Market Segmentation of Credit Card Customers

Matt Hoyle, Shuyuan Shen, Junseok Yang
Dec 15, 2021

Motivation

- Retaining existing customers and finding new ones are crucial for the sustainability and profitability of any business.

- How to identify individuals who are most likely to be long-term credit card customers?

- Market segmentation (Tynan and Drayton 1987; Yankelovich and Meer 2006).

Market Segmentation

- The goal of market segmentation is to identify and delineate market segments or “sets of buyers,” which would then become targets for the company’s marketing plans (Tynan and Drayton 1987).

- Traditional Methods: multiple discriminator analysis, multiple regression analysis, etc.

- Recent Developments: Big data and Machine Learning

Research Questions

1. *Can we perform market segmentation and uncover demographic differences for credit card customers solely by their banking information?*

1. *How well can we distinguish existing customers from the attrited ones using clustering algorithms?*

Data

- About 9000 active credit card holders.
- 20 variables
- Categorical and Numerical

Data Cleaning

```
df = pd.read_csv("BankChurners.csv")
df.head()
```

```
# Drop the first and last 2 columns
# df - Original Dataset without Client Number and 'Naive Bayes ~' columns
df = df.drop(['CLIENTNUM', 'Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_coun
df.head()
```

	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book	Total_Relatior
0	Existing Customer	45	M	3	High School	Married	$60K - 80K	Blue	39	
1	Existing Customer	49	F	5	Graduate	Single	Less than $40K	Blue	44	
2	Existing Customer	51	M	3	Graduate	Married	$80K - 120K	Blue	36	
3	Existing Customer	40	F	4	High School	Unknown	Less than $40K	Blue	34	
4	Existing Customer	40	M	3	Uneducated	Married	$60K - 80K	Blue	21	

Data Cleaning

```
df.isna().sum()
```

```
Attrition_Flag              0
Customer_Age                0
Gender                      0
Dependent_count             0
Education_Level             0
Marital_Status              0
Income_Category             0
Card_Category               0
Months_on_book              0
Total_Relationship_Count    0
Months_Inactive_12_mon      0
Contacts_Count_12_mon       0
Credit_Limit                0
Total_Revolving_Bal         0
Avg_Open_To_Buy             0
Total_Amt_Chng_Q4_Q1        0
Total_Trans_Amt             0
Total_Trans_Ct              0
Total_Ct_Chng_Q4_Q1         0
Avg_Utilization_Ratio       0
dtype: int64
```

Exploratory Data Analysis

Numerical Attributes' Summary Statistics

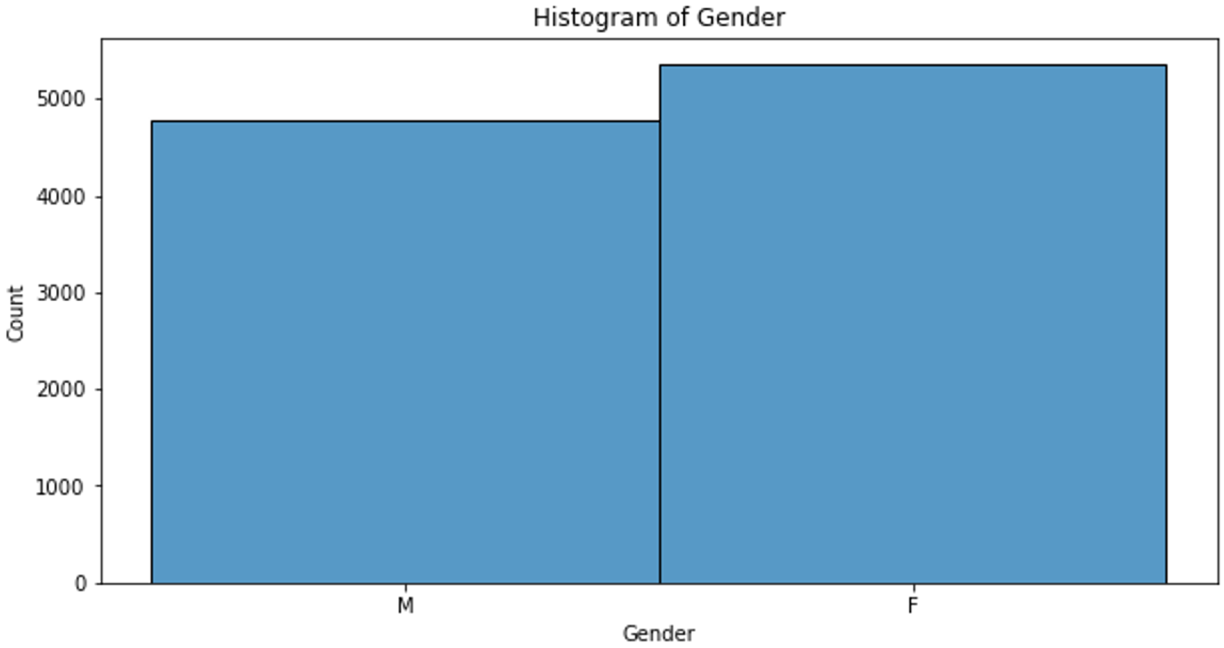
```
df.describe()
```

	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_Limit	Total_Re
count	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10
mean	46.325960	2.346203	35.928409	3.812580	2.341167	2.455317	8631.953698	1
std	8.016814	1.298908	7.986416	1.554408	1.010622	1.106225	9088.776650	
min	26.000000	0.000000	13.000000	1.000000	0.000000	0.000000	1438.300000	
25%	41.000000	1.000000	31.000000	3.000000	2.000000	2.000000	2555.000000	
50%	46.000000	2.000000	36.000000	4.000000	2.000000	2.000000	4549.000000	1
75%	52.000000	3.000000	40.000000	5.000000	3.000000	3.000000	11067.500000	1
max	73.000000	5.000000	56.000000	6.000000	6.000000	6.000000	34516.000000	2

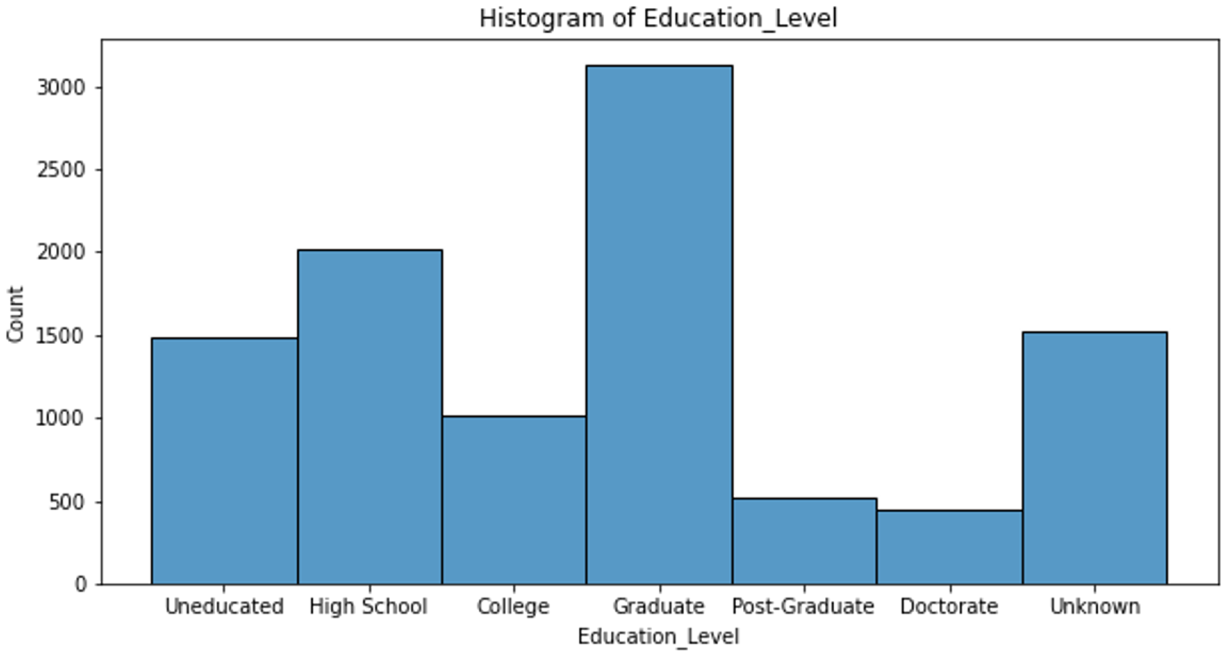
Exploratory Data Analysis

```
cat_list = ['Gender', 'Education_Level', 'Marital_Status', 'Income_Category', 'Card_Category']

df_cat = df[cat_list].copy()
df_cat['Education_Level'] = pd.Categorical(df_cat['Education_Level'], ['Uneducated','High School','College','Graduat
df_cat['Marital_Status'] = pd.Categorical(df_cat['Marital_Status'], ['Single','Married','Divorced','Unknown'])
df_cat['Income_Category'] = pd.Categorical(df_cat['Income_Category'], ['Less than $40K','$40K - $60K','$60K - $80K',
df_cat['Card_Category'] = pd.Categorical(df_cat['Card_Category'], ['Blue','Silver','Gold','Platinum'])

for col in cat_list:
    plt.figure(figsize = (10, 5))
    print(df[col].value_counts())
    sns.histplot(x = col, data = df_cat)
    plt.title("Histogram of %s"%(col))
    plt.show()
    print('----------------------------------------------------------------')
```

