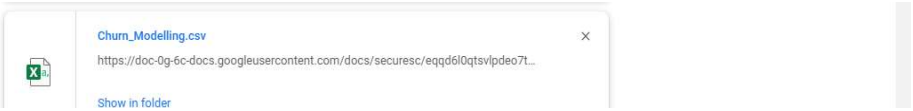
ASSIGNMENT 2

DATE	**26 SEPTEMEBR 2022.**
TEAM ID	**PNT2022TMID38674**
PROJECT NAME	**AI Based Discourse for Banking Industry**
NAME	**Logesh R (TM)**

1. Download the dataset



2. Load the dataset

```
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()
```

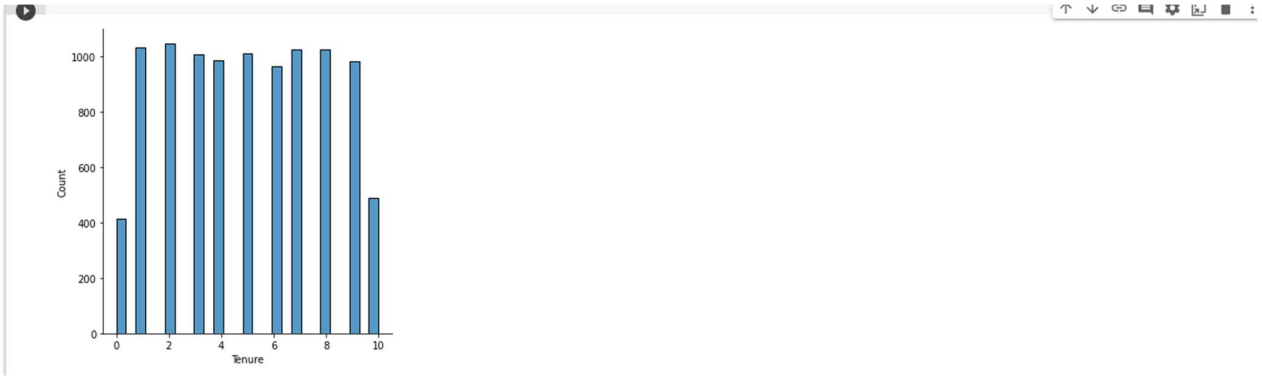
	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France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

3. perform below visualization

- Univariate Analysis

```
## 3a.univariate analysis

df=pd.read_csv('Churn_Modelling.csv')
df.head()
sns.displot(df.Tenure)
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams
```

