PCA and K-Means
See Dropbox for Due Date
Dr. Thomas Kinsman, Ph.D.

Homework is to be programmed in R, Python, Java, or Matlab. (Not Excel.)

As always, assume that the instructor has no knowledge of the language or API calls but can read comments. **Use prolific comments** before each section of code, or complicated function call to explain what the code does, and why you are using it. Make sure your write-up and/or your code are self explanatory. Hand in your write-up, your results, and the well-commented code, in the associated dropbox.

ASSIGNMENT:
As with the previous HW (agglomeration) assume you work at SSS – (Sam's Spiffy Supermarket). SSS tracks each receipt by "Guest ID".

1. You are provided with the file `HW_AG_SHOPPING_CART_...csv`. It contains data for the number of times various categories of items (attributes) were purchased by guests, for 10 different visits.

   QUESTION: Why do we use 10 visits instead of just keeping records of every single visit?

   As was mentioned in the previous homework, we do this to reduce the noise in the data, if we take the data on 10 visits, as we increase the number of visits we use, the more the data of those visits are likely to converge on the expected value (I forget the name for this concept if I'm honest, but a good example is taking the average of a number of coin flips, head being 1, and tails being 0, on low number of coin flips, the number might lean more to one, or 0, purely because it's not unlikely enough that they won't be more towards one side or the other, even if the odds are 50/50, but as we increase the number of coin tosses, it becomes more and more likely that 50% of the total will be heads, and 50% will be tails)

2. Using the package of your choice, compute the covariance matrix. Do not print this out or report it. The matrix should be 20x20.

   Using the package of your choice, compute the eigenvectors and eigenvalues of the full covariance matrix. Do not print them.

   For reference, Dr. Kinsman spent hours computing and checking his results.
   His first two eigenvectors starts off with the values…

```
FIRST_TWO_EIGEN_VECTORS = [ ...
 -0.3 -0.3 -0.4 0.1 -0.1 -0.1 0.1 0.0 -0.3 0.4 0.4 -0.0 0.2 0.0 0.2 -0.0 0.2 0.0 0.1 0.2 ;  0.3 0.1 -0.1
0.2 0.6 -0.1 0.3 0.0 -0.3 0.1 0.1 -0.0 0.1 0.0 -0.0 -0.0 -0.4 -0.0 0.4 -0.1 ;
```

   You should get something similar, but not maybe not exactly the same as these.

   CONSIDER THE FOLLOWING, but you do not need to answer them in your write up.
   a. If you get an eigenvector that looks like the first one here, but it is negative. That is normal, and it is okay. Why? This is because a vector in the exact opposite direction has the sign changed for all of its components. You can swap all of the signs and get the same vector. It does not matter. If you project onto that vector you just get an opposite value. No big deal.

b. Often students get a covariance matrix that is 21x21 instead of 20x20!

What did they forget to do?

They forgot to remove the ID from the dataset

3. Sort the *eigenvalues* in terms of highest to lowest absolute value.

4. Normalize eigenvalues by dividing each by the total of all the **absolute** values, and plot the cumulative sum of these normalized eigenvalues. The plot will start at the origin for no eigenvalues, and end with a y value of 1.0 for all of the eigenvalues. Use something like matplotlib, so you do not need to use some company's product. Plot: Show this plot in your final pdf report. (2)

(…continued…)

5. Print out the first three eigenvectors – the eigenvectors associated with the eigenvalues that have the largest eigenvalues. Look at the components of each one. Each vector should be 1x20.

Q: Why does this tell you about the attributes?

It shows you the 3 vectors that show the direction of the largest spread of the data, as well as show what attributes are more significant than others.

Which attributes are most important?

1, 13, 18

Which can be ignored?

5, 6, 16, and maybe 7

Justify your answers. (2)

If the Eigenvectors represent a direction of spread of the values, then projecting onto them will allow us to reduce the dimensionality, while retaining important general information do be able to cluster the data. The only reason why I thought maybe to ignore 7 is that is only weighs higher because of Eigenvector 1, which I remember being argued that it doesn't tend to be as useful when it comes to projecting onto them to be able to reduce dimensionality, and while in our dataset, it might still be reasonable to use 17 Eigenvectors, I might argue that the diminishing returns might mean that not using it might be warranted.

6. Project all of the original Agglomeration data onto these first two eigenvectors.

PLOT: Generate a 2D plot of these projected points, and show a scatter gram of this 2D plot of points. You have already seen something like this in lecture. Expect to see 2 to 6 clusters. (2)

7. **K-Means using a Package:**

In this new, projected, two-dimensional space, perform k-Means clustering using a package of your choice. For simplicity, use the Euclidean distance. Typically, you will see two, three, or four clusters. However, one semester there were six. Remember: you have already processed this data using Agglomeration. That should tell you something about how many clusters to expect.

8. Find the center of mass (the average values) of each of the k clusters that you got out of k-Means.

Each of these cluster centers is a 2D vectors in PCA space.

Print all k of these 2D vectors. What do they tell you? (1)

9. Multiply these centers of mass back times the first two eigenvectors. **This is re-projection.** What prototype amounts do you get back? What are the relative amounts? Are these completely realistic? Do you notice anything

odd? Did anything go negative? (1)

Different runs produce different outcomes, but the results seem realistic. While It might not make sense at first glance that some of the centers have a position with negative attribute amounts, the center in PCA space is not the same as the center in the data space.

10. Answer this question:

If you projected the data onto all of the eigenvectors, why would this **not** help you with your data understanding? How many dimensions would you have? (1)

That defeats the whole purpose of performing PCA on the data, as you'd end up with the same dimensionality. Not to mention that the diminishing returns of using more and more eigenvectors demonstrate that the benefit to accuracy is vastly outweighed by the cost in computational complexity caused by the high dimensionality.

11. **CONCLUSION:**

Write up an overall summary of what you did, what you learned, and what you found from this experiment. (1)

The purpose of this homework was to demonstrate how PCA might be used in a real world scenario, as well as force us to come to conclusions based off of the data using the techniques learned in class. Being able to reduce the dimensionality of data whilst retaining the information inherent in the data so that we can actually process the features is crucial to any data scientist, not only for our example of K-Means, but for any form of analysis in a high dimensionality data set.

12. Submit all code and your final report in one ZIPPED UP directory named HW_07_FirstName_LastName_DIR.zip. **ZIP UP THE FOLDER, NOT THE CONTENTS!!**