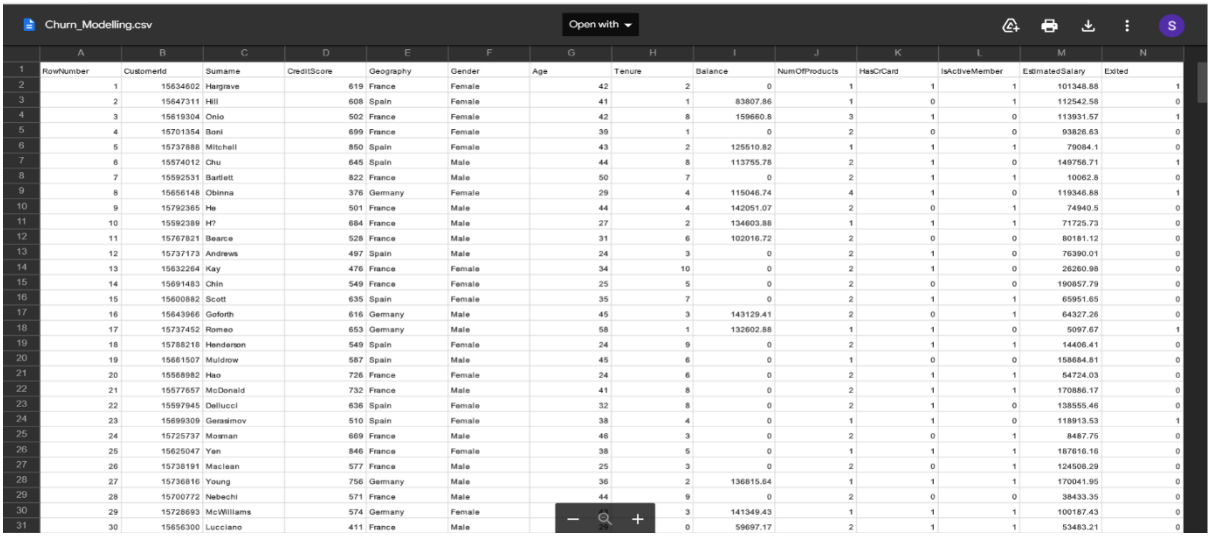
ASSIGNMENT - 2

DATE	24 September 2022
TEAM ID	PNT2022TMID38667
PROJECT NAME	Early Detection of Chronic Kidney Disease using Machine Learning
MAXIMUM MARKS	2 Marks

1. Download the dataset



RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.88	1
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.57	1
4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0
5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.1	0
6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.71	1
7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0
8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88	1
9	15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.5	0
10	15592389	H?	684	France	Male	27	2	134603.88	1	1	1	71725.73	0
11	15767821	Bearce	528	France	Male	31	6	102016.72	2	0	0	80181.12	0
12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01	0
13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.98	0
14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.79	0
15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.65	0
16	15643966	Goforth	616	Germany	Male	45	3	143129.41	2	0	1	64327.26	0
17	15737452	Romeo	653	Germany	Male	58	1	132602.88	1	1	0	5097.67	1
18	15788218	Henderson	549	Spain	Female	24	9	0	2	1	1	14406.41	0
19	15661507	Muldrow	587	Spain	Male	45	6	0	1	0	0	158684.81	0
20	15568982	Hao	726	France	Female	24	6	0	2	1	1	54724.03	0
21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886.17	0
22	15597945	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555.46	0
23	15699309	Gerasimov	510	Spain	Female	38	4	0	1	1	0	118913.53	1
24	15725737	Mosman	669	France	Male	46	3	0	2	0	1	8487.75	0
25	15625047	Yen	846	France	Female	38	5	0	1	1	1	187616.16	0
26	15738191	Maclean	577	France	Male	25	3	0	2	0	1	124508.29	0
27	15736816	Young	756	Germany	Male	36	2	136815.64	1	1	1	170041.95	0
28	15700772	Nebechi	571	France	Male	44	9	0	2	0	0	38433.35	0
29	15728693	McWilliams	574	Germany	Female	43	3	141349.43	1	1	1	100187.43	0
30	15656300	Lucciano	411	France	Male	29	0	59697.17	2	1	1	53483.21	0