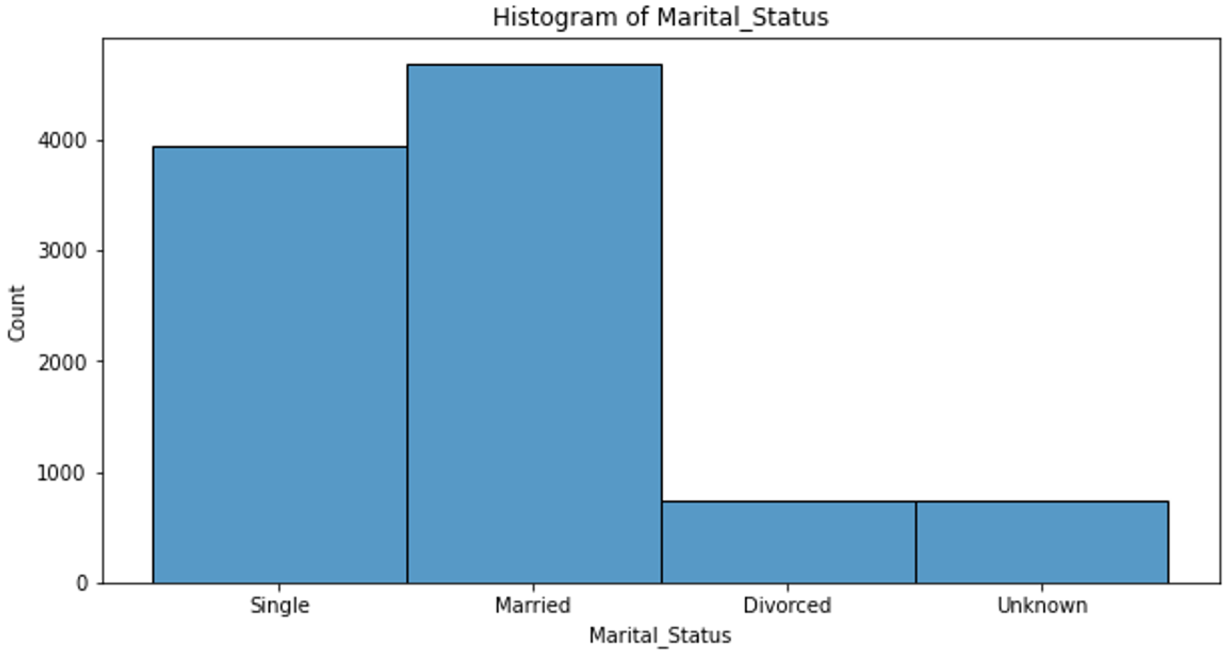
Exploratory Data Analysis



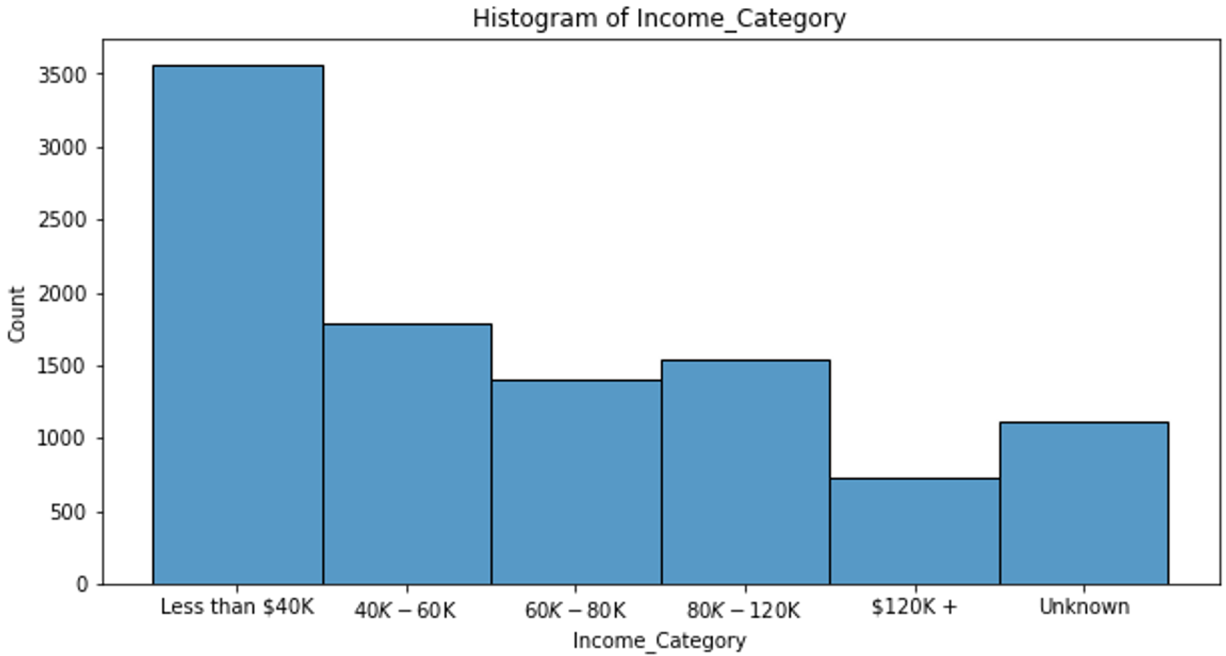
Exploratory Data Analysis



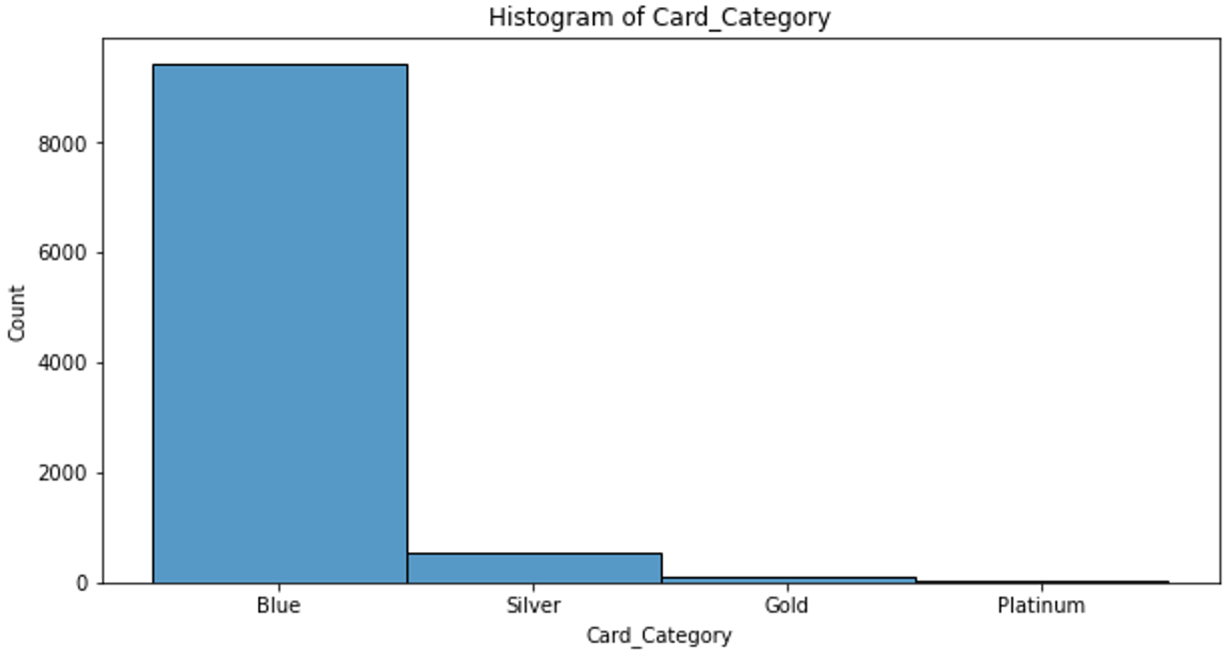
Exploratory Data Analysis



Exploratory Data Analysis



Exploratory Data Analysis



First Research Question

1. *Can we perform market segmentation and uncover demographic differences for credit card customers solely by their banking information?*

Why only use banking information?

- Reduced computational cost
- Privacy concerns
- Impact of demographics in clustering negated
- Issues with demographic data overpowering clustering

Variables Used in Clustering

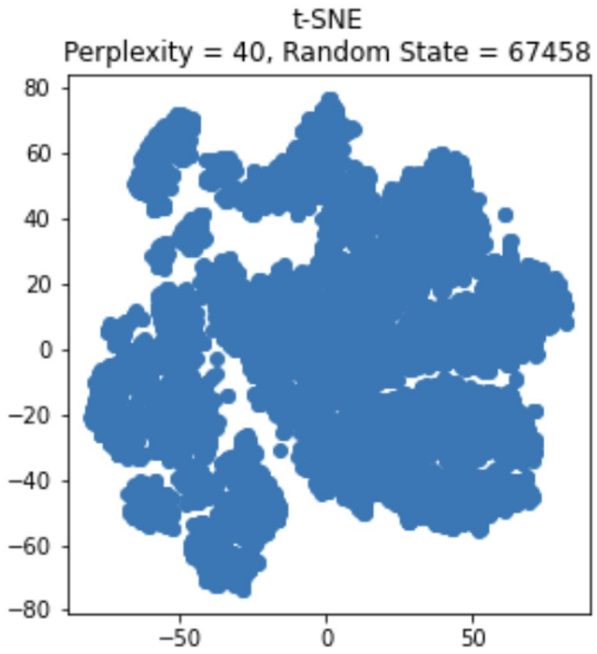
Card Category	Average Open to Buy
Months on the Books	Total Amount Change
Total Relationship Count	Total Count Change
Months Inactive	Total Transaction Amount
Contacts Count	Total Transaction Count
Credit Limit	Average Utilization Ratio
Total Revolving Balance	

Clustering Approach

1. Gower’s distance matrix
