# Clusters

## Finding **Similarities**

**Poul Petersen**
**CIO, BigML, Inc**

# What is Clustering?

- An **unsupervised** learning technique
  - No labels necessary
  - Useful for finding similar instances
  - Smart sampling/labelling
- Finds “self-similar" groups of instances
  - Customer: groups with similar behavior
  - Medical: patients with similar diagnostic measurements
- Defines each group by a “**centroid**”
  - Geometric center of the group
  - Represents the “average” member
  - Number of centroids (**k**) can be specified or determined

# Cluster Centroids

| date | customer | account | auth | class | zip | amount |
|---|---|---|---|---|---|---|
| Mon | Bob | 3421 | pin | clothes | 46140 | 135 |
| Tue | Bob | 3421 | sign | food | 46140 | 401 |
| Tue | Alice | 2456 | pin | food | 12222 | 234 |
| Wed | Sally | 6788 | pin | gas | 26339 | 94 |
| Wed | Bob | 3421 | pin | tech | 21350 | 2459 |
| Wed | Bob | 3421 | pin | gas | 46140 | 83 |
| Thr | Sally | 6788 | sign | food | 26339 | 51 |

# Cluster Centroids

| date | customer | account | auth | class | zip | amount |
|---|---|---|---|---|---|---|
| Mon | Bob | 3421 | pin | clothes | 46140 | 135 |
| Tue | Bob | 3421 | sign | food | 46140 | 401 |
| Tue | Alice | 2456 | pin | food | 12222 | 234 |
| Wed | Sally | 6788 | pin | gas | 26339 | 94 |
| Wed | Bob | 3421 | pin | tech | 21350 | 2459 |
| Wed | Bob | 3421 | pin | gas | 46140 | 83 |
| Thr | Sally | 6788 | sign | food | 26339 | 51 |

similar

**Same:**

auth = pin
amount ~ $100

**Different:**

date: Mon != Wed
customer: Sally != Bob
account: 6788 != 3421
class: clothes != gas
zip: 26339 != 46140

**Centroid:**

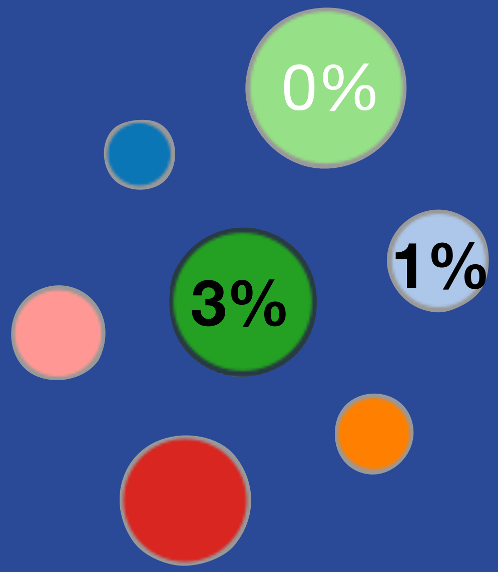
date = Wed (2 out of 3)
customer = Bob
account = 3421
auth = pin
class = gas
zip = 46140
amount = $104

# Use Cases

- Customer segmentation
  - Which customers are similar?
  - How many natural groups are there?

# Customer Segmentation


- Dataset of mobile game users.
- Data for each user consists of usage statistics and a LTV based on in-game purchases
- Assumption: Usage correlates to LTV




***GOAL: Cluster the users by usage statistics. Identify clusters with a higher percentage of high LTV users. Since they have similar usage patterns, the remaining users in these clusters may be good candidates for up-sell.***

# Use Cases

- Customer segmentation
  - Which customers are similar?
  - How many natural groups are there?
- Item discovery
  - What other items are similar to this one?

# Item Discovery


- Dataset of 86 whiskies
- Each whiskey scored on a scale from 0 to 4 for each of 12 possible flavor characteristics.

