# Bank Customer Segmentation

The following is a customer segmentation done for sample datasets of bank customers. The data consists of 1000 randomly sampled customers of the bank. The objective of the segmetation is to identify the customer's propensity to take a personal loan given income and credit card balance. Based on clustering, the sample dataset yielded 3 groups based on income and credit card balance with the following findings:

- Individuals with the highest income also has highest credit card balance. This is also the group that has a higher propensity to get a personal loan.
- There is a slightly higher probability that individuals who also has the lowest income and lowest credit card balance have more propensity to take up a personal loan compared to their next cluster.

In [422…

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

import warnings
warnings.filterwarnings("ignore")
pd.set_option('display.float_format', lambda x: '%.8f' % x)
```

## Background of Dataset

In [358…

```
df = pd.read_csv('c:\\users\\adria\\Downloads\\bank_customers.csv')
```

The dataset consists of 1000 sample customers from the whole portfolio. The attributes consists of the following:

- Location: pertains to NCR, Luzon, Visayas, Mindanao
- Income: declared annual income (in thousands)
- Education: 1=no college degree; 2=college graduate; 3=post graduate studies
- Mortgage: 0=no mortgage; 1=with mortgage
- p_loan: 0=no personal loan; 1=with personal loan
- cc: credit card balance (in thousands)

In [359…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   id         1000 non-null   int64
 1   location   1000 non-null   int64
 2   income     1000 non-null   int64
 3   education  1000 non-null   int64
```

```
 4   mortgage   1000 non-null   int64
 5   p_loan     1000 non-null   int64
 6   cc         1000 non-null   int64
dtypes: int64(7)
memory usage: 54.8 KB
```

## Exploratory Data Analysis

Checking the distribution of both Income and Credit Card Balance, customers have a mean annual income of Php2.041 and an average credit card balance of Php258K.

In [362…
```
df.iloc[:,[2,6]].describe()
```

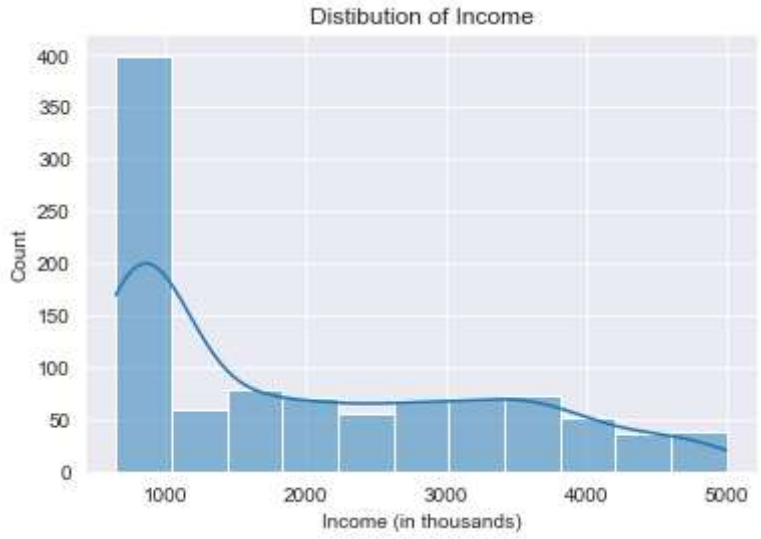
Out[362…

| | income | cc |
|---|---|---|
| **count** | 1000.00000000 | 1000.00000000 |
| **mean** | 2041.53900000 | 258.80400000 |
| **std** | 1297.70164446 | 139.56754882 |
| **min** | 652.00000000 | 10.00000000 |
| **25%** | 861.00000000 | 141.00000000 |
| **50%** | 1620.00000000 | 262.50000000 |
| **75%** | 3086.50000000 | 373.25000000 |
| **max** | 5000.00000000 | 500.00000000 |

Based on the sample, income is significantly skewed with most of the sample have an income of less than Php 1 million. Income ranges from a low of Php 652K to Php 5 million. However, the distribution for credit card balance is somewhat uniform and ranges from Php 10K to Php 500K.

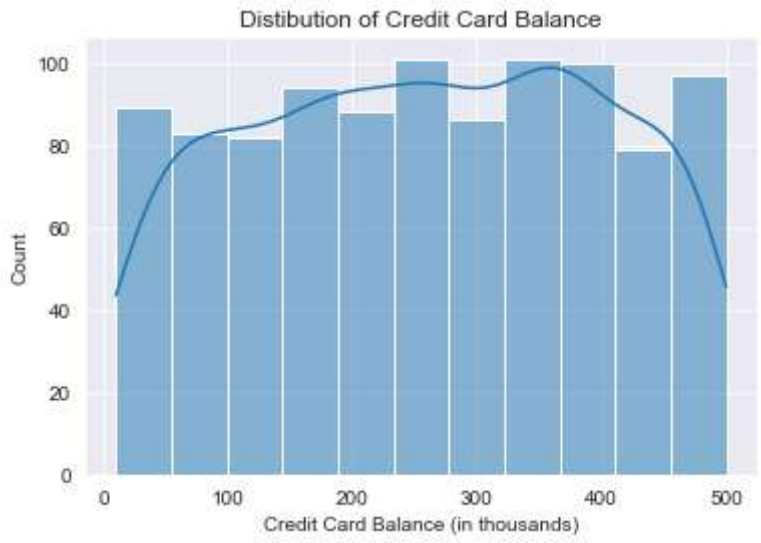
In [365…
```
sns.set_style("darkgrid")
sns.histplot(df['income'], kde=True)
plt.xlabel('Income (in thousands)')
plt.title('Distibution of Income')
```

Out[365… `Text(0.5, 1.0, 'Distibution of Income')`



In [366…
```
sns.set_style("darkgrid")
sns.histplot(df['cc'], kde=True)
plt.xlabel('Credit Card Balance (in thousands)')
plt.title('Distibution of Credit Card Balance')
```

Out[366… `Text(0.5, 1.0, 'Distibution of Credit Card Balance')`




## Scale Income and Credit Card Balance Columns

We will scale both Income and Credit Card Balance and assign them to the income_scaled and cc_scaled columns.

In [367…
```
scaler = StandardScaler()
df[['income_scaled', 'cc_scaled']] = scaler.fit_transform(df[['income','cc']])
df.head(3)
```

Out[367…

| | id | location | income | education | mortgage | p_loan | cc | income_scaled | cc_scaled |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 2834 | 2 | 0 | 1 | 156 | 0.61097058 | -0.73695814 |
| **1** | 2 | 1 | 1361 | 2 | 1 | 1 | 96 | -0.52468110 | -1.16707261 |
| **2** | 3 | 1 | 713 | 2 | 1 | 0 | 291 | -1.02427532 | 0.23079943 |

## KMeans Clustering

We will now initiate the clustering model using KMeans with n_clusters = 3. We will fit the model and predict the clusters of each customer.

In [368…
```
model = KMeans(n_clusters=3, random_state=0)
cluster_cols = ['income_scaled','cc_scaled']

model.fit(df[cluster_cols])
df['Cluster'] = model.predict(df[cluster_cols])
df[['income','income_scaled','cc','cc_scaled','Cluster']].head(3)
```

Out[368…

| | income | income_scaled | cc | cc_scaled | Cluster |
|---|---|---|---|---|---|

| | income | income_scaled | cc | cc_scaled | Cluster |
|---|---|---|---|---|---|
| **0** | 2834 | 0.61097058 | 156 | -0.73695814 | 2 |
| **1** | 1361 | -0.52468110 | 96 | -1.16707261 | 1 |
| **2** | 713 | -1.02427532 | 291 | 0.23079943 | 0 |

To visualize the relationship, we will plot the scaled attributes based on their clustering. Although all clusters consists of more than 300 customers, cluster 2 is noticeably more dispersed than the other two clusters as shown in the graph below.

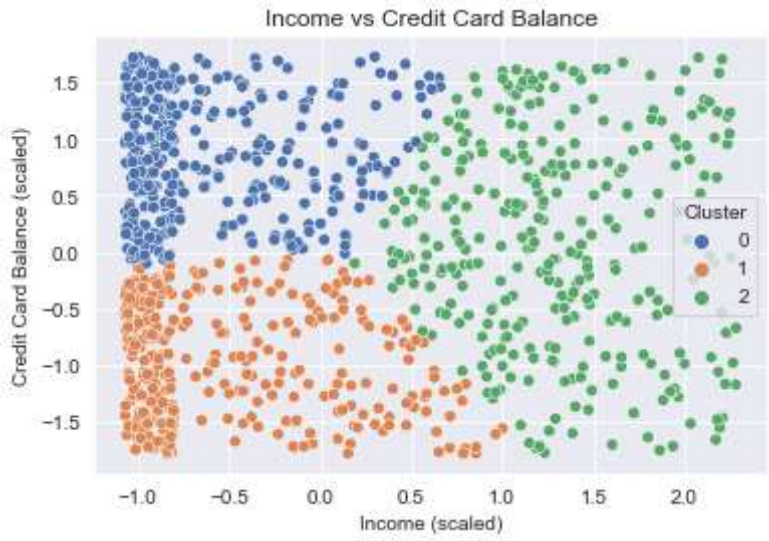
In [420…
```
print('Number of Customers per Cluster:')
df.groupby(['Cluster'])['Cluster'].count()
```

```
Number of Customers per Cluster:
```

