Darshan
UNIVERSITY
योग: कर्मसु कौशलम्

Data Mining

Project Work

137 | Vishal Baraiya | 23010101014

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

1. Read Online Retail Data Set.

```
In [2]: dt = pd.read_csv('D:\\VS_CODES\\DataMining\\ProjectDataMining\\Dataset\\BankChurner
        dt
```

Out[2]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Le
0	768805383	Existing Customer	45	M	3	High Sch
1	818770008	Existing Customer	49	F	5	Gradu
2	713982108	Existing Customer	51	M	3	Gradu
3	769911858	Existing Customer	40	F	4	High Sch
4	709106358	Existing Customer	40	M	3	Uneduca
...	...	...	...	...	...	
10122	772366833	Existing Customer	50	M	2	Gradu
10123	710638233	Attrited Customer	41	M	2	Unkno
10124	716506083	Attrited Customer	44	F	1	High Sch
10125	717406983	Attrited Customer	30	M	2	Gradu
10126	714337233	Attrited Customer	43	F	2	Gradu

10127 rows × 23 columns

2.Read First 10 Data.

In [3]:

```
dt.head(5)
```

Out[3]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level
0	768805383	Existing Customer	45	M	3	High School
1	818770008	Existing Customer	49	F	5	Graduate
2	713982108	Existing Customer	51	M	3	Graduate
3	769911858	Existing Customer	40	F	4	High School
4	709106358	Existing Customer	40	M	3	Uneducated

5 rows × 23 columns

3.Read Last 10 Data.

```
In [4]: dt.tail(5)
```

Out[4]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Le
10122	772366833	Existing Customer	50	M	2	Gradu
10123	710638233	Attrited Customer	41	M	2	Unkno
10124	716506083	Attrited Customer	44	F	1	High Sch
10125	717406983	Attrited Customer	30	M	2	Gradu
10126	714337233	Attrited Customer	43	F	2	Gradu

5 rows × 23 columns

4. Summary of statiscal data

```
In [5]: dt.describe()
```

Out[5]:

	CLIENTNUM	Customer_Age	Dependent_count	Months_on_book	Total_Relationshi
count	1.012700e+04	10127.000000	10127.000000	10127.000000	10127
mean	7.391776e+08	46.325960	2.346203	35.928409	3
std	3.690378e+07	8.016814	1.298908	7.986416	1
min	7.080821e+08	26.000000	0.000000	13.000000	1
25%	7.130368e+08	41.000000	1.000000	31.000000	3
50%	7.179264e+08	46.000000	2.000000	36.000000	4
75%	7.731435e+08	52.000000	3.000000	40.000000	5
max	8.283431e+08	73.000000	5.000000	56.000000	6

In [6]:
```
dt.describe(include='all')
```

Out[6]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_
count	1.012700e+04	10127	10127.000000	10127	10127.000000	
unique	NaN	2	NaN	2	NaN	
top	NaN	Existing Customer	NaN	F	NaN	Gra
freq	NaN	8500	NaN	5358	NaN	
mean	7.391776e+08	NaN	46.325960	NaN	2.346203	
std	3.690378e+07	NaN	8.016814	NaN	1.298908	
min	7.080821e+08	NaN	26.000000	NaN	0.000000	
25%	7.130368e+08	NaN	41.000000	NaN	1.000000	
50%	7.179264e+08	NaN	46.000000	NaN	2.000000	
75%	7.731435e+08	NaN	52.000000	NaN	3.000000	
max	8.283431e+08	NaN	73.000000	NaN	5.000000	

11 rows × 23 columns

5.Data Types of all the Columns

In [7]:
```
dt.dtypes
```

Out[7]:

```
CLIENTNUM
int64
Attrition_Flag
object
Customer_Age
int64
Gender
object
Dependent_count
int64
Education_Level
object
Marital_Status
object
Income_Category
object
Card_Category
object
Months_on_book
int64
Total_Relationship_Count
int64
Months_Inactive_12_mon
int64
Contacts_Count_12_mon
int64
Credit_Limit
float64
Total_Revolving_Bal
int64
Avg_Open_To_Buy
float64
Total_Amt_Chng_Q4_Q1
float64
Total_Trans_Amt
int64
Total_Trans_Ct
int64
Total_Ct_Chng_Q4_Q1
float64
Avg_Utilization_Ratio
float64
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependen
t_count_Education_Level_Months_Inactive_12_mon_1    float64
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependen
t_count_Education_Level_Months_Inactive_12_mon_2    float64
dtype: object
```

6. Number of Rows

In [8]:

```
dt.shape[0]
```

Out[8]:

```
10127
```

7.Number of Columns

```
In [9]:  dt.shape[1]

Out[9]:  23
```

8.Sum of Any Column

```
In [10]:  dt['Dependent_count'].sum()

Out[10]:  23760
```