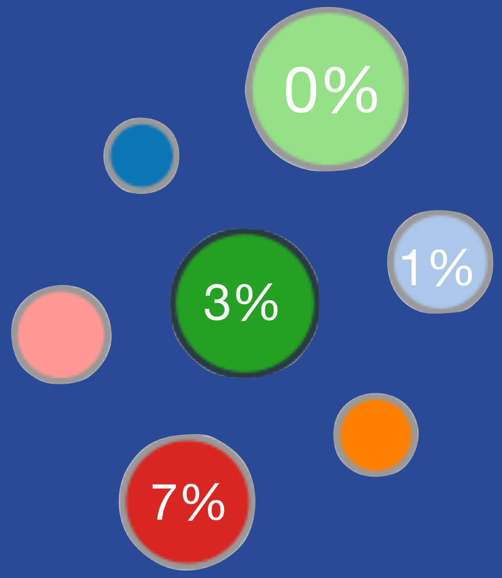

***GOAL: Cluster the whiskies by flavor profile to discover whiskies that have similar taste.***

# Use Cases

- Customer segmentation
  - Which customers are similar?
  - How many natural groups are there?
- Item discovery
  - What other items are similar to this one?
- Similarity
  - What other instances share a specific property?

# Similarity


- Dataset of Lending Club Loans
- Mark any loan that is currently or has even been late as “trouble”

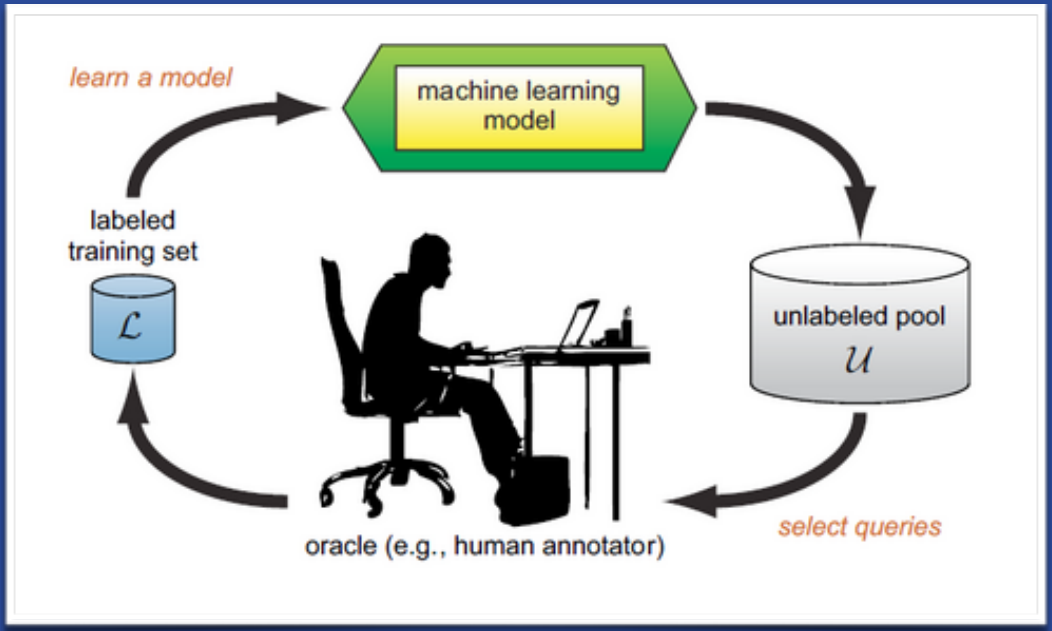



***GOAL:* Cluster the loans by application profile to rank loan quality by percentage of trouble loans in population**

# Use Cases

- Customer segmentation
  - Which customers are similar?
  - How many natural groups are there?
- Item discovery
  - What other items are similar to this one?
- Similarity
  - What other instances share a specific property?
- Recommender (almost)
  - If you like this item, what other items might you like?
- Active learning
  - Labelling unlabelled data efficiently

# Active Learning




- Dataset of diagnostic measurements of 768 patients.
- Want to test each patient for diabetes and label the dataset to build a model but the test is expensive*.

*GOAL:*

***Rather than sample randomly, use clustering to group patients by similarity and then test a sample from each cluster to label the data.***

# Active Learning

*For a more realistic example of high cost, imagine a dataset with a billion transactions, each one needing to be labelled as fraud/not-fraud. Or a million images which need to be labeled as cat/not-cat.

