Problem Statement

Create a k-means model with the assignment dataset using at least 10 features. Experiment with at least 3 k values. Be sure to transform variables into the appropriate format before modeling. Note that a larger k will increase the overhead of interpretation, so it is suggested to keep the k less than 10. What transformations did you apply to the raw dataset? What were different k's chosen? What were the differences in the output with those different k's? Choose a final k that you think reflects the data the best and provide a written interpretation of the different clusters generated by k-means Why did you choose this k and distance metric? Why does it appear these groups have been created? What are the influential features? Are there any inferences you can draw that would be relevant from a business context about the different groups?

ANSWER 1

Create a k-means model with the assignment dataset using at least 10 features. Experiment with at least 3 k values. Be sure to transform variables into the appropriate format before modeling. Note that a larger k will increase the overhead of interpretation, so it is suggested to keep the k less than 10.

Loading the required libraries

```
#@title Loading the required libraries
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
from google.colab import drive

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt # Matplotlib for subplots
%matplotlib inline

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler # Import for feature standardization
from sklearn.metrics import silhouette_samples, silhouette_score # For kmeans evaluation
from sklearn.datasets import load_wine # Used to pull in wine data

pd.options.display.float_format = '{:.2f}'.format
pd.set_option('display.max_columns', 500)
```

Description of why the features were selected for k-means modeling

Create a k-means model with the assignment dataset using at least 10 features.

Selection of features for k means analysi:

Financial & Credit Information

AMT_INCOME_TOTAL Total income of the applicant.

Important to understand affordability and debt-to-income ratio. AMT_CREDIT

Total amount of credit granted for the loan.

High values may indicate higher credit risk unless supported by high income.

AMT_ANNUITY

Monthly installment amount for the loan.

Helps gauge the repayment burden.

AMT_GOODS_PRICE

Price of the goods for which the loan is taken (e.g., car, home).

Indicates loan purpose and risk type (e.g., secured vs unsecured loan).

Demographic & Employment Stability

DAYS_BIRTH

Age of the applicant (negative number of days).

Used to derive risk perception by age; younger applicants may have less credit history.

DAYS_EMPLOYED

Duration of employment (in days; negative means currently employed).

Indicates employment stability, a strong signal of creditworthiness.

External Credit Scores EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3

External risk scores from different sources (scaled between 0 and 1).

Highly predictive features in default prediction and widely used in segmentation.

Household Composition CNT_CHILDREN

Number of children.

Affects financial responsibilities and potential disposable income.

CNT_FAM_MEMBERS

Total number of family members.

Useful in estimating cost of living and resource distribution.

Geographic Rating

REGION_RATING_CLIENT

Rating of the region where the applicant lives.

Captures local economic conditions, infrastructure, and access to financial resources.

Reasons the Features Matter for Clustering These variables collectively:

Reflect income level, debt burden, and ability to repay.

Capture demographic traits that influence financial behavior.

Include proxy indicators for credit risk and socio-economic background.

These features allow the clustering algorithm to group applicants into segments such as:

Low-income high-risk,

High-income low-risk,

Young professionals vs. retired individuals, etc.

Load the dataset

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
from google.colab import drive

# Mount Google Drive
drive.mount('/content/drive')

file_path = '/content/drive/My Drive/Colab Notebooks/aai-510/assignment/train_data.csv'

try:
    # Load the CSV file into a pandas DataFrame
    df = pd.read_csv(file_path)
    # Print the first 5 rows of the DataFrame to verify
    print(df.head())
    plt.show() #display plots

except FileNotFoundError:
    print(f"Error: File not found at {file_path}")
except pd.errors.EmptyDataError:
    print(f"Error: The file at {file_path} is empty.")
```

```
except pd.errors.ParserError:
    print(f"Error: Unable to parse the CSV file at {file_path}. Check the file format.")
except KeyError as e:
    print(f"Error: Column '{e}' not found in the DataFrame. Please check your column names.")
except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

```
   OBS_60_CNT_SOCIAL_CIRCLE  DEF_60_CNT_SOCIAL_CIRCLE  DAYS_LAST_PHONE_CHANGE  \
0                      0.00                      0.00                -1755.00
1                      7.00                      0.00                -3268.00
2                      9.00                      0.00                    0.00
3                      0.00                      0.00                -1971.00
4                      0.00                      0.00                 -689.00

   FLAG_DOCUMENT_2  FLAG_DOCUMENT_3  FLAG_DOCUMENT_4  FLAG_DOCUMENT_5  \
0                0                1                0                0
1                0                1                0                0
2                0                1                0                0
3                0                0                0                0
4                0                1                0                0

   FLAG_DOCUMENT_6  FLAG_DOCUMENT_7  FLAG_DOCUMENT_8  FLAG_DOCUMENT_9  \
0                0                0                0                0
1                0                0                0                0
2                0                0                0                0
3                1                0                0                0
4                0                0                0                0

   FLAG_DOCUMENT_10  FLAG_DOCUMENT_11  FLAG_DOCUMENT_12  FLAG_DOCUMENT_13  \
0                 0                 0                 0                 0
1                 0                 0                 0                 0
2                 0                 0                 0                 0
3                 0                 0                 0                 0
```

```
4                0                0                0                0

   FLAG_DOCUMENT_14  FLAG_DOCUMENT_15  FLAG_DOCUMENT_16  FLAG_DOCUMENT_17  \
0                 0                 0                 0                 0
1                 0                 0                 0                 0
2                 0                 0                 0                 0
3                 0                 0                 0                 0
4                 0                 0                 0                 0

   FLAG_DOCUMENT_18  FLAG_DOCUMENT_19  FLAG_DOCUMENT_20  FLAG_DOCUMENT_21  \
0                 0                 0                 0                 0
1                 0                 0                 0                 0
2                 0                 0                 0                 0
3                 0                 0                 0                 0
4                 0                 0                 0                 0

   AMT_REQ_CREDIT_BUREAU_HOUR  AMT_REQ_CREDIT_BUREAU_DAY  \
0                        0.00                       0.00
1                        0.00                       0.00
2                        0.00                       0.00
3                        0.00                       0.00
4                        0.00                       0.00

   AMT_REQ_CREDIT_BUREAU_WEEK  AMT_REQ_CREDIT_BUREAU_MON  \
0                        0.00                       0.00
1                        0.00                       0.00
2                        0.00                       0.00
3                        0.00                       0.00
4                        0.00                       0.00

   AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
0                       0.00                        0.00
1                       2.00                        1.00
```

one-hot encoding and standardization techniques used for the transforamtion.

```
features = [
    'AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE',
    'DAYS_BIRTH', 'DAYS_EMPLOYED', 'EXT_SOURCE_1', 'EXT_SOURCE_2', 'EXT_SOURCE_3',
    'CNT_CHILDREN', 'CNT_FAM_MEMBERS', 'REGION_RATING_CLIENT', 'TARGET'
]



# Separate features (X) and target (y)
X = df.drop('TARGET', axis=1)
y = df['TARGET']

# Handle categorical features by one-hot encoding
X = pd.get_dummies(X, dummy_na=False) # Use dummy_na=False to avoid creating a column for NaN
```

What were different k's chosen? What were the differences in the output with those different k's?

Decide K based on elbow curve, silhouette_score plot and feature analysis.

