Daniel Leon AMATH 563 Lab 8
May 20, 2022 Clustering with K-means & PCA or SVD

Introduction

The purpose of this lab was to demonstrate unsupervised machine learning methods on high-dimensional low-rank data. We implemented Kmeans clustering, and compared the base method on the data without any dimensionality reduction to those using PCA and SVD low rank approximations. Kmeans is a very popular algorithm for iterative descent clustering because it converges rapidly and it can be used along with other dimensionality reduction methods, such as principal component analysis and singular value decomposition analysis, as we will demonstrate.

Background

This credit card data had 8,950 samples with 17 features each, excluding the customer ID. There were a few instances of missing data, and we addressed those data several ways to see the effect on the analysis. The data was split into test and training sets, with 90% (8,055 samples) in the training set and the remainder in the test set. We produced elbow plots to predict optimal number of clusters. This was carried out on the raw data, as well as in PCA and SVD transformed space. We also carried out pairwise analysis of the data under transformations and not. We implemented Kmeans clustering while looping over both a range (2-5) of principal components/singular vectors as well as a range of clusters (2-10) for Kmeans to look for. The training and test data was plotted in transformed and standard frame with cluster predictions for both sets. Then confusion matrices were produced for both rank-reduced cases using cluster predictions on the non-transformed data as the ground-truth.

Results

After preprocessing the data, which included taking care of missing data (NaN's, will talk about that) and mean-centering and scaling to unit variance, we implemented the elbow graphs for each of the three cases using the full dataset. Below is the elbow plot for the non-reduced data. We can see that the most apparent elbows lie at K=4 and K=6 (though the second arrow points to 10 in figure 1). For both the PCA and SVD approaches, using the range of principal components or singular values from 2 – 5, we see that K=4 is the most obvious elbow across all conditions. It should be noted that setting the NaN's to 0 resulted in the most obvious elbow for all cases, when compared to setting the Nan to the column mean or median value.

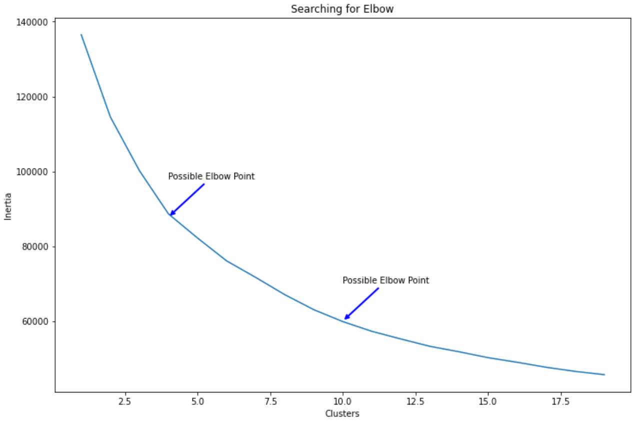
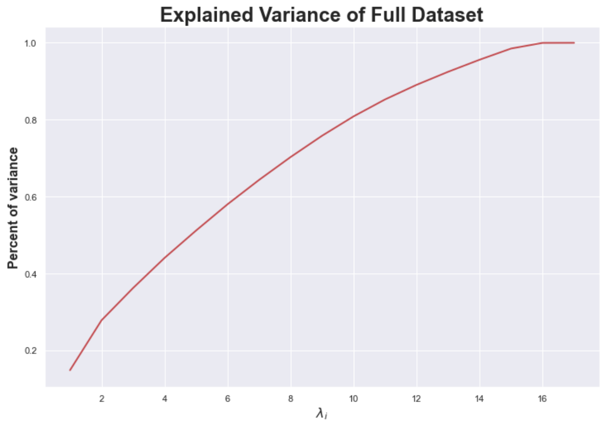


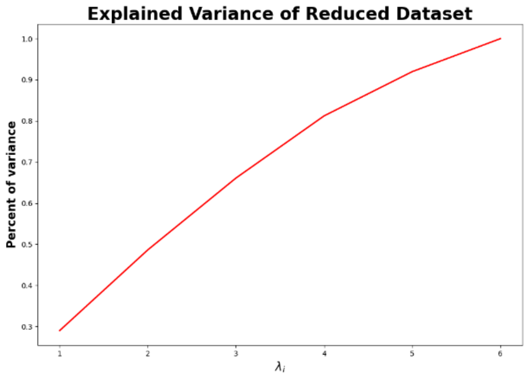
Figure 1: Elbow plot of scaled (non-transformed) data. Elbows at K=4 and K=6 (ignore second elbow pointing at K=10)

By looking at the pairwise plot and the df.describe() data, we determined that the most useful categories to limit the analysis to would be: 'PURCHASES', 'CREDIT_LIMIT', 'PAYMENTS', 'BALANCE', 'MINIMUM_PAYMENTS', 'CASH_ADVANCE'.

