

# **NALAIYA THIRAN**

## **PROJECT REPORT WEEK 3 REPORT**

**Project Title: Fertilizers Recommendation System For Disease Prediction**

**DOMAIN : ARTIFICIAL INTELLIGENCE**

**Mentor Name: Mrs. R Jeena**

**Team ID : PNT2022TMID26182**

**Team Size : 4**

**Team Leader : HARENEE A S**

**Team member-1 : RENNIE SHARON ROSE P**

**Team member-2 : BHAVANI S**

**Team member-3: JANANI PRIYA R**

**PHASE-III DESCRIPTION : EMPATHY MAP**



```

In [ ]: import pandas as pd
import seaborn as sns
import math
import matplotlib.pyplot as plt
import numpy as np
sns.set_style('darkgrid')
sns.set(font_scale=1.3)
%matplotlib inline

1.) reading data from csv

In [ ]: df=pd.read_csv('Churn_Modelling.csv')

In [ ]: df.head()

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

In [ ]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):

```

IBM-Project-3670-1658588290/ x +

github.com/IBM-EPBL/IBM-Project-3670-1658588290/blob/main/Assignment/Team%20Member-1/Assignment%202-Reenie.ipynb

```
Data columns (total 14 columns):
# Column Non-Null Count Dtype
---
0 RowNumber 10000 non-null int64
1 CustomerId 10000 non-null int64
2 Surname 10000 non-null object
3 CreditScore 10000 non-null int64
4 Geography 10000 non-null object
5 Gender 10000 non-null object
6 Age 10000 non-null int64
7 Tenure 10000 non-null int64
8 Balance 10000 non-null float64
9 NumOfProducts 10000 non-null int64
10 HasCrCard 10000 non-null int64
11 IsActiveMember 10000 non-null int64
12 EstimatedSalary 10000 non-null float64
13 Exited 10000 non-null int64
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

### 3.1)univariate analysis