2. Determine clusterability
3. Select clustering algorithm
4. Implement clustering algorithm
5. Select hyperparameters
6. Analyze clusters

Determine Clusterability

- Hopkins Statistic = 0.0866
  - Data is likely clusterable
- t-SNE plot shows clustering
  - Non-separated, non-spherical, non-equal size

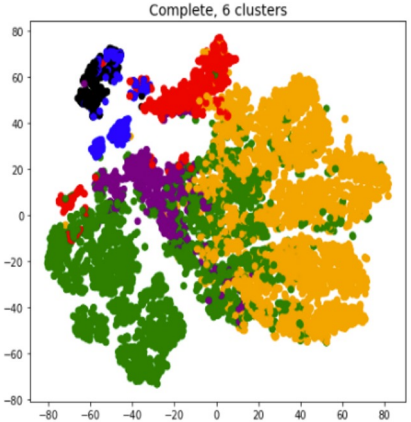


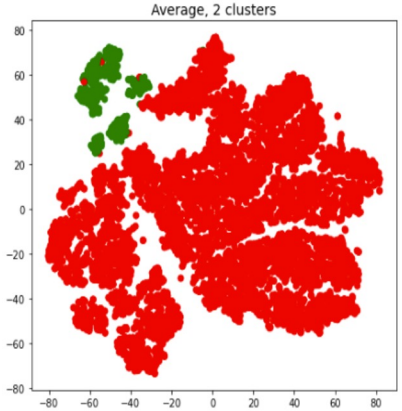
Clustering Algorithm

- Hierarchical Agglomerative Clustering (HAC)
- Test multiple linkages
  - Single
  - Average
  - Complete

HAC Clustering Results

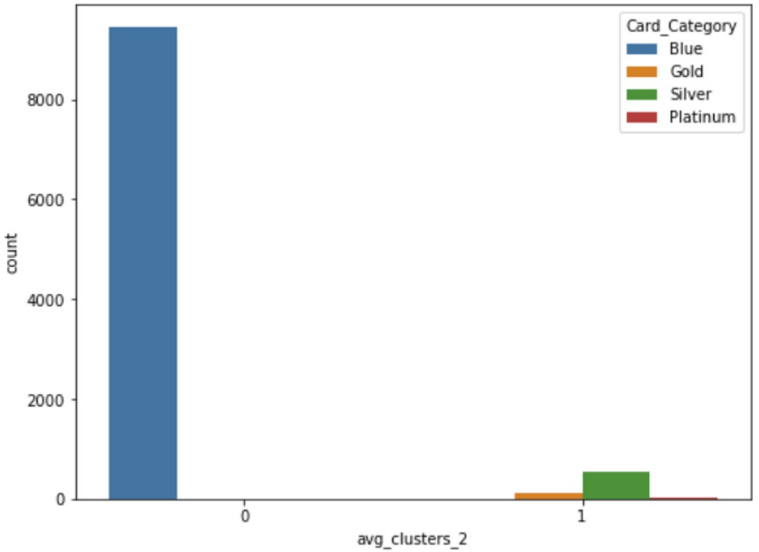
- Single linkage performs poorly
- Silhouette scores suggest 2 clusters
  - Inseparable clusters
- t-SNE plots show similar clustering patterns
- Complete linkage created tiny clusters after 6
- Average linkage maintains two main clusters





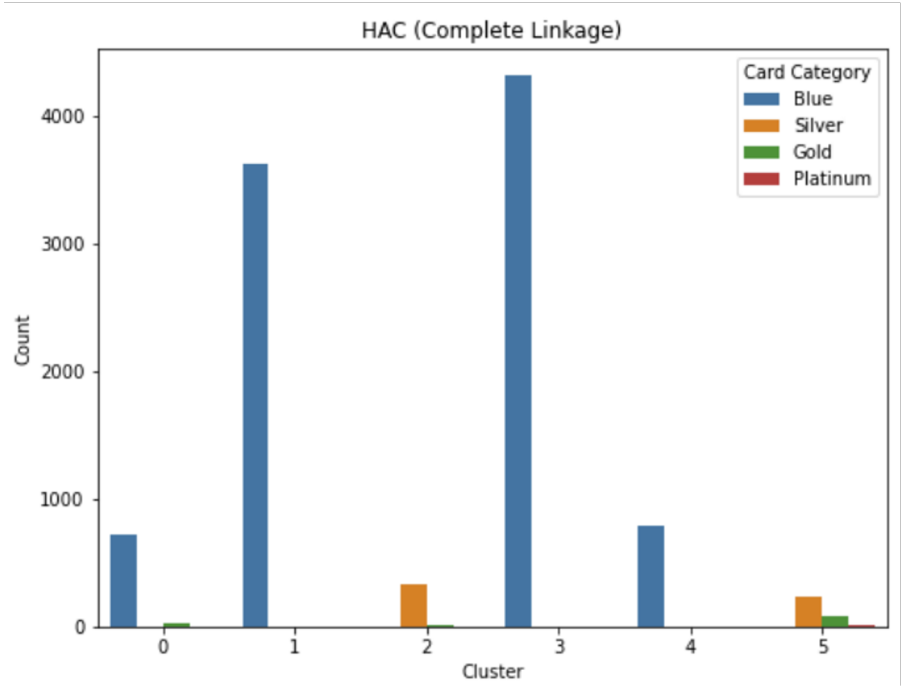
HAC with Average Linkage

- 2 clusters
  - Defined entirely by card category
- Card category is an important characteristic



