

Assignment Date	19 September 2022
Student Name	V.Anusha
Student Roll Number	962719104005
Maximum Mark	

Question –1

1. Download the dataset:

Churn_Modelling.csv														Open with					
	A	B	C	D	E	F	G	H	I	J	K	L	M						
1	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary						
2	1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101344.96						
3	2	15647211	Hill	608	Spain	Female	41	1	83607.86	1	0	1	112942.38						
4	3	15619304	Ono	502	France	Female	42	8	159660.8	3	1	0	113937.51						
5	4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63						
6	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.5						
7	6	15574012	Chiu	645	Spain	Male	44	8	113755.78	2	1	0	149756.71						
8	7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8						
9	8	15656148	Okuma	376	Germany	Female	29	4	115046.74	4	1	0	119346.86						
10	9	15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.5						
11	10	15592389	H7	684	France	Male	27	2	134603.88	1	1	1	71725.73						
12	11	15767821	Beauce	528	France	Male	31	6	102016.72	2	0	0	80181.12						
13	12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01						
14	13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.96						
15	14	15691483	Chen	549	France	Female	25	5	0	2	0	0	196937.79						
16	15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.65						
17	16	15643966	Goforth	616	Germany	Male	45	3	143129.41	2	0	1	64327.28						
18	17	15737452	Romeo	653	Germany	Male	58	1	132602.86	1	1	0	5097.87						
19	18	15788218	Henderson	549	Spain	Female	24	9	0	2	1	1	14406.45						
20	19	15661507	Muldrow	587	Spain	Male	45	6	0	1	0	0	158684.81						
21	20	15668982	Hao	726	France	Female	24	6	0	2	1	1	54724.03						
22	21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886.17						
23	22	15597945	DeLucci	636	Spain	Female	32	8	0	2	1	0	138555.46						
24	23	15699309	Gerasimov	510	Spain	Female	4	4	0	1	1	0	118913.53						
25	24	15725737	Mosman	669	France	Male	3	3	0	2	0	1	8487.75						

Question-2

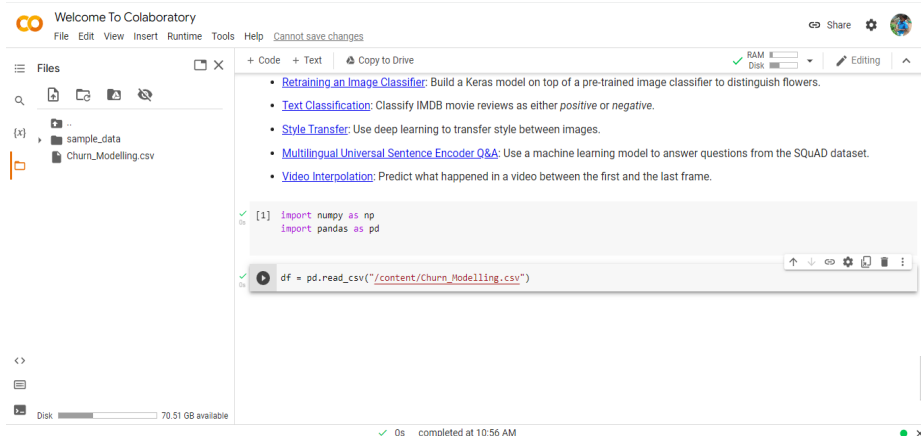
2. Load the dataset.

Solution:

```
import numpy as np
```

```
import pandas as pd
```

```
df = pd.read_csv("/content/Churn_Modelling.csv")
```



Question_3

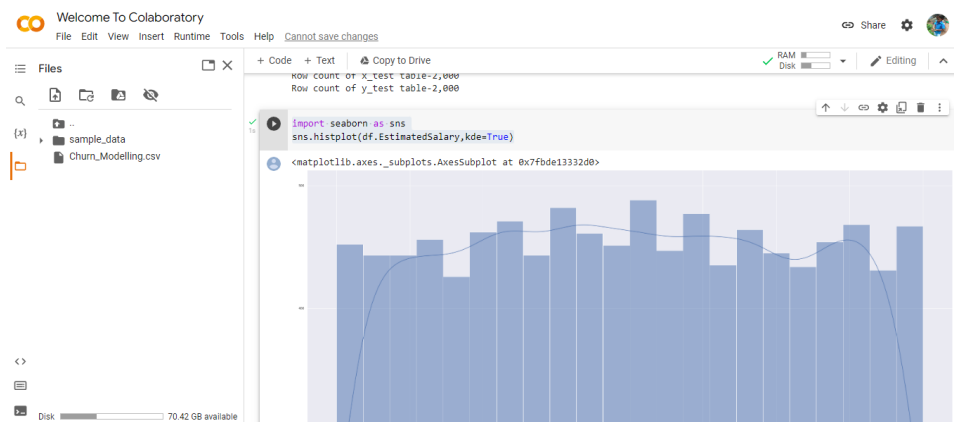
3. Perform Below Visualizations.