2. Load the dataset

```
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

        ## 2.Loading dataset

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

Out[3]:

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

3. Perform Below Visualizations.

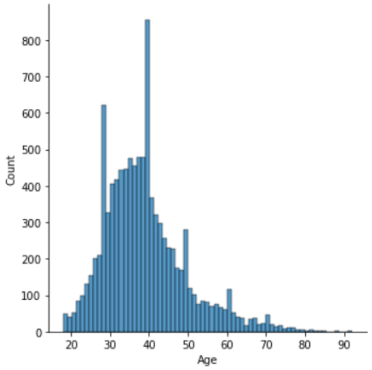
i. Univariate Analysis

In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams

## 3i.univariate analysis

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

Out[4]:
```
<seaborn.axisgrid.FacetGrid at 0x1cf733cce20>
```



ii.Bi - Variate Analysis

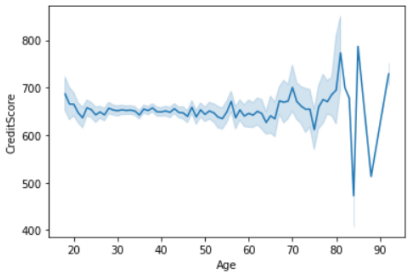
In [8]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams

## 3ii.bivariate analysis

df=pd.read_csv('Churn_Modelling.csv')
df.head()
sns.lineplot(df.Age,df.CreditScore)
```

```
C:\Users\sahan\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='Age', ylabel='CreditScore'>
```



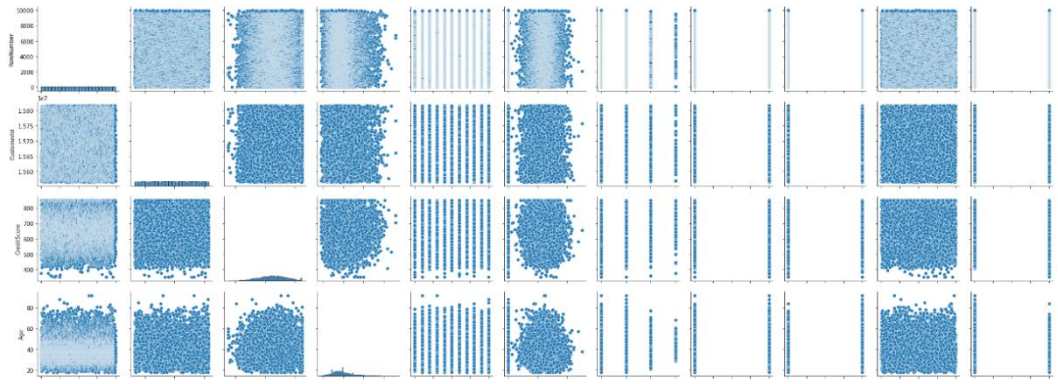
iii.Multi - Variate Analysis

In [25]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams

## 3iii.multi-variate analysis

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

Out[25]:
```
<seaborn.axisgrid.PairGrid at 0x1cf25f01070>
```



4. Perform descriptive statistics on the dataset

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams

## 4.descriptive analysis

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

Out[1]:

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	1(
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	

5. Handle the Missing values.

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams

## 5.no missing values

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 [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams

## 6.find the outlier

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

Out[4]:
```
(32.0, 44.0)
```

In [5]:
```
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.Age.quantile(0.25)
Q3=df.Age.quantile(0.75)
Q1,Q3
IQR=Q3-Q1
IQR
```

Out[5]:
```
12.0
```

In [8]:
```
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.Age.quantile(0.25)
Q3=df.Age.quantile(0.75)
Q1,Q3
IQR=Q3-Q1
IQR
lower_limit = Q1-1.5*IQR
upper_limit = Q3+1.5*IQR
lower_limit, upper_limit
df_no_outlier = df[(df.Age>lower_limit)&(df.Age<upper_limit)]
df_no_outlier
```

Out[8]:

	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
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	7908
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	9627
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	10169
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	4208

7. Check for Categorical columns and perform encoding.

In [10]:
```
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.categorical columns encoding

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

```
C:\Users\sahan\AppData\Local\Temp\ipykernel_33468\2032063202.py:13: UserWarning: Pandas doesn't allow columns to be created via
a new attribute name - see https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access
  df.Geographyr=le.fit_transform(df.Geography)
```

Out[10]:

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

8. Split the data into dependent and independent variables

In [10]:
```
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=['CreditScore'],axis=1)
X.head()
```

Out[10]:

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

In [11]:
```
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=['CreditScore'],axis=1)
X.head()
y=df_main.CreditScore
y
```

Out[11]:
```
0       619
1       608
2       502
3       699
4       850
       ...
9995    771
9996    516
9997    709
9998    772
9999    792
Name: CreditScore, Length: 10000, dtype: int64
```

9. Scale the independent variables

In [12]:
```
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.scaling

X_train = pd.DataFrame(X)
X_train.head()
```

Out[12]:

	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 [19]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

## 10.split train and test data
y=df_main.CreditScore
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,)
```