# Clusters Demo #1

# Human Expert

# Human Expert

# Human Expert

- Jesa used prior knowledge to select possible features that separated the objects.
- “round”, “skinny”, “edges”, “hard”, etc
- Items were then clustered based on the chosen features
- Separation quality was then tested to ensure:
- met criteria of K=3
- groups were sufficiently “distant”
- no crossover

# Human Expert

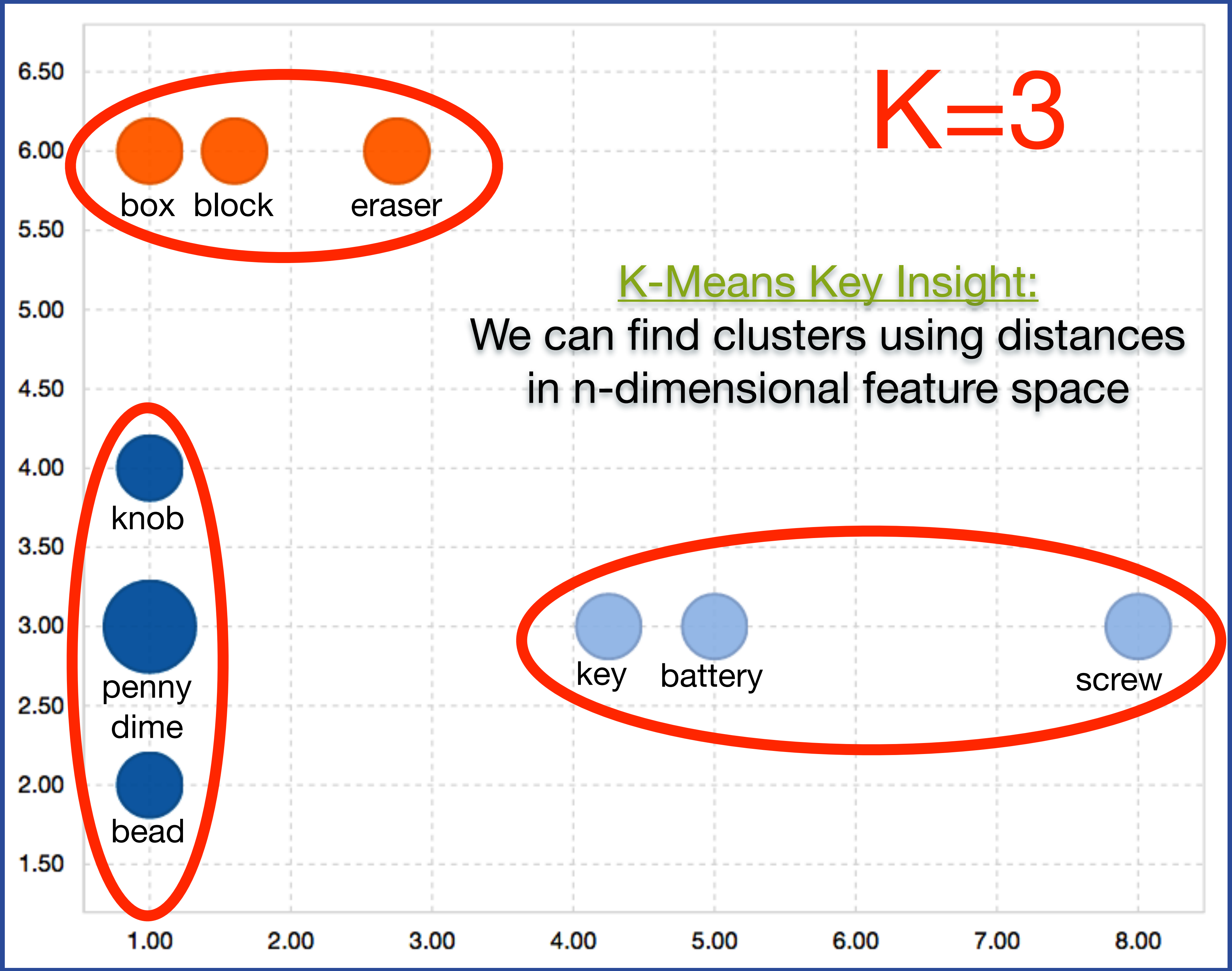
Create **features** that capture these object differences

