GENERAL ASSEMBLY

K MEANS CLUSTERING

Brian Chung

SOMETHING COOL

Is your model smarter than an 8th grader?

https://www.kaggle.com/c/the-allen-ai-science-challenge

http://blog.talla.com/2016/01/how-we-approached-the-allen-a-i-challenge-on-kaggle/

Why Deep Learning Won't Work On A Problem Like This

If you don't know much about machine learning other than what you read in the tech press, your first thought is probably "just throw deep learning at it." Right? Not exactly.

The Allen AI challenge is a difficult use case for deep learning because answering questions is highly dependent on the structure of natural language questions, as well as an accumulated knowledge base. Recently there has been an increased use of 'word vectors' that allow deep learning to effectively perform many NLP tasks. In this case, however, converting question words to vectors is insufficient because the resulting representation does not capture the *compositional* nature of language. For example, take the question "Which of the following is not a way that plants extract energy from their environment?". The usage of *not* changes the meaning of the sentence. "Their environment" is a unit in the sentence that also references the earlier "plants." Learning to parse sentences like these constitutes a learning task in its own right, and people can manually specify grammars for parsing language that are currently hard to beat statistically.

Compounding the difficulty of this task is the problem of using acquired knowledge. A person attempting to answer the questions on the test has an internal model of the the facts that the question corresponds to. *Plants* exist in a given *environment* and *extract energy* in several *ways.* People are able to recall information relevant to the question and organize it in a way that helps answer the question. Research pairing deep learning with memory is an ongoing effort, but similar to natural language processing, shortcuts such as data retrieval are often still more efficient. For example, most search engines encode large bodies of facts in a regularly and easily processable form rather than trying to extract information from a representation in a neural network. Taken together, the problems of natural language processing and utilizing preexisting knowledge bases create a difficult challenge that is often easier to tackle with hand crafted modeling and engineering, rather than end-to-end deep learning. Maybe someday deep learning will solve this problem, but today is not that day.

FROM LAST TIME

Questions:

- L1 and L2 logistic regression comparison
- Do you tune C, and THEN the ROC curve? or do you do both? Or do you tune ROC and then C?
- clarity into C value and modulating C
- how do outliers affect logistic regression

RECAP

Numpy Pandas

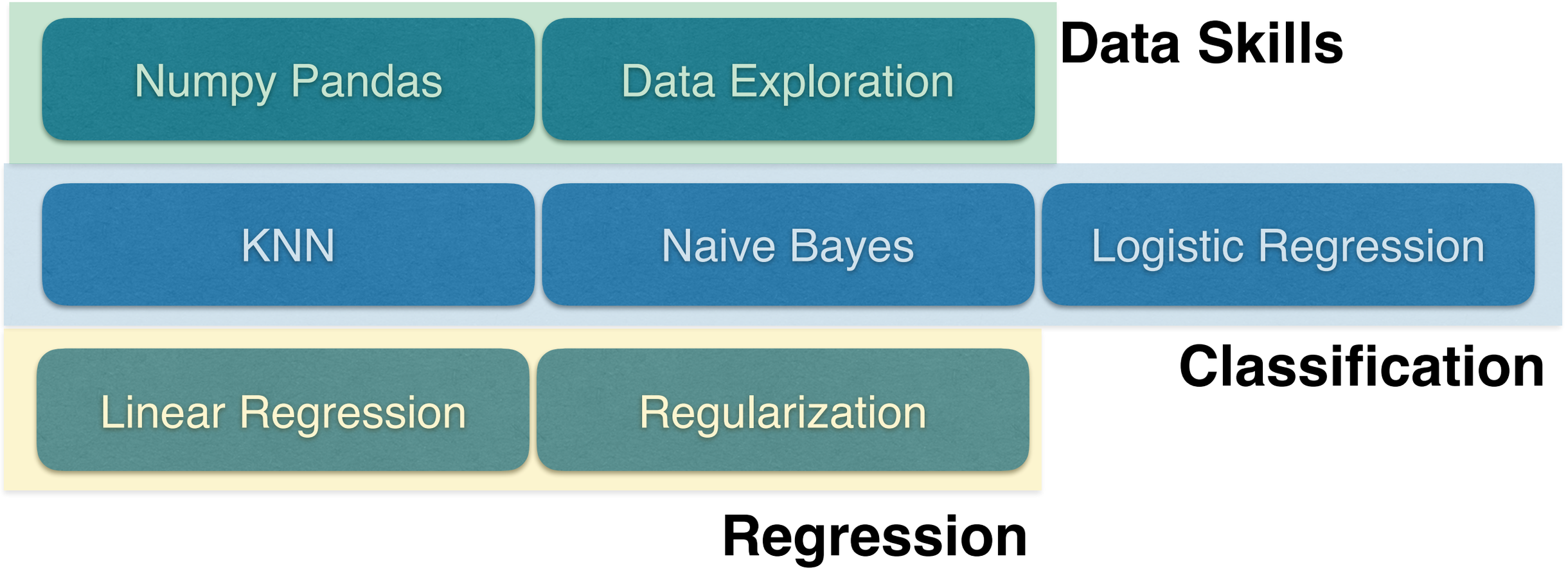
Data Exploration

Data Skills

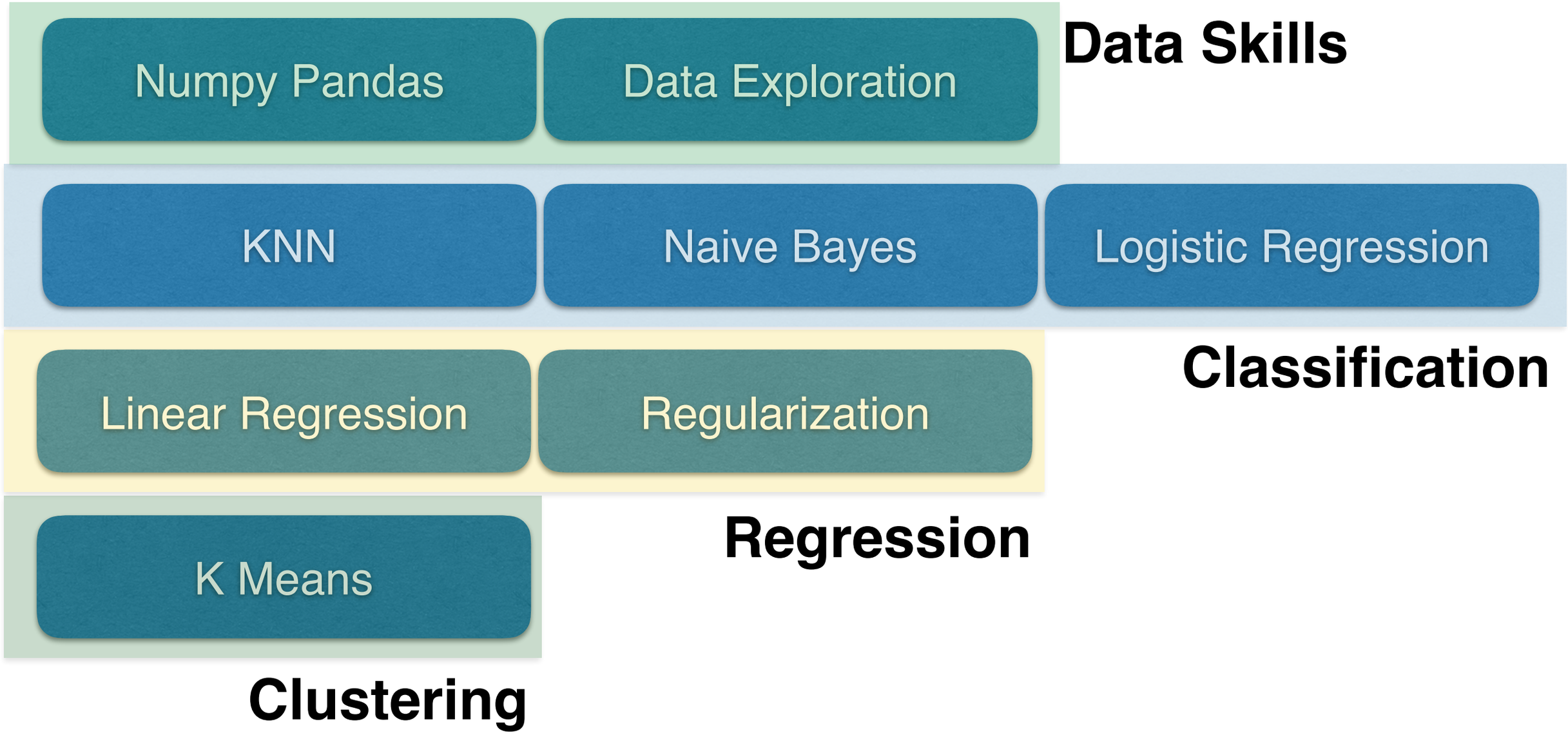
RECAP


Numpy Pandas
Data Exploration
Data Skills
KNN
Naive Bayes
Logistic Regression
Classification

RECAP


Numpy Pandas
Data Exploration
Data Skills
KNN
Naive Bayes
Logistic Regression
Classification
Linear Regression
Regularization
Regression

RECAP


Numpy Pandas
Data Exploration
Data Skills
KNN
Naive Bayes
Logistic Regression
Classification
Linear Regression
Regularization
Regression
K Means
Clustering

K MEANS CLUSTERING AGENDA

I. Cluster Analysis
II. K-Means Clustering
III. Validation

I. CLUSTER ANALYSIS

TYPES OF ML SOLUTIONS

	Continuous	Categorical
Supervised	Regression	Classification
Unsupervised	Dimension Reduction	Clustering

TYPES OF ML SOLUTIONS

	Continuous	*Categorical*
Supervised		
Unsupervised		***Discover patterns***

CLUSTER ANALYSIS

Q: What is a cluster?

CLUSTER ANALYSIS

Q: What is a cluster?

A: A group of **similar** data points

CLUSTER ANALYSIS

Q: What is a cluster?

A: A group of **similar** data points

The concept of similarity is central to the definition of a cluster, and therefore to cluster analysis.

CLUSTER ANALYSIS

Q: What is a cluster?

A: A group of **similar** data points

The concept of similarity is central to the definition of a cluster, and therefore to cluster analysis.

In general, greater similarity between points leads to better clustering

CLUSTER ANALYSIS

Q: What is the purpose of cluster analysis?

CLUSTER ANALYSIS

Q: What is the purpose of cluster analysis?

A: To enhance our understanding of a dataset by dividing data into groups

CLUSTER ANALYSIS

Q: What is the purpose of cluster analysis?

A: To enhance our understanding of a dataset by dividing data into groups

Ex: *You sample 100 people's arm and chest measurements. Then you cluster these measurements into three groups in order to create three t-shirt sizes*

CLUSTER ANALYSIS

Q: What is the purpose of cluster analysis?

