

Project Statistical Methods for Decision Making: GODIGT Bank Customers Data Analysis

Context

A bank generates revenue through interest, transaction fees, and financial advice, with interest charged on customer loans being a significant source of profits. GODIGT Bank, a mid-sized private bank, offers various banking products and cross-sells asset products to existing customers through different communication methods. However, the bank is facing high credit card attrition, leading them to reevaluate their credit card policy to ensure customers receive the right card for higher spending and intent, resulting in profitable relationships.

Objective

As a Data Scientist at the company and the Data Science team has shared some data. You are supposed to find the key variables that have a vital impact on the analysis which will help the company to improve the business.

Data Description

The data contains the different data related to GODIGT bank customers.

Data Dictionary

- userid - Unique bank customer-id
- card_no - Masked credit card number
- card_bin_no - Credit card IIN number
- Issuer - Card network issuer
- card_type - Credit card type
- card_source_date - Credit card sourcing date
- high_networth - Customer category based on their net-worth value (A: High to E: Low)
- active_30 - Savings/Current/Salary etc. account activity in last 30 days
- active_60 - Savings/Current/Salary etc. account activity in last 60 days
- active_90 - Savings/Current/Salary etc. account activity in last 90 days
- cc_active30 - Credit Card activity in the last 30 days
- cc_active60 - Credit Card activity in the last 60 days
- cc_active90 - Credit Card activity in the last 90 days
- hotlist_flag - Whether card is hot-listed(Any problem noted on the card)

- widget_products - Number of convenience products customer holds (dc, cc, net-banking active, mobile banking active, wallet active, etc.)
- engagement_products - Number of investment/loan products the customer holds (FD, RD, Personal loan, auto loan)
- annual_income_at_source - Annual income recorded in the credit card application
- other_bank_cc_holding - Whether the customer holds another bank credit card
- bank_vintage - Vintage with the bank (in months) as on Tthmonth
- T+1_month_activity - Whether customer uses credit card in T+1 month (future)
- T+2_month_activity - Whether customer uses credit card in T+2 month (future)
- T+3_month_activity - Whether customer uses credit card in T+3 month (future)
- T+6_month_activity - Whether customer uses credit card in T+6 month (future)
- T+12_month_activity - Whether customer uses credit card in T+12 month (future)
- Transactor_revolver - Revolver: Customer who carries balances over from one month to the next. Transactor: Customer who pays * off their balances in full every month.
- avg_spends_l3m - Average credit card spends in last 3 months
- Occupation_at_source - Occupation recorded at the time of credit card application
- cc_limit - Current credit card limit

Importing required libraries

```
In [1]: # Import libraries for data manipulation
import numpy as np
import pandas as pd

# Import libraries for data visualization
import matplotlib.pyplot as plt
import seaborn as sns
```

Understanding the structure of data

```
In [4]: df = pd.read_excel('godigt_cc_data.xlsx')

df.head() # Returns first 5 rows
```

```
Out[4]:
```

	userid	card_no	card_bin_no	Issuer	card_type	card_source_date	high_networkh	active
0	1	4384 39XX XXXX XXXX	438439	Visa	edge	2019-09-29	B	
1	2	4377 48XX XXXX XXXX	437748	Visa	prosperity	2002-10-30	A	
2	3	4377 48XX XXXX XXXX	437748	Visa	rewards	2013-10-05	C	
3	4	4258 06XX XXXX XXXX	425806	Visa	indianoil	1999-06-01	E	
4	5	4377 48XX XXXX XXXX	437748	Visa	edge	2006-06-13	B	

5 rows × 28 columns

Number of rows and columns in the dataset

```
In [5]: df.shape # Shape of the dataset
```

```
Out[5]: (8448, 28)
```

Datatypes of the different columns in the dataset

```
In [6]: df.info() # Concise summary of dataset
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8448 entries, 0 to 8447
Data columns (total 28 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   userid                                8448 non-null   int64
1   card_no                               8448 non-null   object
2   card_bin_no                           8448 non-null   int64
3   Issuer                                8448 non-null   object
4   card_type                             8448 non-null   object
5   card_source_date                      8448 non-null   datetime64[ns]
6   high_networth                         8448 non-null   object
7   active_30                             8448 non-null   int64
8   active_60                             8448 non-null   int64
9   active_90                             8448 non-null   int64
10  cc_active30                           8448 non-null   int64
11  cc_active60                           8448 non-null   int64
12  cc_active90                           8448 non-null   int64
13  hotlist_flag                           8448 non-null   object
14  widget_products                       8448 non-null   int64
15  engagement_products                   8448 non-null   int64
16  annual_income_at_source               8448 non-null   int64
17  other_bank_cc_holding                 8448 non-null   object
18  bank_vintage                          8448 non-null   int64
19  T+1_month_activity                   8448 non-null   int64
20  T+2_month_activity                   8448 non-null   int64
21  T+3_month_activity                   8448 non-null   int64
22  T+6_month_activity                   8448 non-null   int64
23  T+12_month_activity                  8448 non-null   int64
24  Transactor_revolver                   8410 non-null   object
25  avg_spends_l3m                       8448 non-null   int64
26  Occupation_at_source                 8448 non-null   object
27  cc_limit                              8448 non-null   int64
dtypes: datetime64[ns](1), int64(19), object(8)
memory usage: 1.8+ MB

```

Statistical summary of the data

```
In [8]: df.describe().T # Summary statistics of the numerical and categorical data
```