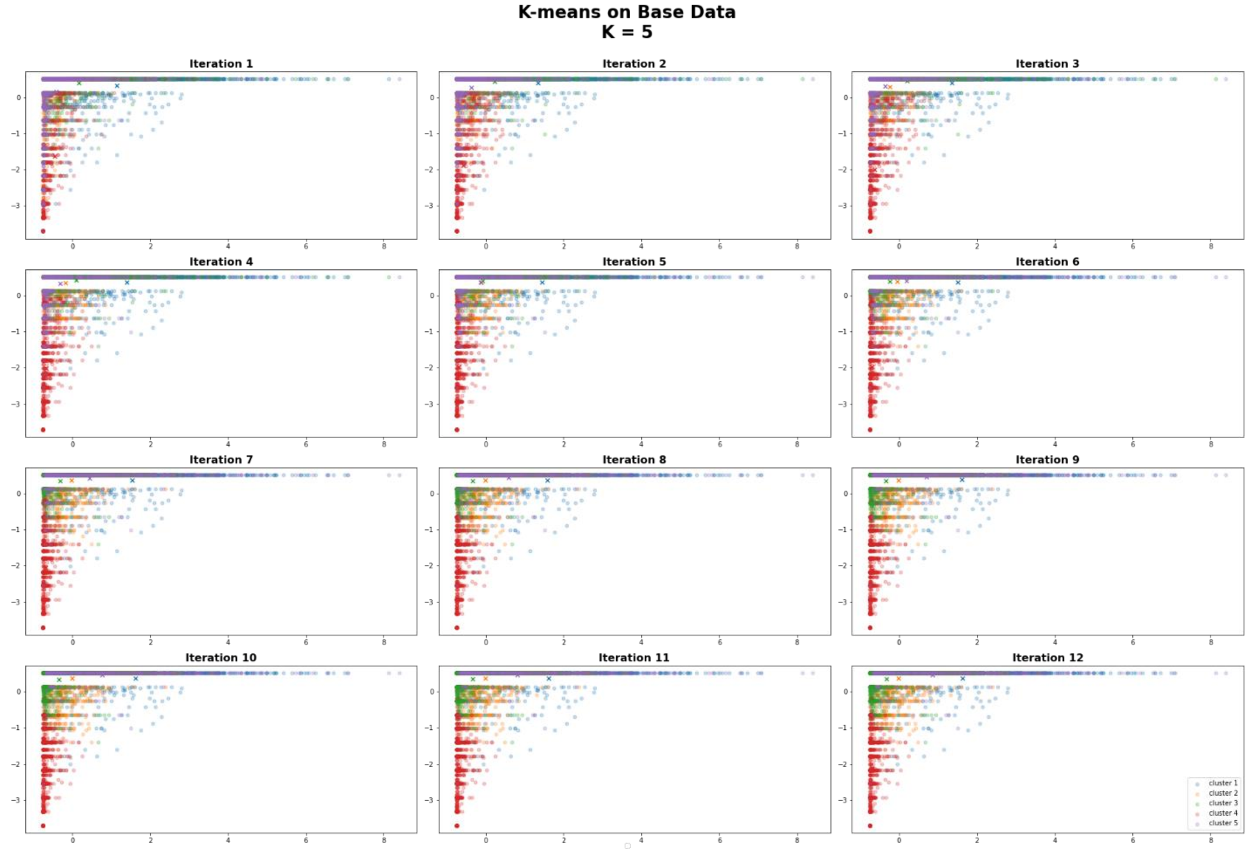
These data were added to a new DataFrame for final analysis, though the clustering was initially performed on the full set

The Kmeans clustering algorithm was implemented by training on the projections of the training data, then making predictions about the projected test data. For the full set of data, using PCA, the explained variance plot is shown in figure 2a. The explained variance for the reduced set is shown in figure 2b. After coordinate transformation, we ran the Kmeans algorithm, sweeping across the whole space of r and K. We will show results for 5 principal components and 5 clusters.