9.Average Of Any Column

```
In [11]:  dt['Dependent_count'].mean()

Out[11]:  2.3462032191172115
```

10. Max in Column

```
In [12]:  dt['Dependent_count'].max()

Out[12]:  5
```

11.Min in Columns

```
In [13]:  dt['Dependent_count'].min()

Out[13]:  0
```

12.Standard deviation of column

```
In [14]:  dt['Dependent_count'].std()

Out[14]:  1.2989083489037916
```

13.location of column using iloc

```
In [15]:  dt.iloc[3]
```

Out[15]:

```
CLIENTNUM
769911858
Attrition_Flag
Existing Customer
Customer_Age
40
Gender
F
Dependent_count
4
Education_Level
High School
Marital_Status
Unknown
Income_Category
Less than $40K
Card_Category
Blue
Months_on_book
34
Total_Relationship_Count
3
Months_Inactive_12_mon
4
Contacts_Count_12_mon
1
Credit_Limit
3313.0
Total_Revolving_Bal
2517
Avg_Open_To_Buy
796.0
Total_Amt_Chng_Q4_Q1
1.405
Total_Trans_Amt
1171
Total_Trans_Ct
20
Total_Ct_Chng_Q4_Q1
2.333
Avg_Utilization_Ratio
0.76
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependen
t_count_Education_Level_Months_Inactive_12_mon_1             0.000134
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependen
t_count_Education_Level_Months_Inactive_12_mon_2              0.99987
Name: 3, dtype: object
```

14.copy

In [16]:

```
x=dt.copy()
x
```

Out[16]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Le
0	768805383	Existing Customer	45	M	3	High Sch
1	818770008	Existing Customer	49	F	5	Gradu
2	713982108	Existing Customer	51	M	3	Gradu
3	769911858	Existing Customer	40	F	4	High Sch
4	709106358	Existing Customer	40	M	3	Uneduca
...	...	...	...	...	...	
10122	772366833	Existing Customer	50	M	2	Gradu
10123	710638233	Attrited Customer	41	M	2	Unkno
10124	716506083	Attrited Customer	44	F	1	High Sch
10125	717406983	Attrited Customer	30	M	2	Gradu
10126	714337233	Attrited Customer	43	F	2	Gradu