HAC with Complete Linkage

- 6 clusters
  - Defined by card category
  - Blue vs non-blue

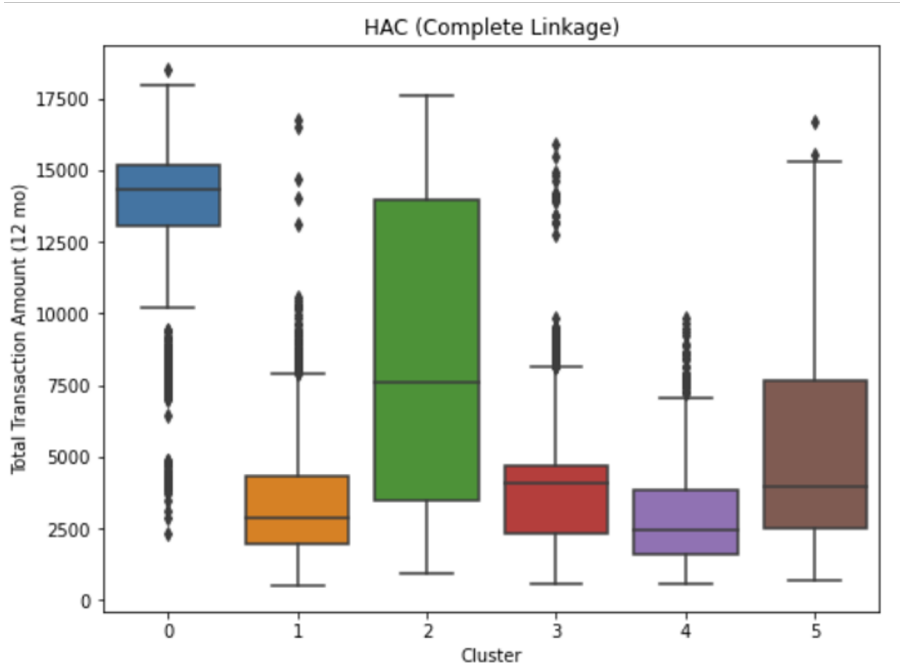


HAC with Complete Linkage

Cluster	Size	Primary Card Type		Secondary Card Type		Third Card Type	
0	747	Blue	96.8%	Gold	2.9%	Platinum	0.27%
1	3618	Blue	100%	-	-	-	-
2	344	Silver	94.2%	Gold	5.2%	Platinum	0.58%
3	4312	Blue	100%	-	-	-	-
4	783	Blue	100%	-	-	-	-
5	323	Silver	71.5%	Gold	23.5%	Platinum	4.95%

Defining the Clusters

- Box plots or count plots for each banking feature
- Total transaction amount
  - Clusters 0 and 2 have high amount
  - Clusters 1 and 4 have low amount



Cluster Definitions

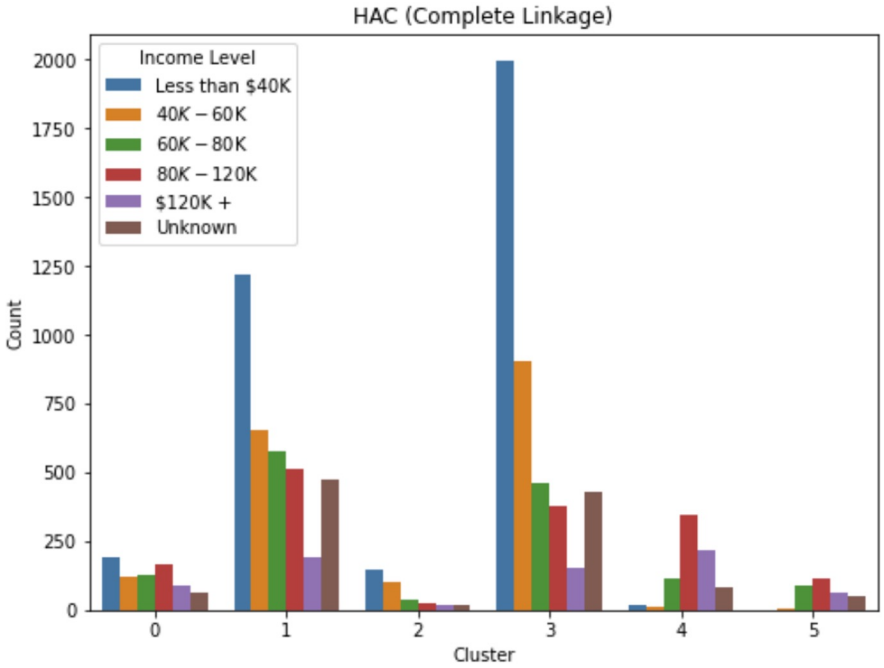
Blue Cluster	Definition
0	High transaction counts and amounts, low number of products, low credit limit
1	Low revolving balance, low credit limit
3	High utilization ratio, low credit limit
4	Low transaction counts and amounts, high credit limit

Non-blue Cluster	Definition
2	High transaction counts and amounts, lower credit limit
5	Low transaction counts and amounts, high credit limit

Cluster Analysis - Demographics

Blue Cluster	Income Levels
0	Mixed income levels
1	Skewed to lower incomes
3	Heavily skewed to lower incomes
4	Heavily skewed to higher incomes

Non-Blue Cluster	Income Levels
0	Lower Income
5	Higher Income



Takeaways - Research Q1

- Credit card customers can be clustered based solely on banking information
  - HAC with complete linkage
- Card category most important banking feature in clustering
  - Transaction count and amount also very important
- Income level most segmented by clustering
  - Banking information heavily shaped by income level
  - Impactful demographics data with reduced privacy concerns
    - Personally identifiable information

Second Research Question

2. How well can we distinguish existing customers from the attrited ones using clustering algorithms?

Data Sampling and Scaling

```
# Check the proportion of Existing vs Attrited
print(len(df[df['Attrition_Flag'] == 'Existing Customer']))
print(len(df[df['Attrition_Flag'] == 'Attrited Customer']))
```

```
8500
1627
```

```
# Equivalent random sample from two different Attrition Flag types

X_1 = df[df['Attrition_Flag'] == 'Existing Customer'].sample(n = 800, replace = False, random_state = 100)
X_2 = df[df['Attrition_Flag'] == 'Attrited Customer'].sample(n = 800, replace = False, random_state = 100)
```

```
# Concatenate two dataframes into one dataframe
X = pd.concat([X_1, X_2])
X
```

	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book	Total_Rela
1237	Existing Customer	38	F	3	Graduate	Married	Less than $40K	Blue	18	
1201	Existing Customer	57	F	1	Unknown	Married	Less than $40K	Blue	47	
6921	Existing Customer	41	F	4	High School	Married	Unknown	Blue	30	
6133	Existing Customer	42	F	4	Graduate	Married	Less than $40K	Blue	36	
4396	Existing Customer	42	F	3	High School	Married	Less than $40K	Blue	36	
...	...	...	...	...	...	...	...	...	...	
2077	Attrited Customer	58	M	2	Uneducated	Single	$120K +	Blue	46	
8757	Attrited Customer	50	M	4	High School	Single	$120K +	Blue	41	