A: To enhance our understanding of a dataset by dividing data into groups

Ex: *You have an image with 32-bit color but you want to reduce the file size of the image. You decide to cluster into 10 groups of colors, and assign each pixel the average of the colors in that group*

CLUSTER ANALYSIS

Q: What is the purpose of cluster analysis?

A: To enhance our understanding of a dataset by dividing data into groups

Ex: *You're an owner of a pizza chain, and you want to open a few stores. You can analyze the* ***clusters*** *from where the pizza is being ordered frequently.*

CLUSTER ANALYSIS

Q: What is the purpose of cluster analysis?

A: To enhance our understanding of a dataset by dividing data into groups

Ex: ***Clustering*** *can help marketers improve their customer base. They can target* ***groups*** *of people based on their similarity (i.e. purchasing preferences, purchasing power, etc.)*

CLUSTER ANALYSIS

Q: What is the purpose of cluster analysis?

A: To enhance our understanding of a dataset by dividing data into groups

Clustering provides a *layer of abstraction* from the individual datapoints, and is one of many **prototype methods**. We define the data in terms of **prototypical points**.

CLUSTER ANALYSIS

Q: What is the purpose of cluster analysis?

A: To enhance our understanding of a dataset by dividing data into groups

Clustering provides a *layer of abstraction* from the individual datapoint, and is one of many **prototype methods**. We define the data in terms of **prototypical points**.

The goal is to extract and enhance the natural structure of the data (and not to impose an arbitrary structure)

CLUSTER ANALYSIS

Q: How do you solve a clustering problem?

CLUSTER ANALYSIS

Q: How do you solve a clustering problem?

A: Think of a cluster as a ‘potential class’, then the solution to a clustering problem is to programmatically determine these classes

CLUSTER ANALYSIS

Q: How do you solve a clustering problem?

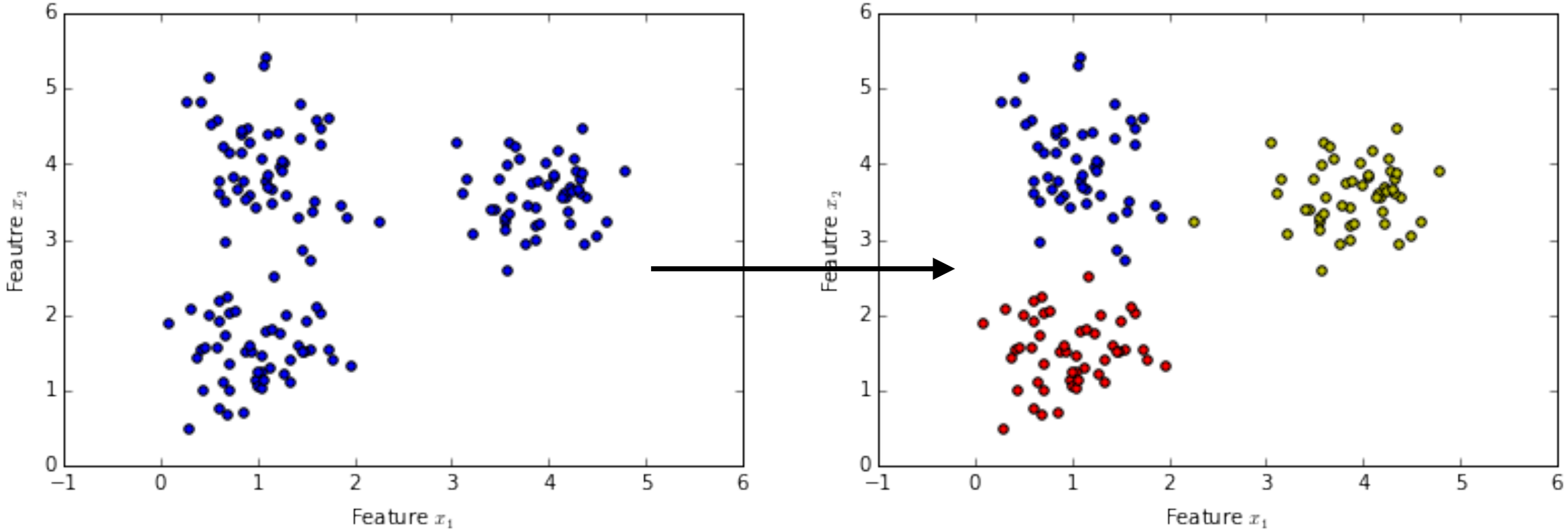
A: Think of a cluster as a ‘potential class’, then the solution to a clustering problem is to programmatically determine these classes

The real purpose of clustering can be data exploration, so a solution is anything that contributes to you understanding.

I. CLUSTER ANALYSIS
II. K-MEANS CLUSTERING

K-MEANS CLUSTERING

K-Means Clustering is a **greedy** learner that **partitions** a dataset into k clusters



K-MEANS CLUSTERING

K-Means Clustering is a **greedy** learner that **partitions** a dataset into k clusters

greedy - *captures local structure (depends on initial conditions)*

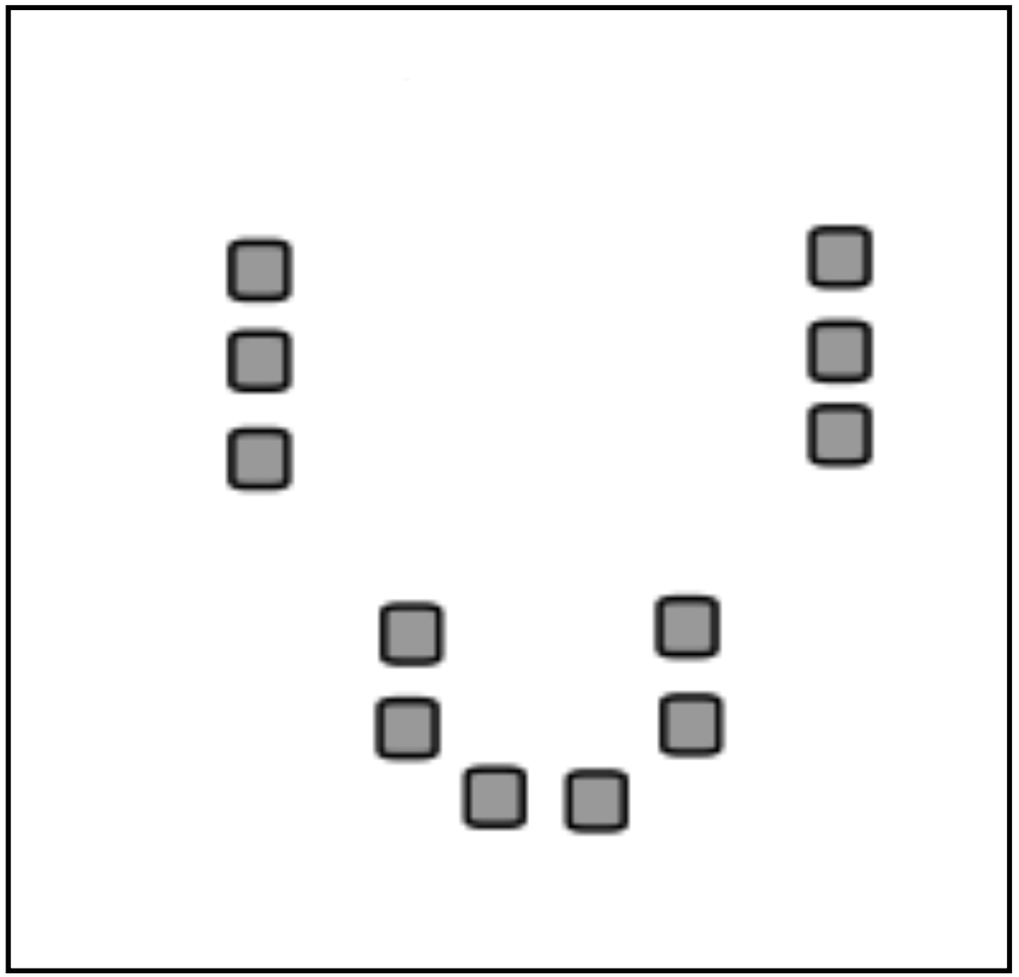
K-MEANS CLUSTERING

K-Means Clustering is a **greedy** learner that **partitions** a dataset into k clusters

greedy - *captures local structure (depends on initial conditions)*
partition *- each point belongs to exactly one cluster*

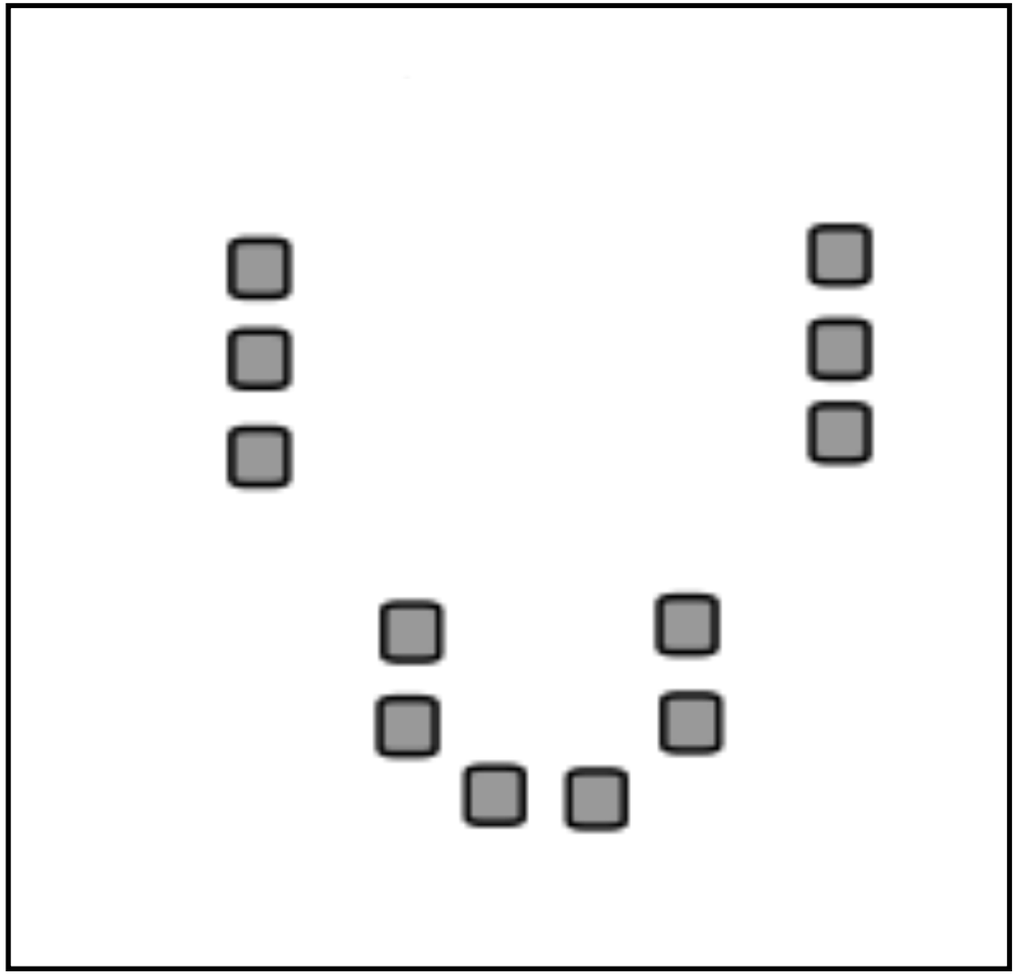
K-Means is algorithmically efficient
(Linear in time and memory by number of records)