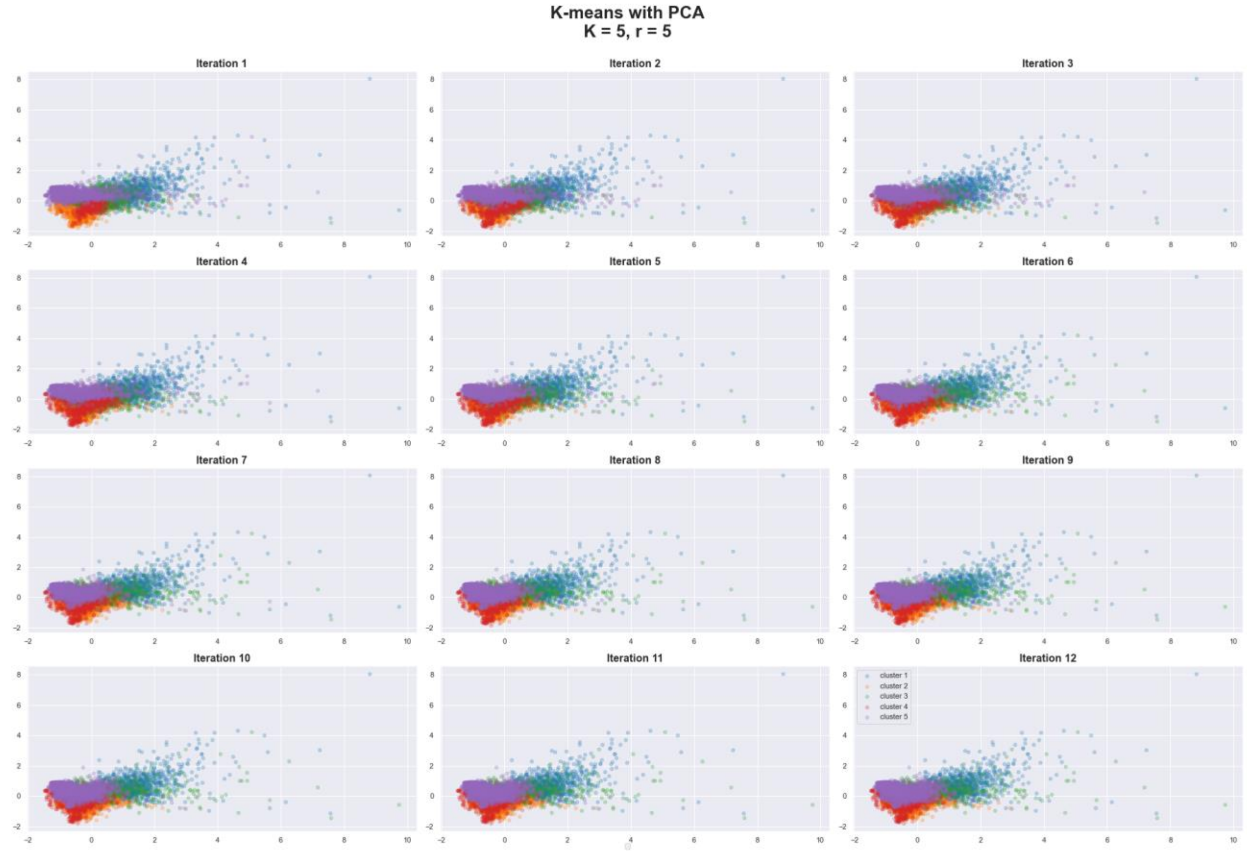




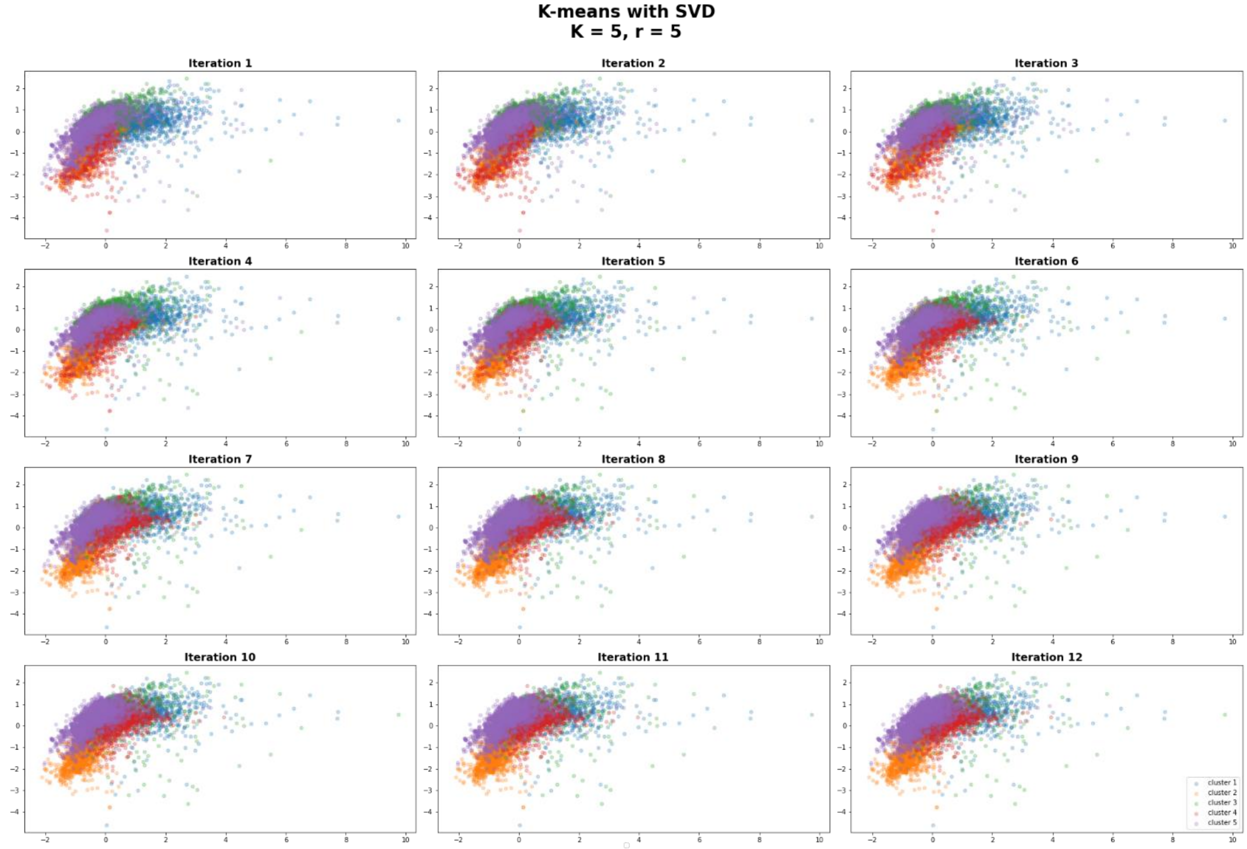
The convergence of the Kmeans algorithm for both transformed cases is shown below. This data included setting the NaN values to 0.



