The University of Texas at Austin
Department of Mathematics
College of Natural Sciences

Foundational Techniques for Machine Learning and Data Science

K-means and Gaussian Mixture Models for Clustering Credit Card Data

Final Project

Addie Duncan, Paulina Hoyos, Liangchen Liu, Will Porteous
Department of Mathematics
April 22, 2022

Outline

- Credit Card Dataset
- Data Preprocessing
- Clustering Algorithms
  - K-means
  - Gaussian Mixture Models
- Clustering Error Metrics
- Implementation and Results
- Conclusions

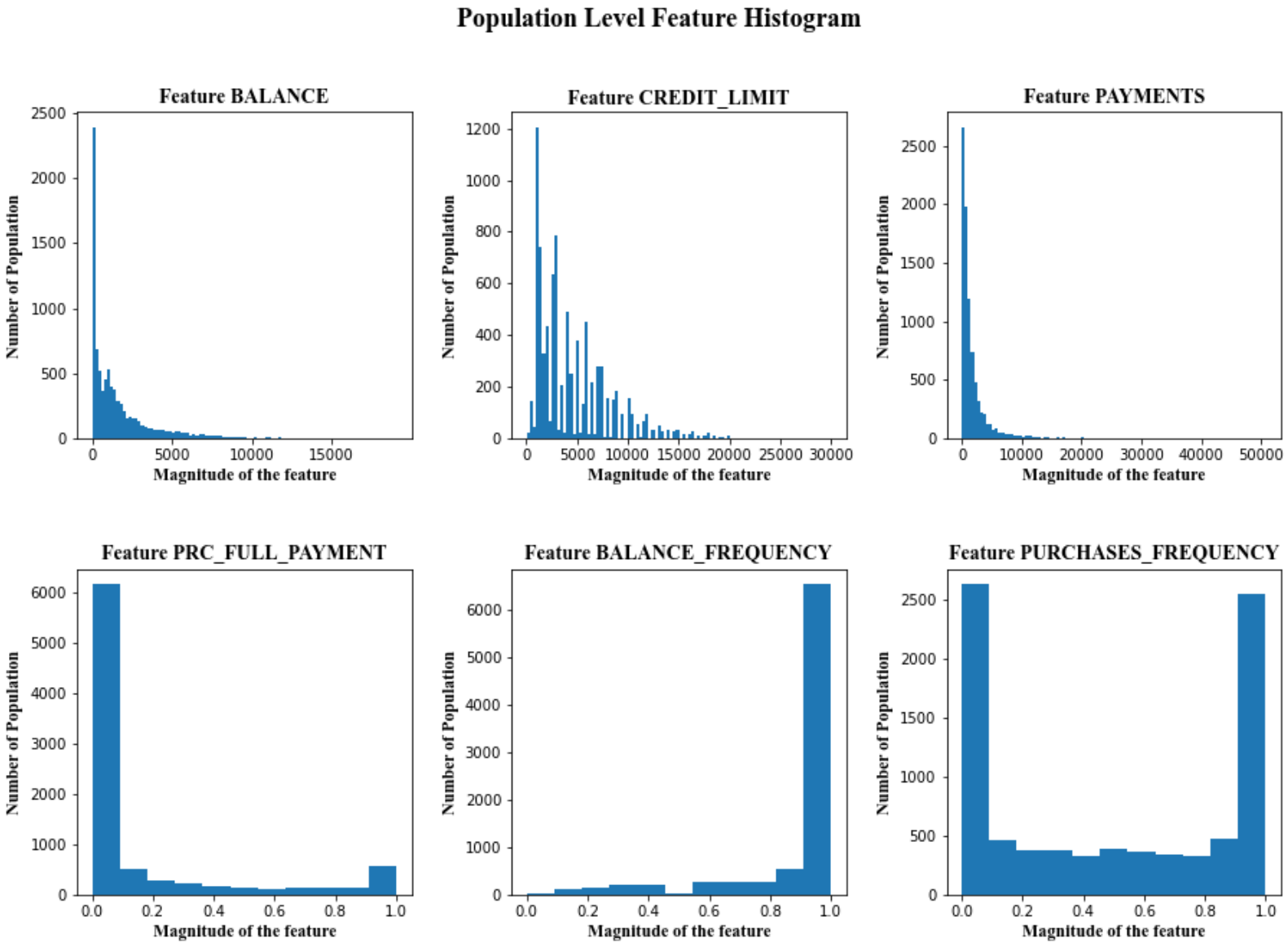
Knowing your data is important

- Anonymized credit card user data over 6 months of time, from Kaggle.
- Data matrix X with $n = 8950$ rows and $d = 18$ columns.
- d features for each customer:

Measured in USD	Credit Limit Purchases Installment Purchases Payments	Balance One off Purchases (Max.) Cash Advance Minimum Payments
Frequency $\in [0, 1]$	Balance Frequency One off Purchase Frequency Cash Advance Frequency	Purchase Frequency Installment Purchase Frequency Percent of Full Payments
Nonnegative integers	Purchases Transactions Tenure	Cash Advance Transactions Customer ID

- We use the first $d = 14$ features in our clustering. We also remove customers with incomplete data and keep the remaining $n \approx 8600$ customers.

But the data can still be naughty…



What can you do?

- No dimension reduction: $d \leq 18$ is small
  - No computational necessity
  - No theoretical guarantees for k-means cost if $d < k$
- Main issues:
  - Outliers, dominate Euclidean metric
  - Scaling across features are different.
- Two solutions:
  - Standardization and outlier removal
  - Quantization

Outlier Removal and Standardization

1. For each feature, remove top and bottom 1% of the data. (The data remaining are within the middle 98% of the data in every feature)
2. Rescale each feature so that they take value in $[0, 1]$.

Although being naive, only $7 - 9\%$ of the total data is removed (as opposed to $2\% * 14$ (features) $= 28\%$) , meaning the data (customers) thrown out are outliers in several features.

Quantization

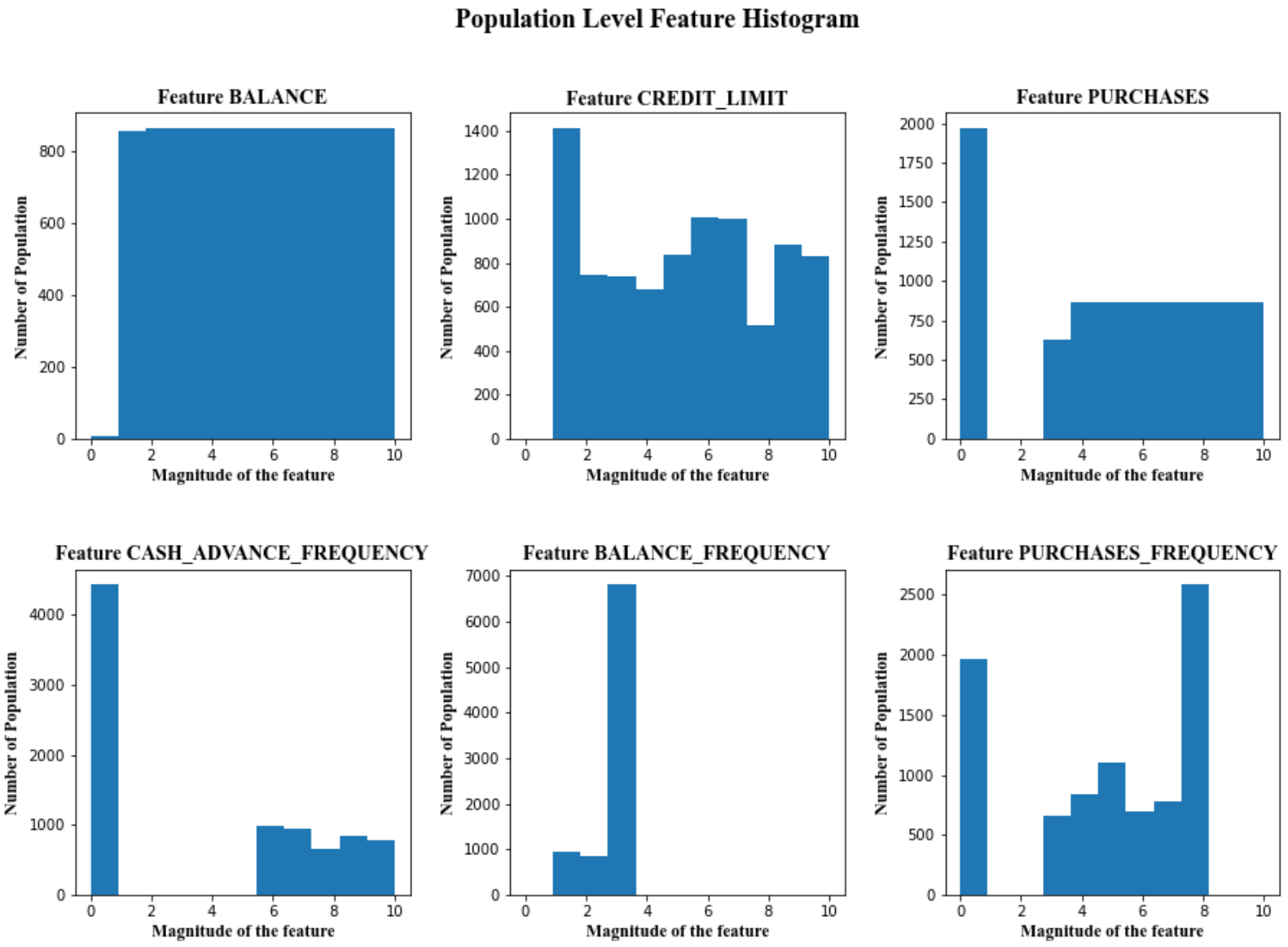
1. Divide each feature into 11 quantiles by *population percentage* and assign each quantile a value $0^*, 1, \ldots, 10$ (the first 10% is assigned 1, the second is 2, etc).
2. If the first k quantiles all have original value 0, assign 0 to those quantiles, and start assigning next non-zero quantile from $k+1$* e.g. $[0, 0, 0, 4, 5, \ldots]$

*Some of the features have more than 50% valued at 0.