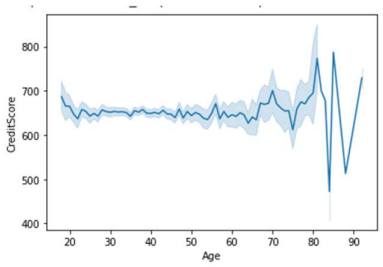


- Bi - Variate Analysis

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

    ## 3b.bivariate analysis

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

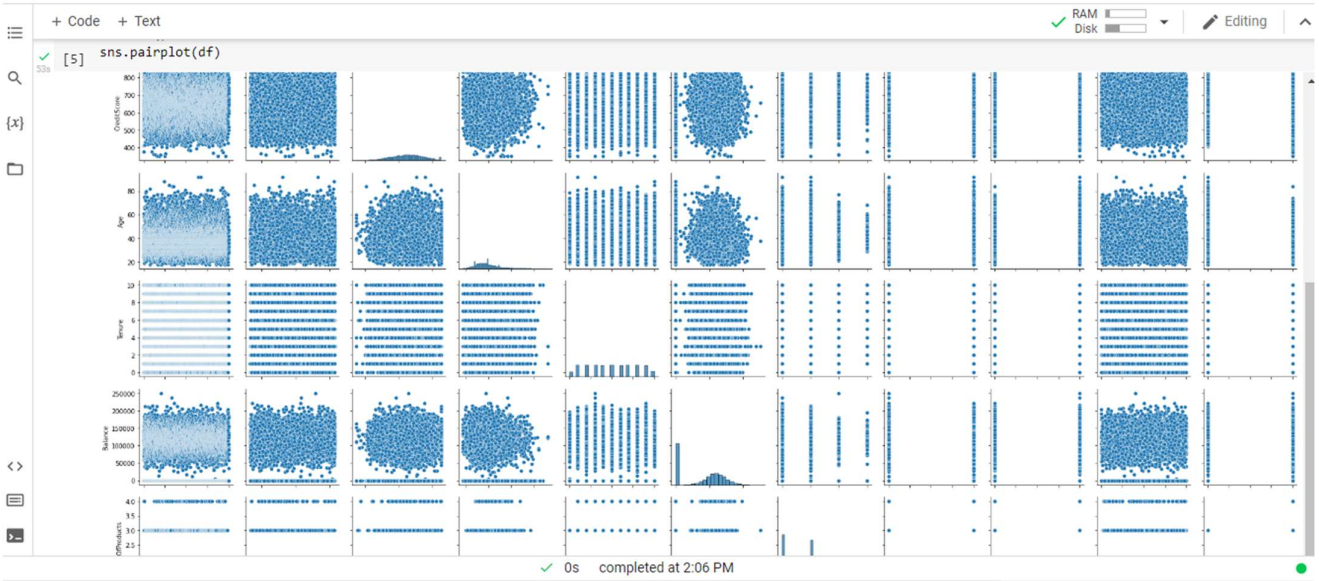


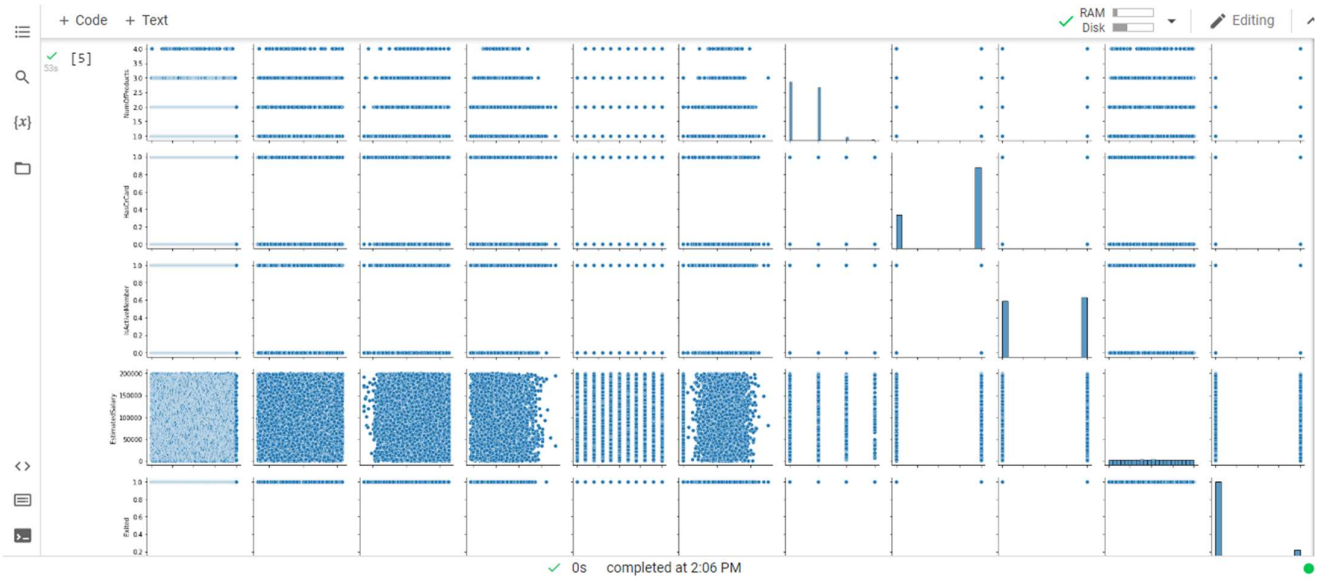
- Multi - Variate Analysis

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

    ## 3c.multi-variate analysis

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





4.Perform the descriptive statistics on the dataset

```
[ ] 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()
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	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	0.203700
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402769
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	0.000000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	0.000000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	0.000000
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	0.000000
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	1.000000

0s completed at 2:06 PM

5. Handle the missing values

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

    ## 5.handle missing values

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

6. Find the outliers and replace the outliers

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

    ## 6. outlier finding

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

    (584.0, 718.0)
```

```
[ ] 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')
```

7.Check the categorical columns and perform encoding

```
[ ] 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 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()

    /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:13: UserWarning: Pandas doesn't allow columns to be created via a new attribute name - see https://p
      del sys.path[0]
```

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

8.Split the dataset into ipdendent and dependent variables.

```
[ ] 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=['Age'],axis=1)
    X.head()
    y=df_main.Age
    y

    0       42
    1       41
    2       42
    3       39
    4       43
            ..
    9995    39
    9996    35
    9997    36
    9998    42
    9999    28
    Name: Age, Length: 10000, dtype: int64
```

9.Scale the independent variable

```
[ ] 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

    ## 9.scale the independent variable

    df_main=pd.read_csv('Churn_Modelling.csv')

    df_main.head()
    X=df_main.drop(columns=['Age'],axis=1)
    X.head()

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

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

✓ 0s completed at 2:06 PM

10.Split the data into training and testing

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

    ## 10.test and train

    y=df_main.Age
    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('X_test.shape:', X_test.shape)
    print('y_train.shape:', y_train.shape)
    print('y_test.shape:', y_test.shape)
```

```
X_train.shape: (7500, 13)
X_test.shape: (2500, 13)
y_train.shape: (7500,)
y_test.shape: (2500,)
```