Kmeans convergence for the non-transformed data. The separation is not as good as the other methods.

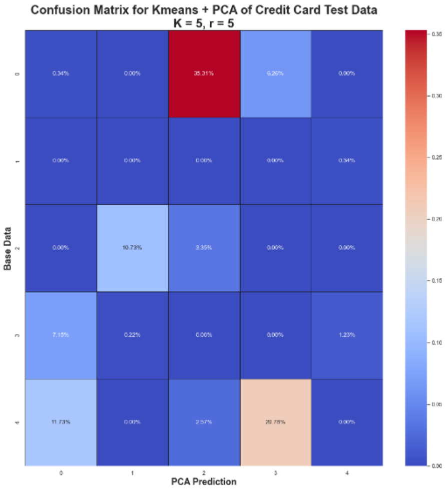


Kmeans convergence for PCA projection of training data. By iteration 7 there isn't much change in clustering.

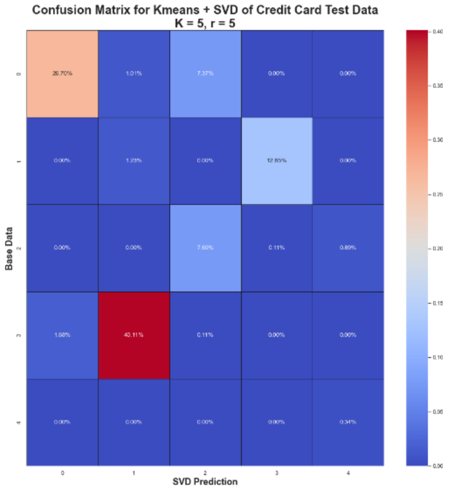


Kmeans convergence for SVD projection of training data. Similar to with PCA, the method converges pretty quickly

As shown, Kmeans converges quickly, and the rank-reduction methods do a much better job of grouping the data together.

