Jeopardy Questions:

**Unsupervised Learning:**

**1. What is the difference between supervised and unsupervised learning?**

Unsupervised learning is where we have, as part of our training data, no observed y values.

Supervised learning is where we have observed y values as part of our training data.

**2. What are two metrics for determining how “good” a cluster is?**

- how much spread is there within a cluster?

- how much distance is there between clusters?

Inertia: within cluster distance - density

Silhouette Score: between cluster distance

**3. Give an example of using unsupervised learning as a stepping stone to supervised learning:**

- cluster stores by particular characteristics, label clusters and and use those as a variable in a supervised model

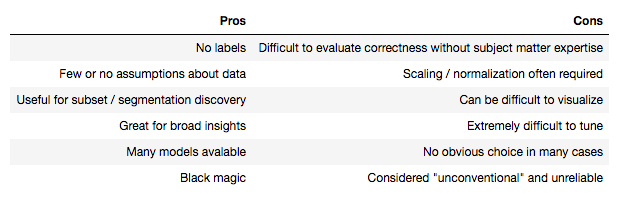
**4. Give an example of an unsupervised learning problem:**

organize 1,400 Target stores by demographic profiles so I can better market to them

**5. What are the two different clustering techniques we learned this week?**

KMeans, DBSCAN

**6. Name 3 pros and 3 cons to clustering?**



**K Means:**

**1. What is K in K means?**

The number of clusters that will be assigned.

**2. Describe how KMeans works:**

K is the number of clusters chosen in advance. The goal is to partition the data into sets such that the total sum of squared distances from each point to the mean point of the cluster is minimized.

The algorithm takes your entire dataset and iterates over its features and observations to determine clusters based around center points. These center points are known as **centroids**.

K-means iterative fitting:

1. Pick a value for k (the number of clusters to create)
2. Initialize k 'centroids' (starting points) in your data
3. Create your clusters. Assign each point to the nearest centroid.
4. Make your clusters better. Move each centroid to the center of its cluster.
5. Repeat steps 3-4 until your centroids converge.

**3. What is a method you can use to try to determine the approximate appropriate number for k in KMeans clustering?**

The elbow method

**4. What must you do to your data when you are going to use a clustering algorithm or PCA?**

SCALE! – this is true MOST OF THE TIME when modeling anthing

**5.What is considered a good Silhouette score? Inertia?**

The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

Negative values generally indicate that a sample has been assigned to the wrong cluster

low inertia = dense cluster (0-1)

**PCA:**

1. **What is PCA used for?**
   1. Dimensionality reduction
2. **What is a good indicator of when we should definitely try to reduce dimensionality?**
   1. When the number of columns exceeds the number of rows
3. **what is the difference between feature extraction and feature elimination?**

Feature extraction: we can apply PCA so that we retain all of the “good” information from all of our features, but discard the redundant information from our features!

Feature reduction: removing entire variables from the model

1. Describe, generally, how PCA works:

### Big picture, what is PCA doing?

1. We are going to look at how all of the X variables relate to one another, then summarize these relationships.

2. Then, we will take this summary and look at which combinations of our X variables are most important.

3. We can also see exactly how important each combination is, then rank these combinations.

4. Once we've taken our original X data and transformed it into Z, we can then drop the columns of Z that are "least important."

* PCA finds *linear combinations* of current predictor variables that...
* create new "principal components". The principal components explain...
* the maximum possible amount of variance in your predictors.

1. **What are 3 benefits to dimensionality reduction?**

#### Dimensionality reduction has a number of advantages:

* Increases computational efficiency when fitting models.
* Can help with addressing a multicollinearity problem.
* Makes visualization simpler (or feasible).

**DBSCAN:**

**1. Give 2 major differences between KMeans and DBSCAN**:

* with DBSCAN, you don’t predetermine the number of clusters being made
* Kmeans deals with distance from the centroid while DBSCAN is density based
* You can’t calculate inertia with DBSCAN
* DBSCAN is a density based clustering algorithm, meaning that the algorithm finds clusters by seeking areas of the dataset that have a higher density of points than the rest of the dataset

**2. What are the 2 parameters you can tune in DBSCAN? And describe what they are.**

Epsilon:  which defines a distance boundary from a point

Min\_samples: The number of samples needed to define a cluster

**3. What does DBSCAN stand for?**

## DBSCAN: Density-based Spatial Clustering of Applications with Noise

**4. describe how DBSCAN works – with an illustration**

1. Choose an “epsilon” and “min\_samples”
2. Pick an arbitrary point (*core sample*), and check if there are at least “min\_samples” points within distance “epsilon”
   * If **yes**, add those points to the cluster and check each of the new points
   * If **no**, choose another arbitrary point (*core sample*) to start a new cluster
3. Stop once all points have been checked

DBSCAN will take the epsilon and minimum points we provided it and cluster all of the points in a neighborhood, first passing the minimum points requirement and then clustering each of the points within epsilon distance to form the clusters. Once one cluster is formed, the algorithm then moves to a new datapoint, and seeks to find related points to form yet another cluster; this will continue until DBSCAN simply runs out of points!

## 5. When might it be advantageous to use DBSCAN over KMeans?

## DBSCAN vs K-Means

* **K-Means** can be thought of as a "general" clustering approach, DBSCAN performs especially well with unevenly distributed, non-gausian clusters.
* **DBSCAN** can be useful to us when we have a lot of dense data. If we used **K-Means** on this data, the algorithm would effectively give us just one large cluster! However with DBSCAN, we can actually break down this cluster into smaller groups to see their attributes.

**Recommender Systems:**

Other than the example we went through in class, what is an example of a user-item recommender system?

Netflix users

What is cosine similarity and how it is it determined? (BRAD)

What sklearn library is cosine similarity in?

Sklearn.metrics\_pairwise

Give an example of item based collaborative filtering:

Using a list comprehension Multiply every item in a list by three and assign it to a new variable name

Using list comprehension, make a list of uppercase strings to lower case strings