1) What is the main difference between Minkowski measures (Euclidean and Manhattan) vs Mahalonobis? The equations are obviously different, that is not the question. The question is that conceptually speaking what is Mahalanobis incorporating as part of distance calculation that Euclidean/Manhattan isn’t? Hint: The answer to this is at least partially on page 13 of the clusteranalysis.pdf.

Note: If you copy/paste the statements from that page as an answer I will give you 0 points for the entire homework.

Mahalanobis distance accounts for the covariance between two vaiables whereas Euclidean distance doesn’t. The only time Euclidean distance is used is when there is zero correlation between variable used.

2) What are the diagonal elements of the covariance matrix (I cover this in my videos in the module)

One grammatically correct statement is needed to answer this question.

They diagonal elements are the variances of the first variable and the second variable (using a 2x2 matrix as an example) within the covariance. These elements will not be identical.

3) What are the off-diagonal elements of the covariance matrix?

One grammatically correct statement is needed to answer this question.

The off-diagonal elements are the covariances between the same features and therefore will always be identical.

4) Assume you are working for a consulting company. You are given a database on a set of countries which has P number of columns and N number of rows. The P columns are the features associated with these countries. N correspond to the number of countries whose information you obtained in the database. 2 of those columns are life expectancy and GDP per ca-pita. Your colleague suggests that you should scale the columns between 0 and 1. Why is this a good suggestion?

Normalizing columns between 0 and 1 is to change the values in the dataset to a common scale. Euclidean distance is not scale in variant and considers all variables equally, therefore it would be easier for us to scale it before calculations.

5) Read in the data named “countries of the world”. Copy/Paste your script that does this.

countries = pd.read\_csv("C:/Users/tyler/OneDrive/Documents/school2k20/Fall2020/CIS3339/Data/countries of the world.csv", sep= ',')

6) The scripts below will run a clustering algorithm with K=2. Explain what each line does. (I am going to do some of this below obviously ignore those). If you do not know what a line does, run it, see what it does. If it does not look like it is doing anything look to see where that object is used and then try to figure it out.

Explain: Tells the program to locate and open the dataset and have it labeled as countries

countries=pd.read\_csv("C:/Users/mmusa/OneDrive/Desktop/teaching/datasets/archive/countries of the world.csv", sep= ',')

Explain: returns a dataframe of “countries of the world” dataset

y=countries.iloc[:,2:]

Explain: returns the dataframe with all null values removed

ynonmiss=y.dropna()

Explain: Creates the kmeans object that can be used to have the clustering algorithm operate on input data. It is specified that the object has 2 clusters.

kmeans = KMeans(n\_clusters=2)

Explain: creates scaler object that tranforms features by scaling each feature to a given range

scaler = MinMaxScaler()

Explain: creates an array from ynomiss dataframe that is scaled for easier calculation. (normalizes)

y\_scaled = scaler.fit\_transform(ynonmiss)

Explain: outputs the definition of kmeans with the scaler transform in effect. The object still has two clusters.

kmeans.fit(y\_scaled)

Explain: Outputs the centroids of the K number of clusters. For each cluster centroid there are M features’ means.

(kmeans.cluster\_centers\_)

7) Run these lines:

sse={}

for k in range(1,15):

kmeans = KMeans(n\_clusters=k, max\_iter=1000).fit(y\_scaled)

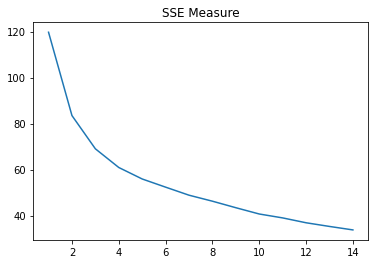
cluster\_labels = kmeans.fit\_predict(y\_scaled)

sse[k] = kmeans.inertia\_ # Inertia: Sum of distances of samples to their closest cluster center

plt.plot(list(sse.keys()), list(sse.values()))

plt.title('SSE Measure')

Copy/Paste and Describe what you see on the plot as well the purpose of the plot.



Generates a plot that shows the sum of squared differences between each observation and the group’s mean. The purpose of this is to visualize and measure of variation within the clusters.

8) Assume K=6. Run the statements necessary to obtain the cluster centroids. Pick any two clusters, and report their centroids. Which of the features are these clusters most different on? (note that the order in which the centroid’s values are reported has the same order of the feature labels (column labels.))

Note that if you wanted to obtain which country belonged to which cluster you would use kmeans.labels\_ script which would give you a list of N numbers going from 1 to K. First number would correspond to the cluster ID of the first Country on the row etc…