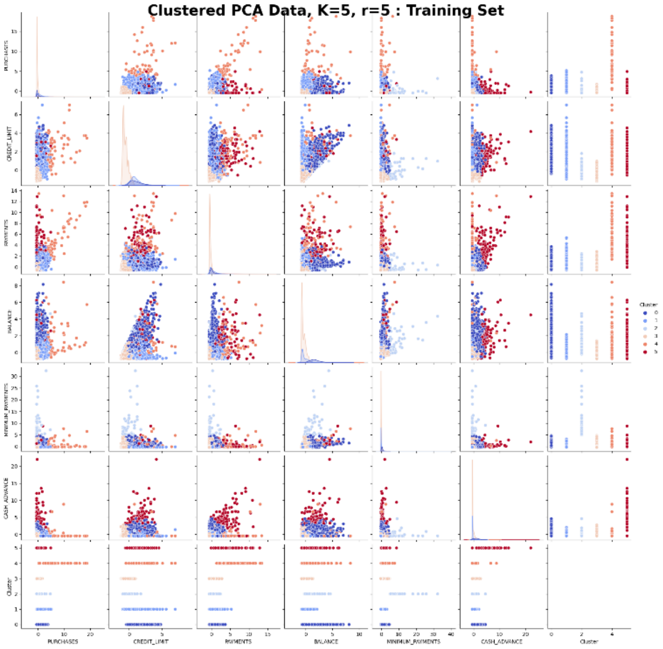


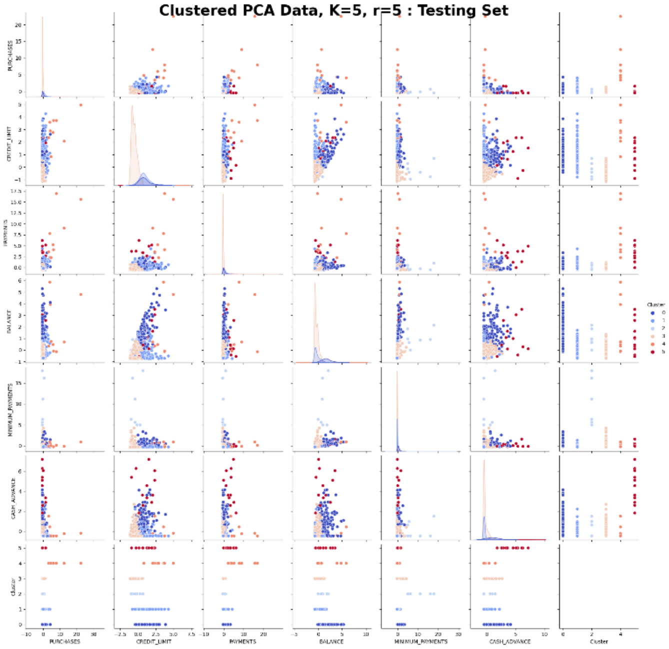
Confusion matrix for PCA method.

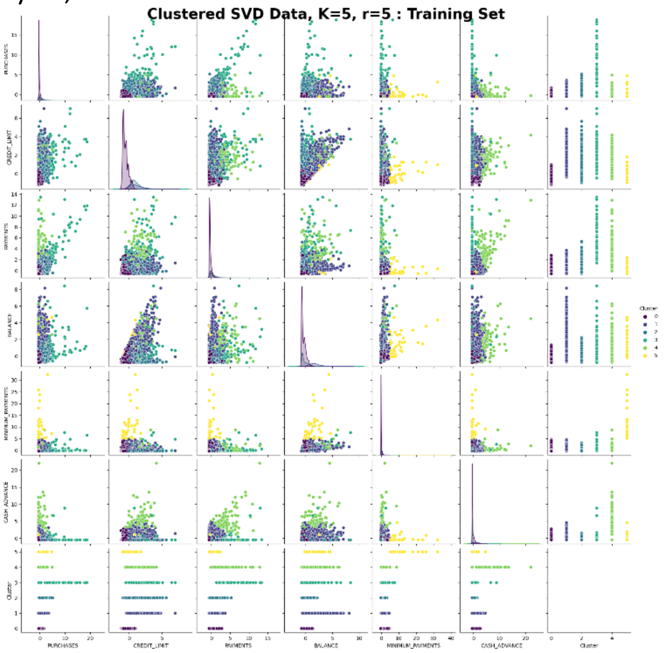


Confusion matrix for SVD method.

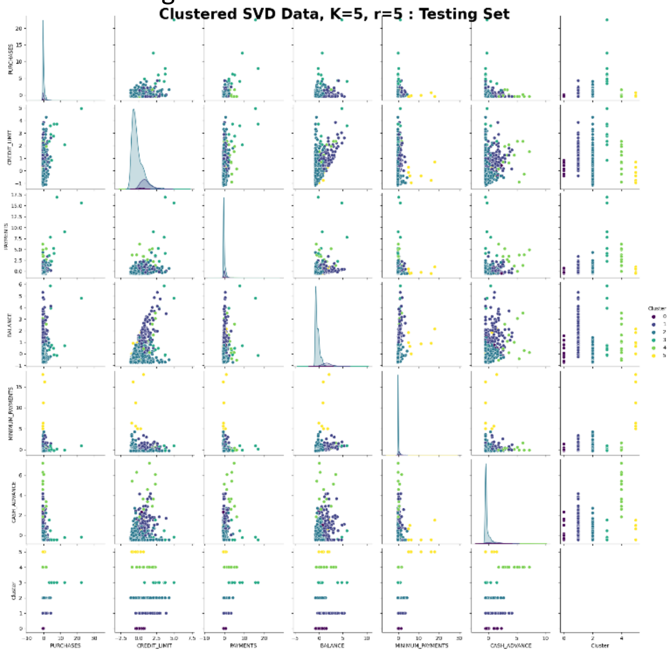
The two methods seem to give quite different results, but both were generated using the full dataset. The confusion matrices aren't as easy to interpret with unlabeled data, so I produced some better seaborn pairwise plots using only the "choice" categories mentioned earlier.







This column has the training data clustered in rank reduced space, then labeled in the regular space.



This column is the test data that was tested using the trained models shown on the left.