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

```
from sklearn.metrics import silhouette_score


# Select features for clustering
features_for_clustering = [
    'AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE',
    'DAYS_BIRTH', 'DAYS_EMPLOYED', 'EXT_SOURCE_1', 'EXT_SOURCE_2', 'EXT_SOURCE_3',
    'CNT_CHILDREN', 'CNT_FAM_MEMBERS', 'REGION_RATING_CLIENT'
]

# Ensure all selected features exist in the dataframe after one-hot encoding
# Filter for columns that exist in the DataFrame
existing_features = [f for f in features_for_clustering if f in X.columns]
X_clustering = X[existing_features]

# Drop rows with NaN values in the selected features for clustering
X_clustering = X_clustering.dropna()


# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_clustering)

# Experiment with different k values (at least 3)
k_values = [2, 3, 4,5,6,7,8]

# Store results
kmeans_results = {}

for k in k_values:
    print(f"\nRunning KMeans for k = {k}")
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
```

```
    kmeans.fit(X_scaled)
    labels = kmeans.labels_
    inertia = kmeans.inertia_
    silhouette_avg = silhouette_score(X_scaled, labels) if k > 1 else None

    # Add cluster labels to the original dataframe (aligned by index)
    X_clustering_with_labels = X_clustering.copy()
    X_clustering_with_labels['Cluster'] = labels

    kmeans_results[k] = {
        'labels': labels,
        'inertia': inertia,
        'silhouette_score': silhouette_avg,
        'cluster_profiles': X_clustering_with_labels.groupby('Cluster')[existing_features].mean()
    }

    print(f"Inertia for k={k}: {inertia:.2f}")
    if silhouette_avg is not None:
        print(f"Silhouette Score for k={k}: {silhouette_avg:.2f}")
    print(f"Cluster Profiles for k={k}:\n{kmeans_results[k]['cluster_profiles']}")


# Analyze the results for different k values
# Inertia: Measures how spread out the clusters are. Lower is better.
# Silhouette Score: Measures how similar a sample is to its own cluster compared to other clusters. Higher is bette


print("\nSummary of Results:")
for k, result in kmeans_results.items():
    # Format the silhouette score conditionally
    silhouette_str = f"{result['silhouette_score']:.2f}" if result['silhouette_score'] is not None else 'N/A'
    print(f"k={k}: Inertia={result['inertia']:.2f}, Silhouette Score={silhouette_str}")
```

```
Running KMeans for k = 2
Inertia for k=2: 554056.40
Silhouette Score for k=2: 0.17
Cluster Profiles for k=2:
         AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY  AMT_GOODS_PRICE  \
Cluster
0               158239.92   412781.07     21679.32        368015.21
1               214379.45   921931.88     37220.71        837143.08

         DAYS_BIRTH  DAYS_EMPLOYED  EXT_SOURCE_1  EXT_SOURCE_2  EXT_SOURCE_3  \
Cluster
0         -13082.11        2304.03          0.42          0.50          0.47
1         -17291.59       77007.01          0.63          0.58          0.54

         CNT_CHILDREN  CNT_FAM_MEMBERS  REGION_RATING_CLIENT
Cluster
0                0.69             2.44                  2.12
1                0.27             2.02                  1.97

Running KMeans for k = 3
Inertia for k=3: 482037.82
Silhouette Score for k=3: 0.21
Cluster Profiles for k=3:
         AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY  AMT_GOODS_PRICE  \
Cluster
0               146808.63   564753.89     24821.67        507863.22
1               157874.65   398416.03     21161.85        355176.83
2               238643.14  1085171.82     42823.38        986811.54
```

```
         DAYS_BIRTH  DAYS_EMPLOYED  EXT_SOURCE_1  EXT_SOURCE_2  EXT_SOURCE_3  \
Cluster
0         -21452.49      363796.95          0.70          0.52          0.55
1         -13493.91       -1752.71          0.45          0.50          0.47
2         -15384.17         146.35          0.57          0.58          0.52

         CNT_CHILDREN  CNT_FAM_MEMBERS  REGION_RATING_CLIENT
Cluster
0                0.05             1.71                  2.11
1                0.60             2.33                  2.11
2                0.49             2.30                  1.94

Running KMeans for k = 4
Inertia for k=4: 424054.86
Silhouette Score for k=4: 0.19
Cluster Profiles for k=4:
         AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY  AMT_GOODS_PRICE  \
Cluster
0               252112.17  1183727.88     45594.05       1080439.76
1               163249.89   423415.58     21969.25        378640.22
2               162547.40   484044.02     24113.94        430660.31
3               145342.22   556965.35     24554.89        500789.59

         DAYS_BIRTH  DAYS_EMPLOYED  EXT_SOURCE_1  EXT_SOURCE_2  EXT_SOURCE_3  \
Cluster
0         -15587.78        2242.30          0.58          0.59          0.53
1         -14106.24       -2045.26          0.47          0.51          0.48
2         -12963.56        -773.73          0.44          0.51          0.48
3         -21470.46      365243.00          0.70          0.52          0.55

         CNT_CHILDREN  CNT_FAM_MEMBERS  REGION_RATING_CLIENT
Cluster
0                0.36             2.18                  1.91
1                0.08             1.68                  2.09
2                1.50             3.47                  2.11
```

```
3                 0.04              1.70                  2.11

Running KMeans for k = 5
Inertia for k=5: 372069.44
Silhouette Score for k=5: 0.19
Cluster Profiles for k=5:
         AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY  AMT_GOODS_PRICE  \
Cluster
0               242233.02  1181484.95     45517.39       1078172.58
1               163154.42   422372.89     21939.25        377735.89
2               162538.85   483710.79     24106.96        430418.83
3               145443.71   557590.30     24572.26        501393.60
4            117000000.00   562491.00     26194.50        454500.00

         DAYS_BIRTH  DAYS_EMPLOYED  EXT_SOURCE_1  EXT_SOURCE_2  EXT_SOURCE_3  \
Cluster
0         -15588.28        2124.57          0.58          0.59          0.53
1         -14100.82       -2043.80          0.47          0.51          0.48
2         -12962.08        -772.13          0.44          0.51          0.48
3         -21470.69      365243.00          0.70          0.52          0.55
4         -12615.00        -922.00          0.46          0.11          0.15

         CNT_CHILDREN  CNT_FAM_MEMBERS  REGION_RATING_CLIENT
Cluster
0                0.36             2.17                  1.92
1                0.08             1.68                  2.09
2                1.50             3.47                  2.11
3                0.04             1.70                  2.11
4                1.00             3.00                  2.00

Running KMeans for k = 6
Inertia for k=6: 340441.38
Silhouette Score for k=6: 0.17
Cluster Profiles for k=6:
         AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY  AMT_GOODS_PRICE  \
Cluster
```

```
0              249866.34  1259698.35     47767.26       1152487.90
1              172362.09   486662.87     23755.56        436078.93
2              145071.22   554535.71     24494.23        498641.21
3           117000000.00   562491.00     26194.50        454500.00
4              158418.10   405375.29     21632.06        361355.53
5              164839.11   507708.63     24986.58        452109.56

         DAYS_BIRTH  DAYS_EMPLOYED  EXT_SOURCE_1  EXT_SOURCE_2  EXT_SOURCE_3  \
Cluster
0         -15303.14        3771.49          0.57          0.58          0.52
1         -16655.35       -2908.50          0.63          0.59          0.56
2         -21489.03      365243.00          0.70          0.52          0.55
3         -12615.00        -922.00          0.46          0.11          0.15
4         -11712.14       -1004.75          0.32          0.42          0.40
5         -12971.96        -667.92          0.44          0.53          0.49

         CNT_CHILDREN  CNT_FAM_MEMBERS  REGION_RATING_CLIENT
Cluster
0                0.39             2.22                  1.93
1                0.12             1.79                  1.97
2                0.04             1.70                  2.11
3                1.00             3.00                  2.00
4                0.17             1.77                  2.19
5                1.57             3.52                  2.10

Running KMeans for k = 7
Inertia for k=7: 324043.82
Silhouette Score for k=7: 0.16
Cluster Profiles for k=7:
         AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY  AMT_GOODS_PRICE  \
Cluster
0               257198.36  1305186.71     48931.74       1197850.04
1               162040.91   427532.57     22535.99        381632.36
2            117000000.00   562491.00     26194.50        454500.00
3               206841.63   923975.13     38639.54        828979.29
4               171964.85   485472.14     23724.00        435079.88
```

```
5              147729.93   341696.28     19047.85        303494.70
6              144485.63   548925.62     24313.77        493439.45

         DAYS_BIRTH  DAYS_EMPLOYED  EXT_SOURCE_1  EXT_SOURCE_2  EXT_SOURCE_3  \
Cluster
0         -15935.67        6780.58          0.59          0.58          0.52
1         -11779.23       -1018.32          0.32          0.44          0.41
2         -12615.00        -922.00          0.46          0.11          0.15
3         -13454.33        -752.24          0.48          0.56          0.51
4         -16788.26       -2939.39          0.64          0.59          0.56
5         -12684.39        -610.48          0.42          0.50          0.48
6         -21493.23      365243.00          0.70          0.52          0.55

         CNT_CHILDREN  CNT_FAM_MEMBERS  REGION_RATING_CLIENT
Cluster
0                0.14             1.93                  1.91
1                0.08             1.66                  2.16
2                1.00             3.00                  2.00
3                1.47             3.41                  2.01
4                0.11             1.79                  1.98
5                1.47             3.41                  2.16
6                0.04             1.70                  2.12

Running KMeans for k = 8
Inertia for k=8: 308990.98
Silhouette Score for k=8: 0.15
Cluster Profiles for k=8:
         AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY  AMT_GOODS_PRICE  \
Cluster
0               259159.47  1320480.64     49427.26       1212663.67
1               208945.68   947666.91     39205.86        851544.58
2               155328.44   460052.67     23398.37        407332.38
3               169672.20   418921.24     22384.35        376040.51
4               150004.53   344495.87     19193.97        306519.52
5               144301.19   547762.71     24289.15        492415.72
```

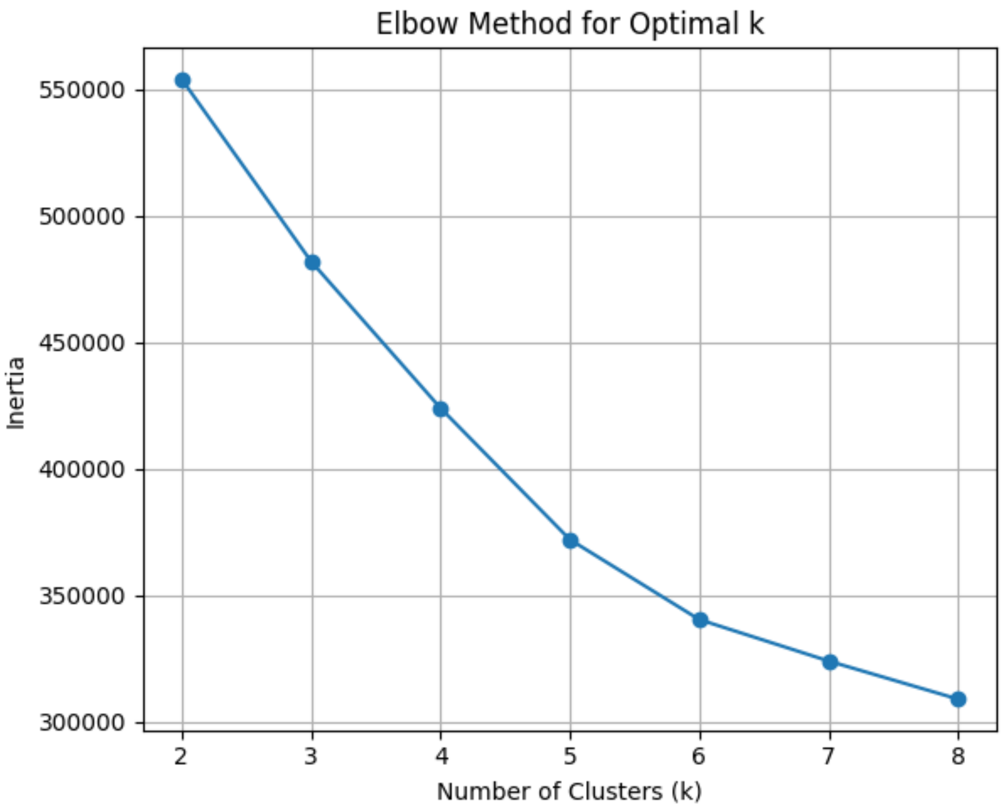
```
6             117000000.00   562491.00     26194.50        454500.00
7                171731.56   503141.89     24103.91        450831.26

         DAYS_BIRTH  DAYS_EMPLOYED  EXT_SOURCE_1  EXT_SOURCE_2  EXT_SOURCE_3  \
Cluster
0         -15963.56        7608.83          0.59          0.59          0.52
1         -13482.80        -774.13          0.49          0.56          0.51
2         -12647.19        -756.44          0.36          0.25          0.41
3         -11792.58       -1433.30          0.34          0.59          0.44
4         -12805.15        -382.69          0.43          0.53          0.49
5         -21500.57      365243.00          0.70          0.52          0.55
6         -12615.00        -922.00          0.46          0.11          0.15
7         -17391.41       -3101.96          0.67          0.59          0.56

         CNT_CHILDREN  CNT_FAM_MEMBERS  REGION_RATING_CLIENT
Cluster
0                0.14             1.93                  1.90
1                1.47             3.41                  2.01
2                0.29             1.97                  2.43
3                0.09             1.64                  1.92
4                1.55             3.50                  2.10
5                0.04             1.70                  2.12
6                1.00             3.00                  2.00
7                0.10             1.78                  2.01

Summary of Results:
k=2: Inertia=554056.40, Silhouette Score=0.17
k=3: Inertia=482037.82, Silhouette Score=0.21
k=4: Inertia=424054.86, Silhouette Score=0.19
k=5: Inertia=372069.44, Silhouette Score=0.19
k=6: Inertia=340441.38, Silhouette Score=0.17
k=7: Inertia=324043.82, Silhouette Score=0.16
k=8: Inertia=308990.98, Silhouette Score=0.15

Choosing final k = 3 for interpretation.
```

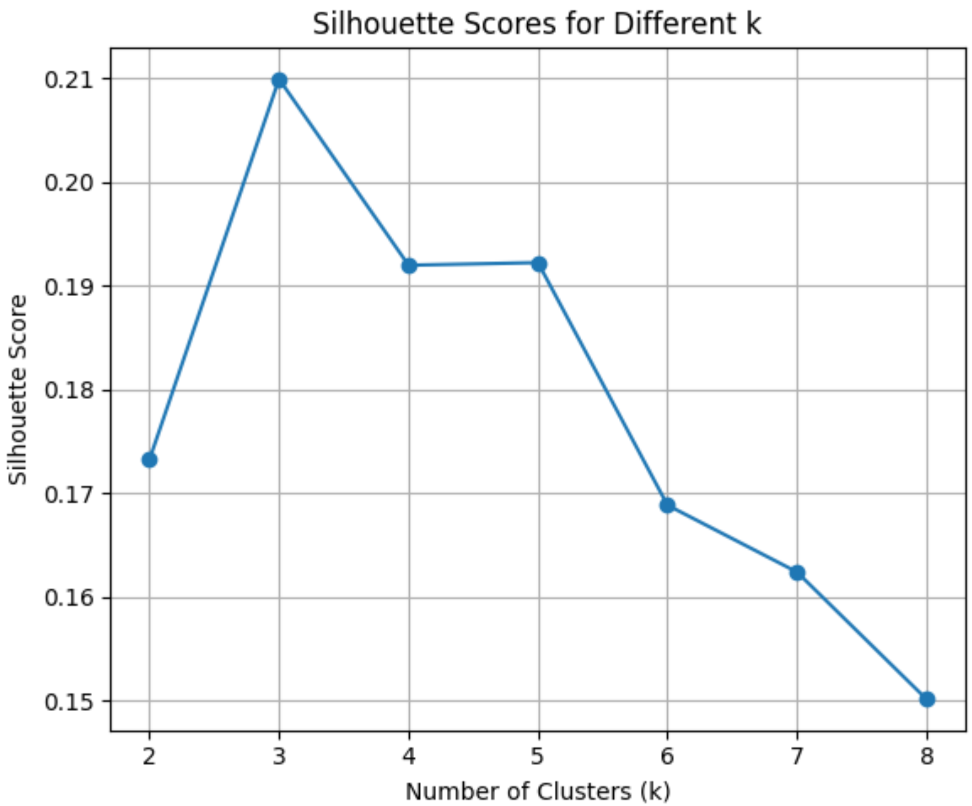
Interpretation of Clusters for k=3:

Cluster	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	DAYS_BIRTH	DAYS_EMPLOYED	EXT_SOURCE_1	EXT_
0	146808.63	564753.89	24821.67	507863.22	-21452.49	363796.95	0.70	
1	157874.65	398416.03	21161.85	355176.83	-13493.91	-1752.71	0.45	
2	238643.14	1085171.82	42823.38	986811.54	-15384.17	146.35	0.57	

```
# Plotting the elbow curve and silhouette score and interpreting the results

import matplotlib.pyplot as plt
# Plot the elbow curve
inertia_values = [result['inertia'] for result in kmeans_results.values()]

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
plt.plot(k_values, inertia_values, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Inertia')
plt.xticks(k_values)
plt.grid(True)

# Plot the silhouette scores
# Filter out k=1 as silhouette score is not defined
silhouette_k_values = [k for k in k_values if k > 1]
silhouette_scores = [kmeans_results[k]['silhouette_score'] for k in silhouette_k_values]
```

```
plt.subplot(1, 2, 2)
plt.plot(silhouette_k_values, silhouette_scores, marker='o')
plt.title('Silhouette Scores for Different k')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Silhouette Score')
plt.xticks(silhouette_k_values)
plt.grid(True)

plt.tight_layout()
plt.show()

# Interpretation of the plots
print("\nInterpretation of Elbow Curve and Silhouette Scores:")
print("- **Elbow Curve:** Look for a point where the rate of decrease in inertia significantly changes (the 'elbow'
print("- **Silhouette Scores:** A higher silhouette score indicates better-defined clusters where samples are well-
"Values close to 1 indicate clear separation, values near 0 indicate overlapping clusters, and negative values indi
print("\nBased on these plots and the business context (interpretability), a final k is  chosen.")
```






```
Interpretation of Elbow Curve and Silhouette Scores:
- **Elbow Curve:** Look for a point where the rate of decrease in inertia significantly changes (the 'elbow'). T
- **Silhouette Scores:** A higher silhouette score indicates better-defined clusters where samples are well-matc

Based on these plots and the business context (interpretability), a final k is  chosen.
```

Based on the elbow plot and silhouette scores:

- **Elbow Method:** Look for the point where the decrease in inertia starts to slow down significantly. In the provided plot, there isn't a perfectly clear "elbow," but the rate of decrease seems to lessen after k=3 or k=4.
- **Silhouette Scores:** The silhouette scores are relatively low for all tested `k` values greater than 1, suggesting that the clusters might not be very distinct. The highest silhouette scores are observed at k=2 and k=3.

Considering both plots and the constraint of keeping `k` less than 10 for interpretability, the most reasonable `k` values to consider based on these metrics are likely **2 or 3**.

3 choice is reasonable given the silhouette score peaks around k=2 and k=3, and an elbow could be argued around these points as well. Choosing 3 allows for a slightly more granular segmentation than 2 while still being reasonably interpretable.

- Describe the characteristics of each cluster based on the mean values of the features.

- Identify influential features by looking at which features show the most significant differences between clusters.
    - Discuss why these groups might have been created (e.g., based on income, credit behavior, age, external scores).
- Draw business inferences (e.g., which cluster might be more risky, which might be good candidates for specific financial products, how to tailor marketing).

Example interpretation structure (replace with actual observations from final_cluster_profiles):

- Cluster 0: Describe characteristics (e.g., lower income, younger, lower external scores). Possible interpretation: Higher risk group.
    - Cluster 1: Describe characteristics (e.g., higher income, older, higher external scores). Possible interpretation: Lower risk group, stable customers.
    - Cluster 2: Describe characteristics (e.g., average income, younger, moderate external scores). Possible interpretation:

Emerging customers.

Why choose this k and distance metric?

- Chosen k based on a balance of inertia/silhouette score and interpretability of the resulting clusters. A smaller k is often preferred for easier interpretation, hence staying below 10.
- The default distance metric for KMeans is Euclidean distance, which is suitable here because the features have been standardized, giving equal weight to each dimension.

  Why these groups were created and influential features:

- Groups were likely created based on a combination of income level, credit history proxies (external scores), age (DAYS_BIRTH is negative, closer to 0 means older), and potentially employment status (DAYS_EMPLOYED is negative, closer to 0 means longer employed).
- Influential features appear to be those with the largest variations in mean values across the clusters (e.g., AMT_INCOME_TOTAL, EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3, DAYS_BIRTH, DAYS_EMPLOYED).

** Business inferences:**

- Identify high-risk and low-risk customer segments based on cluster characteristics and the (excluded) TARGET variable if used for external validation.
- Tailor credit product offerings or marketing strategies to the specific profiles of each cluster.
- Develop targeted risk mitigation strategies for higher-risk clusters.
- Understand the profile of customers who are likely to default (if correlating clusters with the original TARGET variable - note: TARGET was excluded for unsupervised clustering, but can be used for post-hoc analysis of the clusters).

```
choosen_final_k = 3

print(f"\nChoosing final k = {choosen_final_k} for interpretation.")

# Retrieve the results for the final k
final_results = kmeans_results[choosen_final_k]
final_cluster_profiles = final_results['cluster_profiles']

print(f"\nInterpretation of Clusters for k={choosen_final_k}:")
final_cluster_profiles
```

```
Choosing final k = 3 for interpretation.

Interpretation of Clusters for k=3:
```

Cluster	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	DAYS_BIRTH	DAYS_EMPLOYED	EXT_SOURCE_1	EXT_
0	146808.63	564753.89	24821.67	507863.22	-21452.49	363796.95	0.70	
1	157874.65	398416.03	21161.85	355176.83	-13493.91	-1752.71	0.45	
2	238643.14	1085171.82	42823.38	986811.54	-15384.17	146.35	0.57	

```
print(f"\nDecision: Based on the Elbow Method, Silhouette Scores, and the interpretability of the cluster profiles,
```

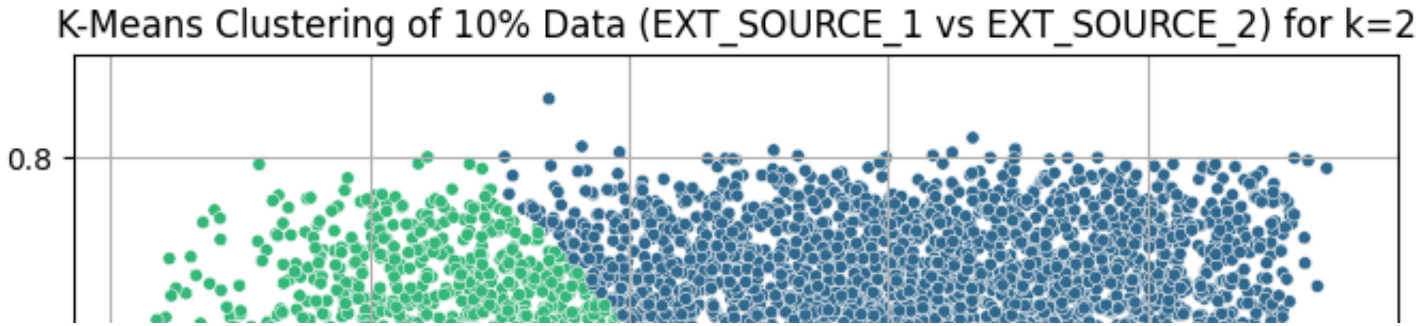
```
Decision: Based on the Elbow Method, Silhouette Scores, and the interpretability of the cluster profiles, the ch
```

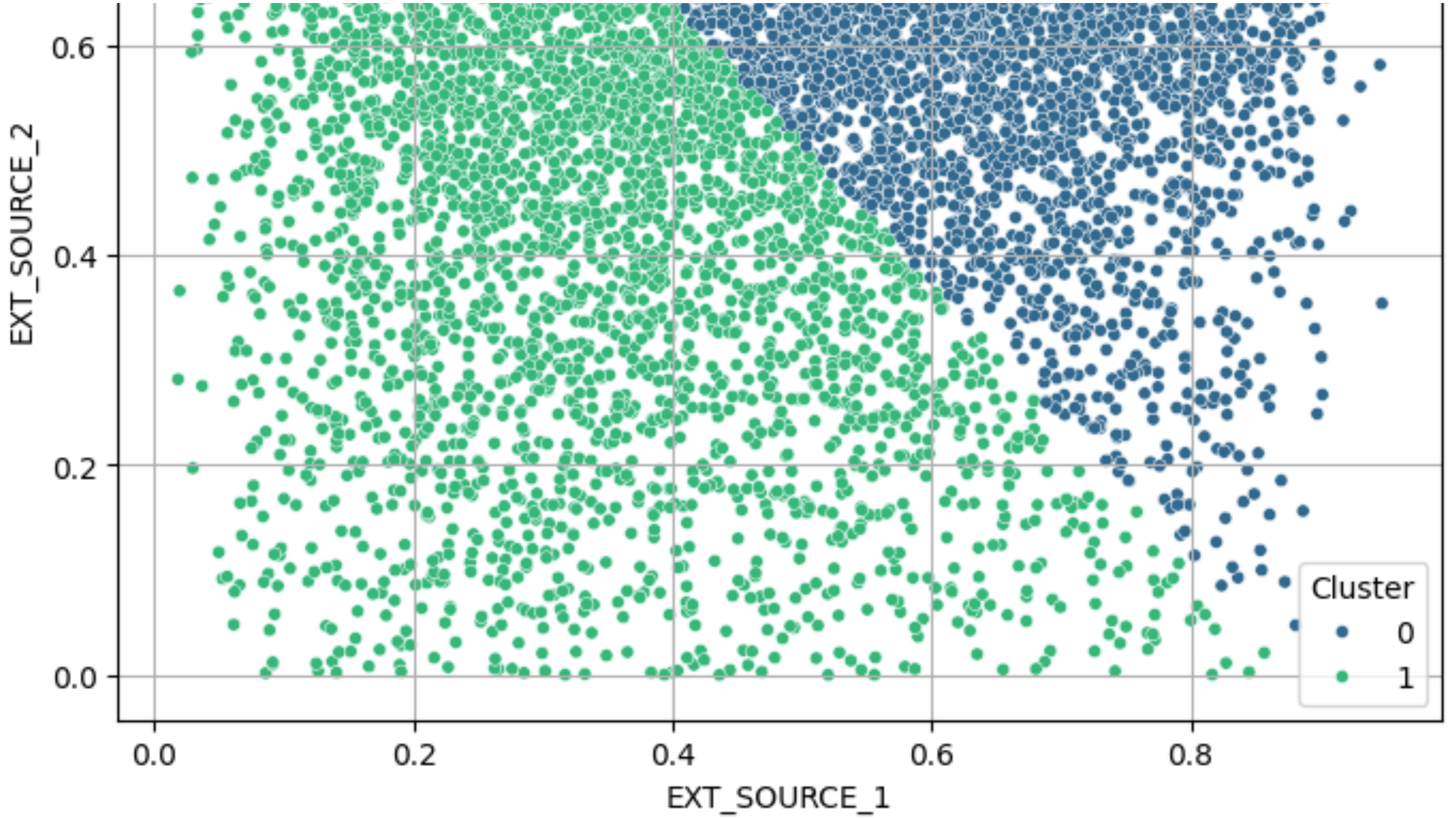
Scatter plots created to understand feaures impact on deciding the cluster size.

```
# EXT_SOURCE_1  EXT_SOURCE_2  use these features and plot scatter for k=2,3,4,5, plot 10% ofpoints remove the centr

import pandas as pd
import matplotlib.pyplot as plt
# Filter out NaN values in the selected features for plotting
df_filtered = df[['EXT_SOURCE_1', 'EXT_SOURCE_2']].dropna().sample(frac=0.1, random_state=42)

# Scale the filtered data for plotting
scaler_plot = StandardScaler()
X_plot_scaled = scaler_plot.fit_transform(df_filtered)

# Assign labels for each k value
for k in [2, 3, 4, 5]:
    print(f"\nRunning KMeans for k = {k} for plotting")
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(X_plot_scaled)
    df_filtered[f'Cluster_k{k}'] = kmeans.labels_
```

```
# Get centroids (scaled)
centroids_scaled = kmeans.cluster_centers_

# Inverse transform centroids to original scale for plotting
centroids_original = scaler_plot.inverse_transform(centroids_scaled)
centroids_df = pd.DataFrame(centroids_original, columns=['EXT_SOURCE_1', 'EXT_SOURCE_2'])


# Plotting
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df_filtered, x='EXT_SOURCE_1', y='EXT_SOURCE_2', hue=f'Cluster_k{k}', palette='viridis', l

# Plot centroids (removed as requested)
# plt.scatter(centroids_df['EXT_SOURCE_1'], centroids_df['EXT_SOURCE_2'], color='red', s=100, marker='X', label=

plt.title(f'K-Means Clustering of 10% Data (EXT_SOURCE_1 vs EXT_SOURCE_2) for k={k}')
plt.xlabel('EXT_SOURCE_1')
plt.ylabel('EXT_SOURCE_2')
plt.legend(title='Cluster')
plt.grid(True)
plt.show()
```

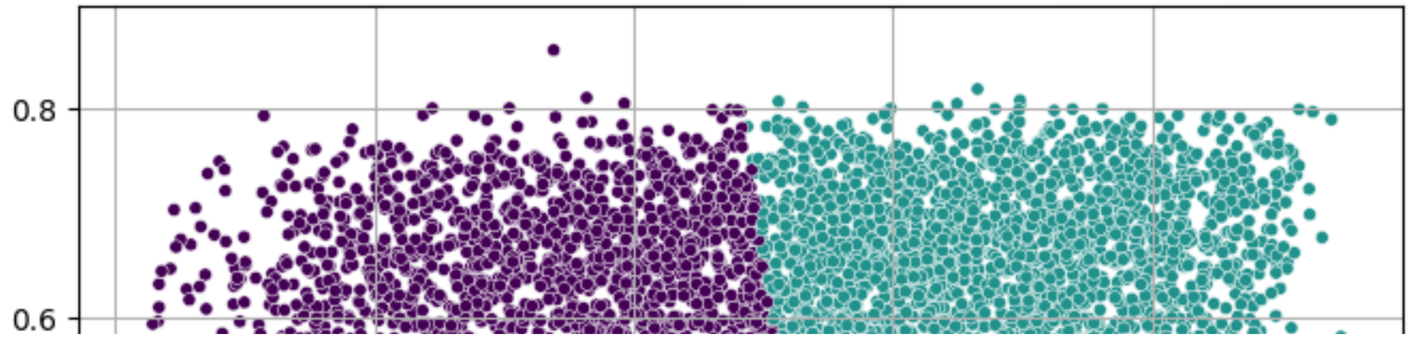
```
Running KMeans for k = 2 for plotting
```

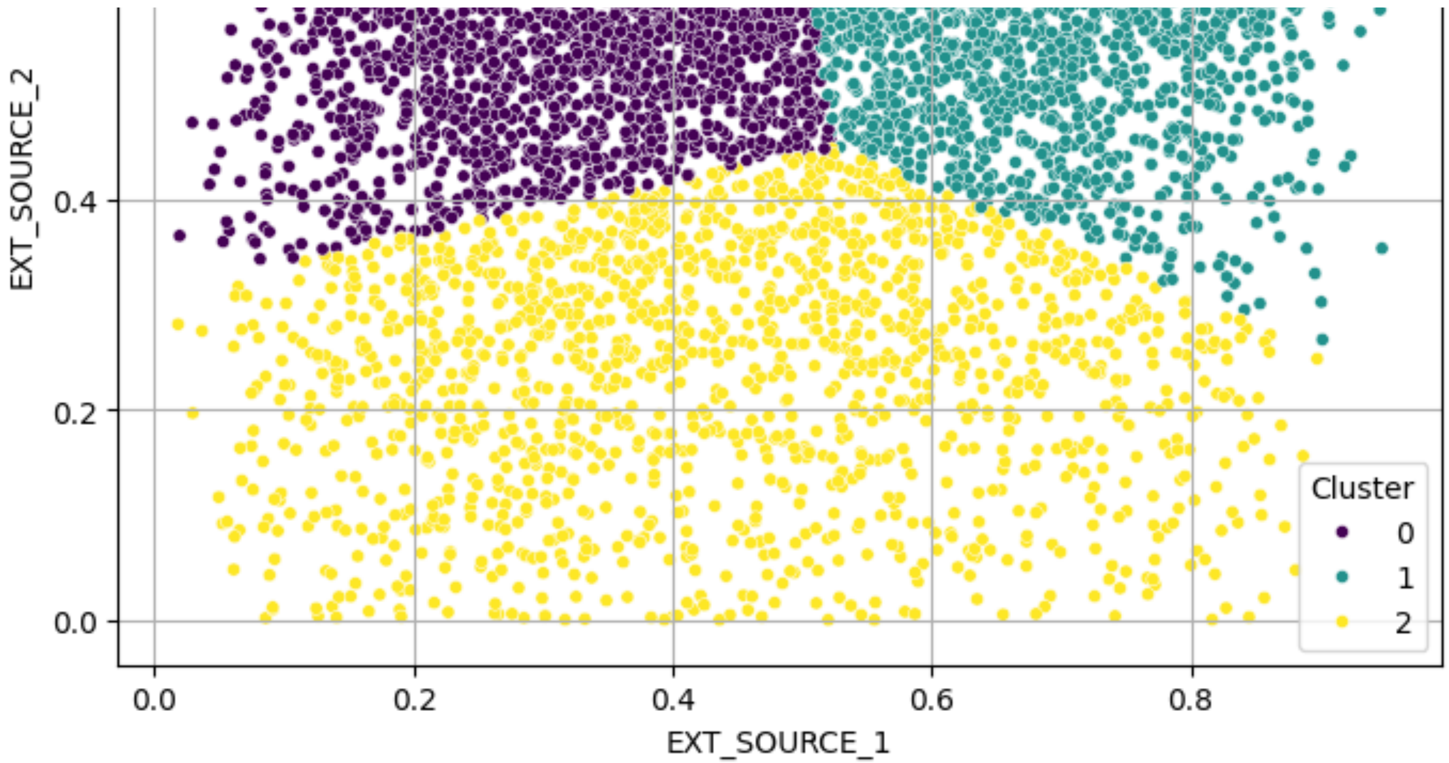




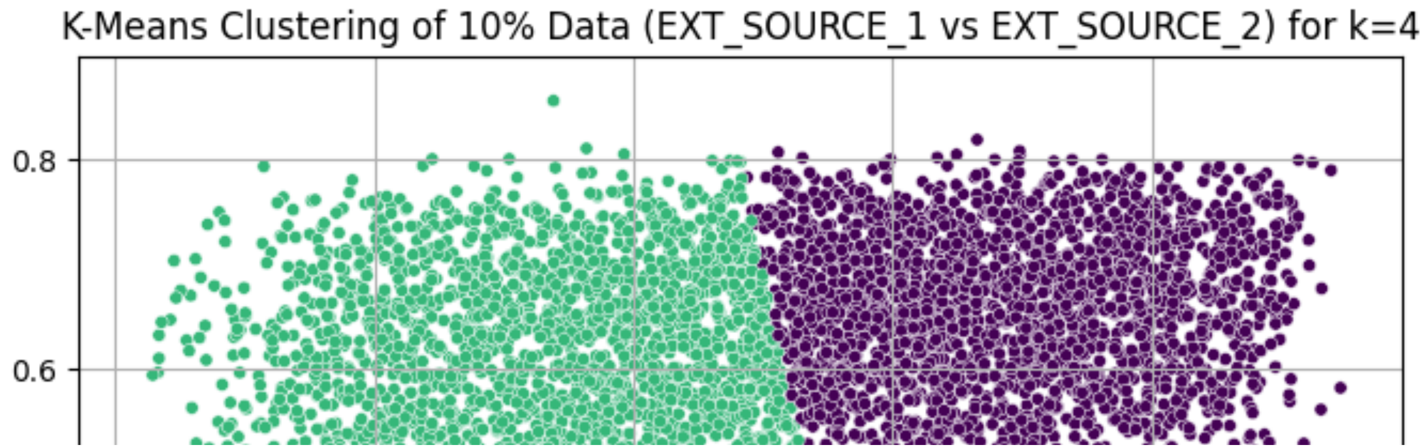

```
Running KMeans for k = 3 for plotting
```

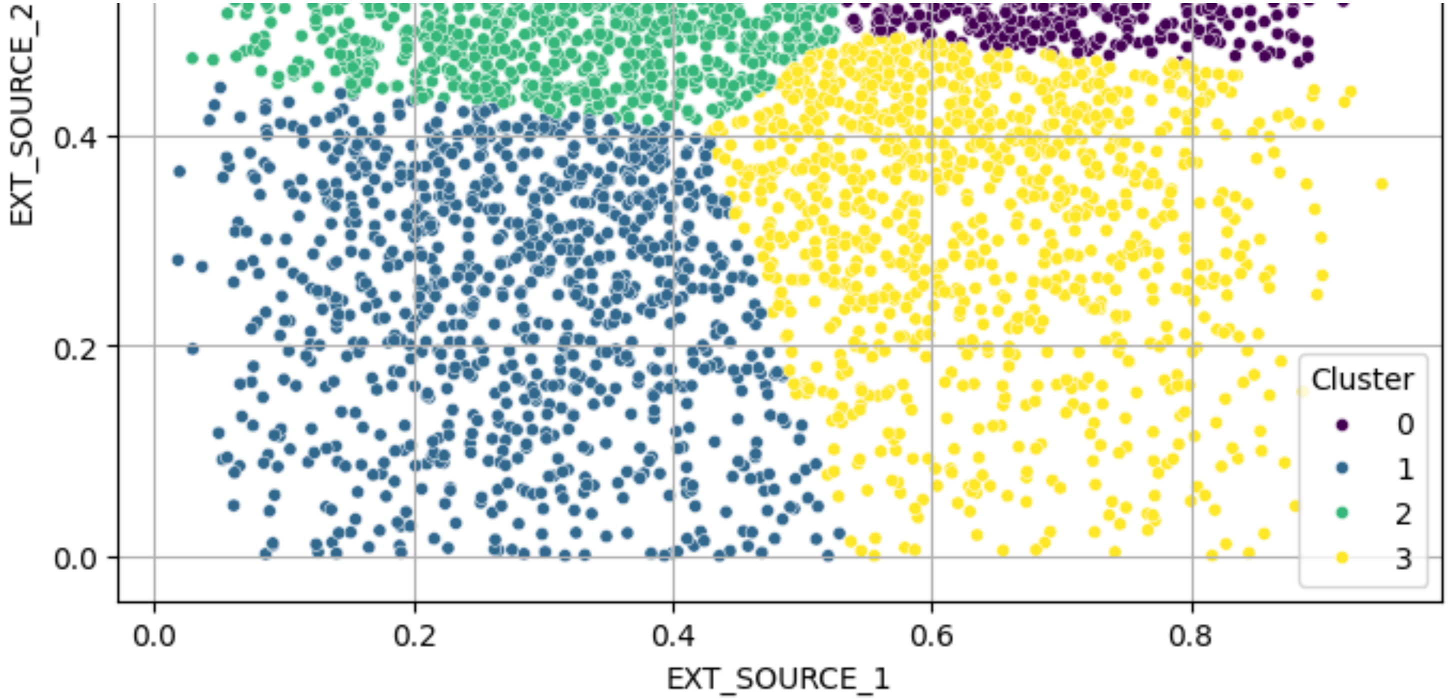
K-Means Clustering of 10% Data (EXT_SOURCE_1 vs EXT_SOURCE_2) for k=3



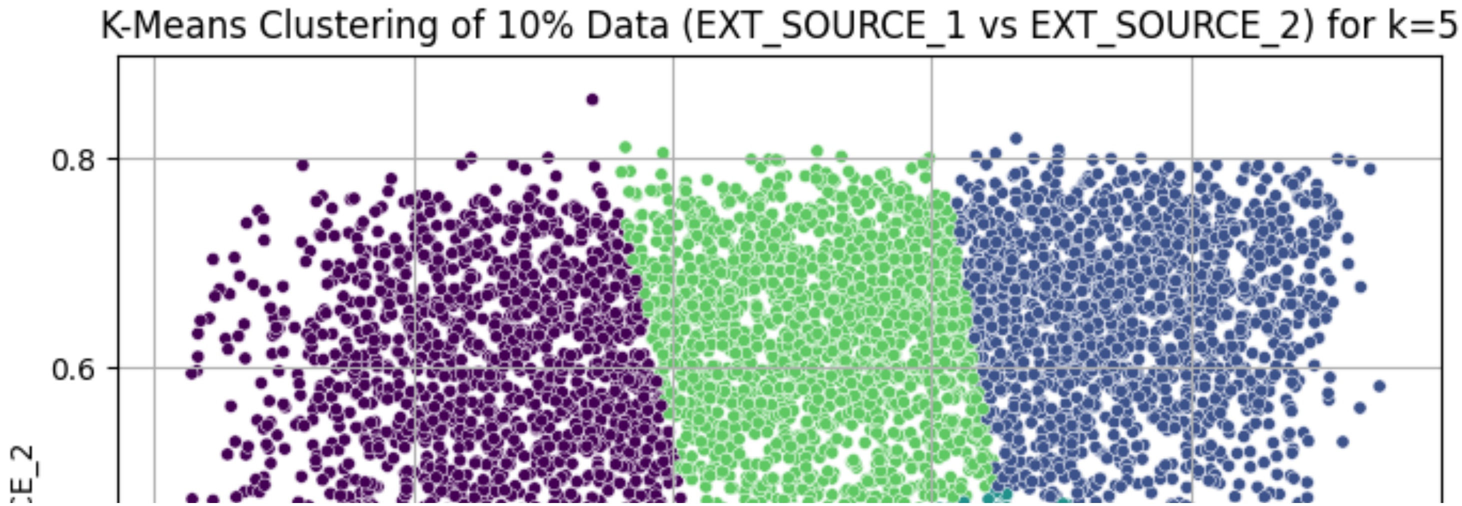


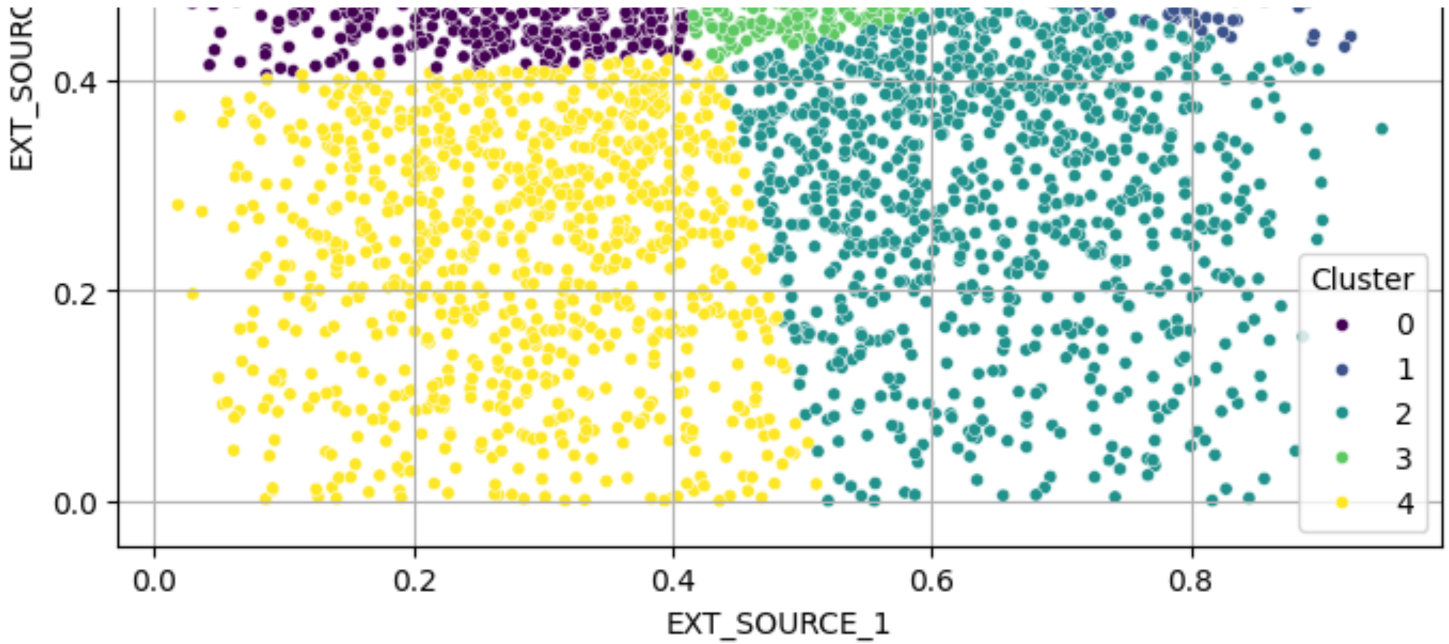
Running KMeans for k = 4 for plotting






```
Running KMeans for k = 5 for plotting
```

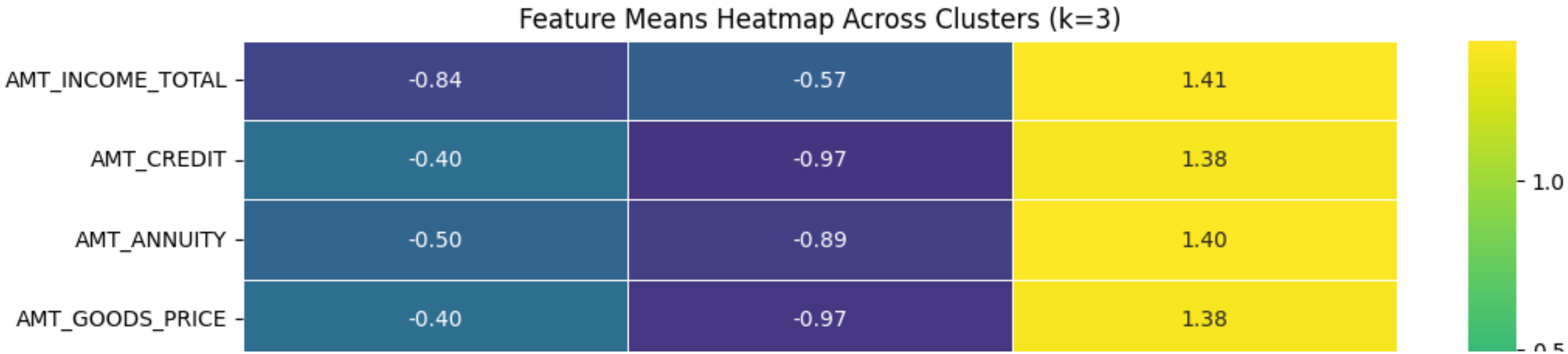


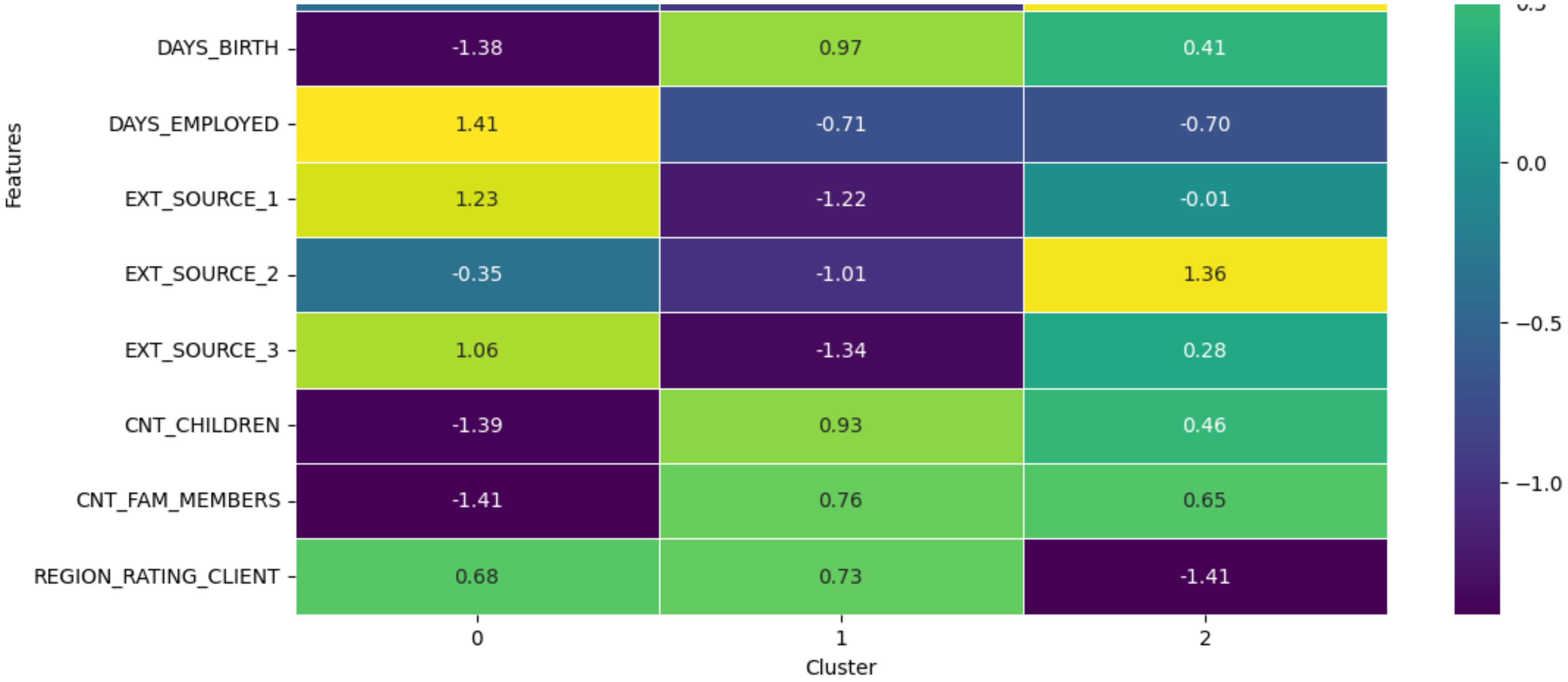



```
# create  heatmap per cluster and analyse

import pandas as pd
import matplotlib.pyplot as plt

final_k = 3
print(f"\nGenerating heatmap for clusters at k = {final_k}")

# Retrieve the cluster labels for the final_k
final_labels = kmeans_results[final_k]['labels']



X_clustering_with_final_labels = X_clustering.copy()
X_clustering_with_final_labels['Cluster'] = final_labels
```

```
# Calculate the mean values of features for each cluster
cluster_heatmap_data = X_clustering_with_final_labels.groupby('Cluster')[existing_features].mean()

# Scale the mean values for better visualization in the heatmap
scaler_heatmap = StandardScaler()
cluster_heatmap_scaled = scaler_heatmap.fit_transform(cluster_heatmap_data)
cluster_heatmap_scaled_df = pd.DataFrame(cluster_heatmap_scaled, columns=cluster_heatmap_data.columns, index=cluster_

# Create the heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(cluster_heatmap_scaled_df.T, annot=True, cmap='viridis', fmt=".2f", linewidths=.5)
plt.title(f'Feature Means Heatmap Across Clusters (k={final_k})')
plt.xlabel('Cluster')
plt.ylabel('Features')
plt.show()
```

```
Generating heatmap for clusters at k = 3
```




Features
DAYS_BIRTH
DAYS_EMPLOYED
EXT_SOURCE_1
EXT_SOURCE_2
EXT_SOURCE_3
CNT_CHILDREN
CNT_FAM_MEMBERS
REGION_RATING_CLIENT
-1.38
0.97
0.41
1.41
-0.71
-0.70
1.23
-1.22
-0.01
-0.35
-1.01
1.36
1.06
-1.34
0.28
-1.39
0.93
0.46
-1.41
0.76
0.65
0.68
0.73
-1.41
0
1
2
Cluster
0.0
−0.5
−1.0

```
print("\n\033[1;34mHeatmap Analysis:\033[0m") # Blue color
print("\033[1;34mThe heatmap visualizes the standardized mean values of each feature across the different clusters 
print("\033[1;34mStandardizing the means helps compare the relative importance of features in differentiating clust
print("\033[1;34mPositive values (warmer colors) indicate features with mean values above the overall average for tl
print("\033[1;34mNegative values (cooler colors) indicate features with mean values below the overall average for tl
print("\033[1;34mLarge absolute values (bright colors) highlight features that significantly distinguish one cluste
```

```
Heatmap Analysis:
The heatmap visualizes the standardized mean values of each feature across the different clusters for the chosen
Standardizing the means helps compare the relative importance of features in differentiating clusters.
Positive values (warmer colors) indicate features with mean values above the overall average for that cluster.
Negative values (cooler colors) indicate features with mean values below the overall average for that cluster.
Large absolute values (bright colors) highlight features that significantly distinguish one cluster from others.
```

```
print("\n\033[1m\033[4m\033[94mInterpretation of the Heatmap:\033[0m")
print("\033[90mThe heatmap displays the scaled mean values of each feature for each cluster at the chosen k (which 
print("\033[90mScaling the means (using StandardScaler) allows us to see how many standard deviations away from the
print("\033[90mThis helps identify features that are particularly high or low within a cluster relative to the rest

# Based on the heatmap generated from the cluster_heatmap_scaled_df:
print("\n\033[1m\033[94mObservations from the Heatmap (assuming k=3 visualization):\033[0m")

print("\n\033[1m\033[92mCluster 0:\033[0m")
print("- \033[90mHigh positive values in `DAYS_EMPLOYED`: Indicates significantly longer time since employed (likel
print("- \033[90mNegative values in `DAYS_BIRTH`: Slightly less negative means older age compared to the overall av
print("- \033[90mHigh positive value in `EXT_SOURCE_1`: Indicates a strong score from this external source.\033[0m"
```

```
print("- \033[90mNegative values in `EXT_SOURCE_2`, `EXT_SOURCE_3`, `CNT_CHILDREN`, `CNT_FAM_MEMBERS`, `REGION_RATII
print("- \033[90mModerate financial values (`AMT_INCOME_TOTAL`, etc.): Closer to the average.\033[0m")
print("\033[1m\033[92mInterpretation: This cluster is characterized by older, potentially long-term unemployed indi


print("\n\033[1m\033[92mCluster 1:\033[0m")
print("- \033[90mHigh negative values in `DAYS_BIRTH`: Indicates significantly younger age.\033[0m")
print("- \033[90mHigh negative values in `DAYS_EMPLOYED`: Indicates currently employed with a shorter tenure.\033[0
print("- \033[90mHigh positive values in `CNT_CHILDREN`, `CNT_FAM_MEMBERS`: Indicates larger families.\033[0m")
print("- \033[90mModerate positive values in financial metrics (`AMT_INCOME_TOTAL`, `AMT_CREDIT`, etc.): Indicates 
print("- \033[90mModerate values in `EXT_SOURCE_1`, `EXT_SOURCE_2`, `EXT_SOURCE_3`, `REGION_RATING_CLIENT`: Closer 
print("\033[1m\033[92mInterpretation: This cluster appears to represent younger, currently employed individuals wit

print("\n\033[1m\033[92mCluster 2:\033[0m")
print("- \033[90mHigh positive values in financial metrics (`AMT_INCOME_TOTAL`, `AMT_CREDIT`, `AMT_ANNUITY`, `AMT_G
print("- \033[90mHigh positive values in `EXT_SOURCE_1`, `EXT_SOURCE_2`: Indicates strong scores from these externa
print("- \033[90mNegative values in `DAYS_BIRTH`: Less negative means older age, but perhaps less pronounced than C
print("- \033[90mVary low negative values in `DAYS_EMPLOYED`: Indicates currently employed with very short tenure (
print("- \033[90mNegative values in `CNT_CHILDREN`, `CNT_FAM_MEMBERS`: Indicates smaller families.\033[0m")
print("- \033[90mHighest value in `REGION_RATING_CLIENT`: Indicates the worst regional rating (closer to 3). *Corre
print("\033[1m\033[92mInterpretation: This cluster is characterized by high-income individuals with large credit ne

print("\n\033[1m\033[4m\033[94mOverall Interpretation Reinforcement from Heatmap:\033[0m")
print("\033[90mThe heatmap clearly shows which features are most influential in separating the clusters.\033[0m")
print("\033[90m`DAYS_EMPLOYED`, `DAYS_BIRTH`, the `EXT_SOURCE` features, and the financial amounts (`AMT_INCOME_TOT
print("\033[90mCluster 0 is distinct due to long-term employment situation and `EXT_SOURCE_1`.\033[0m")
print("\033[90mCluster 1 is characterized by younger age, shorter employment tenure, and larger families.\033[0m")
print("\033[90mCluster 2 stands out with high income, large credit amounts, strong external scores, and better regi

print("\n\033[1m\033[4m\033[94mBusiness Inferences based on Heatmap & Profiles:\033[0m")
print("\033[90m- Cluster 0: May represent a higher-risk segment due to potential long-term unemployment. Further an
print("\033[90m- Cluster 1: Younger families with potential growing financial needs. Could be a good target for fut
```

```
print("\033[90m- Cluster 2: High-value customers with significant borrowing needs and strong credit indicators. Lik

print("\n\033[1m\033[4m\033[94mConclusion:\033[0m")
print("\033[90mThe k=3 clustering reveals distinct segments based on financial health, age, employment status, exte
print("\033[90mThese segments can be used to tailor lending strategies, risk assessment models, and marketing effor
print("\033[90mThe heatmap provides a concise visual summary of the key characteristics of each cluster.\033[0m")
```

```
Interpretation of the Heatmap:
The heatmap displays the scaled mean values of each feature for each cluster at the chosen k (which was set to 3
Scaling the means (using StandardScaler) allows us to see how many standard deviations away from the overall fea
This helps identify features that are particularly high or low within a cluster relative to the rest of the data

Observations from the Heatmap (assuming k=3 visualization):

Cluster 0:
- High positive values in `DAYS_EMPLOYED`: Indicates significantly longer time since employed (likely long-term
- Negative values in `DAYS_BIRTH`: Slightly less negative means older age compared to the overall average.
- High positive value in `EXT_SOURCE_1`: Indicates a strong score from this external source.
- Negative values in `EXT_SOURCE_2`, `EXT_SOURCE_3`, `CNT_CHILDREN`, `CNT_FAM_MEMBERS`, `REGION_RATING_CLIENT`:
- Moderate financial values (`AMT_INCOME_TOTAL`, etc.): Closer to the average.
Interpretation: This cluster is characterized by older, potentially long-term unemployed individuals with high 

Cluster 1:
- High negative values in `DAYS_BIRTH`: Indicates significantly younger age.
- High negative values in `DAYS_EMPLOYED`: Indicates currently employed with a shorter tenure.
- High positive values in `CNT_CHILDREN`, `CNT_FAM_MEMBERS`: Indicates larger families.
- Moderate positive values in financial metrics (`AMT_INCOME_TOTAL`, `AMT_CREDIT`, etc.): Indicates above-averag
- Moderate values in `EXT_SOURCE_1`, `EXT_SOURCE_2`, `EXT_SOURCE_3`, `REGION_RATING_CLIENT`: Closer to the overa
Interpretation: This cluster appears to represent younger, currently employed individuals with larger families a

Cluster 2:
```

- High positive values in financial metrics (`AMT_INCOME_TOTAL`, `AMT_CREDIT`, `AMT_ANNUITY`, `AMT_GOODS_PRICE`)
- High positive values in `EXT_SOURCE_1`, `EXT_SOURCE_2`: Indicates strong scores from these external sources.
- Negative values in `DAYS_BIRTH`: Less negative means older age, but perhaps less pronounced than Cluster 0.
- Vary low negative values in `DAYS_EMPLOYED`: Indicates currently employed with very short tenure (potentially
- Negative values in `CNT_CHILDREN`, `CNT_FAM_MEMBERS`: Indicates smaller families.
- Highest value in `REGION_RATING_CLIENT`: Indicates the worst regional rating (closer to 3). *Correction based

Interpretation: This cluster is characterized by high-income individuals with large credit needs, strong externa

Overall Interpretation Reinforcement from Heatmap:

The heatmap clearly shows which features are most influential in separating the clusters.
`DAYS_EMPLOYED`, `DAYS_BIRTH`, the `EXT_SOURCE` features, and the financial amounts (`AMT_INCOME_TOTAL`, etc.) a
Cluster 0 is distinct due to long-term employment situation and `EXT_SOURCE_1`.
Cluster 1 is characterized by younger age, shorter employment tenure, and larger families.
Cluster 2 stands out with high income, large credit amounts, strong external scores, and better regional ratings

Business Inferences based on Heatmap & Profiles:

- Cluster 0: May represent a higher-risk segment due to potential long-term unemployment. Further analysis on th
- Cluster 1: Younger families with potential growing financial needs. Could be a good target for future product
- Cluster 2: High-value customers with significant borrowing needs and strong credit indicators. Likely lower ri

Conclusion:

The k=3 clustering reveals distinct segments based on financial health, age, employment status, external credit
These segments can be used to tailor lending strategies, risk assessment models, and marketing efforts.
The heatmap provides a concise visual summary of the key characteristics of each cluster.

Business Relevance Summary

Risk Management:

Identify which clusters are **higher risk** and adjust credit limits, interest rates, or approval criteria accordingly.

Personalization:

Tailor product offerings by **demographic and financial needs**.

Operational Efficiency:

Allocate customer service and support more effectively (e.g., proactive outreach to **high-risk segments**).

Marketing Strategy:

Target **stable clusters** with upselling or cross-selling; educate and retain **young borrowers**.

```
#END
#END
```

```
#END
```