

```
In [8]: ##  import required libraries

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

        ## 3 Bi-Variate analysis

        df=pd.read_csv('churn_modelling.csv')
        df.head()
        sns.lineplot(df.Geography,df.HasCrCard)
```

```
C:\Users\91733\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword ar
gs: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

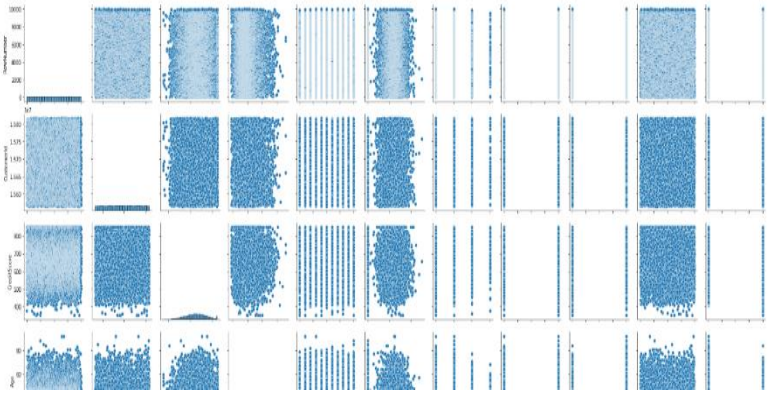
```
Out[8]: <AxesSubplot:xlabel='Geography', ylabel='HasCrCard'>
```



- Multi-variate Analysis

In [7]:
```
##  import required libraries

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

## 3 Multi-Variate analysis

df=pd.read_csv('churn_modelling.csv')
df.head()
sns.pairplot(df)
```

Out[7]: <seaborn.axisgrid.PairGrid at 0x131c33a1e80>



4. Perform descriptive statistics on the dataset.

In [1]:
```
##  import required libraries

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

## 4 Descriptive statistics

df=pd.read_csv('churn_modelling.csv')
df.head()
df.describe()
```

Out[1]:

wNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
)00.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.000000
)00.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	0.203700
386.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402769
1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	0.000000
500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	0.000000
)00.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	0.000000
500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	0.000000
)00.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	1.000000

5. Handle the Missing values

```
In [2]: ##  import required libraries

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

        ## 5 NO Missing Value

        df=pd.read_csv('churn_modelling.csv')
        df.head()
        df.isnull().any()
```

```
Out[2]: 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
```

6. Find the outliers and replace the outliers

```
In [3]: ##  import required libraries

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

        ## Find Outlier

        df=pd.read_csv('churn_modelling.csv')
        df.head()
        Q1=df.CreditScore.quantile(0.25)
        Q3=df.CreditScore.quantile(0.75)
        Q1,Q3
```

```
Out[3]: (584.0, 718.0)
```

```
In [4]: ##  import required libraries

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

        ## 6 Replace The Outlier

        df=pd.read_csv('churn_modelling.csv')
        df.head()
        Q1=df.CreditScore.quantile(0.25)
        Q3=df.CreditScore.quantile(0.75)
        Q1,Q3
        IQR=Q3-Q1
        IQR
```

```
Out[4]: 134.0
```

```
In [11]: ##  import required libraries

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

         ## 6 Replace The Outlier

         df=pd.read_csv('churn_modelling.csv')
         df.head()
         Q1=df.CreditScore.quantile(0.25)
         Q3=df.CreditScore.quantile(0.75)
         Q1,Q3
         IQR=Q3-Q1
         IQR
         lower_limit = Q1-1.5*IQR
         upper_limit = Q1+1.5*IQR
         lower_limit,upper_limit
         df_no_outlier = df[(df.CreditScore> lower_limit) & (df.CreditScore<upper_limit)]
         df_no_outlier
```

Out[11]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSa
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	10134
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	11254
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	11393
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	9382
5	6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	14975

7. Check for Categorical columns and perform encoding.

In [13]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams
from sklearn.preprocessing import LabelEncoder

## 7 Catergorical Encoder

df=pd.read_csv('churn_modelling.csv')
le=LabelEncoder()
df.Gender=le.fit_transform(df.Gender)
df.Geography=le.fit_transform(df.Geography)
df.head()
```

Out[13]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	1	15634602	Hargrave	619	0	0	42	2	0.00	1	1	1	101348.88
1	2	15647311	Hill	608	2	0	41	1	83807.86	1	0	1	112542.58
2	3	15619304	Onio	502	0	0	42	8	159660.80	3	1	0	113931.57
3	4	15701354	Boni	699	0	0	39	1	0.00	2	0	0	93826.63
4	5	15737888	Mitchell	850	2	0	43	2	125510.82	1	1	1	79084.10

In [ ]:

8. Split the data into dependent and independent variables.

In [21]:
```
## import required libraries

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

## 8 Independent Variable-X

df_main=pd.read_csv('churn_modelling.csv')
df_main.head()
x=df_main.drop(columns=['Tenure'],axis=1)
x.head()
```

Out[21]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	125510.82	1	1	1	79084.10	0

In [23]:
```
## import required libraries

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

## 8 dependent Variable-y

df_main=pd.read_csv('churn_modelling.csv')
df_main.head()
x=df_main.drop(columns=['Tenure'],axis=1)
x.head()
y=df_main.Surname
y
```

Out[23]:
```
0       Hargrave
1           Hill
2           Onio
3           Boni
4       Mitchell
          ...
9995     Obijiaku
9996    Johnstone
9997          Liu
9998    Sabbatini
9999       Walker
Name: Surname, Length: 10000, dtype: object
```

9. Scale the independent variables

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

df_main=pd.read_csv('churn_modelling.csv')
df_main.head()
x=df_main.drop(columns=['Tenure'],axis=1)
x.head()

## 9 scale the Independent Variables

x_train = pd.DataFrame(x)
x_train.head()
```

Out[2]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	125510.82	1	1	1	79084.10	0

10. Split the data into training and testing

In [8]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

## 10 Data into Training and Testing
y=df_main.Surname
y
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.25,random_state=0)
print('x_train.shape : ' ,x_train.shape)
print('y_train.shape : ' ,y_train.shape)
print('x_test.shape : ' ,x_test.shape)
print('y_test.shape : ' ,y_test.shape)
```

```
x_train.shape :  (7500, 13)
y_train.shape :  (7500,)
x_test.shape :  (2500, 13)
y_test.shape :  (2500,)
```