Why Quantization?

- Removes influence of outliers without removing them.
- Sets all features to take values $\in [0, \ldots, 10]$ so that each feature is weighted equally in the clustering.

Now the data is nicely behaved



K-means

- $\mathcal{X} = \{x_1, x_2, \ldots, x_n\} \subseteq \mathbb{R}^d$ data points.
- We want to find a partition $P = C_1 \cup \cdots \cup C_k$ of $\mathcal{X}$ and a set of means $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$ such that the cost function

$$\text{Cost}_{\mathcal{X}}(C_1, \ldots, C_k) = \min_{C_1,\ldots,C_k} \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

is minimized.

Gaussian Mixture Models (GMMs)

- A multivariate Gaussian random variable $X \in \mathrm{R}^n$ has pdf

$$f(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right).$$

  We denote this as $X \sim \mathcal{N}(\mu,\Sigma)$.
- A Gaussian Mixture Model is a finite convex combination of multivariate Gaussians, denoted as

$$X \sim \sum_{j=1}^{m} \lambda_j \mathcal{N}(\mu_j,\Sigma_j),$$

  with $\lambda_j \in [0,1]$ and $\sum_{j=1}^{m} \lambda_j = 1$.
- λ_j is the probability of drawing from the j-th component $\mathcal{N}(\mu_j,\Sigma_j)$.

Parameter Estimation of a GMM

- Consider the data points $\{x_1, x_2, \ldots, x_n\}$ as n independent realizations a GMM $X \sim \sum_{j=1}^{m} \lambda_j \mathcal{N}(\mu_j, \Sigma_j)$ whose parameters $\theta = \{(\lambda_j, \mu_j, \Sigma_j)\}_{j=1}^{m}$ we ignore.
- We want to efficiently estimate the parameters θ so that the joint pdf of $x_1, \ldots, x_n$, called the likelihood function,

$$\mathcal{L}(\theta; x_1, \ldots, x_n) = \prod_{j=1}^{n} f_\theta(x_j),$$

  where f_θ is the pdf of X, is maximized.
- The Maximum Likelihood Estimate (MLE) of θ is

$$\begin{aligned} \theta_{MLE} &= \arg\max \mathcal{L}(\theta; x_1, \ldots, x_n) \\ &= \arg\max \log \mathcal{L}(\theta; x_1, \ldots, x_n). \end{aligned}$$

Expectation Maximization

- The log-likelihood function is
$$\log \mathcal{L}(\theta; x_1, \ldots, x_n) = \sum_{i=1}^{n} \log f_\theta(x_i).$$
- We lower bound each term
$$\log f_\theta(x_i) \geq \sum_{j=1}^{m} w_{j,i}^{(0)} \log f(x_i; \mu_j, \Sigma_k) = Q_{x_i}(\theta \mid \theta^{(0)}).$$

Expectation Maximization Algorithm

Input: Implementations of the E-step $E(\cdot)$ and M-step $M(\cdot, \cdot)$
Output: A parameter θ_{MLE}

1. Initialize $\theta^{(1)}$
2. For $t = 1, 2, \ldots, T$ do
3. $\quad Q(\cdot \mid \theta^{(t)}) \leftarrow E(\theta^{(t)})$
4. $\quad \theta^{(t+1)} \leftarrow M(\theta^{(t)}, Q_t)$
5. return $\theta^{(T+1)}$

Expectation Maximization

- E-step: Given the current estimate of the parameters $\theta^{(t)}$, we compute weights as

$$w_{j,i}^{(t)} = \frac{\lambda_j^{(t)} f(x_i; \mu_j^{(t)}, \Sigma_j^{(t)})}{\sum_{k=1}^{m} \lambda_k^{(t)} f(x_i; \mu_k^{(t)}, \Sigma_k^{(t)})},$$

and then compute the Q-function as

$$Q(\theta \mid \theta^{(t)}) = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{j,i}^{(t)} \log f(x_i; \mu_j, \Sigma_j).$$

- M-step: Maximizes the Q-function to generate the next iterate of the parameters

$$\theta^{(t+1)} = \arg\max Q(\theta \mid \theta^{(t)}).$$

Cluster Assignment with GMM

- Clusters are determined by the Gaussian components $\mathcal{N}(\mu_j, \Sigma_j)$ for $j \in \{1, \ldots, m\}$.

- The E-step finds the weight $w_{j,i}^{(T)}$ encoding the probability that the data point x_i belongs to the cluster C_j given by the jth Gaussian $\mathcal{N}(\mu_j, \Sigma_j)$:

$$w_{j,i}^{(T)} = \frac{\lambda_j^{(T)} f(x_i; \mu_j^{(t)}, \Sigma_j^{(T)})}{\sum_{k=1}^{m} \lambda_k^{(T)} f(x_i; \mu_k^{(T)}, \Sigma_k^{(T)})}$$

- We select the cluster C_j that maximizes this probability.

Geometry of GMM Clusters

GMM clusters have an ellipsoidal shape that varies according to the covariance matrices Σ_j. There are three cases:

1. $\Sigma_j = \sigma_j \cdot \mathrm{Id}$
2. $\Sigma_j = \mathrm{diag}(\sigma_{ji})$
3. Full Σ

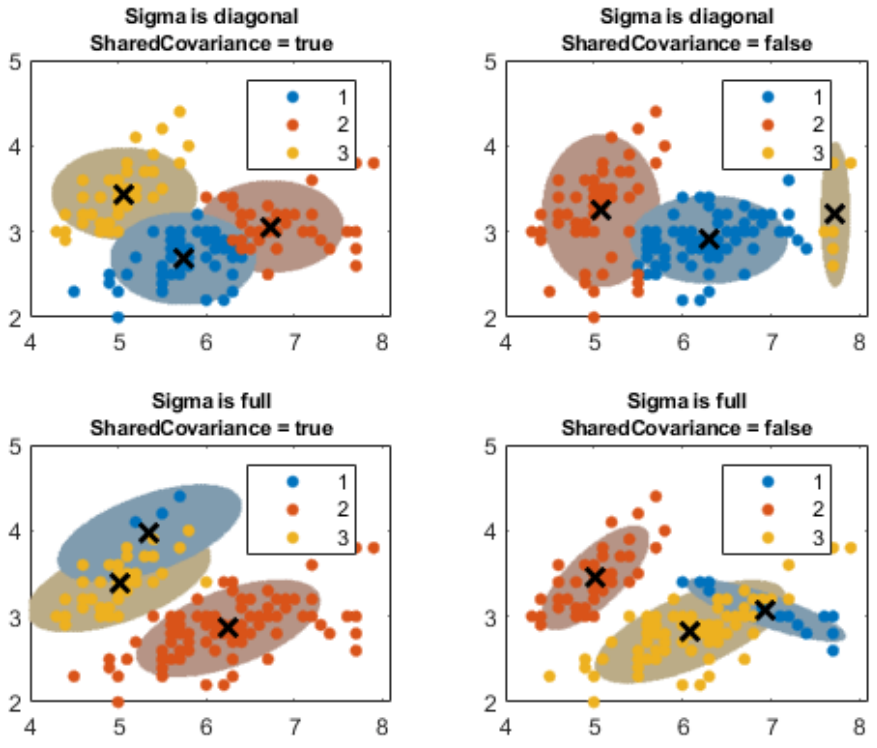


Figure from https://www.mathworks.com/help/stats/clustering-using-gaussian-mixture-models.html

GMM and Data Quantization

Quantization defines a sample-dependent map

$$Q_{\mathcal{D}} : \mathbb{R}^d \to (\{0, \ldots, 10\})^d \text{ via } Q(x_1, \ldots, x_d) = (Q^1(x_1), \ldots, Q^d(x_d))$$

Quantized Problem

Given data $\mathcal{D}' = \{Q(x_i)\}_{i=1}^p \in (\{0, \ldots, 10\})^d$ sampled from a random variable $Q(X) \in \{0, \ldots, 10\}^d \subset \mathbb{R}^d$ we seek

$$f_{\theta_0} = \sum_{j=1}^{m} \lambda_j \mathcal{N}(\mu_j, \Sigma_j) \sim X \in \mathbb{R}^d \quad \text{for } \theta_0 \in \arg\max \log \mathcal{L}(\theta; \mathcal{D})$$

Clustering Error Metrics - Inertia

Inertia: Within-Cluster-Sum-of-Squares

$$\sum_{j=0}^{k} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

where μ_j is the center of the j^{th} cluster, C_j.

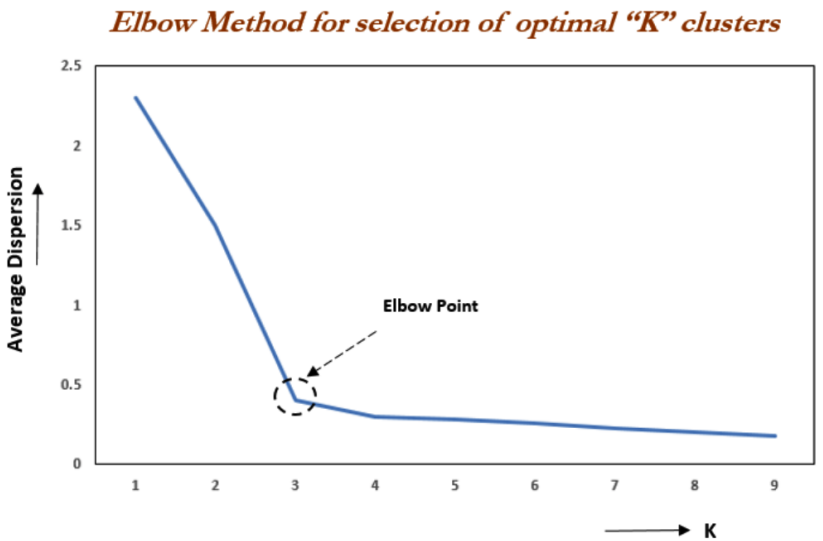
- This is the objective function we aim to minimize in the k-means algorithm.
- Measures how dense the clusters are.
- We want to find the cluster number that gives us low inertia AND low number of clusters.
  (What happens when #clusters = #data points?)
- Drawbacks: Favors convex, spherical clusters. k-means specific.

Clustering Error Metrics - Inertia

How can we use inertia to find the optimal value for k?

Elbow/Knee Method

Plot the values of k vs k-means inertia. Find the value of k for which this graph has maximum curvature.



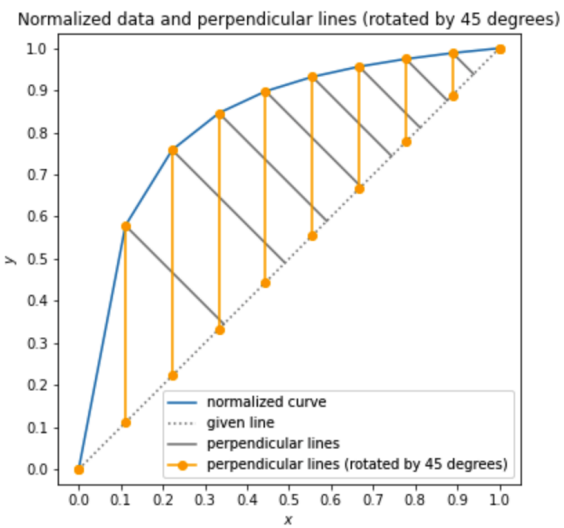
Problem: Data is noisy and discrete, so using derivatives to find the point of maximum curvature is not accurate.

Figure from https://www.oreilly.com/library/view/statistics-for-machine/9781788295758/c71ea970-0f3c-4973-8d3a-b09a7a6553c1.xhtml

Clustering Error Metrics - Inertia

Solution: Kneedle Method

1. Invert the graph if the function is concave up. Standardize the data.
2. Draw a vertical line from each point to the diagonal.
3. Graph the lengths of these lines and choose the k corresponding to the maximum of this difference graph.



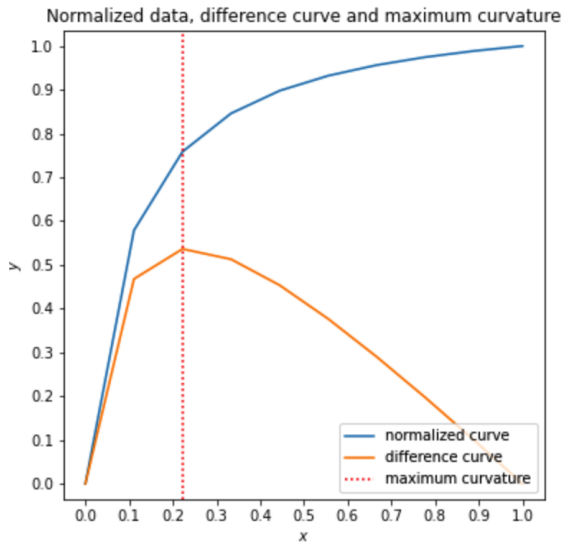


Figure from https://towardsdatascience.com/detecting-knee-elbow-points-in-a-graph-d13fc517a63c

Clustering Error Metrics - Silhouette Score

Silhouette Score

Let x_s be a sample point assigned to cluster C_j.

$$a := \frac{1}{|C_j| - 1} \sum_{x_i \in C_j, x_i \neq x_s} \|x_j - x_s\|$$

$$= \text{mean inter-cluster distance from } x_s$$

$$b := \min_{m \neq j} \frac{1}{|C_m|} \sum_{x_i \in C_m} \|x_i - x_s\|$$

$$= \text{mean distance of next nearest cluster points from } x_s$$

$$\text{Silhouette score for } x_s = \frac{b - a}{\max(a, b)}$$

- Measures how dense AND well-separated the clusters are.
- Can be used for k-means and GMM.
- Drawbacks: Favors convex clusters.

Clustering Error Metrics - Silhouette Score

- In general we want to maximize the Silhouette score.
- We can also plot the Silhouette scores for each sample to visually analyze our clusters.
  - Are large clusters dense? Are small clusters sparse?
  - Negative Silhouette scores may indicate inaccurate assignments.
  - Favor uniformity in Silhouette cluster averages.

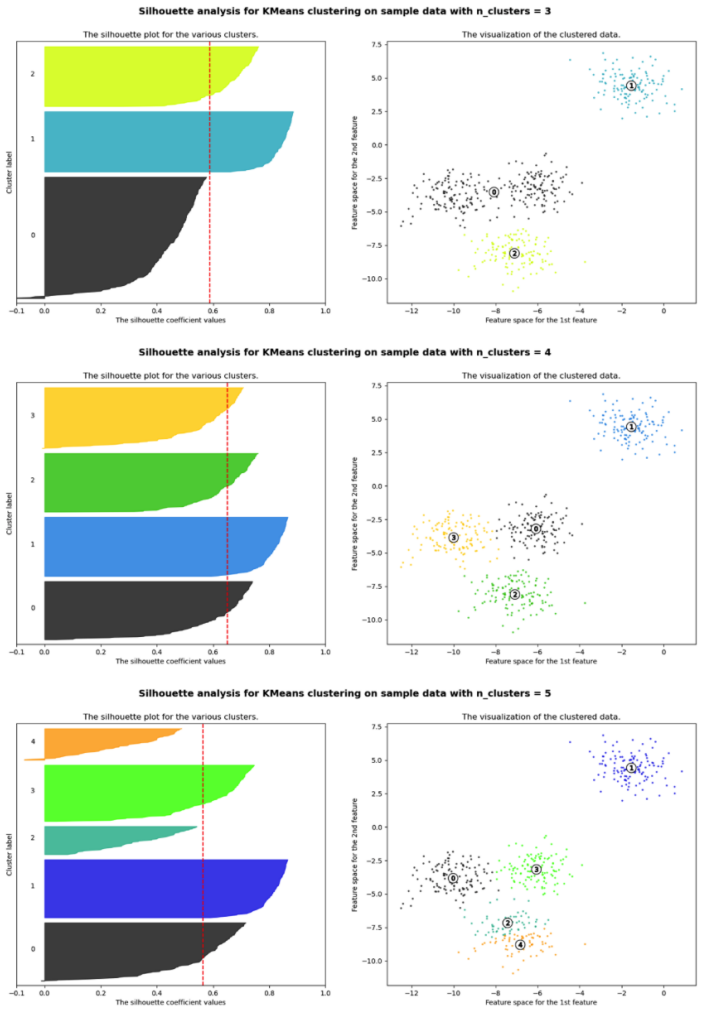


Figure from

https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Clustering Error Metrics - CH Index & DB Index

Calinski-Harabasz Index

CH score is the ratio of the between-cluster dispersion and the inter-cluster dispersion.
Higher scores indicate dense and well-separated clusters.

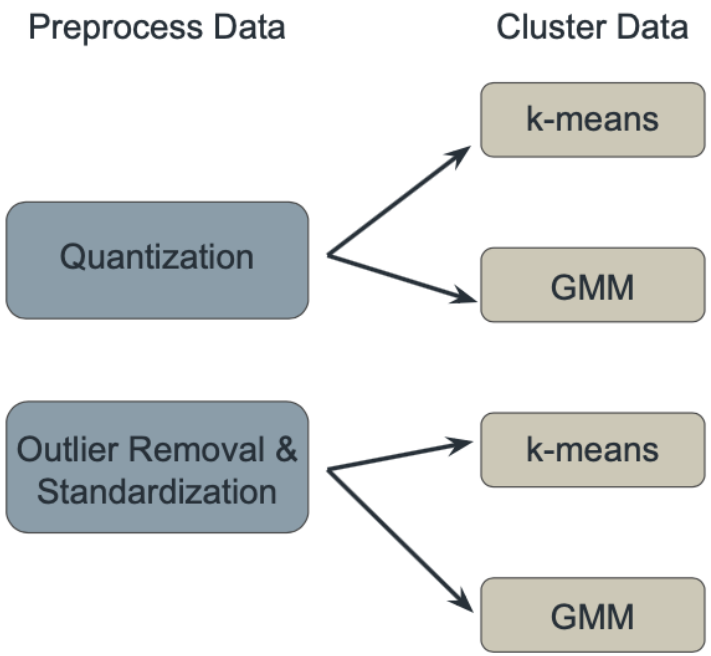
Davies-Bouldin Index

DB score is the ratio of the “cluster diameter" and the distance between cluster centers.
Lower scores indicate dense and well-separated clusters.

- Both can be used for k-means and GMM.
- Drawbacks: Both favor convex geometry.

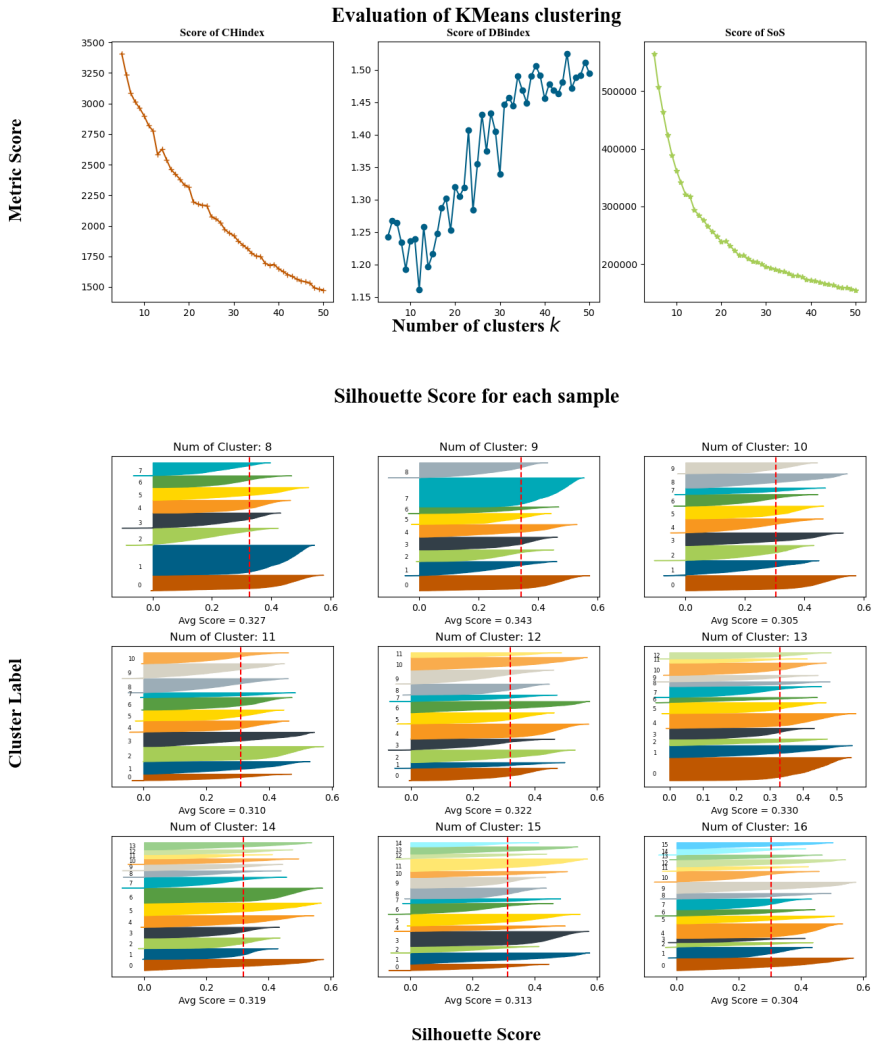
Implementation

- Use scikit-learn *k*-means++ and GMM algorithms.
- Our code allows us to select the costumer features we want to use for clustering. In our final results we cluster the first 14 features (dollar amounts and frequencies)
- We obtain 4 different clustering results:

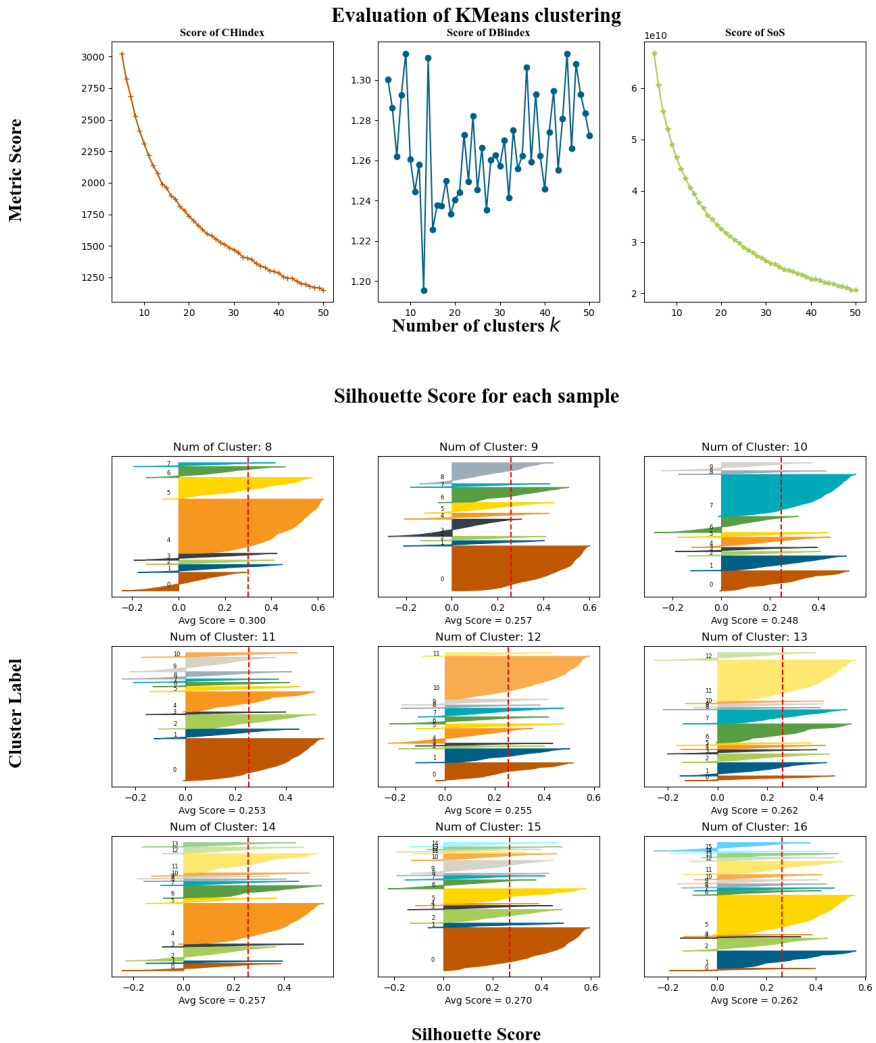


Choosing the Number of Clusters - *k*-means

Quantization: optimal *k* is 9

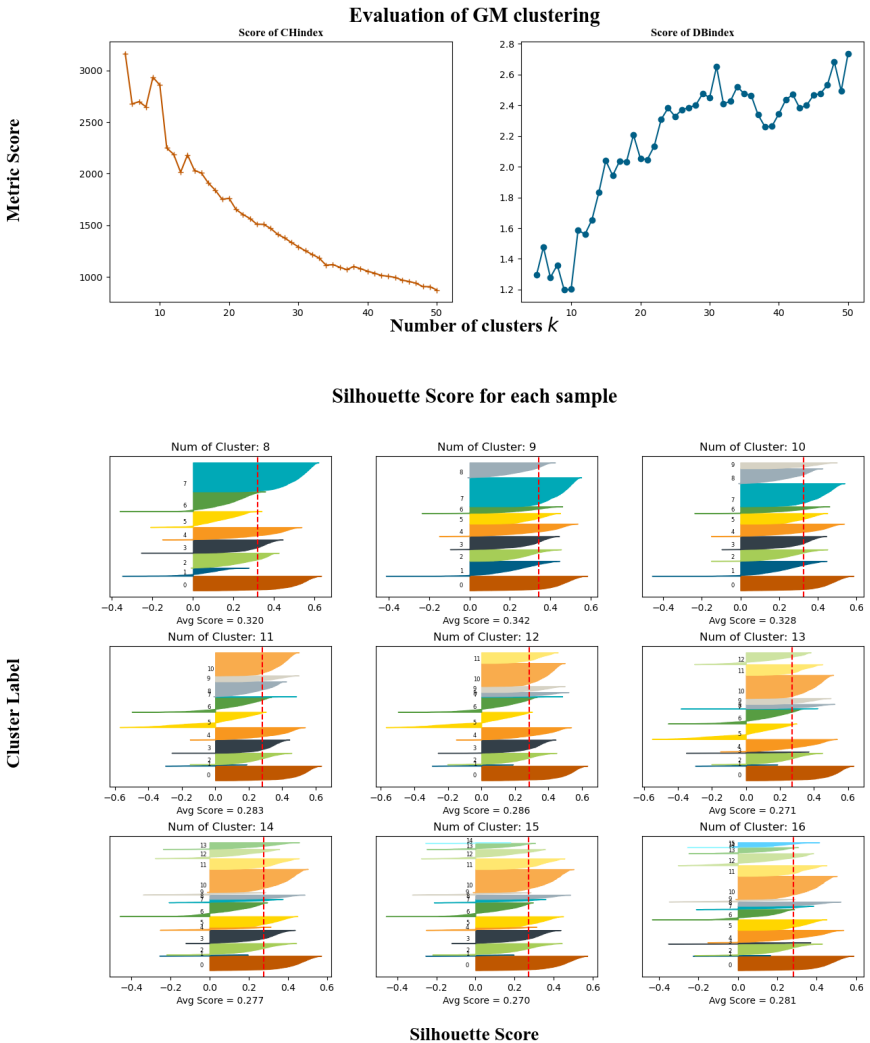


Standardization: optimal *k* is 13

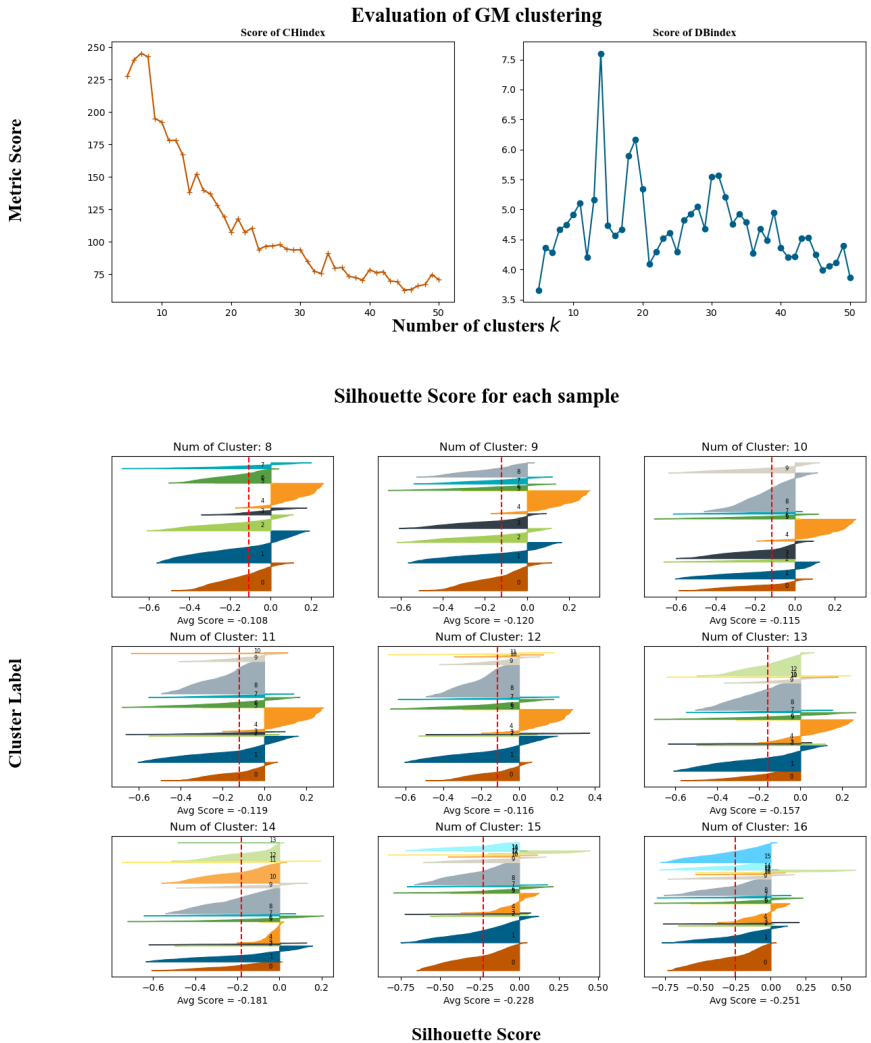


Choosing the Number of Clusters - GMM

Quantization: optimal *k* is 9



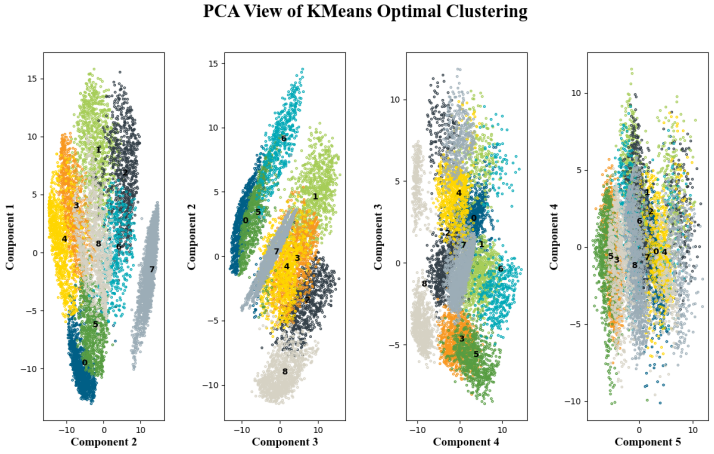
Standardization: optimal *k* is 8

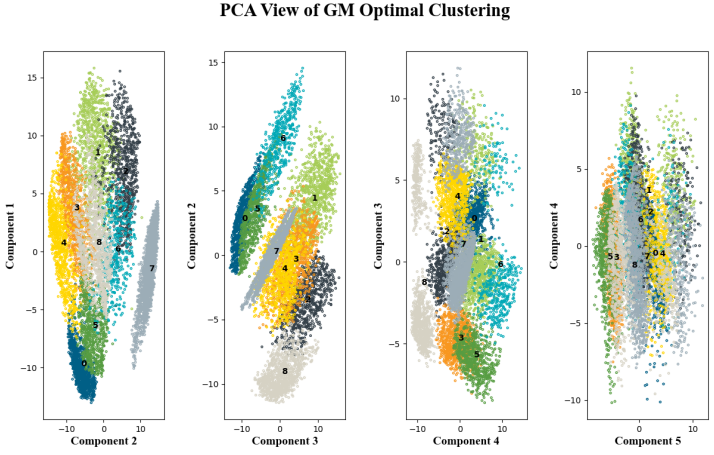


Results

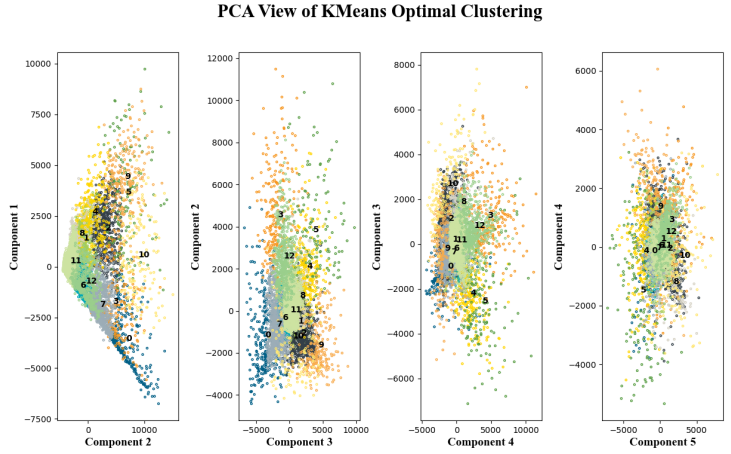
Preprocessing	Clustering	Optimal k	CH Index	DB Index	Silhouette Score
Quantization	k-means	9	2962	1.1926	0.343
Quantization	GMM	9	2934	1.1974	0.342
Standardization	k-means	13	2074	1.1956	0.262
Standardization	GMM	8	242	4.6616	−0.108

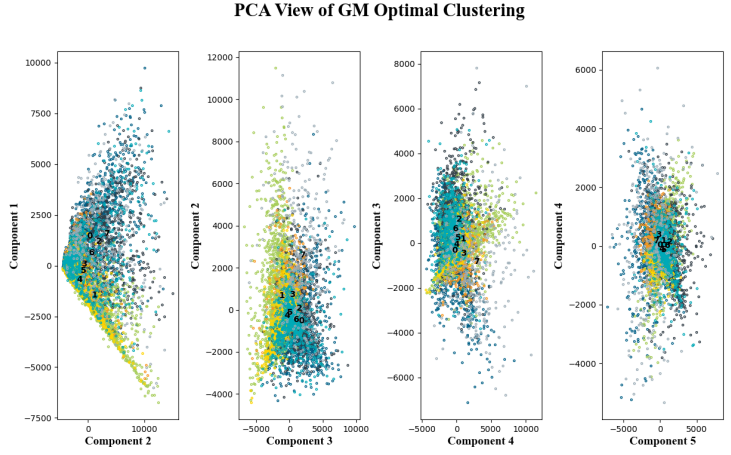
Quantization:



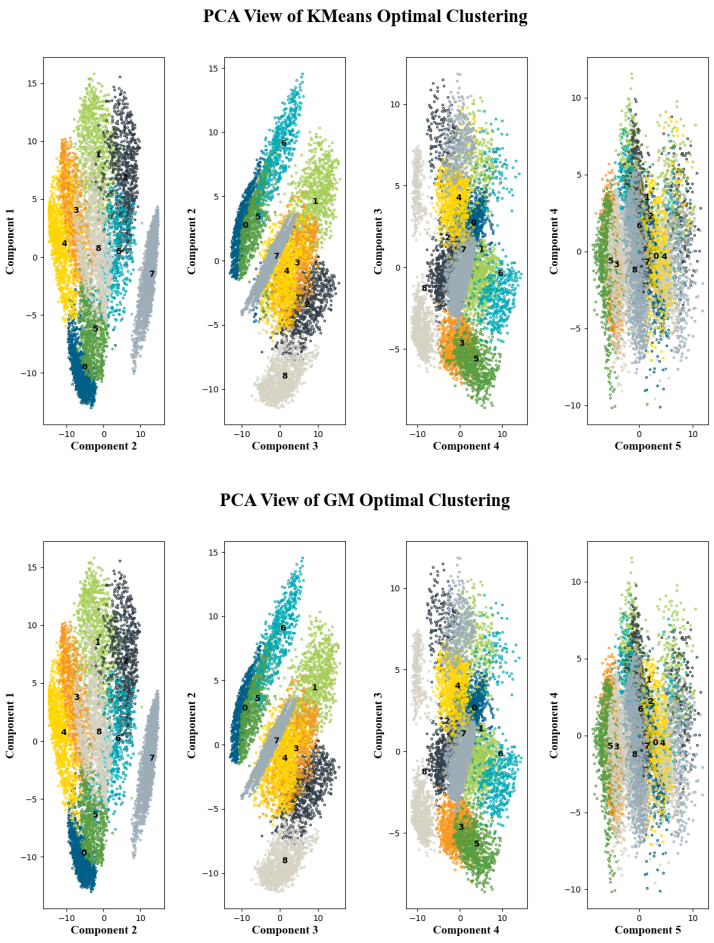


Standardization:



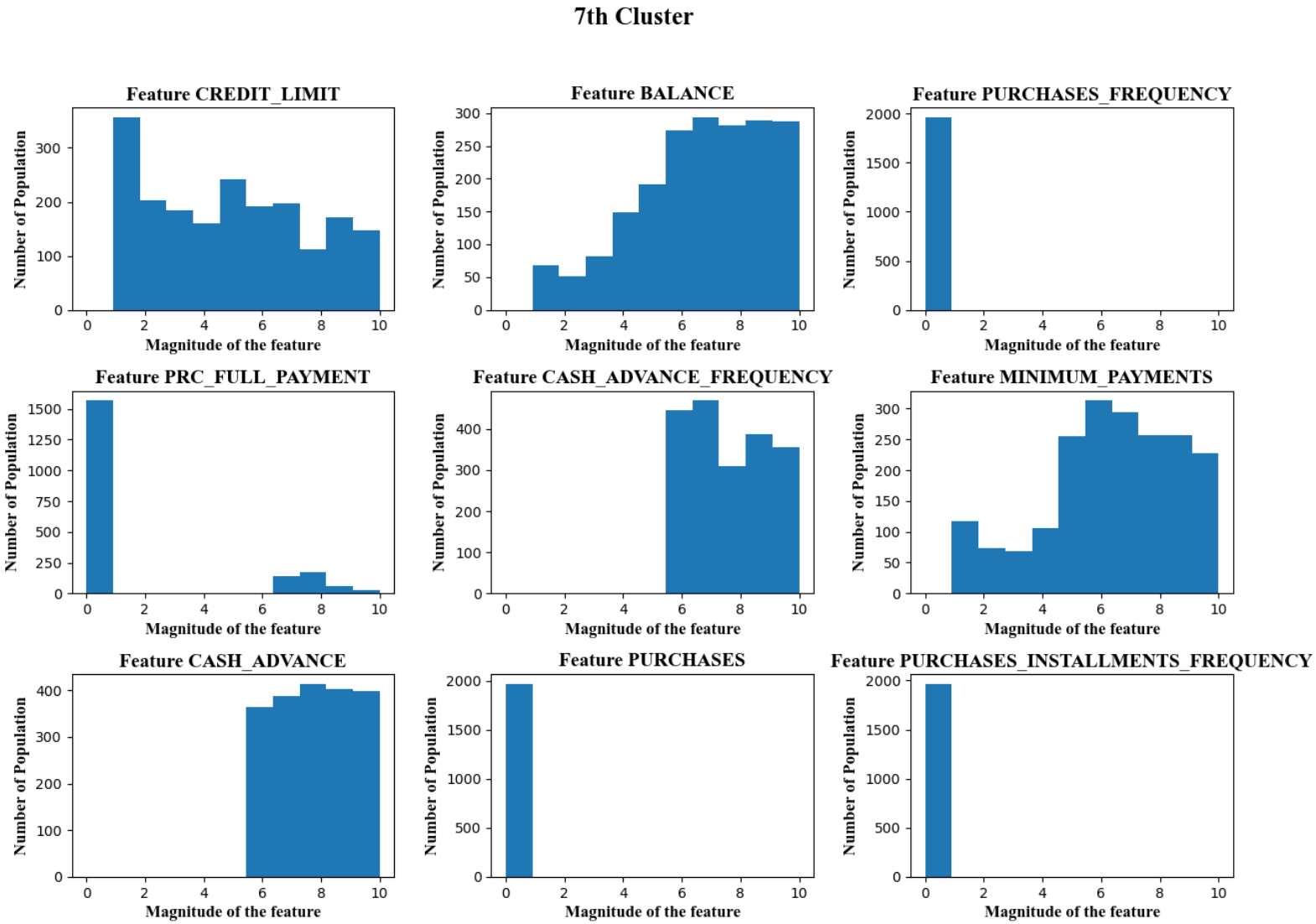


Conclusions on Customer Behavior



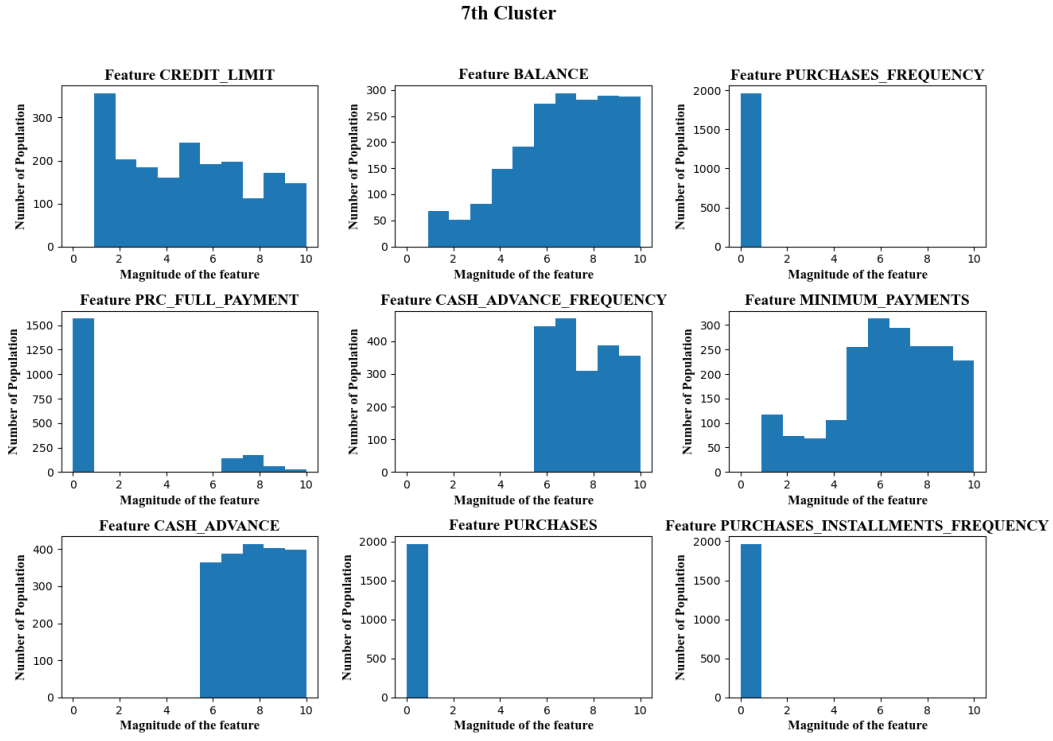
- Believe your eyes: optimal Kmeans and GMM are close
- GMM: likelihood interpretation and predictive distribution function $X \sim f(x;\theta)$
- KMeans: provides convex region assignment with piece-wise linear boundary
- These clusters persist (with some rearrangement) across both model sets

Customer Behavior Analysis



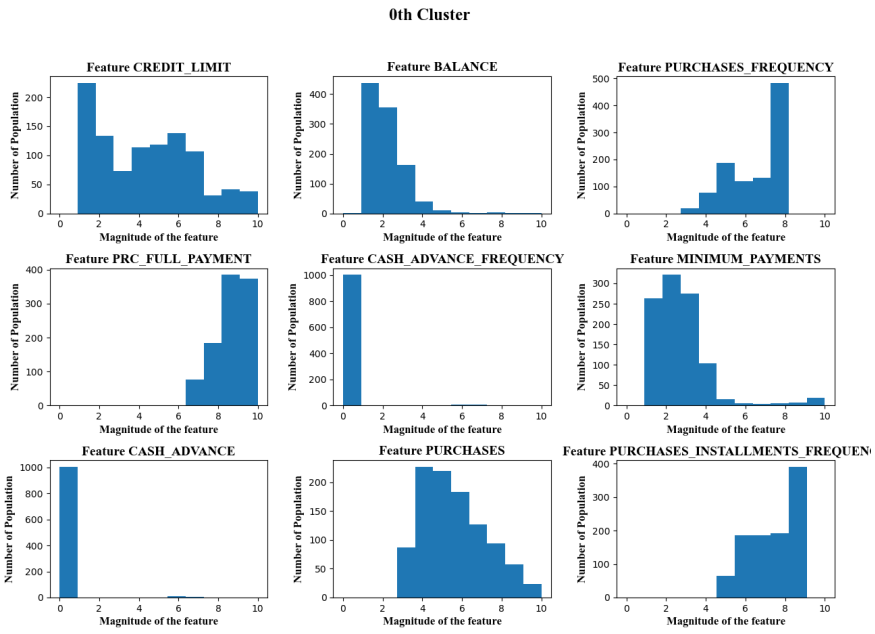
Customer Behavior Analysis

- If you are in the top 10% cash advance users (by freq.) it is probable that you belong to cluster 7; in that case, you do not use your card for regular purchases, irrespective of your credit

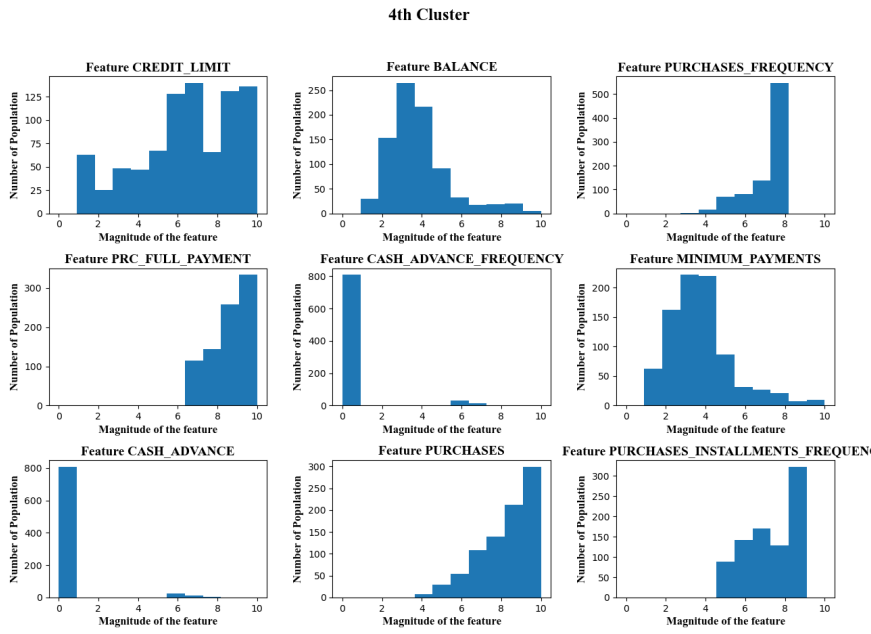


Customer Behavior Analysis

Cluster ’0’:

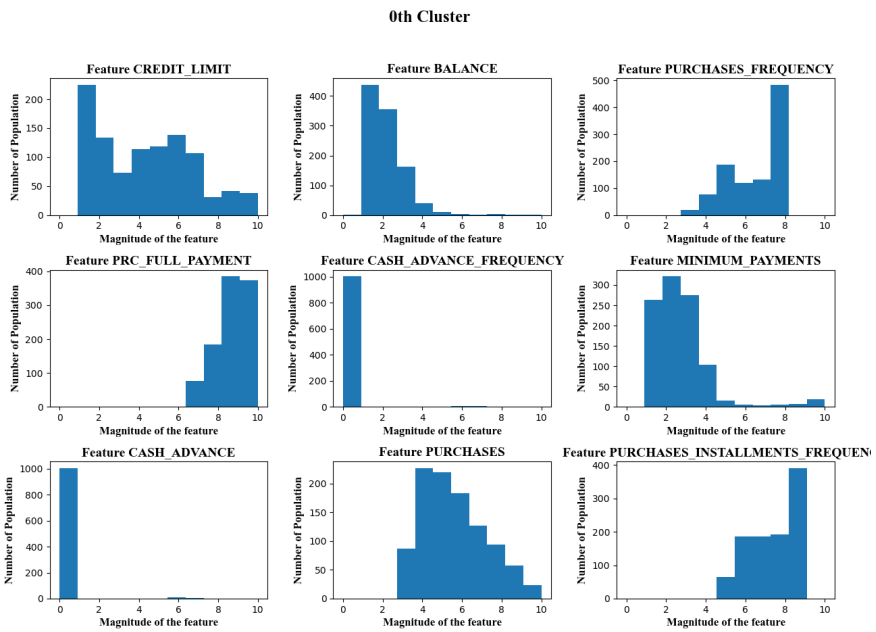


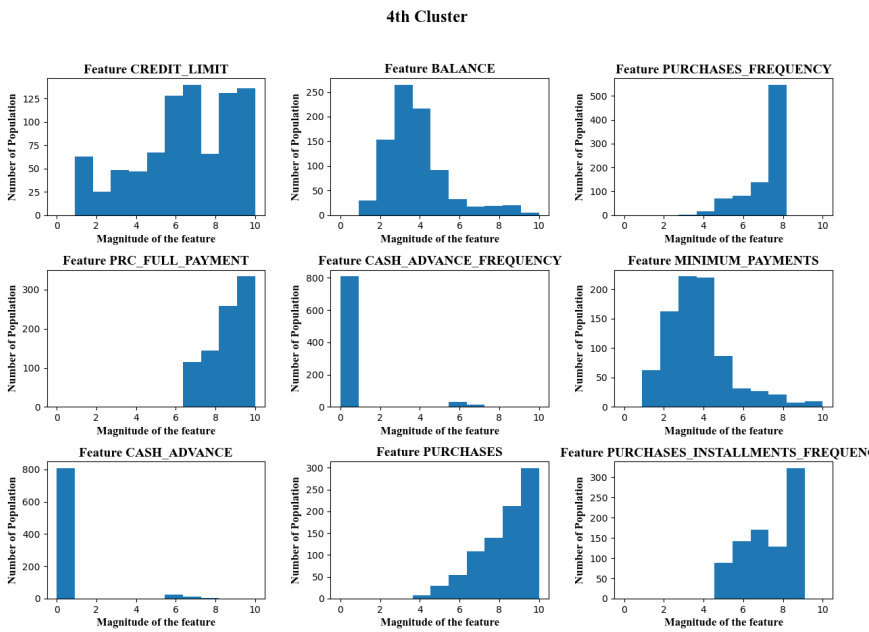
Cluster ’4’:



Customer Behavior Analysis

- If you are in the top 20% strata of card purchase users (clusters 0 4, 3), it is probable that you make most of your payments in full, and do not use cash advances, irrespective of your limit distribution





References

[1] *2.3.10 Clustering performance evaluation*. URL: https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation.