K-MEANS ALGORITHM



Suppose we have data with no labels (i.e. unsupervised learning)

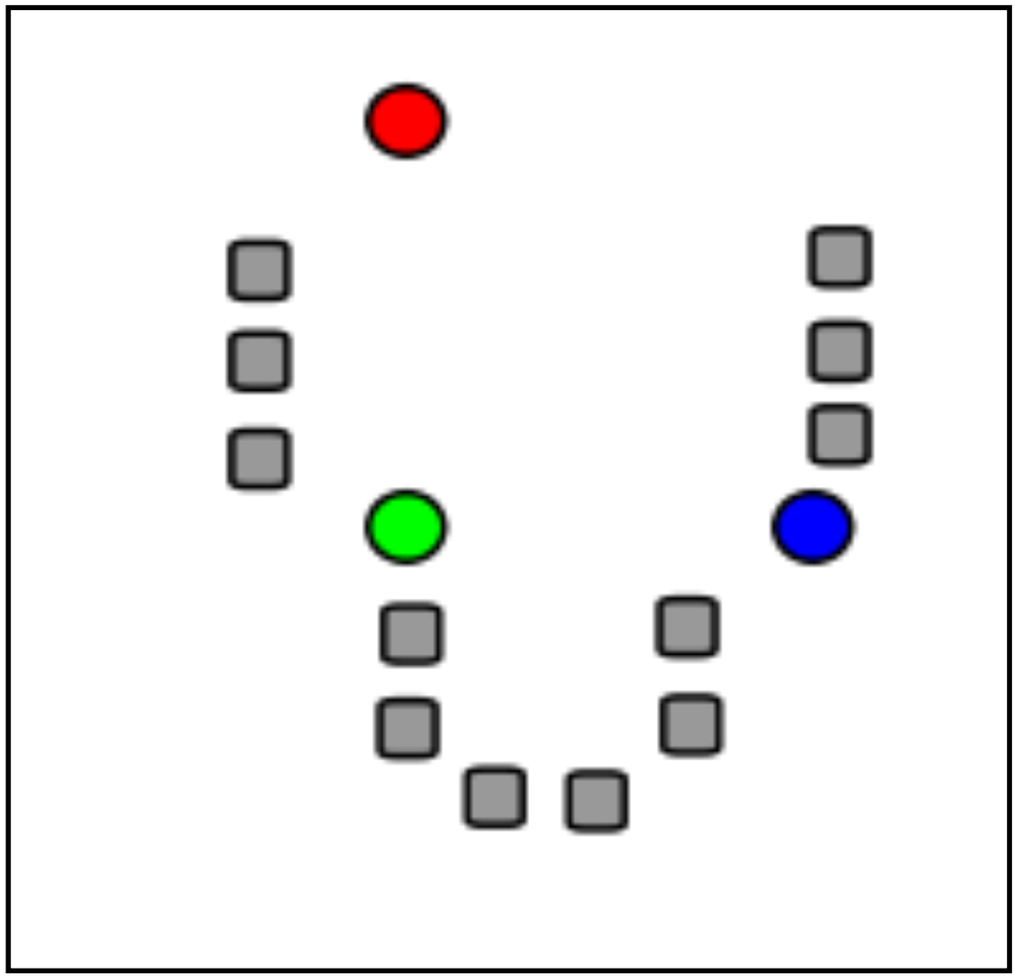
K-MEANS ALGORITHM



Suppose we have data with no labels (i.e. unsupervised learning)

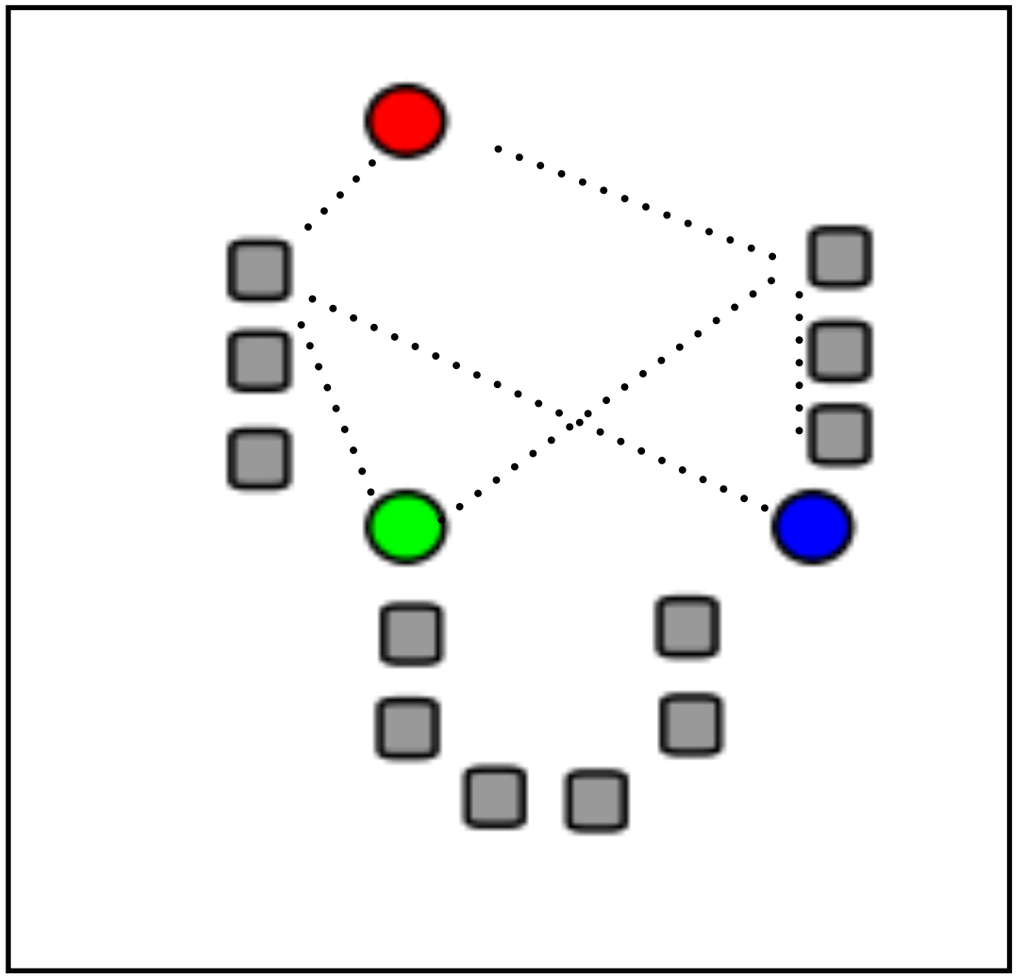
We would like to infer class labels for the data, or assign each data point to a class label

K-MEANS ALGORITHM



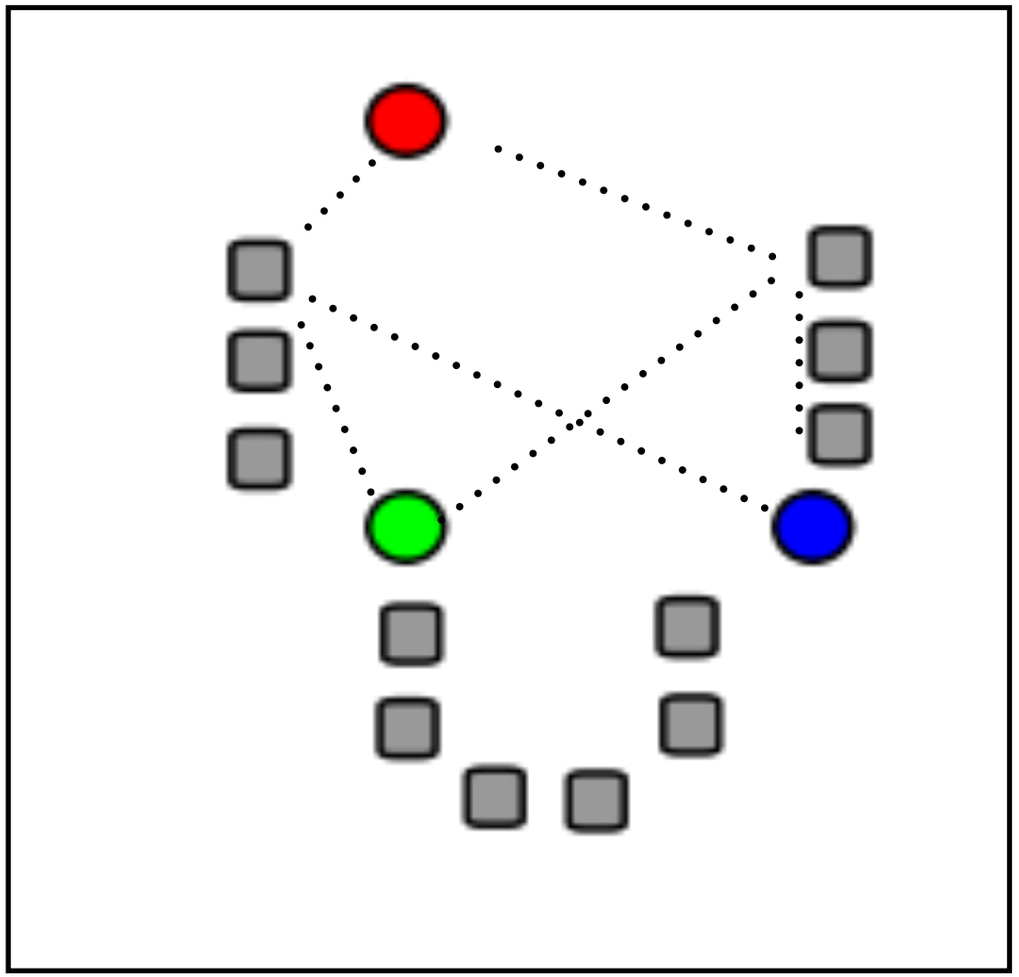
1. Start with **k** cluster centers (centroids) chosen at random amongst the points