10127 rows × 23 columns

15.Uniqu data

In [17]:

```
dt['Dependent_count'].nunique()
```

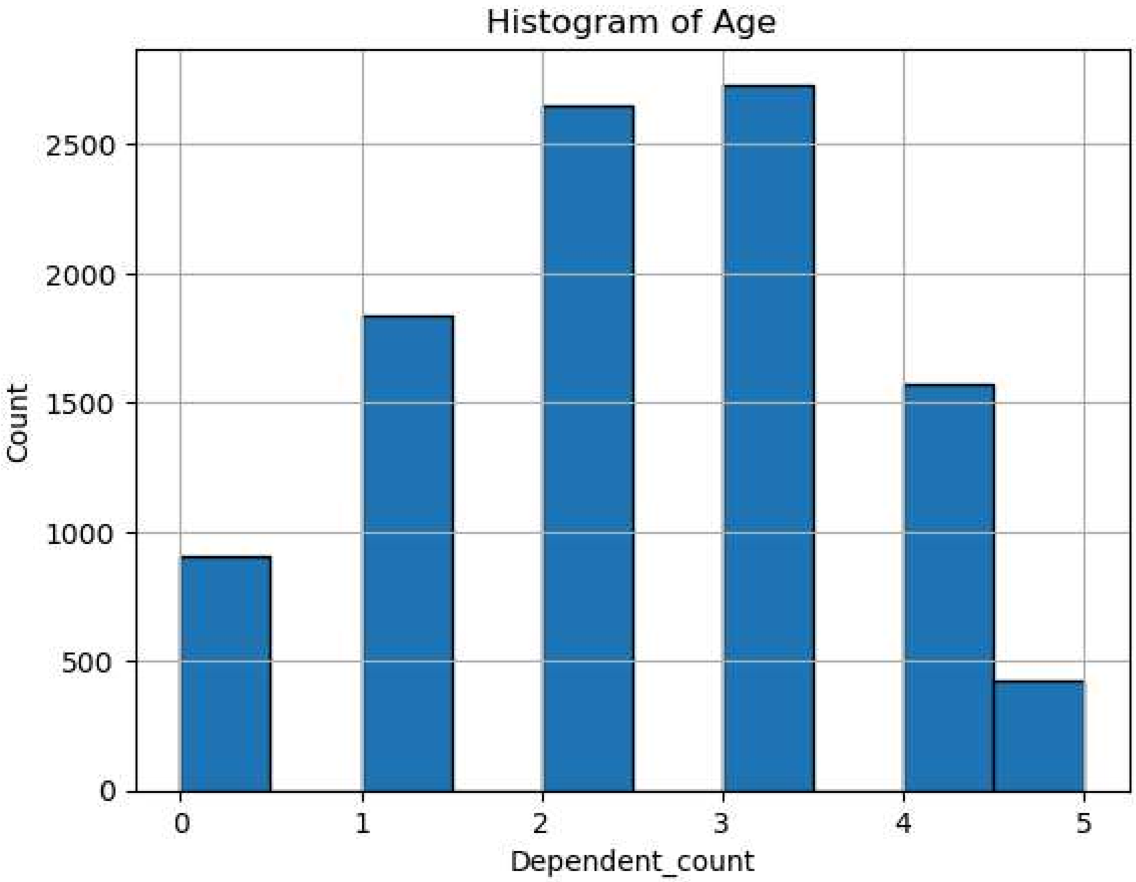
Out[17]:

```
6
```

16.Histogram of Age

In [18]:

```
dt['Dependent_count'].hist(edgecolor = 'black')
plt.title("Histogram of Age")
plt.xlabel('Dependent_count')
plt.ylabel('Count')
plt.show()
```



17.return value having 424 quantity

```
In [19]: q_counts = dt['Dependent_count'].value_counts()
         q_counts[q_counts == 424]
```

```
Out[19]: Dependent_count
         5    424
         Name: count, dtype: int64
```

19.drop

```
In [20]: dt.drop("Dependent_count", axis=1, inplace=True)
```

```
In [21]: dt
```

Out[21]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Education_Level	Marital_Status
0	768805383	Existing Customer	45	M	High School	Married
1	818770008	Existing Customer	49	F	Graduate	Single
2	713982108	Existing Customer	51	M	Graduate	Married
3	769911858	Existing Customer	40	F	High School	Unknown
4	709106358	Existing Customer	40	M	Uneducated	Married
...	...	...	...	...	...	...
10122	772366833	Existing Customer	50	M	Graduate	Single
10123	710638233	Attrited Customer	41	M	Unknown	Divorced
10124	716506083	Attrited Customer	44	F	High School	Married
10125	717406983	Attrited Customer	30	M	Graduate	Unknown
10126	714337233	Attrited Customer	43	F	Graduate	Married

10127 rows × 22 columns

18.Condition

In [22]:

```
dt[dt["Gender"] == 'M']
```

Out[22]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Education_Level	Marital_Status
0	768805383	Existing Customer	45	M	High School	Married
2	713982108	Existing Customer	51	M	Graduate	Married
4	709106358	Existing Customer	40	M	Uneducated	Married
5	713061558	Existing Customer	44	M	Graduate	Married
6	810347208	Existing Customer	51	M	Unknown	Married
...	...	...	...	...	...	...
10118	713755458	Attrited Customer	50	M	Unknown	Unknown
10120	710841183	Existing Customer	54	M	High School	Single
10122	772366833	Existing Customer	50	M	Graduate	Single
10123	710638233	Attrited Customer	41	M	Unknown	Divorced
10125	717406983	Attrited Customer	30	M	Graduate	Unknown

4769 rows × 22 columns

20.length

In [23]:

```
len(dt[dt["Gender"] == 1])
```

Out[23]:

```
0
```

21.Groupby

In [24]:

```
quant = dt.groupby('Gender')['Customer_Age'].sum()
total = quant[quant > 100]
total
```

Out[24]:

```
Gender
F    248916
M    220227
Name: Customer_Age, dtype: int64
```

22.return Index

```
In [25]: dt.index
```

```
Out[25]: RangeIndex(start=0, stop=10127, step=1)
```

23.return highest column value

```
In [26]: dt['CLIENTNUM'].value_counts().head(1)
```

```
Out[26]: CLIENTNUM
         768805383    1
         Name: count, dtype: int64
```

24.using loc

```
In [27]: dt.loc[1,'CLIENTNUM']
```

```
Out[27]: 818770008
```

25.set Index

```
In [28]: dt.set_index('CLIENTNUM')
```

Out[28]:

	Attrition_Flag	Customer_Age	Gender	Education_Level	Marital_Status	Incom
CLIENTNUM						
768805383	Existing Customer	45	M	High School	Married	
818770008	Existing Customer	49	F	Graduate	Single	Les
713982108	Existing Customer	51	M	Graduate	Married	
769911858	Existing Customer	40	F	High School	Unknown	Les
709106358	Existing Customer	40	M	Uneducated	Married	
...	...	...	...	...	...	
772366833	Existing Customer	50	M	Graduate	Single	
710638233	Attrited Customer	41	M	Unknown	Divorced	
716506083	Attrited Customer	44	F	High School	Married	Les
717406983	Attrited Customer	30	M	Graduate	Unknown	
714337233	Attrited Customer	43	F	Graduate	Married	Les

10127 rows × 21 columns

In [ ]: