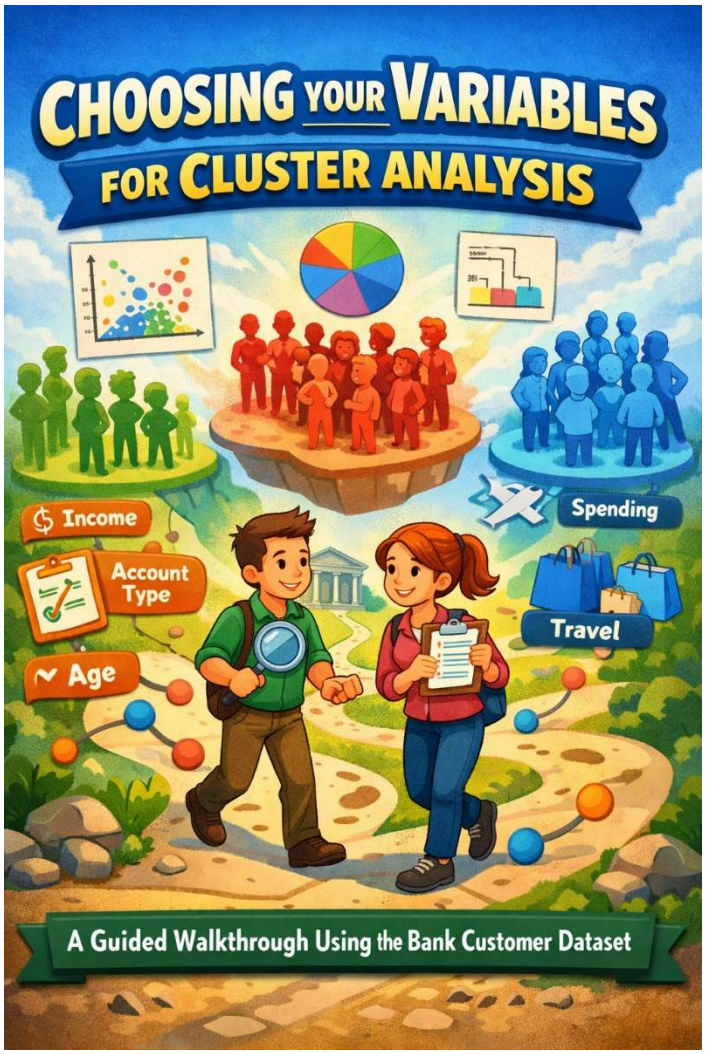
CHOOSING YOUR VARIABLES FOR CLUSTER ANALYSIS

A Guided Walkthrough Using the Bank Customer Dataset

In-Class Activity & Demo

Predictive Analytics & Data Mining

Dahlkemper School of Business | Gannon University



Developed by Dr. Benyawarath “Yaa” Nithithanatchinnapat

Spring 2026

Contents

Part 1: Your Turn First

Before I show you my thinking, I want you to wrestle with this decision yourself. That's where the real learning happens.

> **The Scenario**
> You're a data analyst at a regional bank. Leadership wants to stop treating all 800 customers the same and start offering personalized services. Your job: use clustering to find natural customer segments. You have 28 variables to work with. You need to decide which ones go into the clustering algorithm.

Here's What You Have

Below is the complete list of variables in the bank customer dataset. Your task: decide which ones to include in your cluster analysis and which ones to leave out.

#	Variable	Type	Description
1	**Customer_ID**	Text	Unique identifier (e.g., CUST-00142)
2	**Age**	Numeric	Customer's age in years
3	**Gender**	Categorical	Male, Female, or Non-Binary
4	**Marital_Status**	Categorical	Single, Married, Divorced, Widowed
5	**Education**	Ordinal	High School through Doctorate
6	**Region**	Categorical	Northeast, Southeast, Midwest, Southwest, West
7	**Annual_Income**	Numeric	Estimated annual income ($)
8	**Credit_Score**	Numeric	FICO credit score (550–850)
9	**Account_Type**	Categorical	Checking Only, Savings Only, or Both
10	**Tenure_Years**	Numeric	Years as a customer
11	**Account_Balance**	Numeric	Current balance ($)
12	**Num_Products**	Numeric	Number of bank products held (1–5)
13	**Has_Credit_Card**	Binary	1 = Yes, 0 = No
14	**Has_Mortgage**	Binary	1 = Yes, 0 = No
15	**Has_Personal_Loan**	Binary	1 = Yes, 0 = No
16	**Has_Auto_Loan**	Binary	1 = Yes, 0 = No
17	**Has_Investment_Account**	Binary	1 = Yes, 0 = No
18	**Monthly_Transactions**	Numeric	Number of transactions per month
19	**Avg_Transaction_Amount**	Numeric	Average dollar value per transaction
20	**Monthly_Deposits**	Numeric	Total deposits per month ($)

21	**Monthly_Withdrawals**	Numeric	Total withdrawals per month ($)
22	**Net_Monthly_Flow**	Numeric	Deposits minus withdrawals ($)
23	**Overdraft_Count_12mo**	Numeric	Overdrafts in the past 12 months
24	**Online_Logins_Month**	Numeric	Online banking logins per month
25	**Mobile_Sessions_Month**	Numeric	Mobile app sessions per month
26	**Branch_Visits_Year**	Numeric	In-person branch visits per year
27	**Customer_Service_Calls_Year**	Numeric	Service calls in the past year
28	**Satisfaction_Score**	Numeric	Self-reported satisfaction (1–10)

✍ Team Exercise (10 minutes)

With your team, make three lists:

IN — Variables you'd include in the clustering algorithm
OUT — Variables you'd exclude entirely
MAYBE — Variables you're unsure about

For each decision, write a one-sentence reason WHY.
Be ready to defend your choices to the class!

Part 2: The Six-Question Filter

There's no magic button that picks variables for you. But there IS a systematic way to think about it. For every variable in your dataset, run it through these six questions. If a variable can't pass all six, it doesn't make the cut.

> **Why Not Just Throw Everything In?**
> Two big reasons. First, the curse of dimensionality: with too many features, distances between points become more uniform, and the algorithm can't tell who's similar to whom. Second, noisy or irrelevant variables dilute the signal from the variables that actually matter, producing clusters that are fuzzy and hard to act on.

Question 1: Does It Relate to My Business Question?

Our question is: *"What distinct groups exist among our customers based on how they bank with us?"* Every variable we include should help distinguish different banking behaviors, needs, or value levels.

The test: If two customers score differently on this variable, does that suggest they should be served differently?

✓ **PASS:** Account_Balance — A customer with $5,000 and a customer with $150,000 absolutely need different services.

✗ **FAIL:** Customer_ID — It's just a label. CUST-00142 vs. CUST-00587 tells us nothing about behavior.

Question 2: Does It Actually Vary Across Customers?

A variable that's nearly the same for everyone can't help separate groups. If 98% of customers have a credit card, the Has_Credit_Card flag barely moves the needle — it's almost a constant.

The test: Check the distribution. If one value dominates (say, 90%+), the variable won't contribute much to cluster formation.

✓ **PASS:** Age — Ranges from 22 to 78 with a healthy spread. Plenty of variation to work with.

⚠ **BORDERLINE:** Individual product flags (Has_Credit_Card, Has_Mortgage, etc.) — Each one alone is binary and somewhat skewed, but combined they become meaningful.

Question 3: Is It Redundant With Something I'm Already Using?

If two variables are highly correlated ($r > 0.8$), they're essentially voting twice for the same dimension. That gives that one aspect of your data double the weight in the distance calculation.

The test: Run a correlation matrix. If you see a pair above 0.80, keep the one that's more interpretable or more complete — or combine them into a single composite.

⚠ **REDUNDANCY ALERT:** Monthly_Deposits and Monthly_Withdrawals are both strongly correlated with Annual_Income. And we already have Net_Monthly_Flow which captures the difference. Including all three would triple-count the "money movement" dimension.

⚠ **REDUNDANCY ALERT:** Online_Logins_Month and Mobile_Sessions_Month are correlated — digitally engaged customers use both. Combine them into a single Digital_Engagement score.

Question 4: Is It Numeric (or Can I Make It Numeric Meaningfully)?

K-Means calculates Euclidean distance — it needs numbers. Categorical variables like Region or Gender don't have a natural distance. "Northeast" isn't 3 units away from "Midwest."

The test: Is it already a number? If categorical, can you encode it meaningfully (like Education's ordinal scale), or is it better used for post-hoc profiling?

✓ **PASS:** All our numeric and binary variables work directly with K-Means.

✗ **FAIL FOR K-MEANS:** Gender, Marital_Status, Region, Account_Type — No natural distance metric. Better used to describe clusters after the fact.

> **The Profile-After Trick**
> Excluding Gender and Region from clustering doesn't mean we ignore them. After we find clusters, we cross-tabulate: "Cluster 2 is 65% female, mostly from the Southeast." This gives us demographic color without distorting the distance math.

Question 5: Is It an ID, Timestamp, or Near-Unique Field?

This sounds obvious, but it trips up students every semester. Any variable that's unique (or nearly unique) per row will cause the algorithm to treat every customer as completely different from every other customer. That defeats the whole purpose of clustering.

✗ **ALWAYS EXCLUDE:** Customer_ID, account numbers, email addresses, timestamps, row indices.

Question 6: Am I Covering Multiple Dimensions Without Overloading Any One?

The best clustering features span different aspects of customer behavior. If you include five income-related variables and one engagement variable, you're essentially telling the algorithm that income matters five times more than engagement. The clusters will just be income tiers — not very insightful.

The test: Group your candidate variables into dimensions (value, behavior, channel, risk, experience). Aim for roughly equal representation.

Dimension	What It Captures	Target: 2–3 Variables
Demographics	Lifecycle stage, life situation	Age, (Education optional)
Financial Capacity	Ability to use products and manage credit	Annual_Income, Credit_Score
Relationship Depth	How embedded the customer is with the bank	Tenure_Years, Total_Products
Account Value	How much the customer is worth to the bank	Account_Balance, Net_Monthly_Flow
Activity Level	How actively they use banking services	Monthly_Transactions, Avg_Txn_Amount
Channel Preference	How they prefer to interact with the bank	Digital_Engagement, Branch_Visits
Financial Stress	Signs of difficulty or risk	Overdraft_Count_12mo
Experience	How they feel about the bank	Satisfaction_Score

Part 3: The Full Walkthrough — Every Variable, Every Decision

Now let's apply the six questions to all 28 variables. I'll show my reasoning for each one. Compare this to the choices your team made — where did you agree? Where did you disagree? Why?

Variable	Verdict	Reasoning
Customer_ID	**OUT**	Fails Q5. It's a unique label, not a behavior. Including it would make every customer look maximally different from every other customer.
Gender	**OUT**	Fails Q4. No natural numeric distance for K-Means. More useful as a profiling variable after clustering. We want behavior-driven clusters, not demographic buckets.
Marital_Status	**OUT**	Fails Q4. Same issue as Gender — categorical with no ordinal meaning. "Married" isn't 2 units from "Single." Use for profiling after.
Education	**OUT**	Borderline. It IS ordinal (High School < Bachelor's < Doctorate), so encoding is possible. But it's partly redundant with Income and already well-represented by that dimension. Keep it for profiling.
Region	**OUT**	Fails Q4. Nominal with no distance meaning. Also fails Q1 partially — region alone doesn't tell you how someone banks. Profile after.
Account_Type	**OUT**	Fails Q4. Three categories with no natural distance. Account depth is already captured by Num_Products and Account_Balance.
Monthly_Deposits	**OUT**	Fails Q3. Highly correlated with Annual_Income ($r \approx 0.85$). And the difference (deposits – withdrawals) is already captured by Net_Monthly_Flow. Including it triple-counts the money dimension.
Monthly_Withdrawals	**OUT**	Fails Q3. Same redundancy issue as Monthly_Deposits. Net_Monthly_Flow already captures the saving-vs-spending behavior we care about.
Customer_Service_Calls	**OUT**	Borderline, but OUT. Partially redundant with Satisfaction_Score (unhappy customers call more). Also ambiguous — high calls could mean problems OR could mean complex, high-value customers needing more touch. Keep it for profiling.
Age	**IN**	Passes all six. Strong lifecycle indicator: a 25-year-old and a 65-year-old have fundamentally different banking needs, product preferences, and channel habits.
Annual_Income	**IN**	Passes all six. Core financial capacity metric. A $35K earner and a $200K earner need completely different product suites. Represents the "financial capacity" dimension.