Scaling

```
# First, drop all categorical variables
drop_list = ['Attrition_Flag','Gender', 'Education_Level', 'Marital_Status', 'Income_Category', 'Card_Category']

# x - Scaled Dataset
x = X.copy()
x = x.drop(drop_list, axis = 1)
x.head()
```

```
# Scale numerical variables
from sklearn.preprocessing import StandardScaler

ss = StandardScaler()
ss_array = ss.fit_transform(x)
ss_array
```

```
# Need to match the index
x = pd.DataFrame(ss_array, columns = x.columns, index = x.index)
x.head()
```

```
# Put categorical variables back into the dataset
for col in cat_list:
    x[col] = X[col]
x.head()
```

Q4_Q1	Total_Trans_Amt	Total_Trans_Ct	Total_Ct_Chng_Q4_Q1	Avg_Utilization_Ratio	Gender	Education_Level	Marital_Status	Income_Category	Card_Category
388372	-0.671670	-0.432105	-0.317525	1.236877	F	Graduate	Married	Less than $40K	Blue
329120	-0.756892	-0.654053	-0.362418	1.763467	F	Unknown	Married	Less than $40K	Blue
171440	0.260265	0.943973	1.890371	-0.188856	F	High School	Married	Unknown	Blue

Clusterability

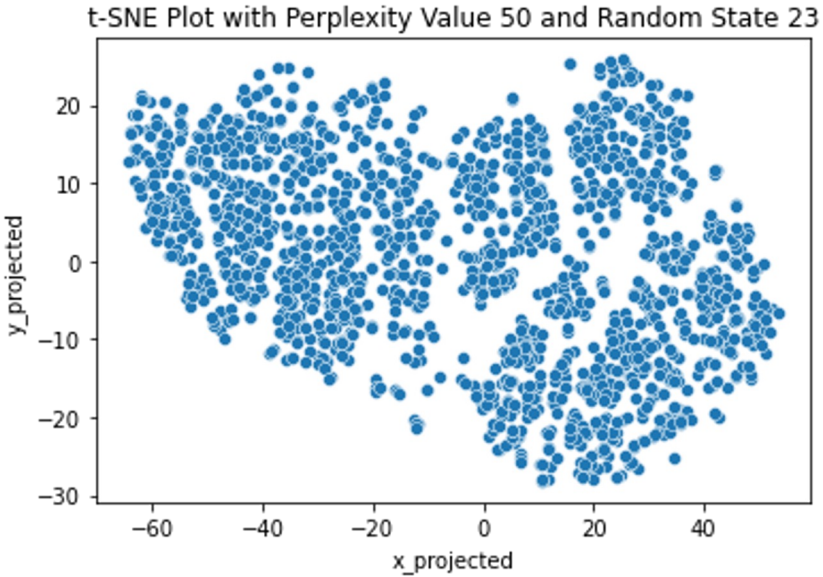
```
# X - Unscaled Dataset, x - Scaled Dataset
from gower import gower_matrix
dist_mat = gower_matrix(x)
dist_mat
```

```
array([[0.        , 0.22388041, 0.23646867, ..., 0.3480109 , 0.4914987 ,
        0.26662782],
       [0.22388041, 0.        , 0.30855188, ..., 0.41026807, 0.41364172,
        0.27406055],
       [0.23646867, 0.30855188, 0.        , ..., 0.3848821 , 0.39061034,
        0.26088417],
       ...,
       [0.3480109 , 0.41026807, 0.3848821 , ..., 0.        , 0.29175434,
        0.3959032 ],
       [0.4914987 , 0.41364172, 0.39061034, ..., 0.29175434, 0.        ,
        0.2725757 ],
       [0.26662782, 0.27406055, 0.26088417, ..., 0.3959032 , 0.2725757 ,
        0.        ]], dtype=float32)
```

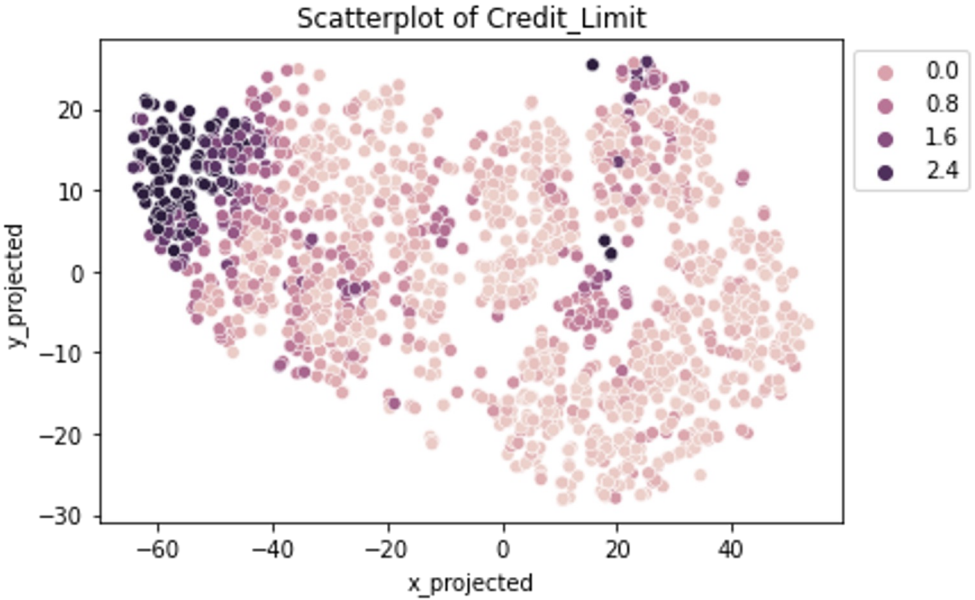
Clusterability

- Hopkins Statistic ≈ 0.16
  - Data are likely clusterable

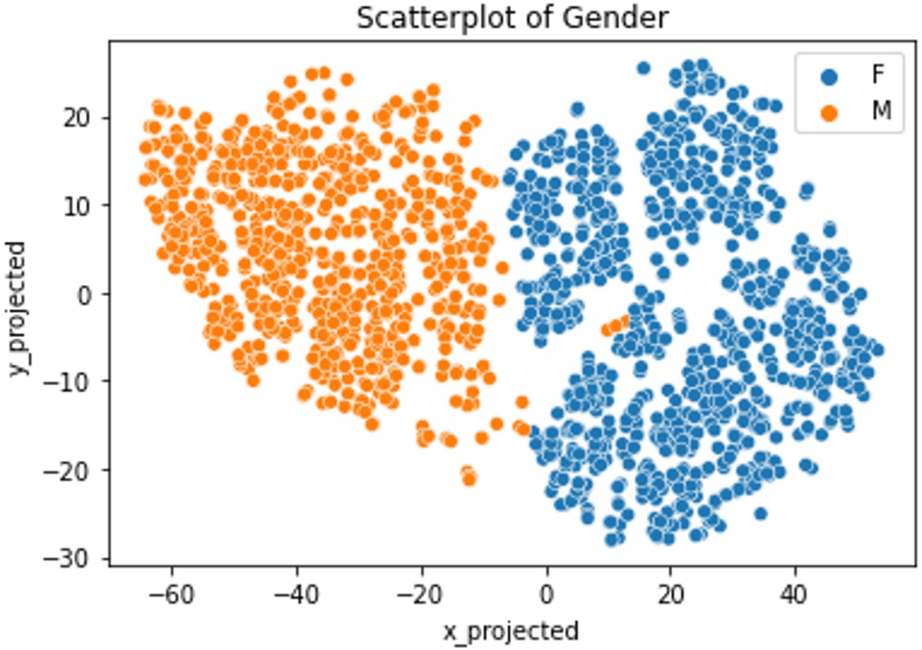
- t-SNE plot shows clustering



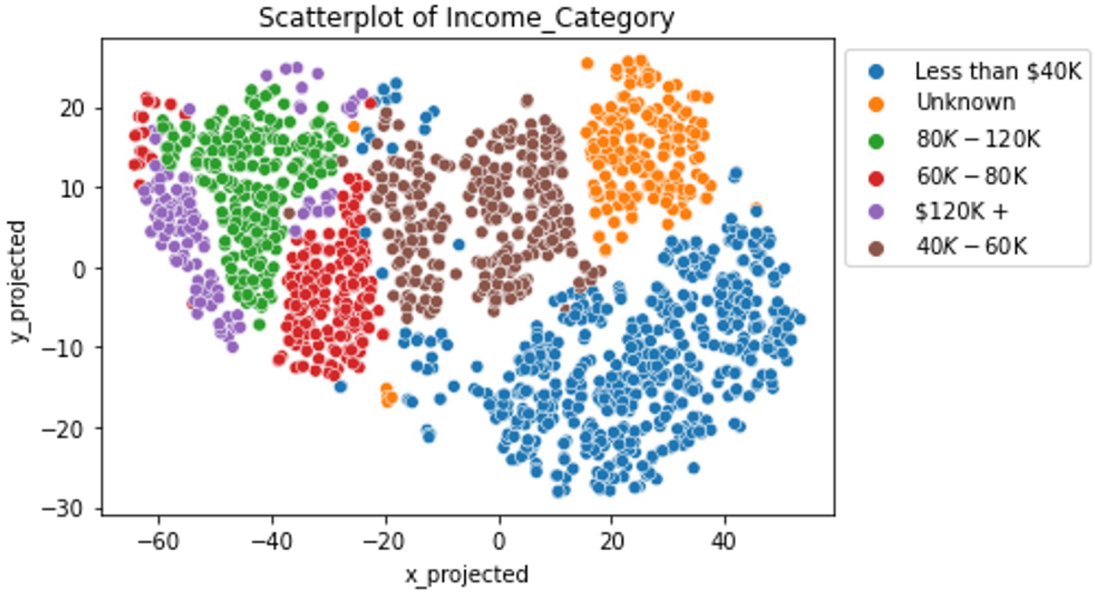
Some Explorations



Some Explorations



Some Explorations

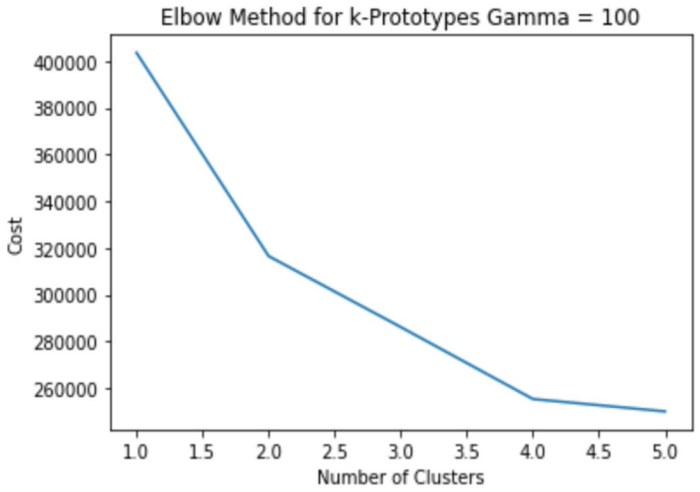


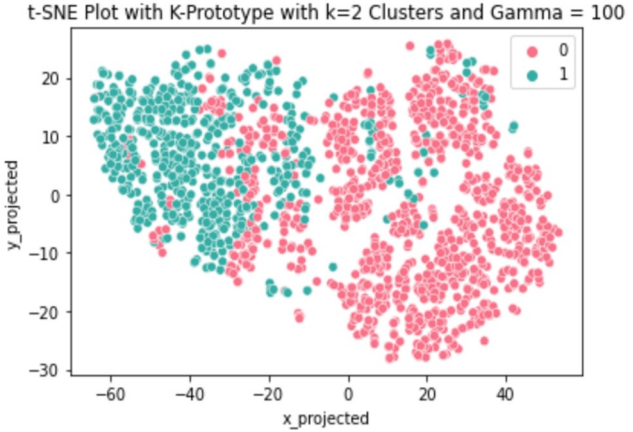
Algorithm Selection Motivation

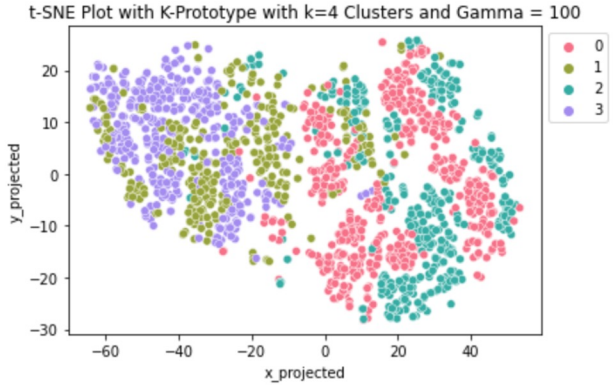
Considering that the dataset is mixed of categorical and numerical attributes, we can use:

- K-Prototype Algorithm
- Hierarchical Agglomerative Clustering using Gower's Distance Matrix

K-PROTOTYPE ALGORITHM

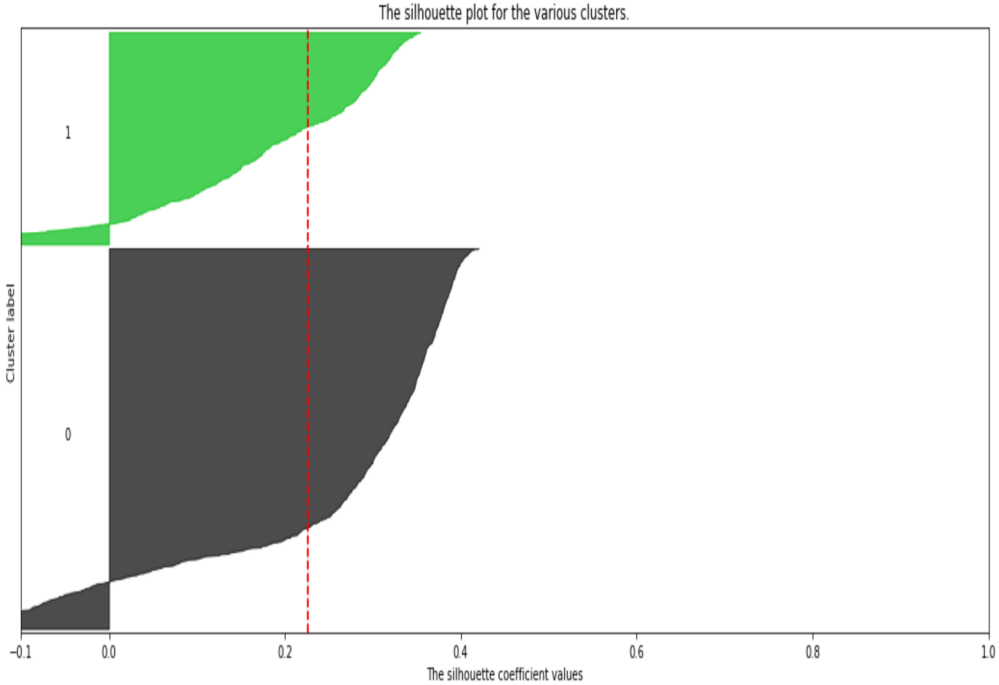






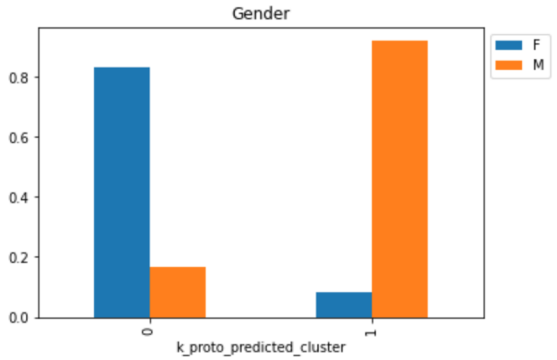
COHESION AND SEPARATION

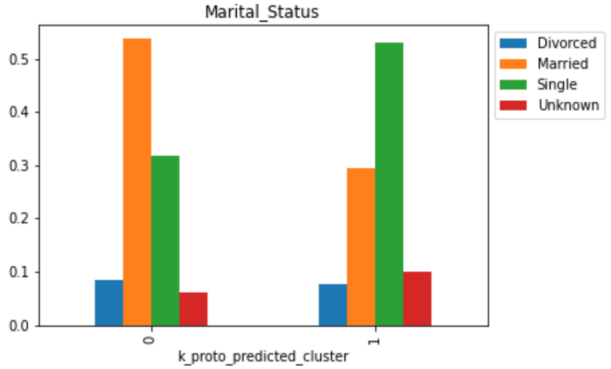
- Relatively Low Silhouette Scores
- Negative Values - Observation “Closer” to the other cluster than the cluster that was originally assigned to
- Not Cohesive and Not Well Separated