```
In [ ]: #statistical analysis
df.describe()
```

```
Out[ ]:
```

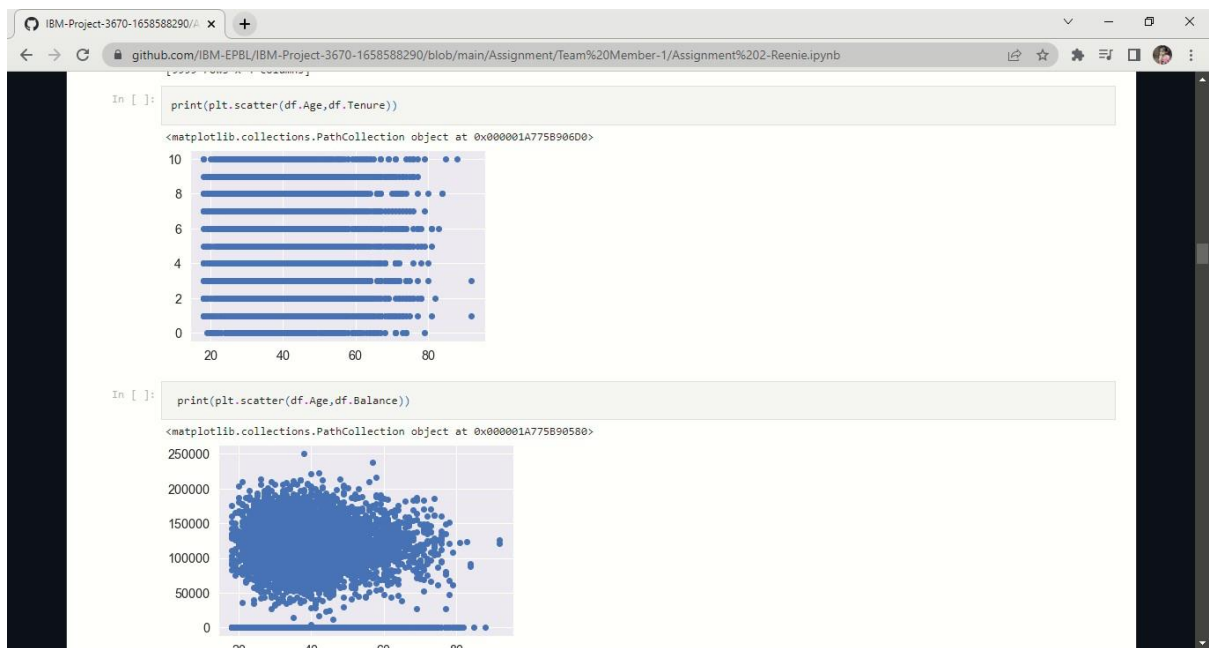
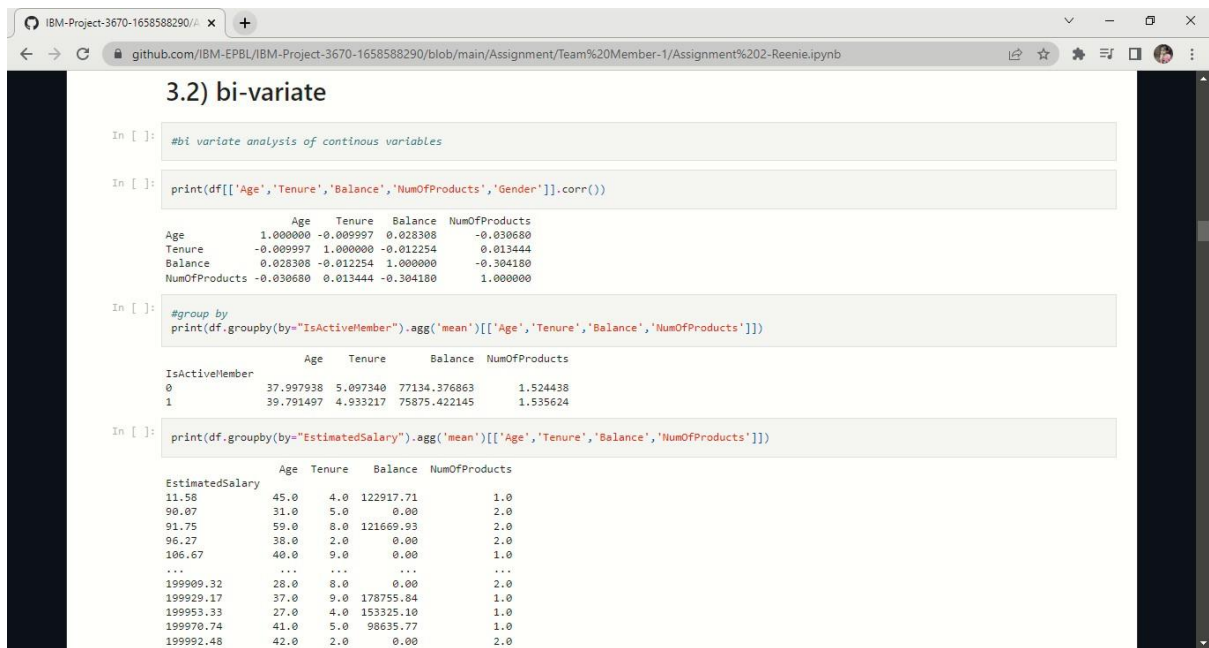
	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.705500	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	0.000000	0.000000	11.580000	0.000000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.000000	