K-MEANS ALGORITHM



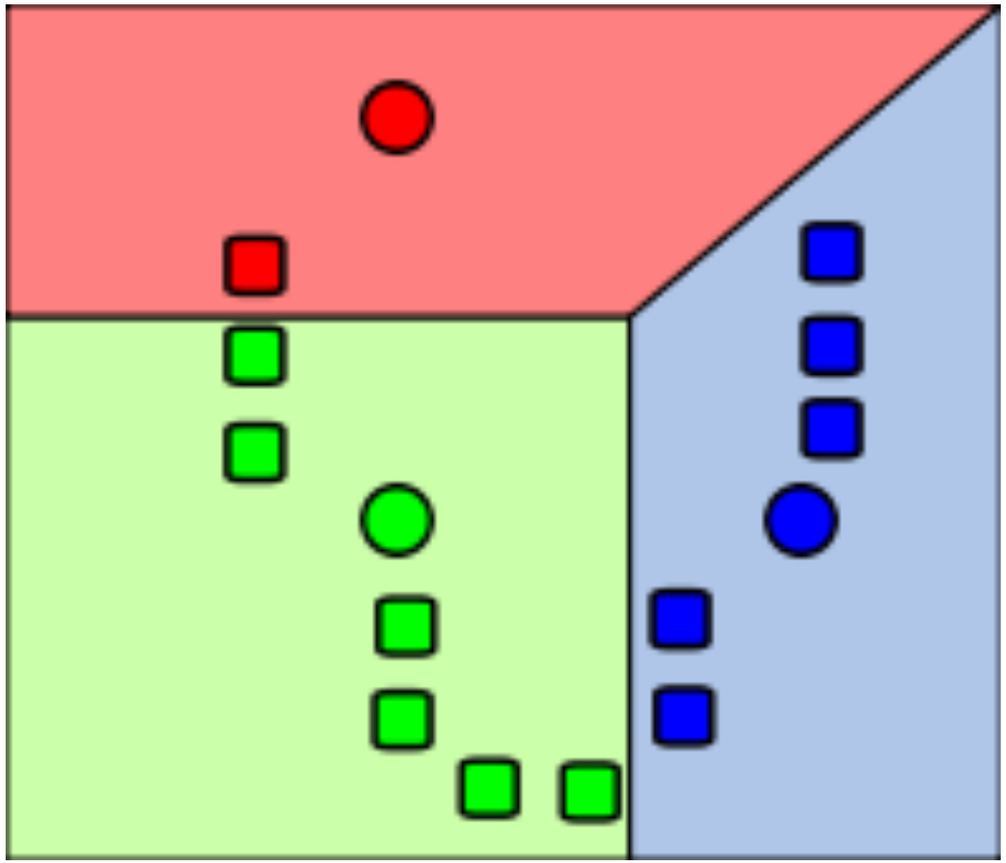
1. Start with **k** cluster centers (centroids) chosen at random amongst the points
2. Compute distances to each centroid from every point

K-MEANS ALGORITHM



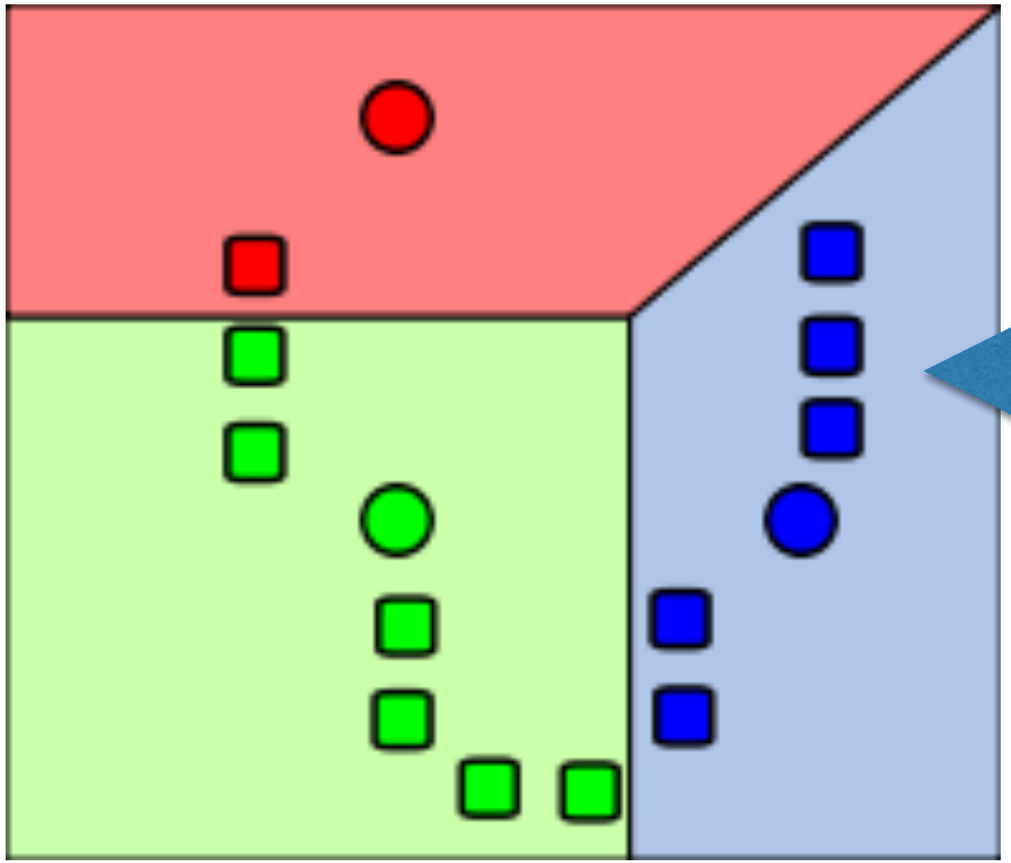
1. Start with **k** cluster centers (centroids) chosen at random amongst the points
2. Compute distances to each centroid from every point **Just like KNN, we need to scale the features! (i.e. x / sd(x))**

K-MEANS ALGORITHM



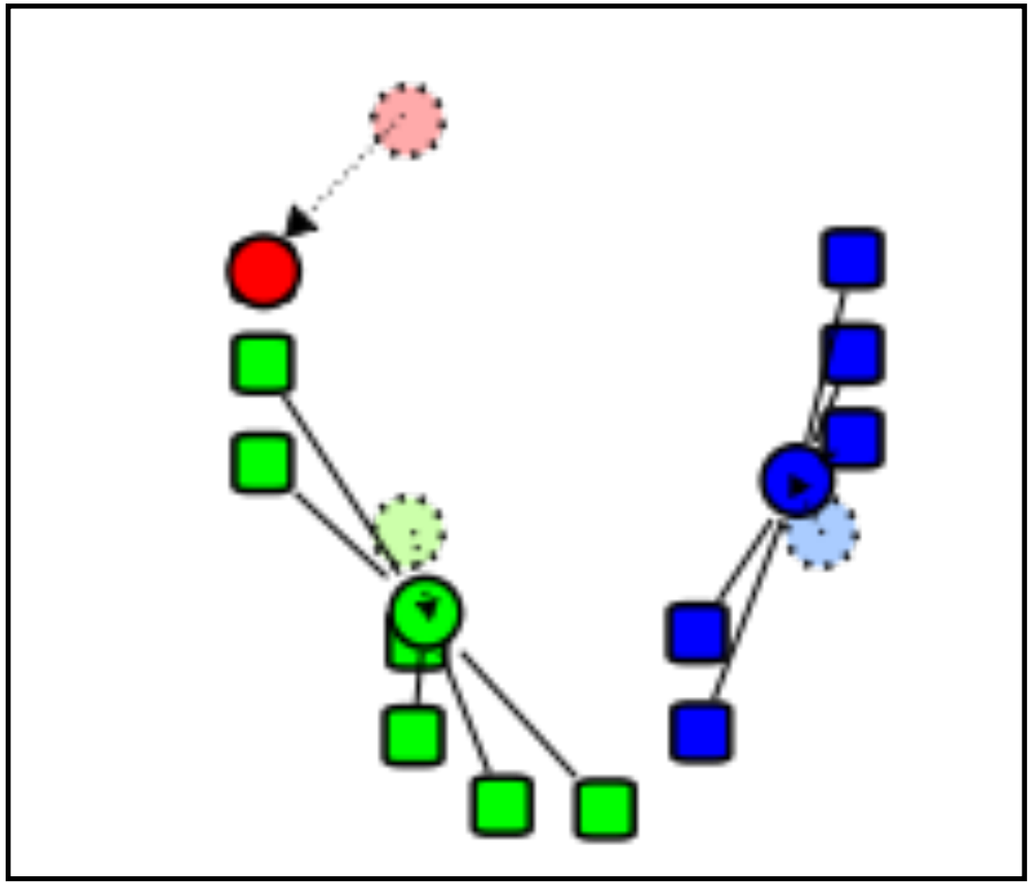
1. Start with **k** cluster centers (centroids), chosen at random amongst the points
2. Compute distances to each centroid from every point **Just like KNN, we need to scale the features! (i.e. x / sd(x))**
3. Label each point according to its closest centroid

K-MEANS ALGORITHM



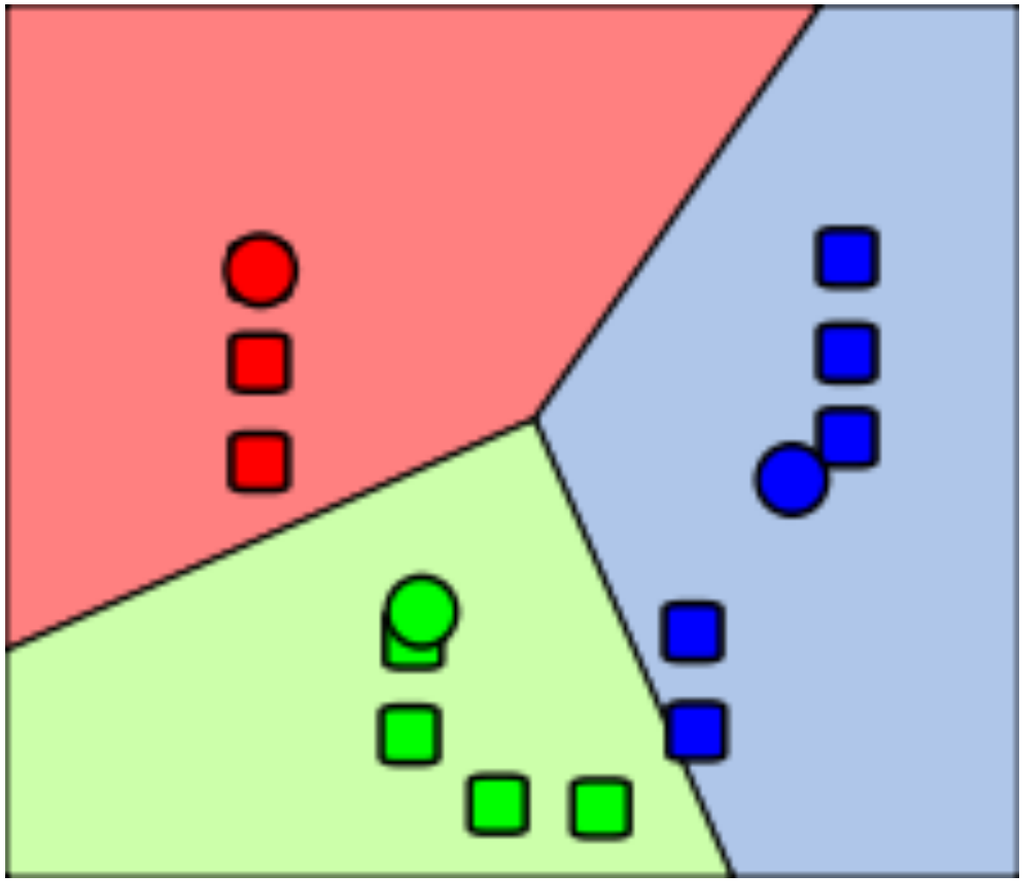
This is called a Voronoi diagram, which indicates graphically the partitions of the feature space into their respective clusters