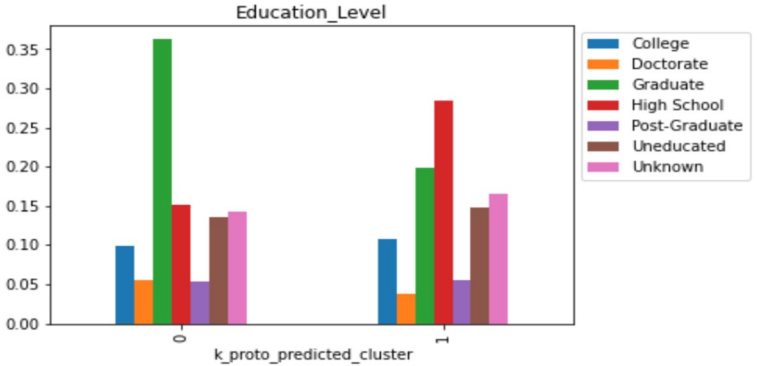


DISTINCTIONS - DEMOGRAPHICS

- Gender
- Education Level
- Marital Status

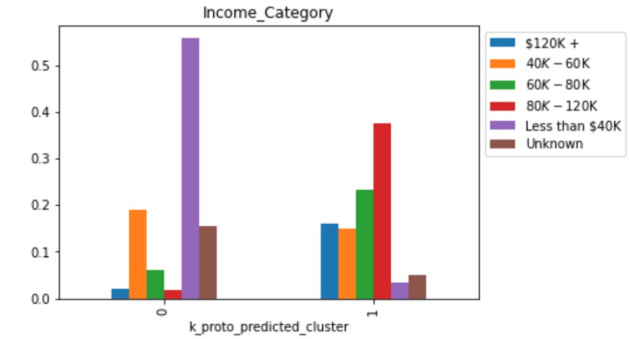


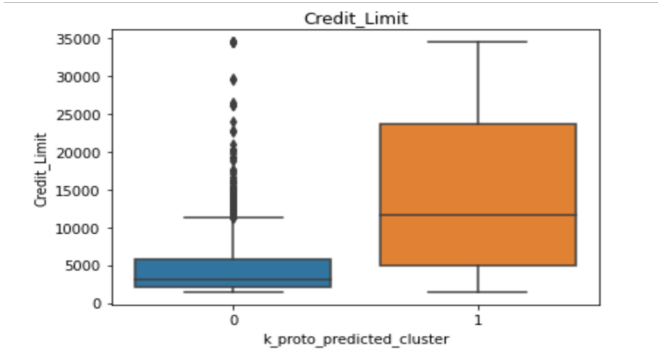


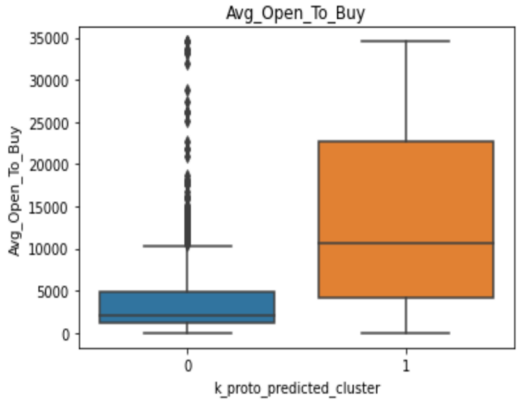


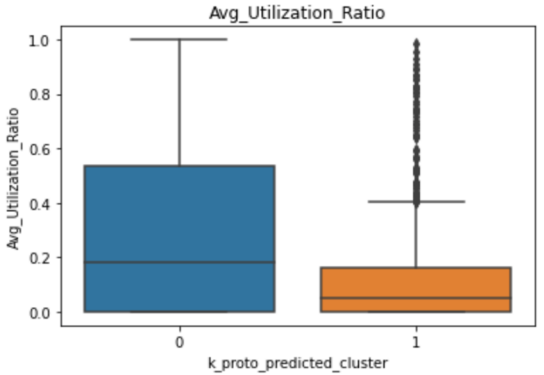
DISTINCTIONS - BANK INFO

- Income
- Credit Limit
- Average Open to Buy
- Average Utilization Ratio









CLUSTER COMPARISON

Cluster 0

- Female
- Married
- Graduate
- Income < $40K
- Low Credit Limit
- Low Average Open to Buy
- High Average Utilization Ratio

Low Income / Less Economically Active

➢ Attrited Customer

Cluster 1

- Male
- Single
- High School
- Income > $40K
- High Credit Limit
- High Average Open to Buy
- Low Average Utilization Ratio