0.000000	51002.110000	0.000000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.000000	1.000000	100193.915000	0.000000

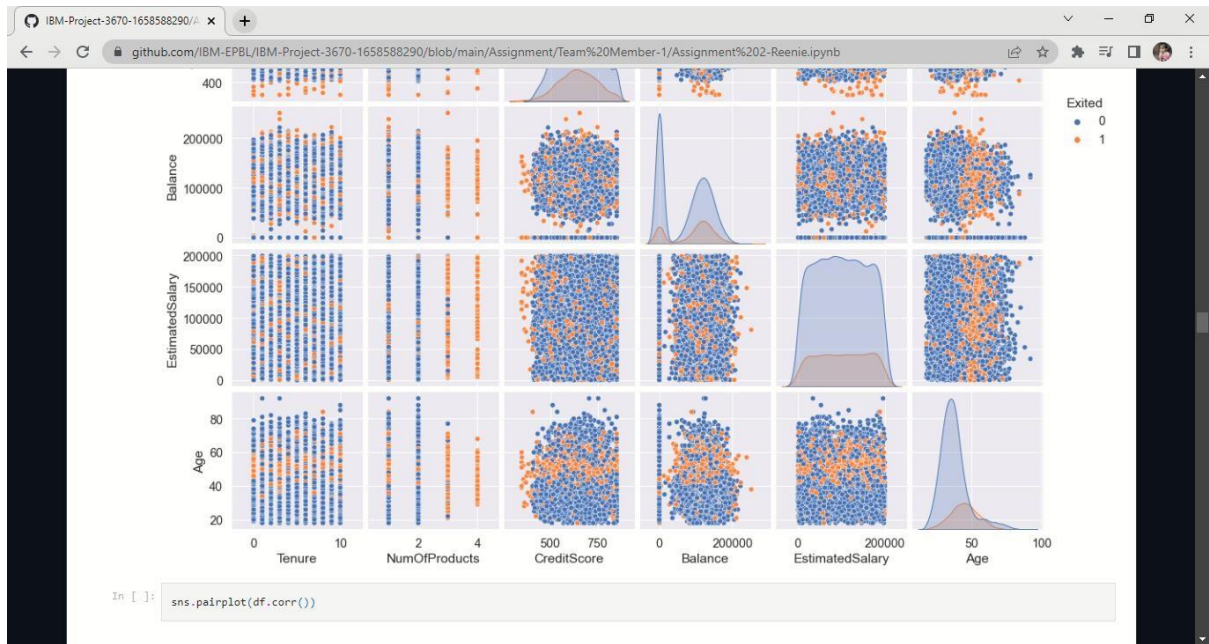
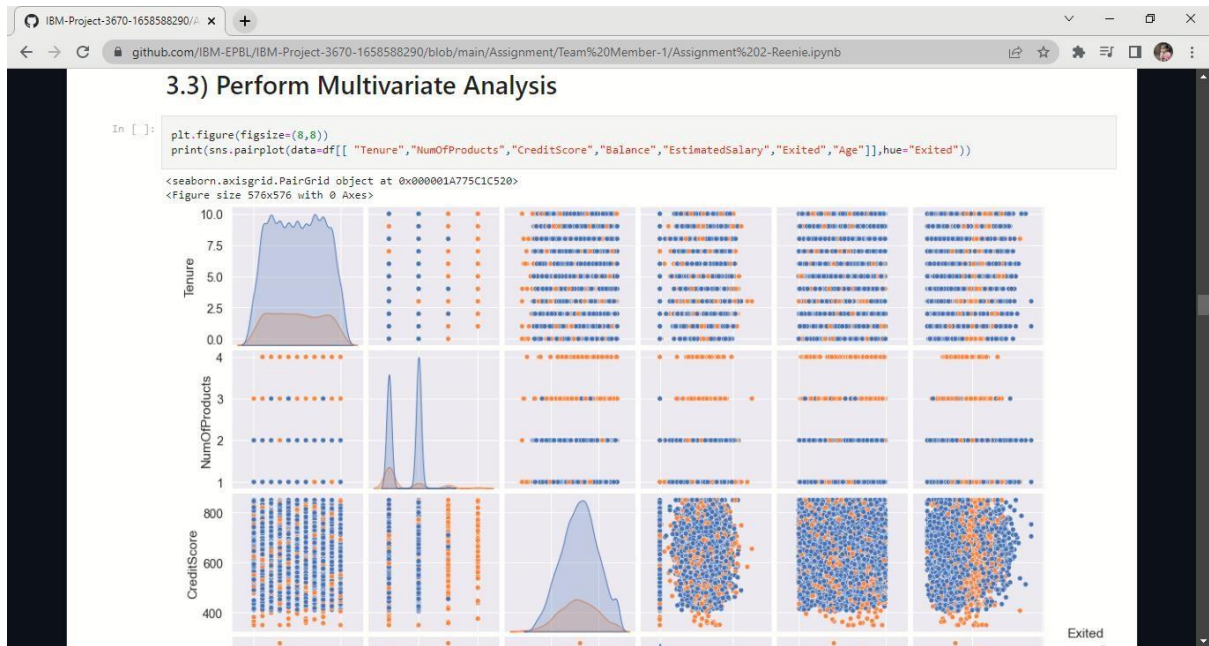
IBM-Project-3670-1658588290/ x +

github.com/IBM-EPBL/IBM-Project-3670-1658588290/blob/main/Assignment/Team%20Member-1/Assignment%202-Reenie.ipynb