K-MEANS ALGORITHM



1. Start with **k** cluster centers (centroids), chosen at random amongst the points
2. Compute distances to each centroid from every point **Just like KNN, we need to scale the features! (i.e. x / sd(x))**
3. Label each point according to its closest centroid
4. Recompute each cluster centroid with the datapoints within that cluster

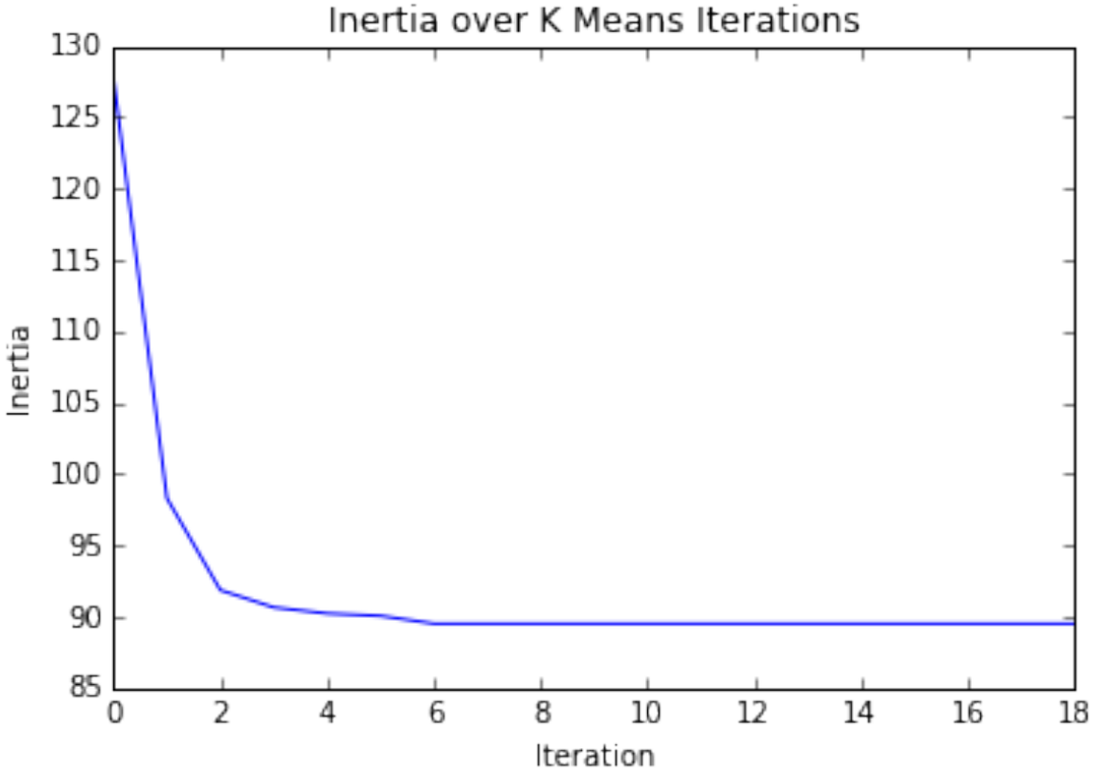
K-MEANS ALGORITHM



1. Start with **k** cluster centers (centroids), chosen at random amongst the points
2. Compute distances to each centroid from every point **Just like KNN, we need to scale the features! (i.e. x / sd(x))**
3. Label each point according to its closest centroid
4. Recompute each cluster centroid with the datapoints within that cluster
5. Repeat steps 2-4 until convergence. i.e. For n iterations, or until the centroids do not change much

Let’s try an example…

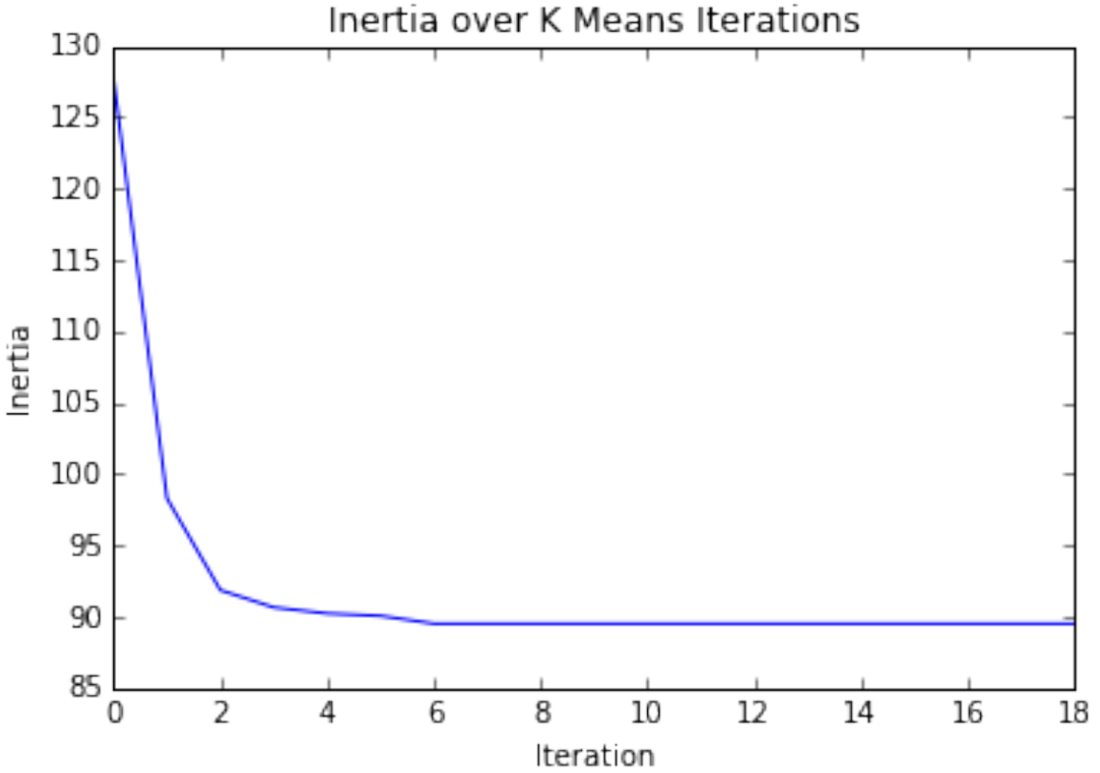
K-MEANS – ANOTHER VIEW



K Means clustering could be thought of in a single line as minimizing the **squared euclidean distance** from within cluster points to the centroids (aka cluster means).

$$\sum_{i}^{n} \min_{\mu_j \in C} \left(\left|\left| x_j - \mu_i \right|\right|^2 \right)$$

K-MEANS – ANOTHER VIEW



K Means clustering could be thought of in a single line as minimizing the **squared euclidean distance** from within cluster points to the centroids (aka cluster means).

$$\sum_{i}^{n} \min_{\mu_j \in C} \left(\left|\left| x_j - \mu_i \right|\right|^2 \right)$$

This is also called **inertia**

K-MEANS STRENGTHS AND WEAKNESSES

✓K Means is very efficient and is linear in time and space by the number of records

K-MEANS STRENGTHS AND WEAKNESSES

✓K Means is very efficient and is linear in time and space by the number of records

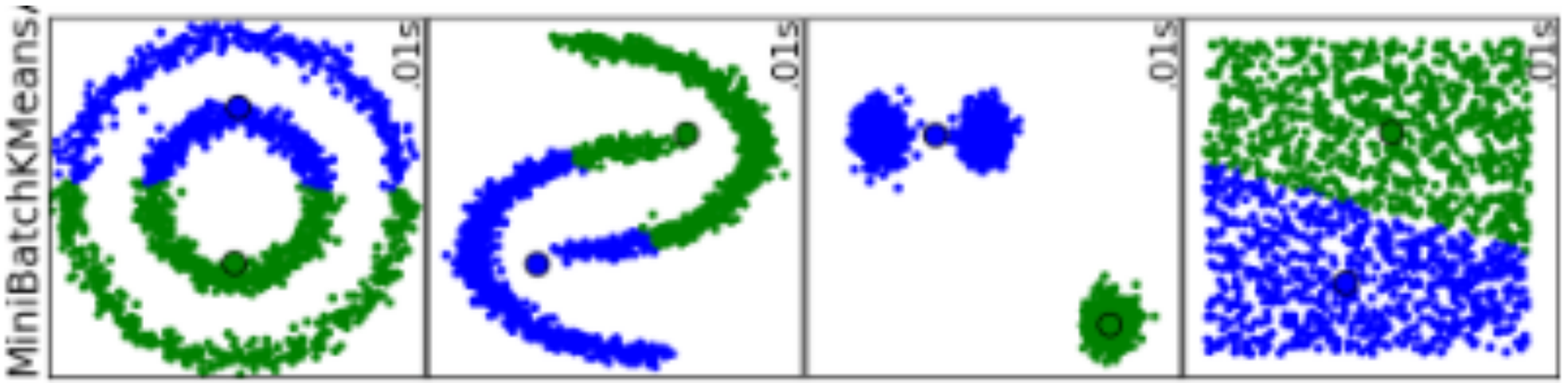
✓K Means is also very simple to code compared to other algorithms and easy to debug

K-MEANS STRENGTHS AND WEAKNESSES

✓K Means is very efficient and is linear in time and space by the number of records

✓K Means is also very simple to code compared to other algorithms and easy to debug

- K Means has a hard time dealing with non-convex clusters, or data with widely varying shapes and densities



(from sklearn documentation)

K-MEANS STRENGTHS AND WEAKNESSES

✓K Means is very efficient and is linear in time and space by the number of records
✓K Means is also very simple to code compared to other algorithms and easy to debug
✓Will Always converge with enough time

- K Means has a hard time dealing with non-convex clusters, or data with widely varying shapes and densities (though you can somewhat mitigate this by throwing on more clusters)
- Each points' features need to be scaled correctly
- May converge to a local minima

K-MEANS STRENGTHS AND WEAKNESSES

✓K Means is very efficient and is linear in time and space by the number of records
✓K Means is also very simple to code compared to other algorithms and easy to debug
✓Will Always converge with enough time