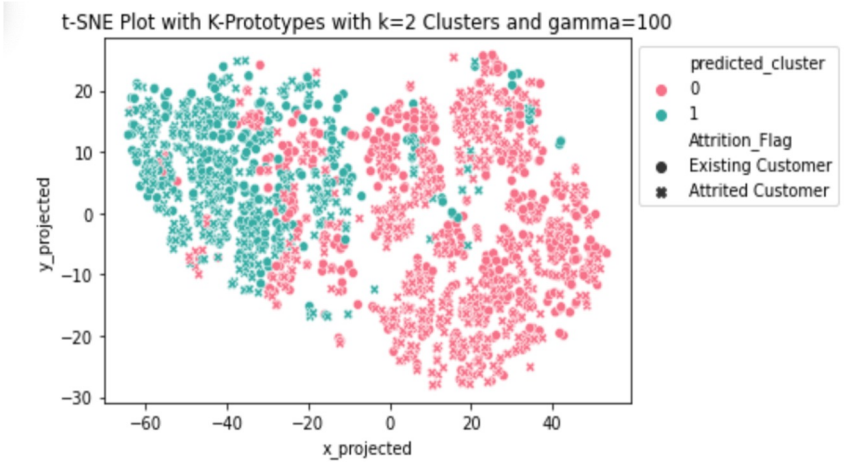
High Income / Economically Active

➢ Existing Customer

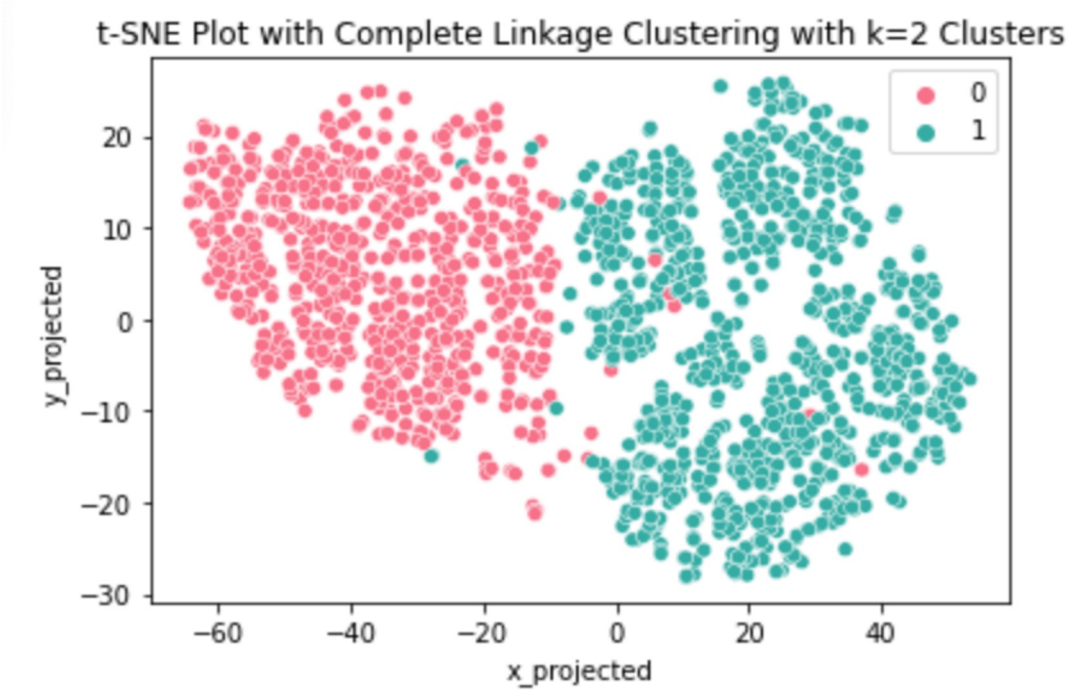
SUPERVISED CLUSTERING EVALUATION METRICS

- Homogenous Score = 0.0000306
- Completeness Score = 0.0000326
- V-Score = 0.0000316
- Adjusted Rand Index = -0.000536

Poor Performance!

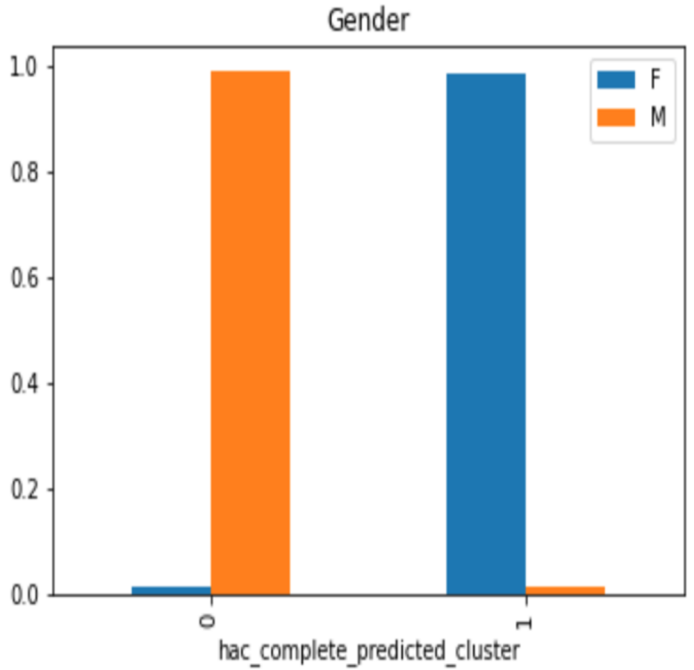


HIERARCHICAL AGGLOMERATIVE CLUSTERING



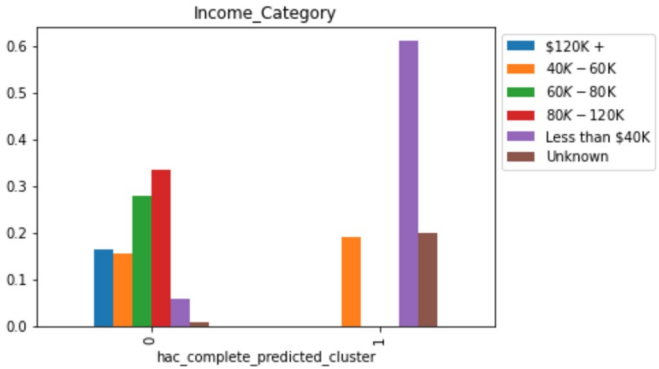
DISTINCTION - DEMOGRAPHIC

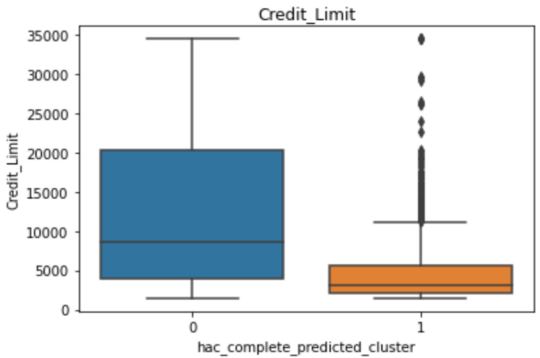
- Gender

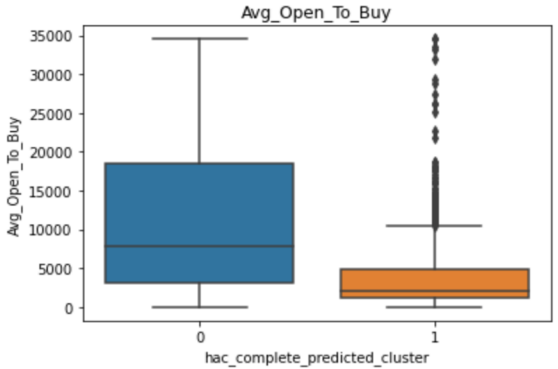


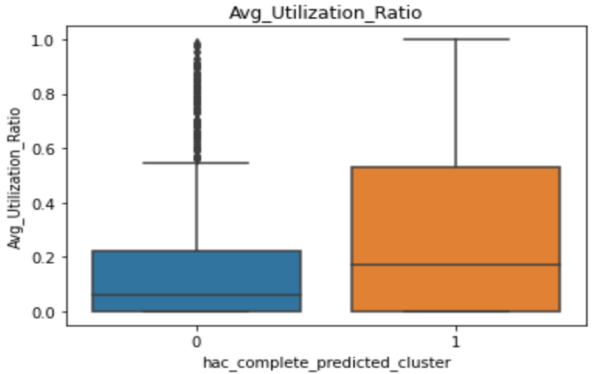
DISTINCTION - BANK INFO

- Income
- Credit Limit
- Average Open to Buy
- Average Utilization Ratio









CLUSTER COMPARISON

Cluster 0

- Male
- Income > $40K
- High Credit Limit
- High Average Open to Buy
- Low Utilization Ratio

High Income / Economically Active

➢ Existing Customer

Cluster 1

- Female
- Income < $40K
- Low Credit Limit
- Low Average Open to Buy
- High Utilization Ratio

Low Income / Less Economically Active

➢ Attrited Customer

SUPERVISED CLUSTERING EVALUATION METRICS

- Homogenous Score = 0.001486
- Completeness Score = 0.0015
- V-Score = 0.001496
- Adjusted Rand Index = 0.0014

Poor Performance!

Takeaway - Research Q2

- Clustering from both K-Prototype and HAC fails to match the pre-assigned Attrition Flag label.

- ‘Gender’, ‘Income’, ‘Credit Limit’ are better aligned with the clustering structure.

Final Summary

- Research Question 1
  - Clustering of banking information is effective via HAC with complete linkage
  - Card category, transaction count, and transaction amount are important in clustering
  - Income level most segmented demographic
- Research Question 2
  - Clustering from both K-Prototype and HAC fails to match the pre-assigned Attrition Flag label.
  - ‘Gender’, ‘Income’, ‘Credit Limit’ are better aligned with the clustering structure.

Limitations and Potential Improvements

- Research Question 1
  - Computational cost
    - Compare multiple methods
  - Include income in clustering post-analysis

- Research Question 2
  - Computational cost
  - Balanced vs unbalanced comparison