```
In [ ]: plt.figure(figsize=(4,5))
sns.countplot(x='Tenure',data=df)
plt.xlabel('0:Customers with Bank, 1: exited from bank')
plt.ylabel('No.of.Customers')
plt.title("Bank Customers viz")
plt.show()
```

```
In [ ]: sns.histplot(x='NumOfProducts', data=df);
```





IBM-Project-3670-1658588290/ x +

github.com/IBM-EPBL/IBM-Project-3670-1658588290/blob/main/Assignment/Team%20Member-1/Assignment%202-Reenie.ipynb

## 4.) Descriptive Statistics

```
In [ ]: #descriptive analysis
df=pd.DataFrame(df)
print(df.sum())
```

RowNumber	50005000
CustomerId	156909405694
Surname	HargraveHillOnioBoniMitchellChuBartlettObinnaH...
CreditScore	6505288
Geography	FranceSpainFranceFranceSpainSpainFranceGermany...
Gender	FemaleFemaleFemaleFemaleFemaleMaleMaleFemaleMa...
Age	389218
Tenure	50128
Balance	764858892.88
NumOfProducts	15302
HasCrCard	7055
IsActiveMember	5151
EstimatedSalary	1000902398.81
Exited	2037
dtype:	object

```
In [ ]: print(df.mode())
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	\
0	1	15565701	Smith	850.0	France	Male	37.0	
1	2	15565706	NaN	NaN	NaN	NaN	NaN	
2	3	15565714	NaN	NaN	NaN	NaN	NaN	
3	4	15565779	NaN	NaN	NaN	NaN	NaN	
4	5	15565796	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	
9995	9996	15815628	NaN	NaN	NaN	NaN	NaN	
9996	9997	15815645	NaN	NaN	NaN	NaN	NaN	
9997	9998	15815656	NaN	NaN	NaN	NaN	NaN	
9998	9999	15815660	NaN	NaN	NaN	NaN	NaN	
9999	10000	15815690	NaN	NaN	NaN	NaN	NaN	

IBM-Project-3670-1658588290/ x +

github.com/IBM-EPBL/IBM-Project-3670-1658588290/blob/main/Assignment/Team%20Member-1/Assignment%202-Reenie.ipynb

## 5.) Rows with Missing Values

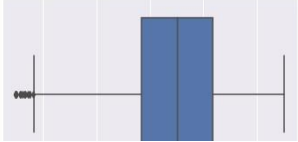
```
In [ ]: print(df.isnull().sum())
```

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

## Dispersion of data

```
In [ ]: print(sns.boxplot(x=df['CreditScore']))
```

AxesSubplot(0.125,0.125;0.775x0.755)

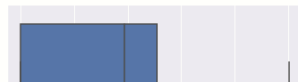
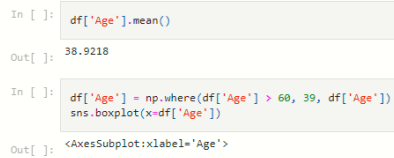




10000 rows × 14 columns

```
In [ ]: sns.boxplot(x=df['Age'])

Out[ ]: <AxesSubplot:xlabel='Age'>
```



IBM-Project-3670-1658588290/ x +

github.com/IBM-EPBL/IBM-Project-3670-1658588290/blob/main/Assignment/Team%20Member-1/Assignment%202-Reenie.ipynb