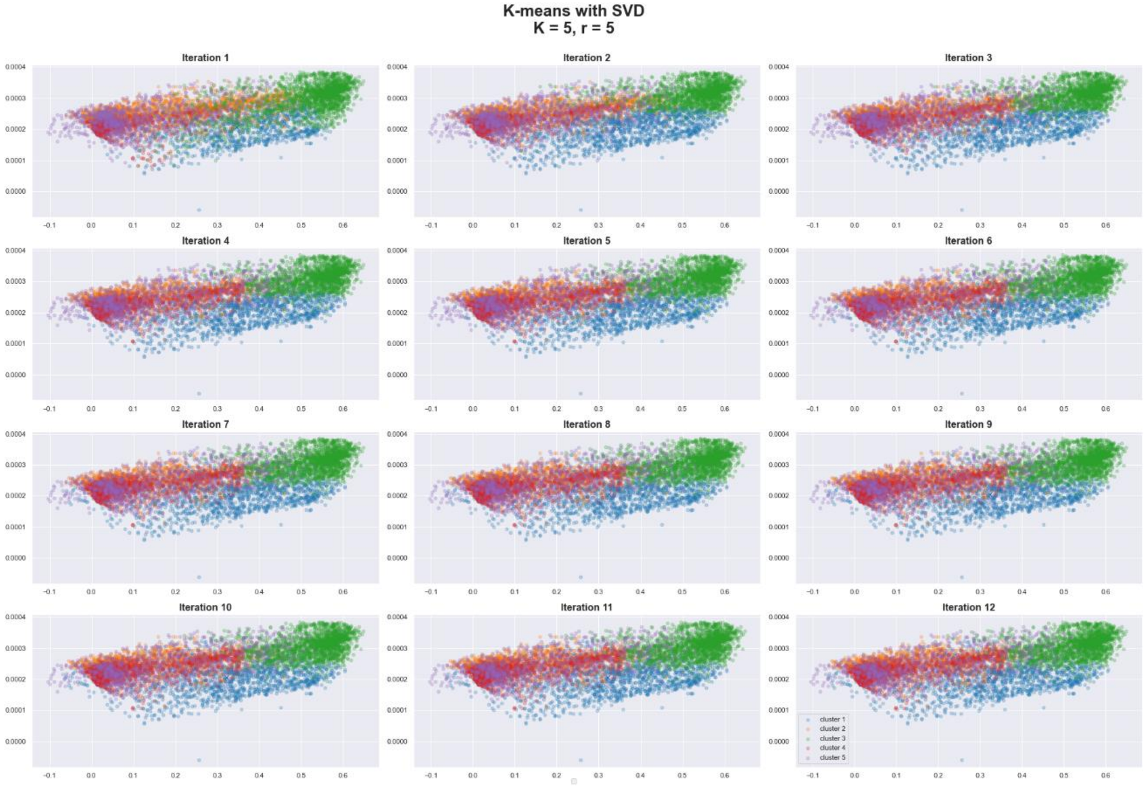
This method of presentation ended up being the most interpretable to me. Both the bottom row and right-most column show the distribution of each category and associated cluster. Since the data are unlabeled, the different methods assign a different number/color to different clusters, but they are largely grouping the same points.
For example, in test results:

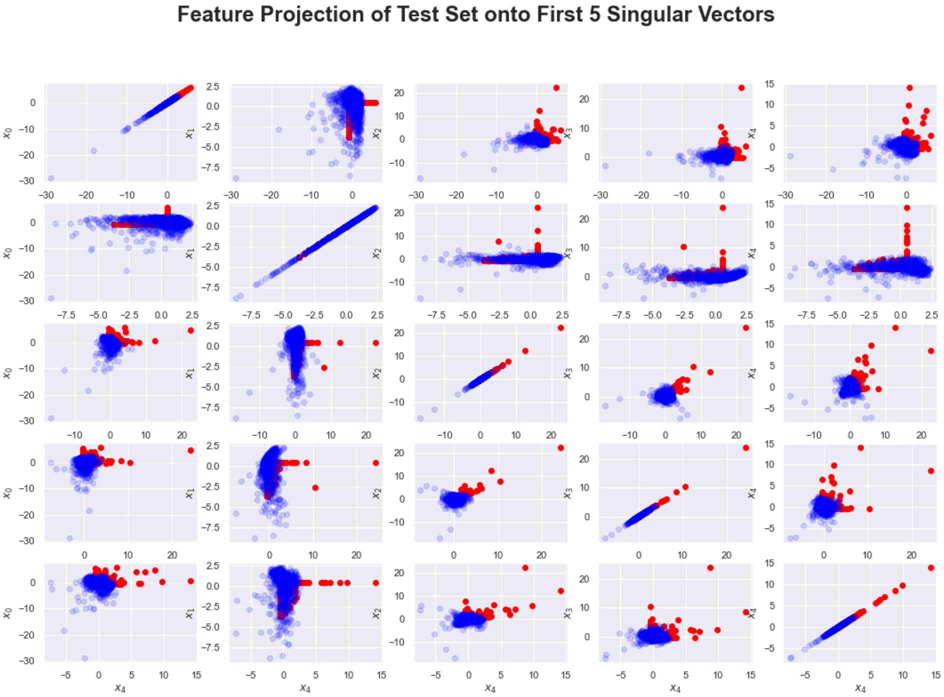
1. Cluster 5 in PCA (red) and cluster 4 in SVD have the highest cash advance average, they have low payments, they make few purchases with credit, they maintain a moderate balance, and their credit limit is on the lower average. I would expect this group to be mostly living paycheck to paycheck.
2. Cluster 4 in PCA and cluster 3 in SVD seems like the higher earning, higher spending type based on the high value of purchases, credit limit, payments, and balance. They also have low level of cash advance.
3. Etc

That is how we can make sense of unlabeled data by using clustering. One thing that I forgot to do was to un-normalize the data. Next time!