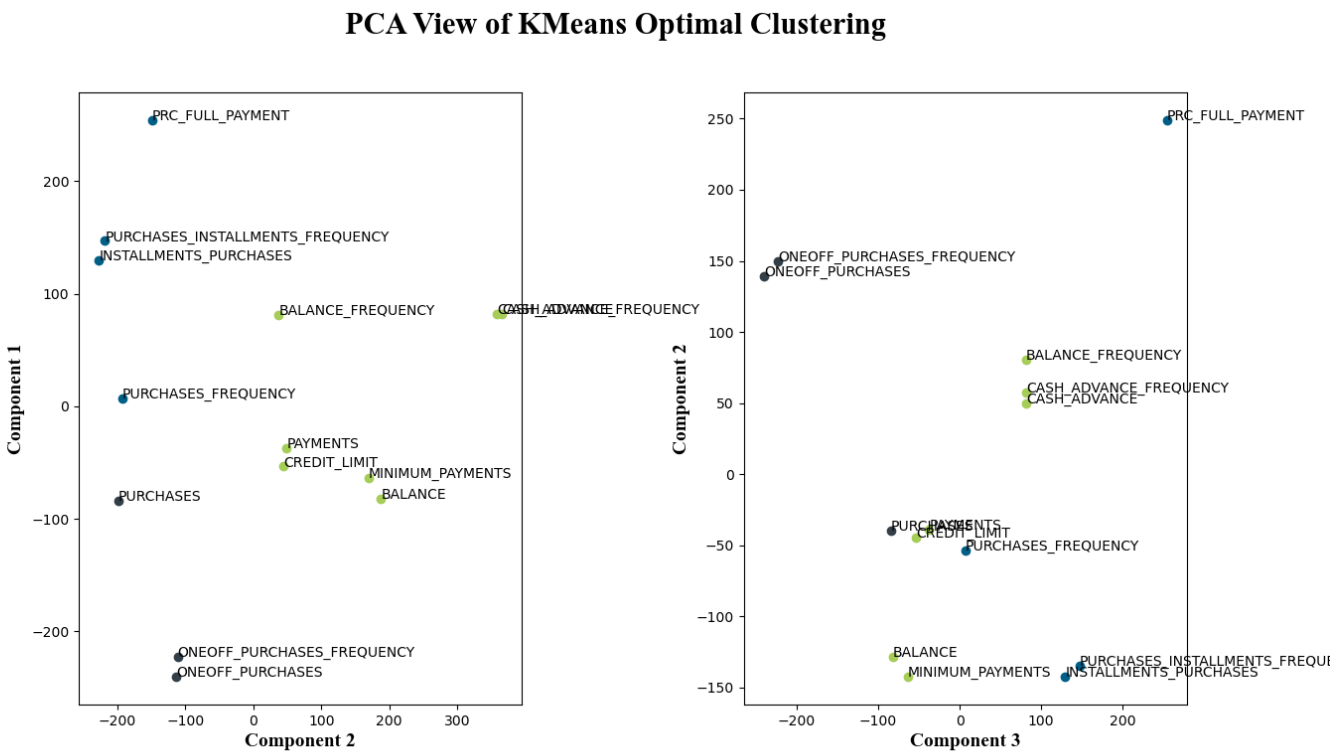
[2] Barath Raghaven, Ville Satopaa, Jeannie Albrecht, and David Irwin. Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior. In: *IEEE ICDCS SIMPLEX Workshop* (June 2011).

[3] Arjun Bhasin. *Credit card dataset for Clustering*. Mar. 2018. URL: https://www.kaggle.com/datasets/arjunbhasin2013/ccdata.

Clustering the Transpose Data

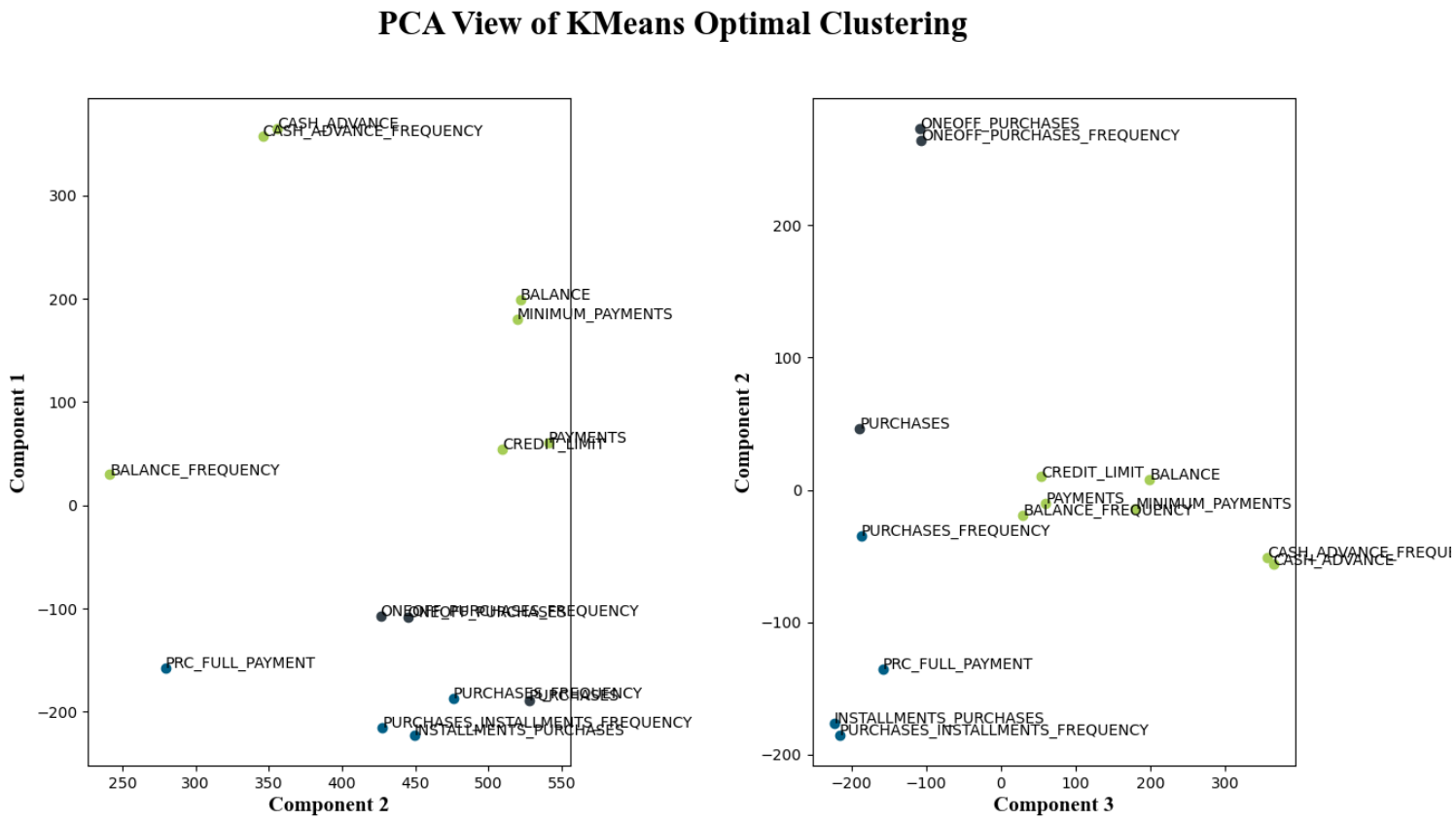
- If we transpose our data matrix (to a 14×8636 matrix) we can think about clustering on our features.
- Using 3 clusters, we now have theoretical guarantees to reduce dimension to 3.

Full dimension:



Clustering the Transpose Data

SVD reduction to 3 dimensions:



We observe the clustering is consistent.

The University of Texas at Austin

Department of Mathematics

College of Natural Sciences