Credit_Score	**IN**	Passes all six. Measures creditworthiness independently from income. Some high earners have poor credit; some moderate earners have excellent credit. This distinction matters for product eligibility.
Tenure_Years	**IN**	Passes all six. Loyalty and relationship maturity. Partially correlated with Age, but not perfectly — a 35-year-old who's banked here 12 years is very different from a 35-year-old who joined 6 months ago.
Account_Balance	**IN**	Passes all six. The single best measure of customer value to the bank. Ranges from $500 to $250K — enormous variation that should absolutely drive segmentation.
Num_Products → Total_Products_Held	**IN***	IN as a composite. Instead of five separate binary flags (credit card, mortgage, etc.), we sum them into one Total_Products_Held score. This captures cross-sell depth in a single dimension rather than giving product-mix five votes.
Has_Credit_Card through Has_Investment_Account	**COMBINE**	Each binary flag alone is too thin (just 0 or 1). Together they form Total_Products_Held. The individual flags are still useful for profiling: "Cluster 1 is 85% credit card holders but only 15% have investments."
Monthly_Transactions	**IN**	Passes all six. Activity volume — how often the customer uses the bank. A customer making 8 transactions/month vs. 60 transactions/month has a very different relationship with the bank.
Avg_Transaction_Amount	**IN**	Passes all six. Spending power per transaction. Complementary to Monthly_Transactions — someone who makes 10 large purchases is different from someone who makes 60 small ones.
Net_Monthly_Flow	**IN**	Passes all six. Captures the savings trajectory: positive = growing their account, negative = drawing down. More informative than raw deposits/withdrawals because it shows the NET behavior.
Online_Logins + Mobile_Sessions → Digital_Engagement	**IN***	IN as a composite. Both measure digital channel preference and are correlated with each other. Combining them into one Digital_Engagement score prevents double-counting the "digital" dimension.
Branch_Visits_Year	**IN**	Passes all six. The counterpart to Digital_Engagement. Together they capture the full channel-preference spectrum from all-digital to all-branch. NOT redundant — they're actually negatively correlated.
Overdraft_Count_12mo	**IN**	Passes all six. Financial stress signal. Most customers have 0–1, but the ones with 3–4 represent a distinct at-risk group that the bank needs to handle differently. Low variance, HIGH business value.
Satisfaction_Score	**IN**	Passes all six. Experience quality metric. A high-balance customer with satisfaction of 3/10 is a flight risk. A low-

		balance customer with 9/10 is a future growth candidate. This dimension is unique.

Part 4: The Final Feature Set

After running every variable through the six-question filter, here's what we end up with — 13 clustering features covering 8 distinct dimensions:

#	Clustering Feature	Dimension	What It Captures
1	**Age**	Demographics	Lifecycle stage
2	**Annual_Income**	Financial Capacity	Earning power
3	**Credit_Score**	Financial Capacity	Creditworthiness
4	**Tenure_Years**	Relationship Depth	Loyalty and maturity
5	**Account_Balance**	Account Value	Customer dollar value
6	**Total_Products_Held ***	Relationship Depth	Cross-sell breadth
7	**Monthly_Transactions**	Activity Level	Usage frequency
8	**Avg_Transaction_Amount**	Activity Level	Spending power per use
9	**Net_Monthly_Flow**	Account Value	Saving vs. spending trend
10	**Digital_Engagement ***	Channel Preference	Online + mobile usage
11	**Branch_Visits_Year**	Channel Preference	Traditional channel usage
12	**Overdraft_Count_12mo**	Financial Stress	Risk / distress signal
13	**Satisfaction_Score**	Experience	Happiness with the bank

** Composite features we engineered: Total_Products_Held = sum of 5 product flags; Digital_Engagement = Online_Logins + Mobile_Sessions*

The Scorecard

28 original variables → 13 clustering features. We excluded 10 (1 ID, 5 categoricals, 2 redundant flows, 1 ambiguous, 1 redundant with composite). We combined 7 into 2 composites. 13 features across 8 dimensions gives us balanced coverage without overloading any one aspect.