Bonus Content

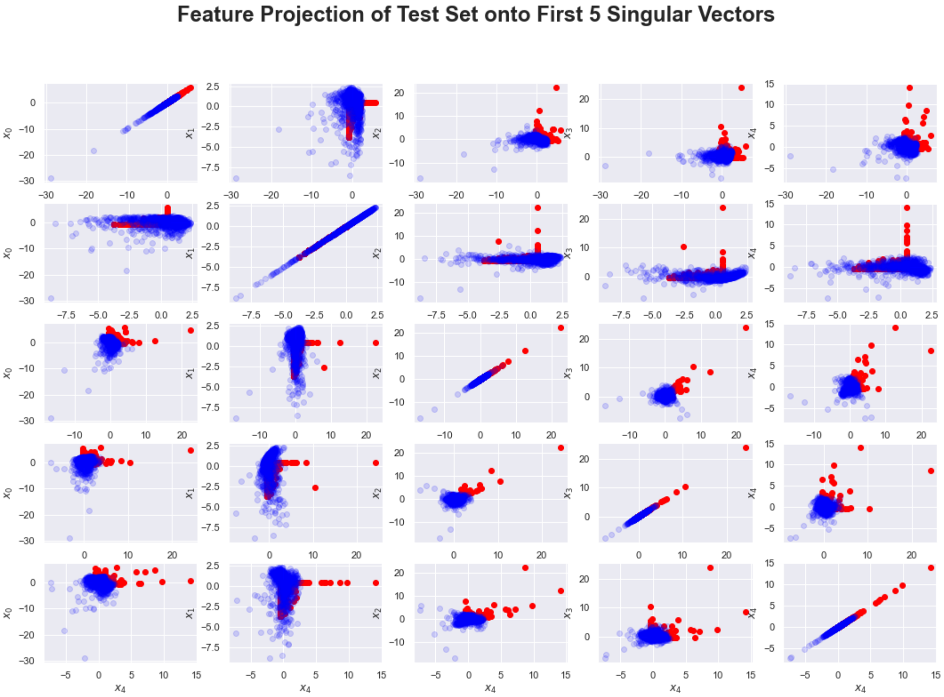
I ran into a few situations that I would like to show. I remembered that PCA needed centered data, and thought that was not the case for SVD. So initially I was only normalizing the SVD data. It did produce quite different-looking clustering (though I hadn't gotten my analysis to the point where my final results graphs were, so I don't know what the predictions would have been like). Compare the clustering graph for SVD with non-centered data to the one that was shown earlier. Initially it looks like it's getting better separation, but when I overlaid the projections of the data with the original, it was clear that the variance was not well retained.


K-means with SVD
K = 5, r = 5
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Iteration 7
Iteration 8
Iteration 9
Iteration 10
Iteration 11
Iteration 12
0.0004
0.0003
0.0002
0.0001
0.0000
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5



The projection of the data onto non-centered SVD space.

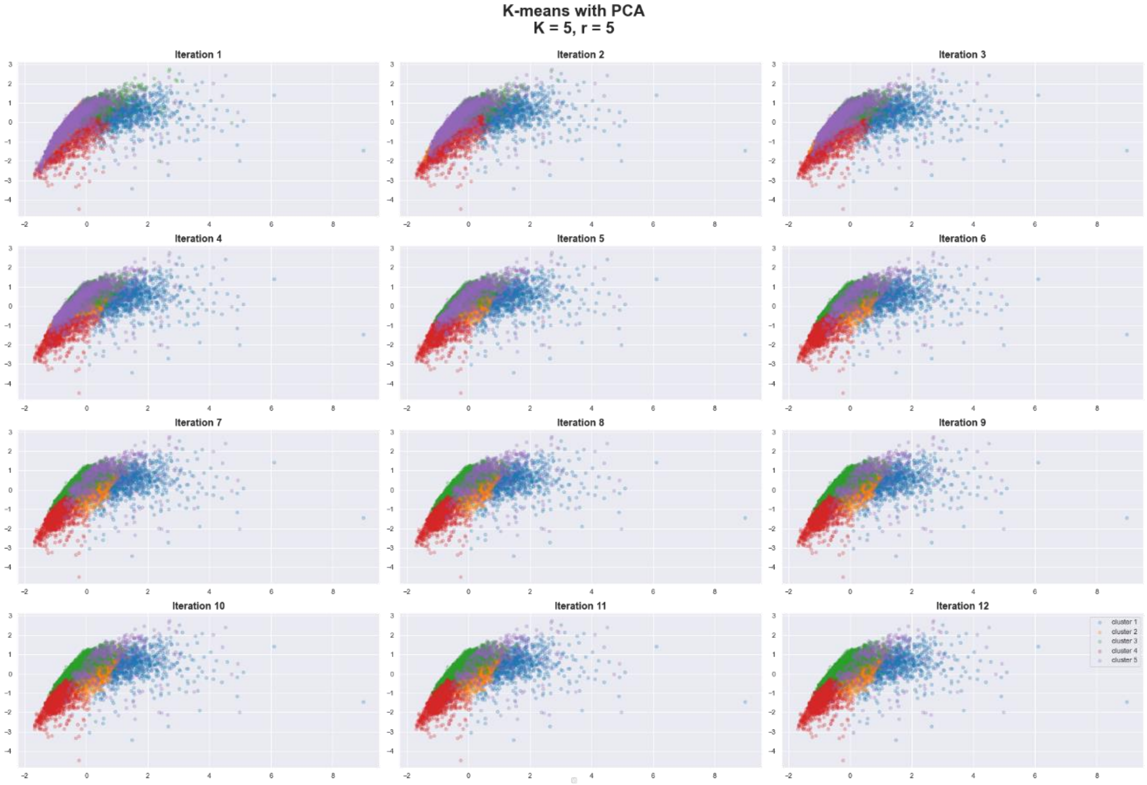
Compare those projections to the case with centered data shown below.



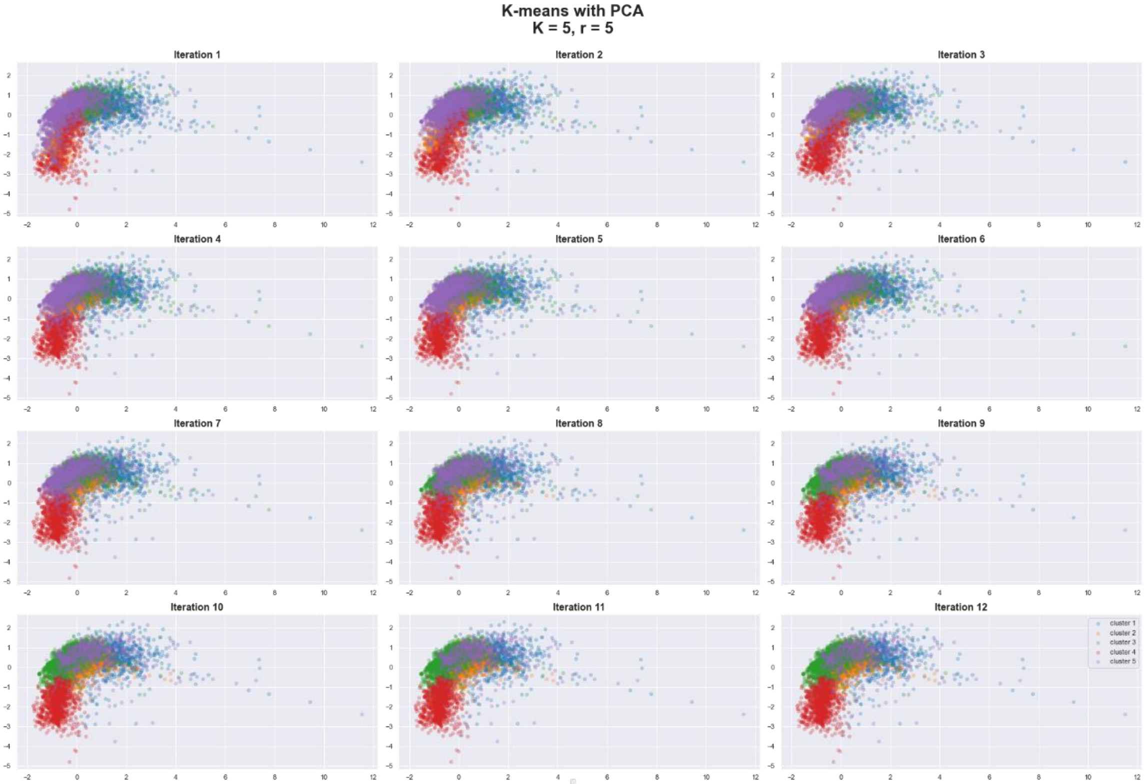
Actually, after looking at these plots again, the non-centered data projections don't look very different, and the SVD clustering seems much better. So, I think I just learned something, and it would be interesting to relook at results with non-centered data for the SVD approach.

One other thing that I looked at was the effect that using different methods of filling in the NaN values. The main analysis went forward with setting the values to zero, which I thought made the most sense to do for the 'minimum payment' category, which was >99% of them. Below I will show the PCA clustering for the other two methods that haven't been shown yet, under the same conditions as before.

This is NaN set to column median value, which was about 350.



This is NaN set to column mean value, which was about 800.



The method of filling in the NaN values does indeed seem to have a pretty big effect on the presentation of the clustering, but as with the last situation, I had not figured out the final presentation mode that I ended with yet when I did these, so I don't know what the real impact would be on classification. Also, it is interesting to note that the NaN-to-mean PCA clustering spread looks very much like the SVD clustering that was presented. Food for thought!