Out[8]:

	count	mean	min	25%	50%	
userid	8448.0	4224.5	1.0	2112.75	4224.5	633
card_bin_no	8448.0	436747.044508	376916.0	426241.0	437551.0	4384
card_source_date	8448	2009-06-19 13:35:17.045454592	1998-07-24 00:00:00	2004-01-07 18:00:00	2009-05-24 12:00:00	2014-06-27 06:00:00
active_30	8448.0	0.292377	0.0	0.0	0.0	
active_60	8448.0	0.494792	0.0	0.0	0.0	
active_90	8448.0	0.642045	0.0	0.0	1.0	
cc_active30	8448.0	0.284091	0.0	0.0	0.0	
cc_active60	8448.0	0.484493	0.0	0.0	0.0	
cc_active90	8448.0	0.632339	0.0	0.0	1.0	
widget_products	8448.0	3.614583	0.0	2.0	4.0	
engagement_products	8448.0	3.991122	0.0	2.0	4.0	
annual_income_at_source	8448.0	1674594.738991	200095.0	1061104.0	1372133.5	188173
bank_vintage	8448.0	33.164181	6.0	19.0	33.0	
T+1_month_activity	8448.0	0.111269	0.0	0.0	0.0	
T+2_month_activity	8448.0	0.04794	0.0	0.0	0.0	
T+3_month_activity	8448.0	0.080374	0.0	0.0	0.0	
T+6_month_activity	8448.0	0.008878	0.0	0.0	0.0	
T+12_month_activity	8448.0	0.00947	0.0	0.0	0.0	
avg_spends_13m	8448.0	49527.36553	0.0	17110.0	37943.0	6609
cc_limit	8448.0	251706.912879	0.0	90000.0	150000.0	3500

Exploratory Data Analysis (EDA)

Univariate Analysis

Issuer

```
In [9]: # Check unique Issuer
df['Issuer'].value_counts() # Frequency of each distinct value in the Issuer column
```

```
Out[9]: Issuer
      Visa      7279
      Mastercard  728
      Amex      441
      Name: count, dtype: int64
```

Card Type

```
In [12]: # check unique Card Type
df['card_type'].value_counts() # Frequency of each distinct value in the card_type
```

```
Out[12]: card_type
      rewards      1502
      prosperity  1007
      edge        980
      chartered   923
      smartearn   765
      shoprite    688
      indianoil   680
      cashback    676
      aura        652
      gold        145
      prime       112
      pulse       101
      elite       96
      centurion   62
      platinum    59
      Name: count, dtype: int64
```

High Network

```
In [11]: # check unique High Network
df['high_network'].value_counts() # Frequency of each distinct value in the high_n
```

```
Out[11]: high_network
      A      1740
      D      1696
      E      1693
      B      1660
      C      1659
      Name: count, dtype: int64
```

Hotlist Flag

```
In [13]: # Check unique Hotlist Flag
df['hotlist_flag'].value_counts() # Frequency of each distinct value in the hotlist
```

```
Out[13]: hotlist_flag
      N      8410
      Y       38
      Name: count, dtype: int64
```

Other Bank CC Holding

```
In [14]: # Check unique Other Bank CC Holding
df['other_bank_cc_holding'].value_counts() # Frequency of each distinct value in the

Out[14]: other_bank_cc_holding
Y      4728
N      3720
Name: count, dtype: int64
```

Transactor Revolver

```
In [15]: # Check unique Transactor Revolver
df['Transactor_revolver'].value_counts() # Frequency of each distinct value in the

Out[15]: Transactor_revolver
T      7115
R      1295
Name: count, dtype: int64
```

Occupation At Source

```
In [17]: # Check unique Occupation At Source
df['Occupation_at_source'].value_counts() # Frequency of each distinct value in the

Out[17]: Occupation_at_source
Salaried      3918
Self Employed  2175
Retired       1089
Student        621
Housewife      384
0              261
Name: count, dtype: int64
```

Analyse the dataset and list down the top 5 important variables, along with the business justifications.

1. card_type

Business Justification: Customers are divided mainly based on the card type used by them. Customer Service Representatives need to know the different card types offered by the bank to suggest a right card to a customer.

2. high_networth

Business Justification: Customers can be suggested card type based on their high networth category. Customers with high net worth category can be suggested type of credit cards having maximum benefits which suits their need and for which bank can charge some fees. Customers with low net worth can be suggested type of credit cards having good benefits which suits their need and for which bank do not charge any fee. Add-on credit cards can be offered to customers as well based on their high net worth category.

3. avg_spends_l3m

Business Justification: Customers who are spending high amount on their purchases every quarter can be rewarded. These rewards can include additional cash back, points or miles. Customers can be suggested to take add-on credit cards (rewards, cashback) as well based on their spending behaviour. When customers will spend high amount on purchases than bank will get high transaction fee from merchants resulting in revenue increase. It will help bank to retain existing customers who are spending high amount on their purchases every quarter.

4. annual_income_at_source

Business Justification: Customers can be suggested credit card type based on annual income at source. Customers with high annual income at source can be suggested type of credit cards having good benefits which suits their need and for which bank can charge some fees. Add-on credit cards can be offered to customers as well based on their annual income at source.

5. other_bank_cc_holding

Business Justification: It is quite possible that customer attrition is happening due to the additional benefits provided by other banks. Customer Service Representatives can connect with these customers holding other banks credit card to know the reasons and offer them GODIGT Bank credit cards (if available) with similar benefits. They can share reasons with GODIGT Bank higher management as well. There is high probability that customer will remain with bank if they do not have any other bank credit card.

6. Transactor_revolver

Business Justification: Customers who are paying off their balances in full every month can also be rewarded. These rewards can include cash back, points or miles. It will help bank to retain existing customers.

6. cc_limit

Business Justification: Credit limit can be increased for the customers based on card type who are spending high amount on their purchases every quarter. This will help in customers to avoid overdraft in case balance amount becomes more than credit limit.