## 7.) Check for Categorical columns and perform encoding

```
In [ ]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Surname'] = le.fit_transform(df['Surname'])
df['Geography'] = le.fit_transform(df['Geography'])
df['Gender'] = le.fit_transform(df['Gender'])
df.head()
```

```
Out [ ]: RowNumber  CustomerId  Surname  CreditScore  Geography  Gender  Age  Tenure  Balance  NumOfProducts  HasCrCard  IsActiveMember  EstimatedSalary  Exited
0          1    15634602    1115         619         0         0  42         2         0.00             1             1             1    101348.88         1
1          2    15647311    1177         608         2         0  41         1    83807.86             1             0             1    112542.58         0
2          3    15619304    2040         502         0         0  42         8    76500.00             3             1             0    113931.57         1
3          4    15701354     289         699         0         0  39         1         0.00             2             0             0     93826.63         0
4          5    15737888    1822         850         2         0  43         2    76500.00             1             1             1     79084.10         0
```

```
In [ ]: from sklearn.preprocessing import OneHotEncoder
oe_style = OneHotEncoder()
oe_results = oe_style.fit_transform(df[["Surname"]])
pd.DataFrame(oe_results.toarray(), columns=oe_style.categories_).head()
```

```
Out [ ]: 0 1 2 3 4 5 6 7 8 9 ... 2922 2923 2924 2925 2926 2927 2928 2929 2930 2931
0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
3 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

IBM-Project-3670-1658588290/ x +

github.com/IBM-EPBL/IBM-Project-3670-1658588290/blob/main/Assignment/Team%20Member-1/Assignment%202-Reenie.ipynb

## 8.) Split the data into dependent and independent variables

```
In [ ]: print("-----Dependent Variables-----")
print(df.iloc[:, 0:4])
```

```
-----Dependent Variables-----
   RowNumber  CustomerId  Surname  CreditScore
0          1    15634602    1115         619
1          2    15647311    1177         608
2          3    15619304    2040         502
3          4    15701354     289         699
4          5    15737888    1822         850
...
9995      9996    15606229    1999         771
9996      9997    15569892    1336         516
9997      9998    15584532    1570         709
9998      9999    15682355    2345         772
9999     10000    15628319    2751         792
```

```
[10000 rows x 4 columns]
```

```
In [ ]: print("-----Independent Variables-----")
Y=df.iloc[:,4]
print(Y)
```

```
-----Independent Variables-----
0      0
1      2
2      0
3      0
4      2
..
9995   0
9996   0
9997   0
9998   1
9999   0
Name: Geography, Length: 10000, dtype: int32
```



IBM-Project-3670-1658588290/ x +github.com/IBM-EPBL/IBM-Project-3670-1658588290/blob/main/Assignment/Team%20Member-1/Assignment%202-Reenie.ipynb

### 9.)Scale the independent variables

```
In [ ]: from sklearn.preprocessing import StandardScaler, MinMaxScaler
object= StandardScaler()
# standardization
scaled = sc.fit_transform(x)
print(scaled)

[[-1.73187761 -0.78321342 -0.46418322 -0.32622142]
 [-1.7315312  -0.60653412 -0.3909112  -0.44003595]
 [-1.73118479 -0.99588476  0.62898807 -1.53679418]
 ...
 [ 1.73118479 -1.47928179  0.07353887  0.60498839]
 [ 1.7315312  -0.11935577  0.98943914  1.25683526]
 [ 1.73187761 -0.87855909  1.4692527  1.46377078]]
```

```
In [ ]: #Split the data into train & test
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 3, random_state = 6)
x_train
```

Out[ ]:

	RowNumber	CustomerId	Surname	CreditScore
5880	5881	15710231	1062	537
9114	9115	15605737	989	541
1060	1061	15650933	1621	490
1841	1842	15788539	924	501
5105	5106	15710465	2347	671
...	...	...	...	...
9040	9041	15653952	2567	581
8527	8528	15586931	1252	694