Part 5: What Happens to the Excluded Variables?

Excluding a variable from clustering doesn't mean throwing it away. The excluded variables play a crucial role AFTER clustering as profiling variables.

The Two-Phase Approach

Phase 1 — Clustering (behavioral features only): Let the algorithm find groups based on what customers DO. This produces behavior-driven segments that the bank can actually act on.

Phase 2 — Profiling (add demographics back in): Cross-tabulate the clusters with the excluded variables. Now you can say: “Cluster 3 tends to be married, college-educated, and from the Southeast.” This adds richness without distorting the clustering.

Example of Phase 2 profiling output:

Profiling Variable	Cluster 0	Cluster 1	Cluster 2	Cluster 3
% Female	52%	48%	55%	46%
% Married	23%	72%	68%	33%
Top Region	West	Northeast	Southeast	Midwest
Top Education	Bachelor’s	Master’s	Some College	Bachelor’s

See the difference? The clusters were formed based on behavior (balance, transactions, digital engagement). Then the demographics layer on top to give each cluster a face. Marketing can now say: “Our Digital-First Achievers tend to be unmarried, college-educated professionals in the West — let’s target them through Instagram and app-based promotions.”

Part 6: Common Mistakes to Avoid

Mistake 1: "More Variables = Better Clusters"

It's tempting to think that giving the algorithm more data is always better. In supervised learning (classification, regression), more relevant features often do help. But clustering is unsupervised — there's no target variable to keep things on track. Extra irrelevant or redundant features just add noise and make every customer look equally far apart.

The sweet spot: 5–15 well-chosen features for most business clustering problems.

Mistake 2: Leaving Customer_ID in the Feature Set

This happens more often than you'd think. The ID is unique per row, so the algorithm sees 800 completely different data points. Result: either 800 clusters of 1, or clusters that make no sense because they're driven by alphanumeric ID patterns.

Mistake 3: Mixing Categoricals Into K-Means Without Encoding

K-Means uses Euclidean distance. If you've assigned Gender as Male = 1, Female = 2, Non-Binary = 3, the algorithm thinks Non-Binary is "farther" from Male than Female is. That's mathematically meaningless. Either one-hot encode properly (and accept the sparsity), or keep categoricals out and profile after.

Mistake 4: Not Scaling Before Clustering

If Annual_Income is in the thousands and Satisfaction_Score is on a 1–10 scale, income will dominate every distance calculation. A $1,000 income difference will outweigh a 5-point satisfaction difference, even though the satisfaction gap is huge on its scale. Always standardize (z-score) first.

Mistake 5: Ignoring Correlated Features

Including Monthly_Deposits, Monthly_Withdrawals, Annual_Income, and Account_Balance all at once means the "money" dimension gets four votes while "channel preference" gets two. Your clusters will essentially just be income tiers dressed up as segments.

Part 7: Debrief Discussion

Now that you've seen the full walkthrough, let's compare notes.

Team Discussion Questions

1. Where did your team's variable choices agree with this walkthrough? Why do you think those were obvious?

2. Where did you disagree? Can you make a case for your choice being equally valid?

3. Which of the “six questions” was hardest for your team to apply? Why?

4. If the business question changed — say, we’re now segmenting for credit risk instead of marketing — which variables would you add or remove?

5. A colleague argues: “Why not just throw everything in and let PCA reduce it?” What’s the advantage of deliberate selection BEFORE running PCA?

🚀 The Bottom Line

Variable selection isn’t a math problem — it’s a judgment call. The algorithm doesn’t know your business question. It doesn’t know that overdraft count matters more than shoe size. YOU bring that judgment. The six-question filter is your tool for making that judgment systematically instead of guessing.