● Univariate Analysis

Solution:

```
import seaborn as sns
```

```
sns.histplot(df.EstimatedSalary,kde=True)
```



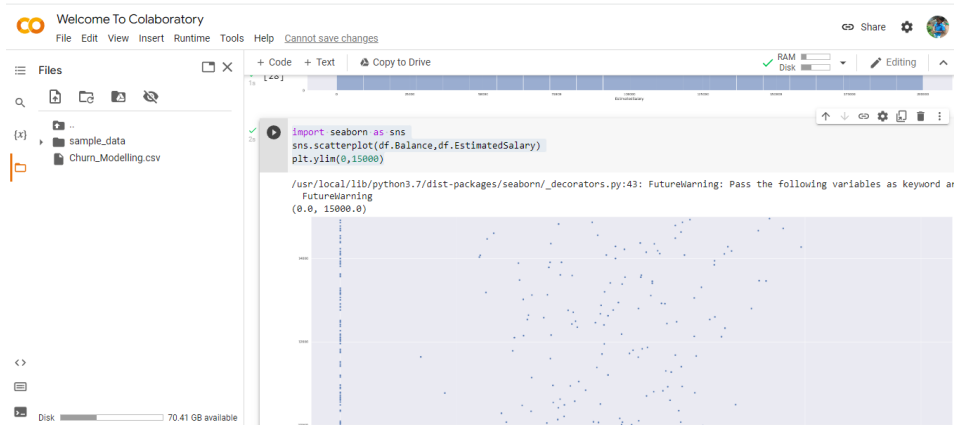
● Bi - Variate Analysis

Solution:

```
import seaborn as sns
```

```
sns.scatterplot(df.Balance,df.EstimatedSalary)
```

```
plt.ylim(0,15000)
```



• Multi - Variate Analysis

Solution:

```
import seaborn as sns
```

```
df=pd.read_csv("/content/Churn_Modelling.csv")
```

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



Question_4

4. Perform descriptive statistics on the data set

Solution:

```
df=pd.read_csv("/content/Churn_Modelling.csv")
```

```
df.describe(include='all')
```

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive RAM Disk Editing

Files

sample_data
Churn_Modelling.csv

```
df=pd.read_csv("/content/Churn_Modelling.csv")
df.describe(include='all')
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOf
count	10000.000000	1.000000e+04	10000	10000.000000	10000	10000	10000.000000	10000.000000	10000.000000	10000.000000
unique	NaN	NaN	2932	NaN	3	2	NaN	NaN	NaN	NaN
top	NaN	NaN	Smith	NaN	France	Male	NaN	NaN	NaN	NaN
freq	NaN	NaN	32	NaN	5014	5457	NaN	NaN	NaN	NaN
mean	5000.500000	1.569094e+07	NaN	650.528800	NaN	NaN	38.921800	5.012800	76485.889288	NaN
std	2886.89568	7.193619e+04	NaN	96.653299	NaN	NaN	10.487806	2.892174	62397.405202	NaN
min	1.000000	1.556570e+07	NaN	350.000000	NaN	NaN	18.000000	0.000000	0.000000	0.000000
25%	2500.750000	1.562853e+07	NaN	584.000000	NaN	NaN	32.000000	3.000000	0.000000	0.000000
50%	5000.500000	1.569074e+07	NaN	652.000000	NaN	NaN	37.000000	5.000000	97198.540000	NaN
75%	7500.250000	1.575323e+07	NaN	718.000000	NaN	NaN	44.000000	7.000000	127644.240000	NaN
max	10000.000000	1.581569e+07	NaN	850.000000	NaN	NaN	92.000000	10.000000	250898.090000	NaN

Disk 70.50 GB available

0s completed at 11:28 AM

Question_5

5. Handle the Missing values.

Solution:

```
from ast import increment_lineno
```

```
import pandas as pd
```

```
import numpy as np
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
%matplotlib inline
```

```
sns.set(color_codes=True)
```

```
df=pd.read_csv("/content/Churn_Modelling.csv")
```

```
df.head()
```

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help Cannot save changes

RAM Disk

Copy to Drive

Files

sample_data

Churn_Modelling.csv

```

from ast import Increment_lineno
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(color_codes=True)
df=pd.read_csv("/content/Churn_Modelling.csv")
df.head()

```

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

Disk 70.48 GB available

0s completed at 11:36 AM

Question_6

6. Find the outliers and replace the outliers

Solution:

```
import pandas as pd
```

```
import matplotlib
```

```
from matplotlib import pyplot as pyplot
```

```
%matplotlib inline
```

```
matplotlib.rcParams['figure.figsize']=(10,6)
```

```
df=pd.read_csv("/content/Churn_Modelling.csv")
```

```
df.sample(5)
```

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help Cannot save changes

RAM Disk

Copy to Drive

Files

sample_data

Churn_Modelling.csv

```

import pandas as pd
import matplotlib
from matplotlib import pyplot as pyplot
%matplotlib inline
matplotlib.rcParams['figure.figsize']=(10,6)
df=pd.read_csv("/content/Churn_Modelling.csv")
df.sample(5)

```

	RowNumber	CustomerId	Surname	Creditscore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
2471	2472	15595537	Trout	626	Germany	Male	49	9	171787.84	2	1	
7525	7526	15770406	Watson	580	Germany	Male	35	9	121355.19	1	0	
9560	9561	15658409	Mao	686	France	Male	41	5	128876.71	3	1	
259	260	15607178	Welch	850	Germany	Male	38	3	54901.01	1	1	
7495	7496	15589641	Sutherland	557	France	Female	27	2	0.00	2	0	

Disk 70.47 GB available

Question_7

7. Check for Categorical columns and perform encoding.

Solution:

```
df=pd.read_csv("/content/Churn_Modelling.csv")

df.columns

import pandas as pd

import numpy as np

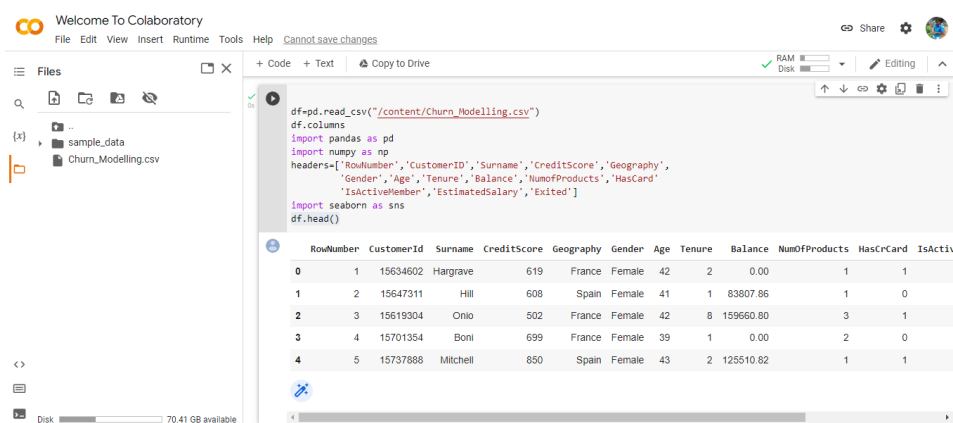
headers=['RowNumber','CustomerID','Surname','CreditScore','Geography',

         'Gender','Age','Tenure','Balance','NumofProducts','HasCard'

         'IsActiveMember','EstimatedSalary','Exited']

import seaborn as sns

df.head()
```



df=pd.read_csv("/content/Churn_Modelling.csv")

df.columns

import pandas as pd

import numpy as np

headers=['RowNumber','CustomerID','Surname','CreditScore','Geography',

'Gender','Age','Tenure','Balance','NumofProducts','HasCard'

'IsActiveMember','EstimatedSalary','Exited']

import seaborn as sns

df.head()

	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	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	

Question_8

8. Split the data into dependent and independent variables.

Solution:

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

print(x)

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

print(y)
```



Question_9

9. Scale the independent variables

Solution:

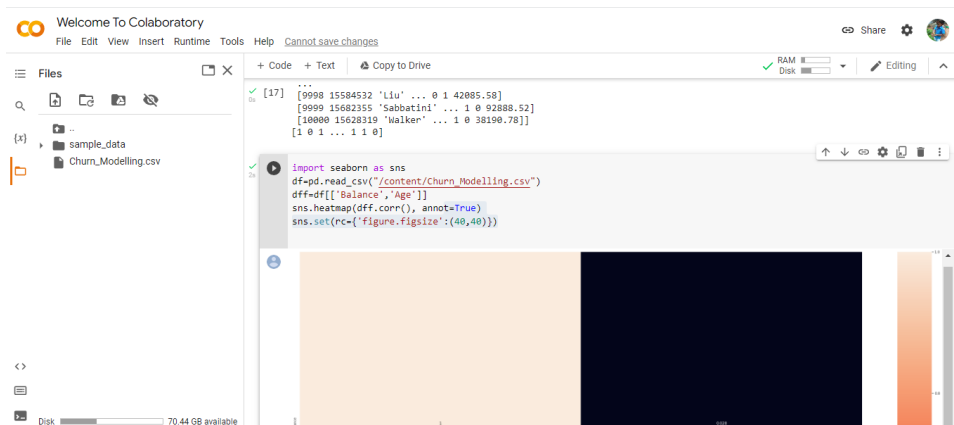
```
import seaborn as sns
```

```
df=pd.read_csv("/content/Churn_Modelling.csv")
```

```
dff=df[['Balance','Age']]
```

```
sns.heatmap(dff.corr(), annot=True)
```

```
sns.set(rc={'figure.figsize':(40,40)})
```



Question_10

10. Split the data into training and testing

Solution:

```
from scipy.sparse.construct import random
```

```
x=df.iloc[:, 1:2].values
```

```
y=df.iloc[:,2].values
```

```
from sklearn.model_selection import train_test_split
```

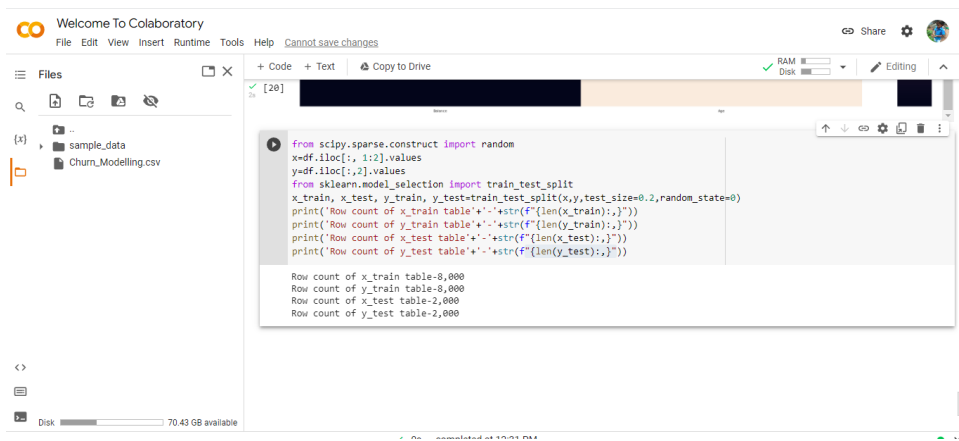
```
x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

```
print('Row count of x_train table'+ '-' +str(f"{len(x_train):,}"))
```

```
print('Row count of y_train table'+ '-' +str(f"{len(y_train):,}"))
```

```
print('Row count of x_test table'+ '-' +str(f"{len(x_test):,}"))
```

```
print('Row count of y_test table'+ '-' +str(f"{len(y_test):,}"))
```



Colaboratory interface showing the code execution. The code defines features (x) and labels (y) from a dataset, splits them into training and testing sets using `train_test_split` with a test size of 0.2 and random state 0. It then prints the row counts for each table.

```
from scipy.sparse.construct import random
x=df.iloc[:, 1:2].values
y=df.iloc[:,2].values
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)
print('Row count of x_train table'+ '-' +str(f"{len(x_train):,}"))
print('Row count of y_train table'+ '-' +str(f"{len(y_train):,}"))
print('Row count of x_test table'+ '-' +str(f"{len(x_test):,}"))
print('Row count of y_test table'+ '-' +str(f"{len(y_test):,}"))
```

Row count of x_train table-8,000
Row count of y_train table-8,000
Row count of x_test table-2,000
Row count of y_test table-2,000