Out[420…
```
Cluster
0    362
1    330
2    308
Name: Cluster, dtype: int64
```

In [410…
```
sns.scatterplot(data=df, x="income_scaled", y="cc_scaled", hue='Cluster', palette="deep
plt.title('Income vs Credit Card Balance')
plt.xlabel('Income (scaled)')
plt.ylabel('Credit Card Balance (scaled)')
```

Out[410…
```
Text(0, 0.5, 'Credit Card Balance (scaled)')
```




To verify if the number of clusters is optimal, we will verify clusters from 1 to 15 using the Elbow Method. The sum of squared distance will be tested for each cluster count and determine the inflection point for their sum of squared distance. The graph below shows the inflection point at three (3) clusters.

In [424…
```
# Finding the optimal cluster size using elbow method
sosd = []

# Run clustering for size 1 to 15
```

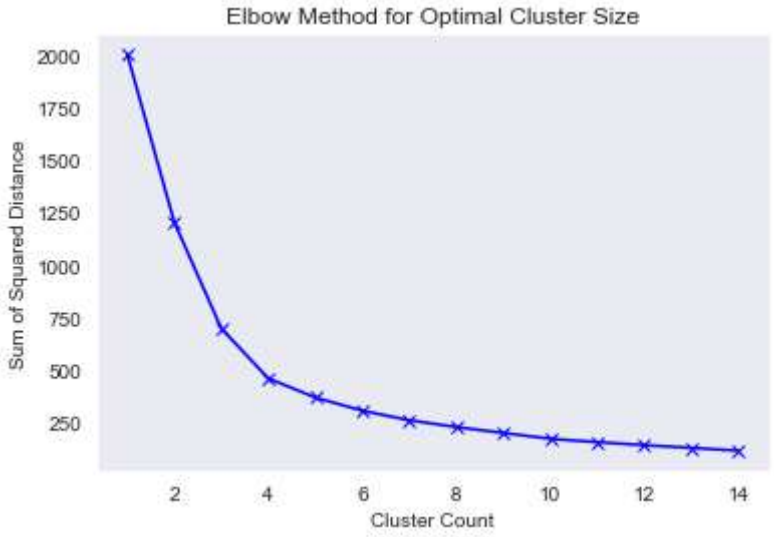
```
K = range(1,15)

for k in K:
    km = KMeans(n_clusters=k)
    km = km.fit(df[['income_scaled','cc_scaled']])
    sosd.append(km.inertia_)

print('Sum of squared distances: ', sosd)

# Plot sosd against number of clusters
import matplotlib.pyplot as mpLib
mpLib.plot(K, sosd, 'bx-')
mpLib.xlabel('Cluster Count')
mpLib.ylabel('Sum of Squared Distance')
mpLib.title('Elbow Method for Optimal Cluster Size')
mpLib.grid()
mpLib.show()
```

```
Sum of squared distances:  [1999.9999999999989, 1202.9822983037807, 698.3159800871821, 4
60.7814941474864, 369.75701117177624, 307.1252018748727, 261.68626076056984, 229.4011970
7538855, 201.37798260197033, 173.88706977188346, 157.6688778177592, 143.5023726867954, 1
30.32112870659006, 117.32888763864307]
```




## Conclusion

The following are the key findings:

- Customers with higher annual income (average of Php 3.7M) and higher credit card balance (average of Php 271K) have a higher propensity to take up a personal loan.
- It is surprising to see that cluster 0 or those with an average income of Php 1.2M and an average credit card balance of Php 375K have a slightly higher propensity to avail of personal loans compared to cluster 2.

In [425…

```
ploans = ['income', 'cc', 'p_loan']
df.groupby('Cluster')[ploans].mean()
```

Out[425…

| Cluster | income | cc | p_loan |
|---|---|---|---|

| Cluster | income | cc | p_loan |
|---|---|---|---|
| **0** | 1224.82596685 | 372.28176796 | 0.01657459 |
| **1** | 1372.97272727 | 122.71212121 | 0.00909091 |
| **2** | 3717.76298701 | 271.24350649 | 0.04220779 |