- K Means has a hard time dealing with non-convex clusters, or data with widely varying shapes and densities (though you can somewhat mitigate this by throwing on more clusters)
- Each points' features need to be scaled correctly
- May converge to a local minima
- Outliers can screw your centroids up

K-MEANS STRENGTHS AND WEAKNESSES

✓K Means is very efficient and is linear in time and space by the number of records
✓K Means is also very simple to code compared to other algorithms and easy to debug
✓Will Always converge with enough time

- K Means has a hard time dealing with non-convex clusters, or data with widely varying shapes and densities (though you can somewhat mitigate this by throwing on more clusters)
- Each points' features need to be scaled correctly
- May converge to a local minima
- Outliers can screw your centroids up

Try preprocessing or k-medians

K-MEANS STRENGTHS AND WEAKNESSES

✓K Means is very efficient and is linear in time and space by the number of records
✓K Means is also very simple to code compared to other algorithms and easy to debug
✓Will Always converge with enough time

- K Means has a hard time dealing with non-convex clusters, or data with widely varying shapes and densities (though you can somewhat mitigate this by throwing on more clusters)
- Each points' features need to be scaled correctly
- May converge to a local minima
- Outliers can screw your centroids up
- Lastly, need continuous variables

K-MEANS STRENGTHS AND WEAKNESSES

✓K Means is very efficient and is linear in time and space by the number of records
✓K Means is also very simple to code compared to other algorithms and easy to debug
✓Will Always converge with enough time

- K Means has a hard time dealing with non-convex clusters, or data with widely varying shapes and densities (though you can somewhat mitigate this by throwing on more clusters)
- Each points' features need to be scaled correctly
- May converge to a local minima
- Outliers can screw your centroids up
- Lastly, need continuous variables

Can try binarizing inputs, or Jacquard coefficient

I. CLUSTER ANALYSIS
II. K-MEANS CLUSTERING
III. VALIDATION

ASSESSING ML PERFORMANCE

How do we assess *supervised* learning algorithms?

ASSESSING ML PERFORMANCE

How do we assess *supervised* learning algorithms?

We test our predictions and compare. We have quality measures such as R squared or accuracy

ASSESSING ML PERFORMANCE

How do we assess *supervised* learning algorithms?

We test our predictions and compare. We have quality measures such as R squared or accuracy

How do we assess *unsupervised* learning algorithms?

ASSESSING ML PERFORMANCE

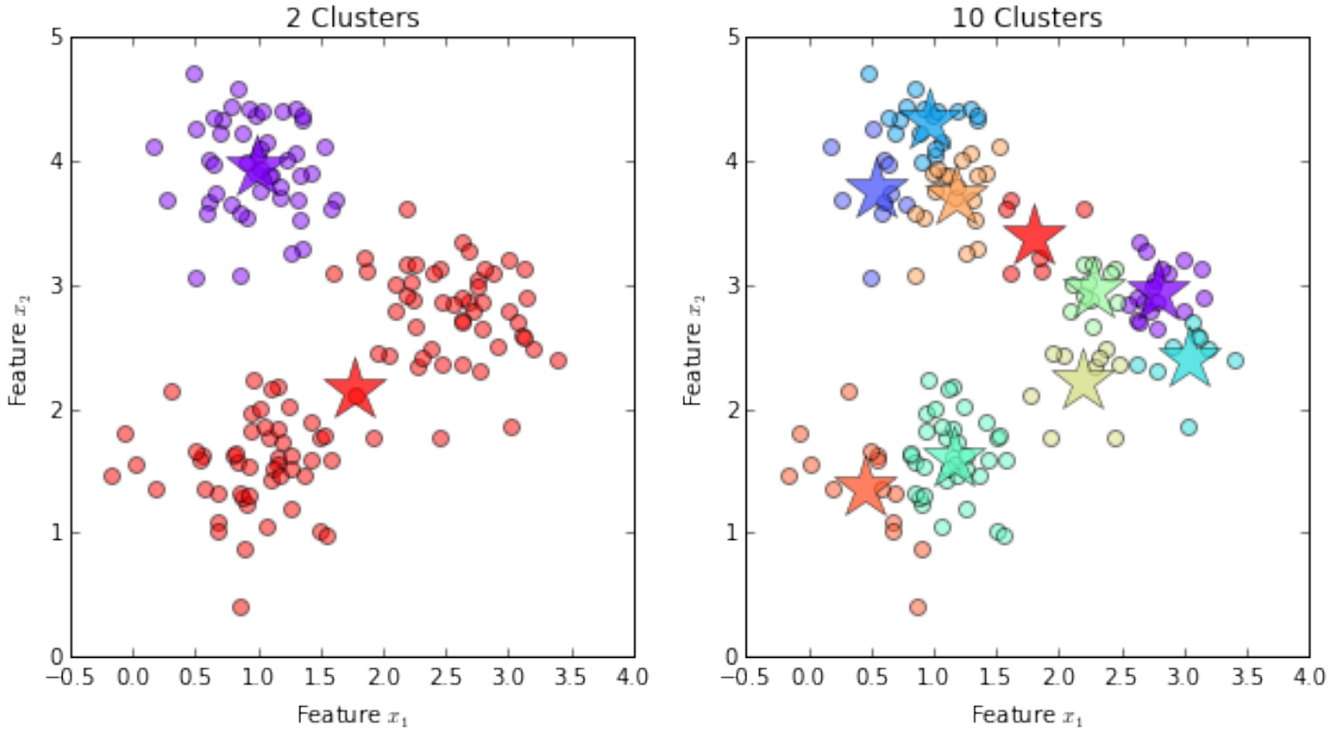
How do we assess *supervised* learning algorithms?

We test our predictions and compare. We have quality measures such as R squared or accuracy

How do we assess *unsupervised* learning algorithms?

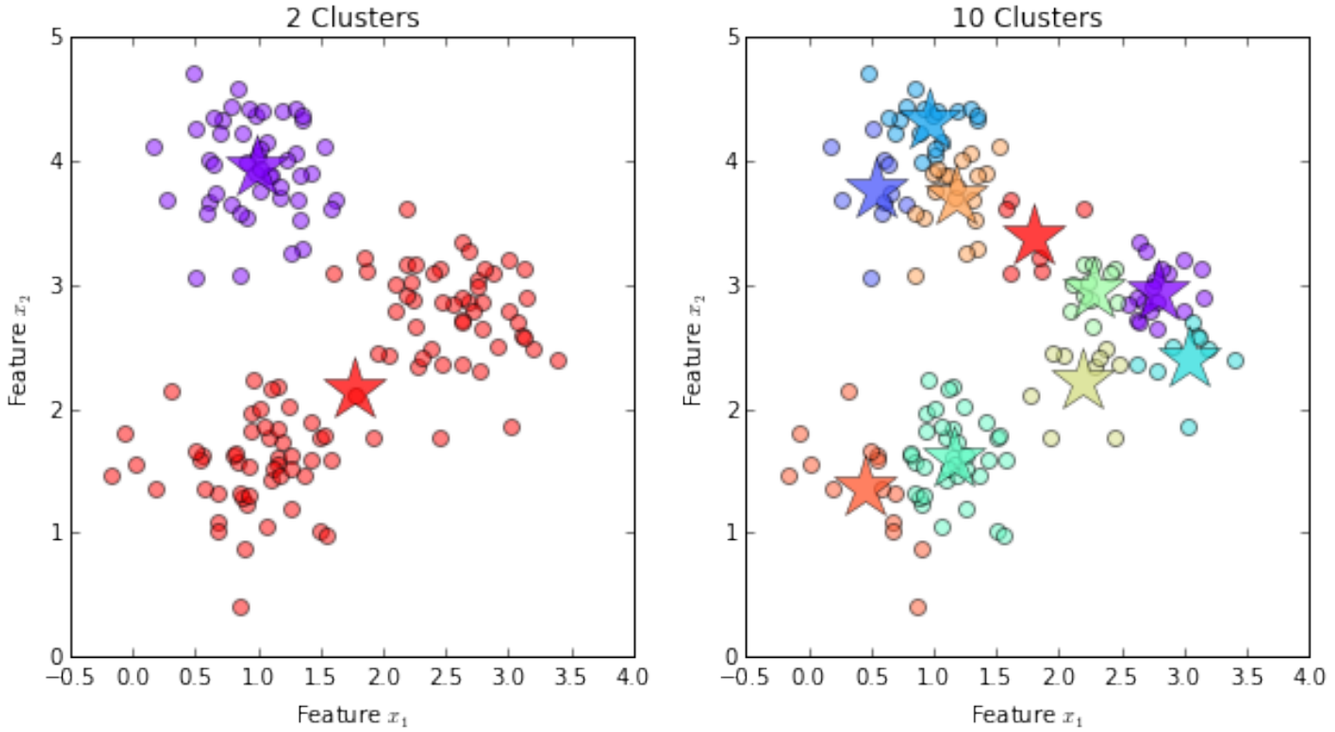
Sort of cant…

So then … how do we choose k?



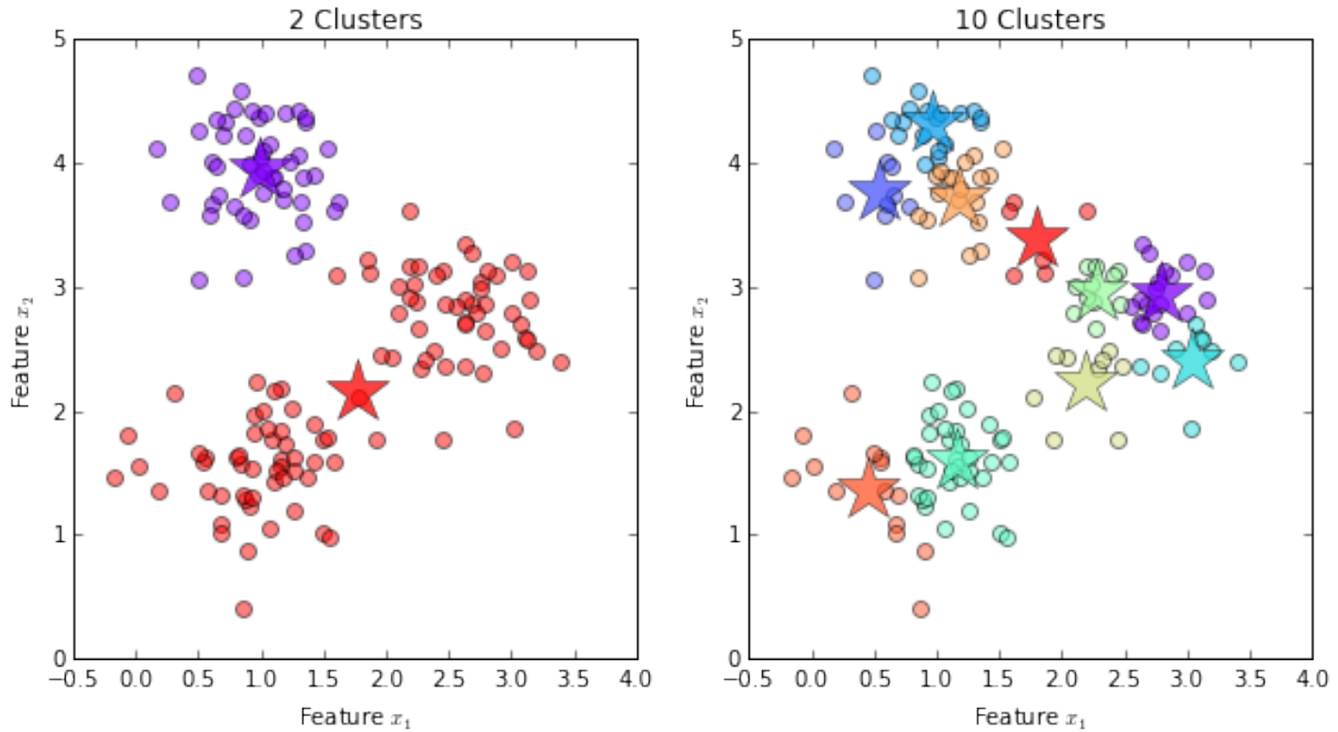
ASSESSING ML PERFORMANCE

Too few *k* clusters and you don't partition your data enough
Too many *k* clusters and you risk making clusters where no natural clusters exist in the data

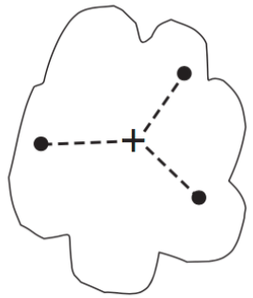


ASSESSING ML PERFORMANCE

We will look at two metrics available to help assess the correct *k*, called cohesion and separation



Cohesion measures clustering effectiveness within a cluster

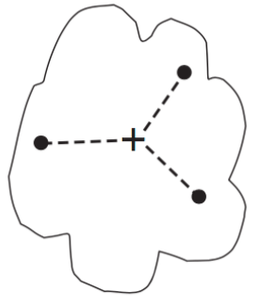


(a) Cohesion.

$$\hat{C}(C_i) = \sum_{x \in C_i} d(x, c_i)$$

CLUSTER VALIDATION

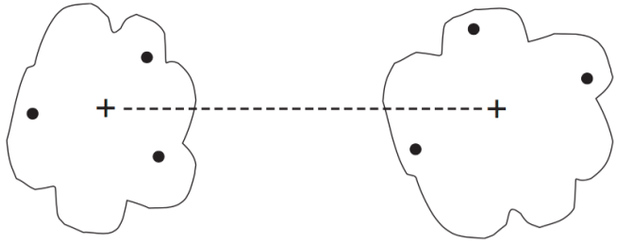
Cohesion measures clustering effectiveness within a cluster



(a) Cohesion.

$$\hat{C}(C_i) = \sum_{x \in C_i} d(x, c_i)$$

Separation measures clustering effectiveness between clusters

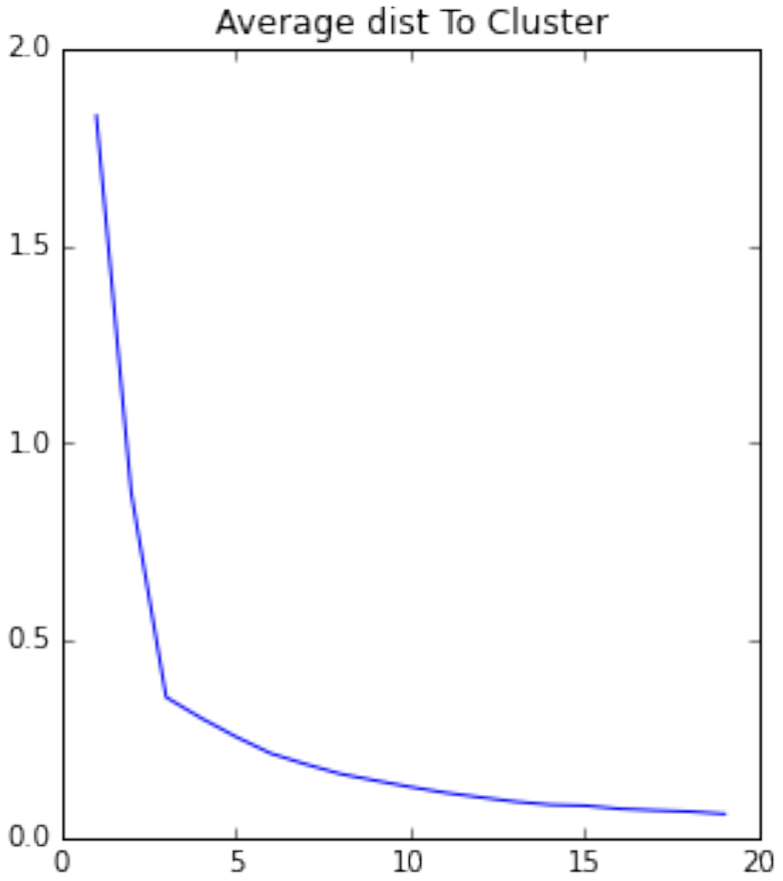


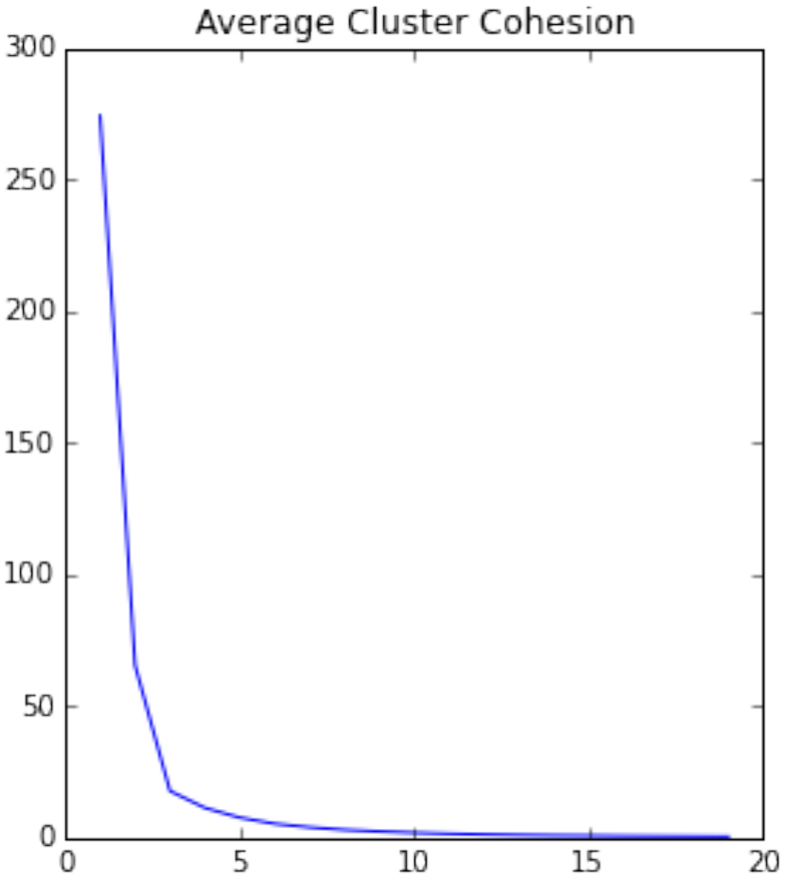
(b) Separation.

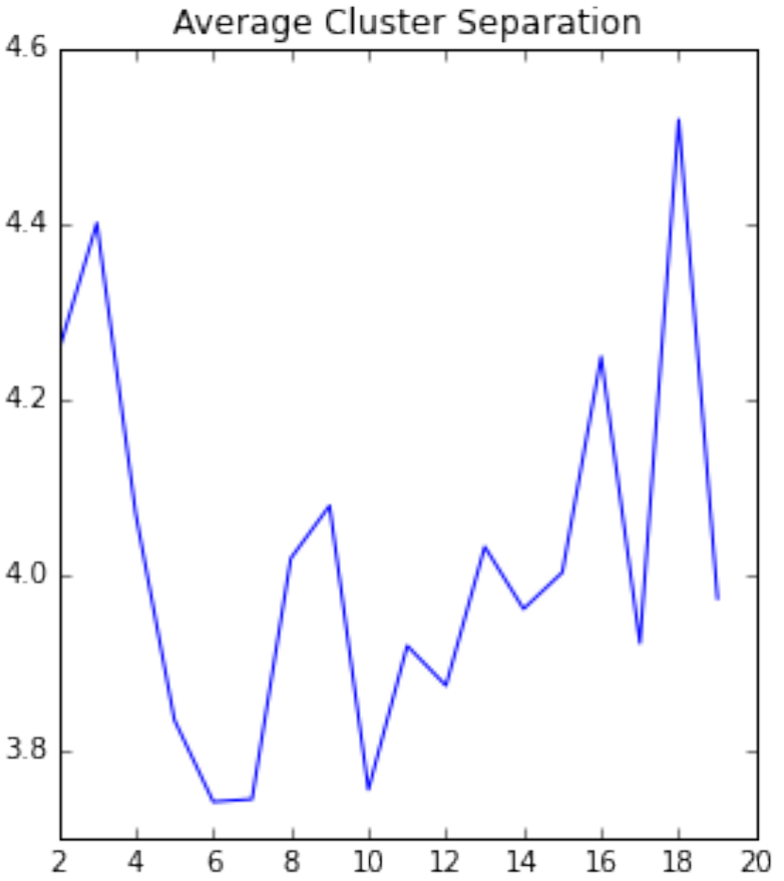
$$\hat{S}(C_i, C_j) = d(c_i, c_j)$$

http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

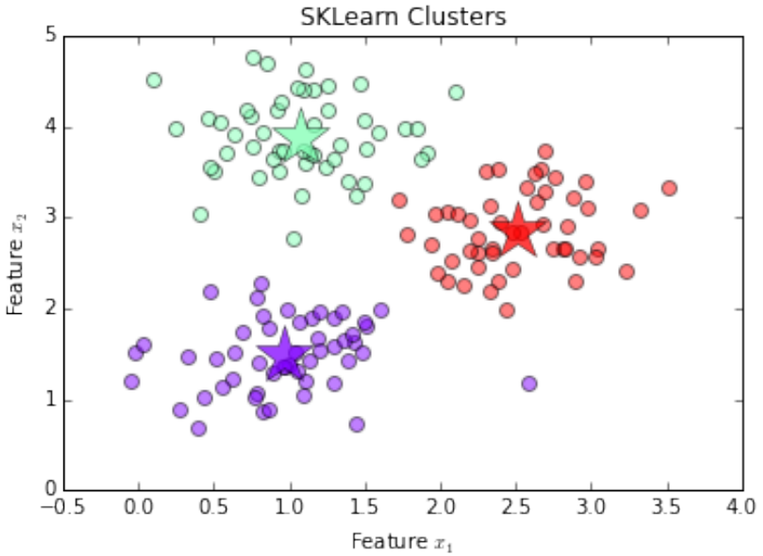
CLUSTER VALIDATION





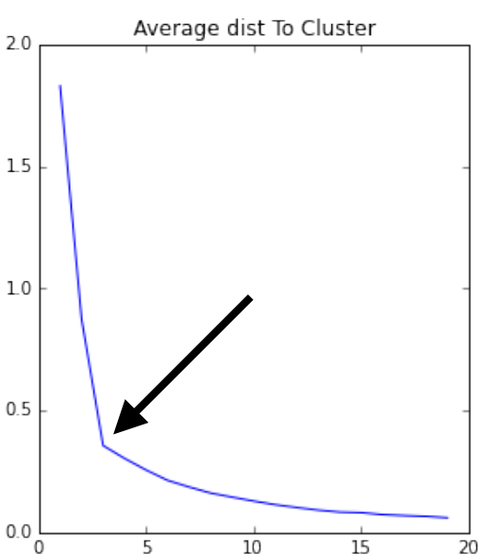


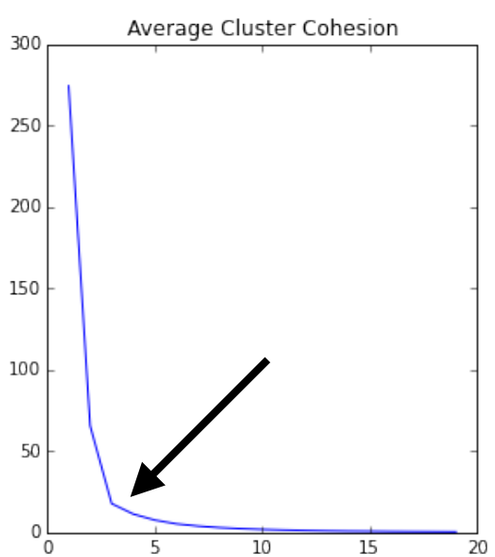
CLUSTER VALIDATION

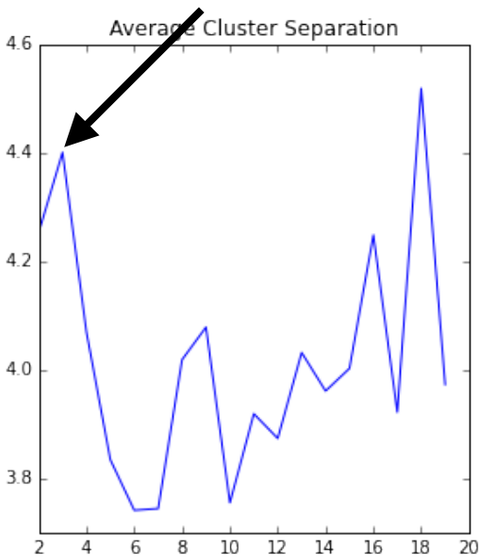


This example has been created from 3 natural clusters (3 fixed centroids plus gaussian noise)

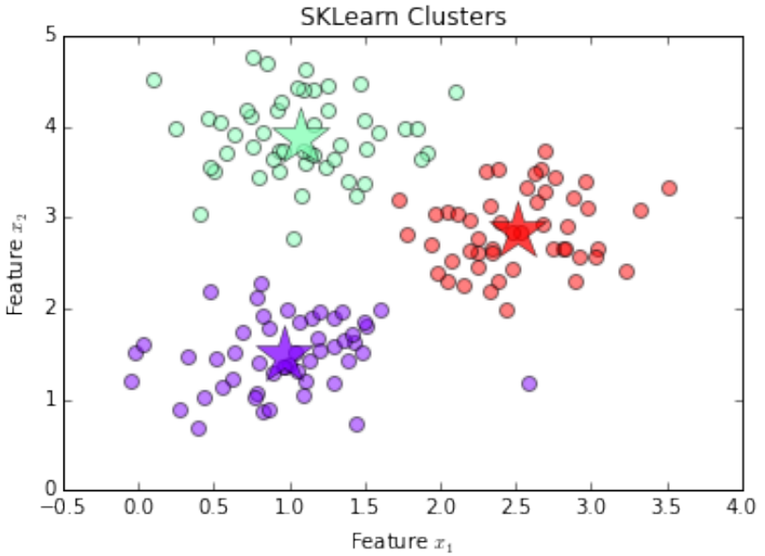
Notice the shape of the curves below







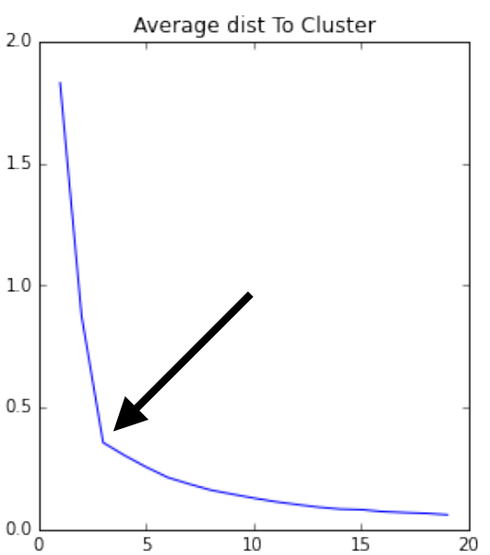
CLUSTER VALIDATION

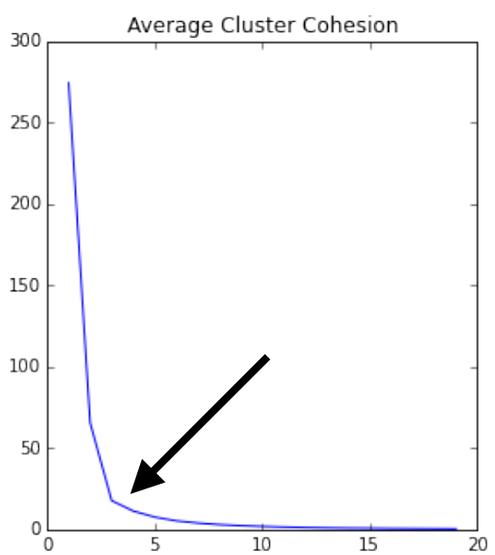


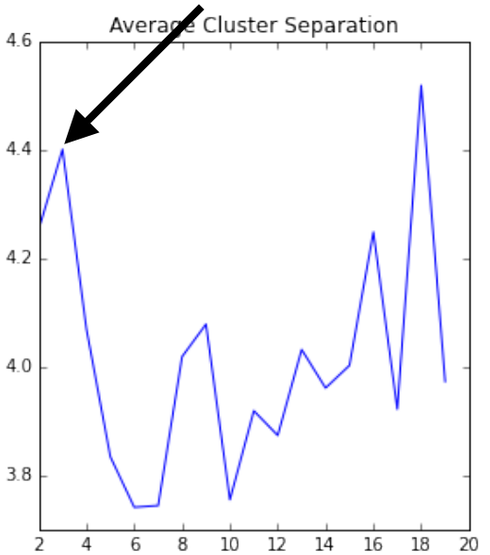
This example has been created from 3 natural clusters (3 fixed centroids plus gaussian noise)

Notice the shape of the curves below

Choosing the k at which the curve kinks is called the "elbow method"







CLUSTER VALIDATION

Cluster validation is certainly an open problem. Since we don't have "labels", and it's hard to call one method right or wrong, many variants of "optimal k selection" have arisen.

http://stats.stackexchange.com/questions/23472/how-to-decide-on-the-correct-number-of-clusters

CLUSTER VALIDATION

Cluster validation is certainly an open problem. Since we don't have "labels", and it's hard to call one method right or wrong, many variants of "optimal k selection" have arisen.

http://stats.stackexchange.com/questions/23472/how-to-decide-on-the-correct-number-of-clusters

However, a good rule of thumb is to choose the number of clusters that intuition or requirements provide you

CLUSTER VALIDATION

- For instance, if you have three Dominos locations you could build in various locations, you could set k = 3

- If you work for Armani and you want to create prototypical sizes for S, M, L, XL you would probably set k = 4

- If you have a product dashboard, and you want to create 4 different templates for different types of users, you could set k =5. Then you can view the average usage patterns within each cluster, and perhaps define them as “power users”, “beginners”, “light-weight” and “medium usage”.

CLUSTER VALIDATION

Again, clustering requires human intervention and intuition to really provide strength

Human intuition required

I. CLUSTER ANALYSIS
II. K-MEANS CLUSTERING
III. VALIDATION
IV. ADDITIONAL CONSIDERATIONS

K-MEANS ADDITIONAL CONSIDERATIONS

Q: *How do we define **distance** from a point to a centroid?*
A: *We've so far used the **euclidean** distance (squared euclidean distance to be exact). This is the typical metric, and is essentially the cluster variance. This is one of many **similarity metrics** that we can use to define closeness.*

$$\mathrm{d(x,c)} = ||x - c||^2 = \sum_i (x_i - c_i)^2$$

K-MEANS ADDITIONAL CONSIDERATIONS

Q: *How do we define **distance** from a point to a centroid?*
A: *There is also a variant that uses **binary** features, called the **Jaccard coefficient**. It is a metric useful for problems with sparse binary data, such as text mining, basket analysis, etc.*

$$J(\mathrm{x},\mathrm{c}) = \frac{\left| x \cap c \right|}{\left| x \cup c \right|}$$

K-MEANS ADDITIONAL CONSIDERATIONS

Customer X has bought items A, B, C, D.
Customer Y has bought items B, B, C, E, F (bought B twice)
Their intersection of purchases is B, C
Their union of purchases is A, B, C, D, E, F

$$J(\mathrm{x},\mathrm{y}) = \frac{|x \cap y|}{|x \cup y|} = \frac{2}{6}$$

K-MEANS ADDITIONAL CONSIDERATIONS

Q: *How do we define **distance** from a point to a centroid?*
A: *There's also manhattan distance, which is the sum of absolute values along each dimension from the point to the centroid. This is essentially an L1 variant of k means (minimize the within cluster median, rather than within cluster mean)*

$$\mathrm{d(x,c)} = \left|\left|x - c\right|\right|_1 = \sum_i \left|x_i - c_i\right|$$

K-MEANS VARIATIONS

- Select Different points
    - Random within space
    - Random within points
    - KMeans++ (Choose initial centroids far from each other)
- K Medians
- K Medioids

THAT'S IT!

- Exit Tickets: DAT1 - Lesson 9 - K Means Clustering
- Homework 5 is due Jan 20
- Milestone 2 is due Jan 25
- You'll have partial opportunity to work on it during class Jan 20th (as well as review)

- For more info on kmeans read up on http://www-users.cs.umn.edu/%7Ekumar/dmbook/ch8.pdf
- Check out more examples on the clustering user guide on sklearn website