- **Length/Width**
  - greater than 1 => “skinny”
  - equal to 1 => “round”
  - less than 1 => invert
- **Number of Surfaces**
  - distinct surfaces require “edges” which have corners
  - easier to count

# Clustering Features

| Object | Length / Width | Num Surfaces |
|---|---|---|
| penny | 1 | 3 |
| dime | 1 | 3 |
| knob | 1 | 4 |
| eraser | 2.75 | 6 |
| box | 1 | 6 |
| block | 1.6 | 6 |
| screw | 8 | 3 |
| battery | 5 | 3 |
| key | 4.25 | 3 |
| bead | 1 | 2 |

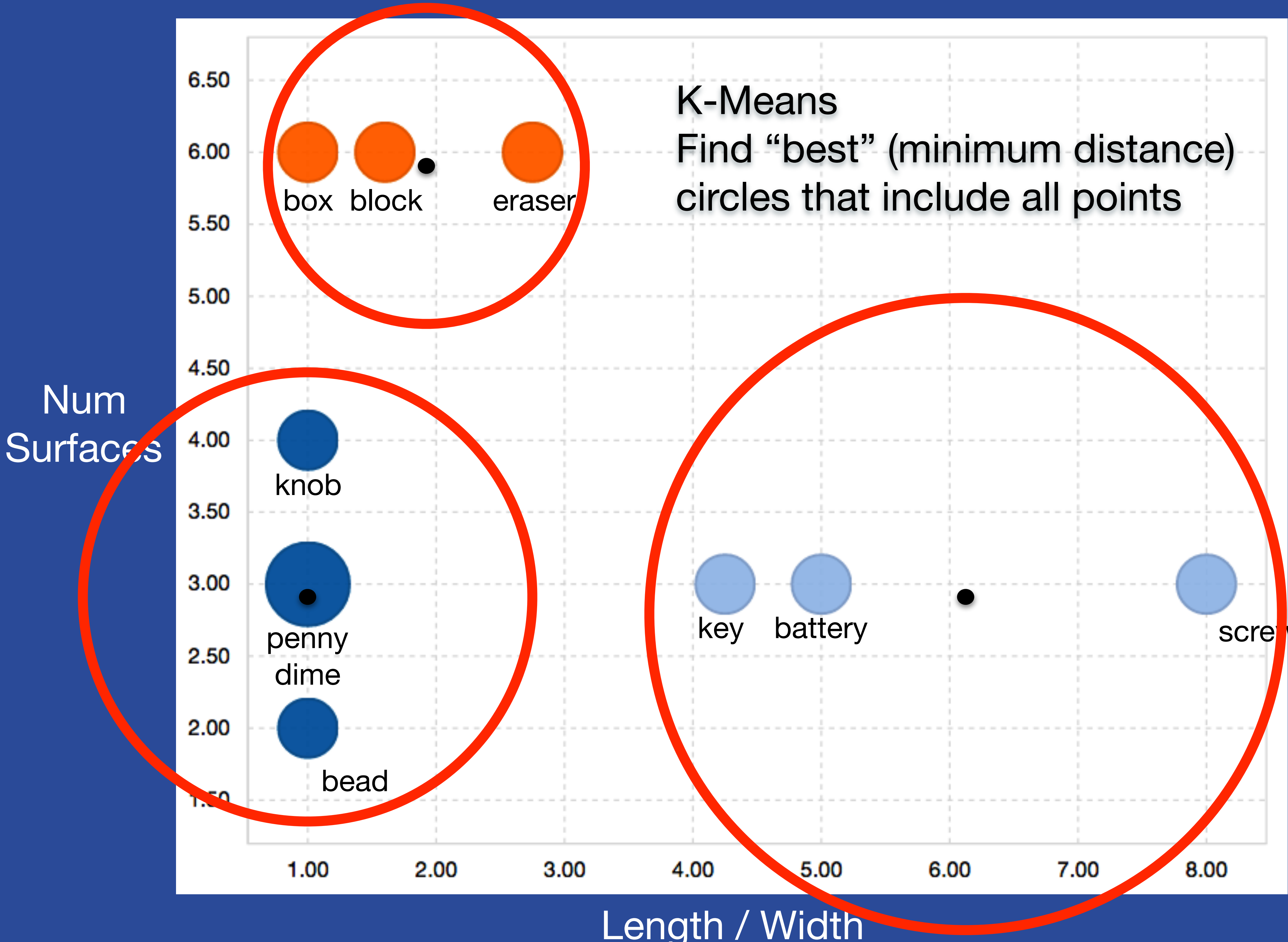
# Plot by Features



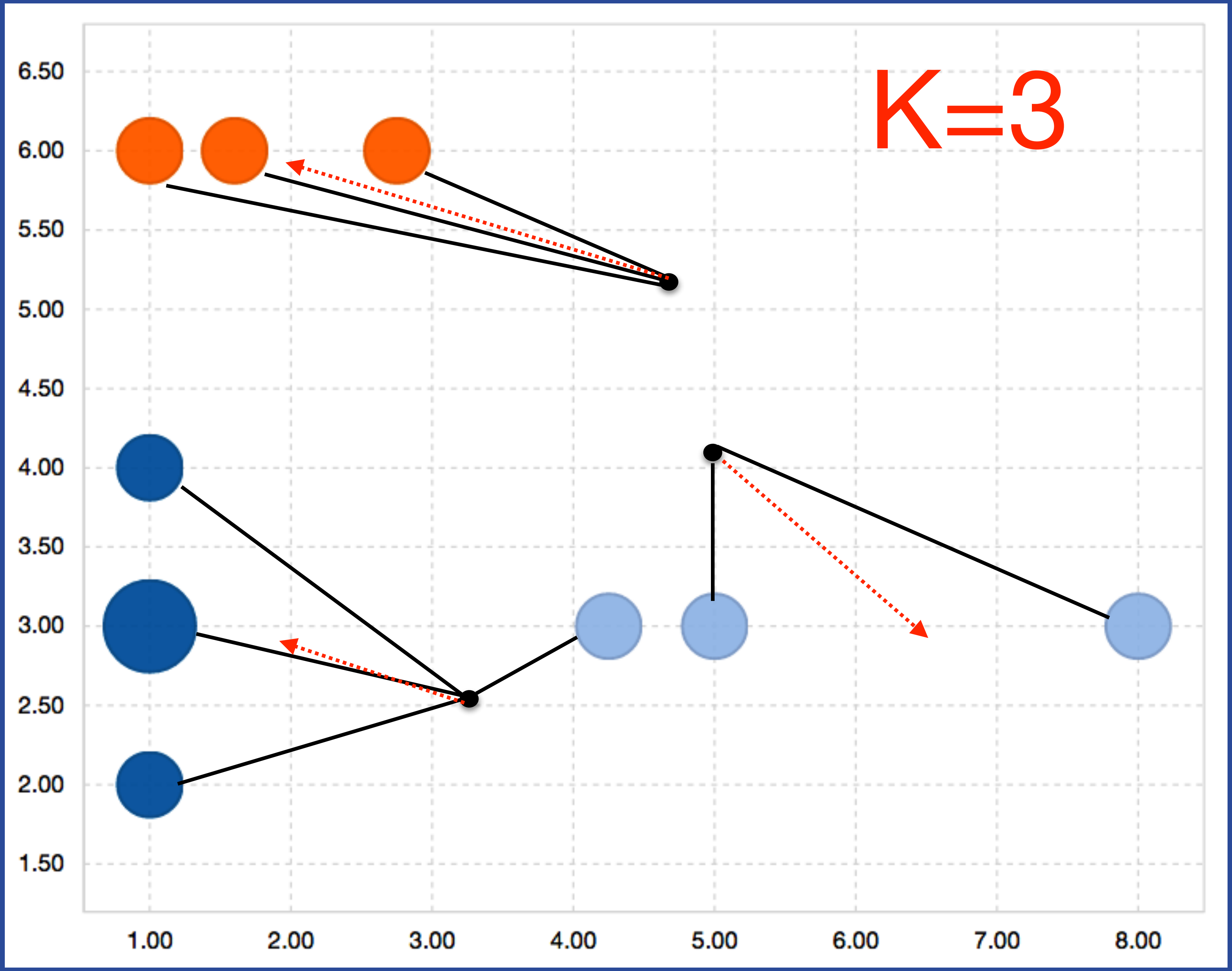

Num Surfaces

Length / Width

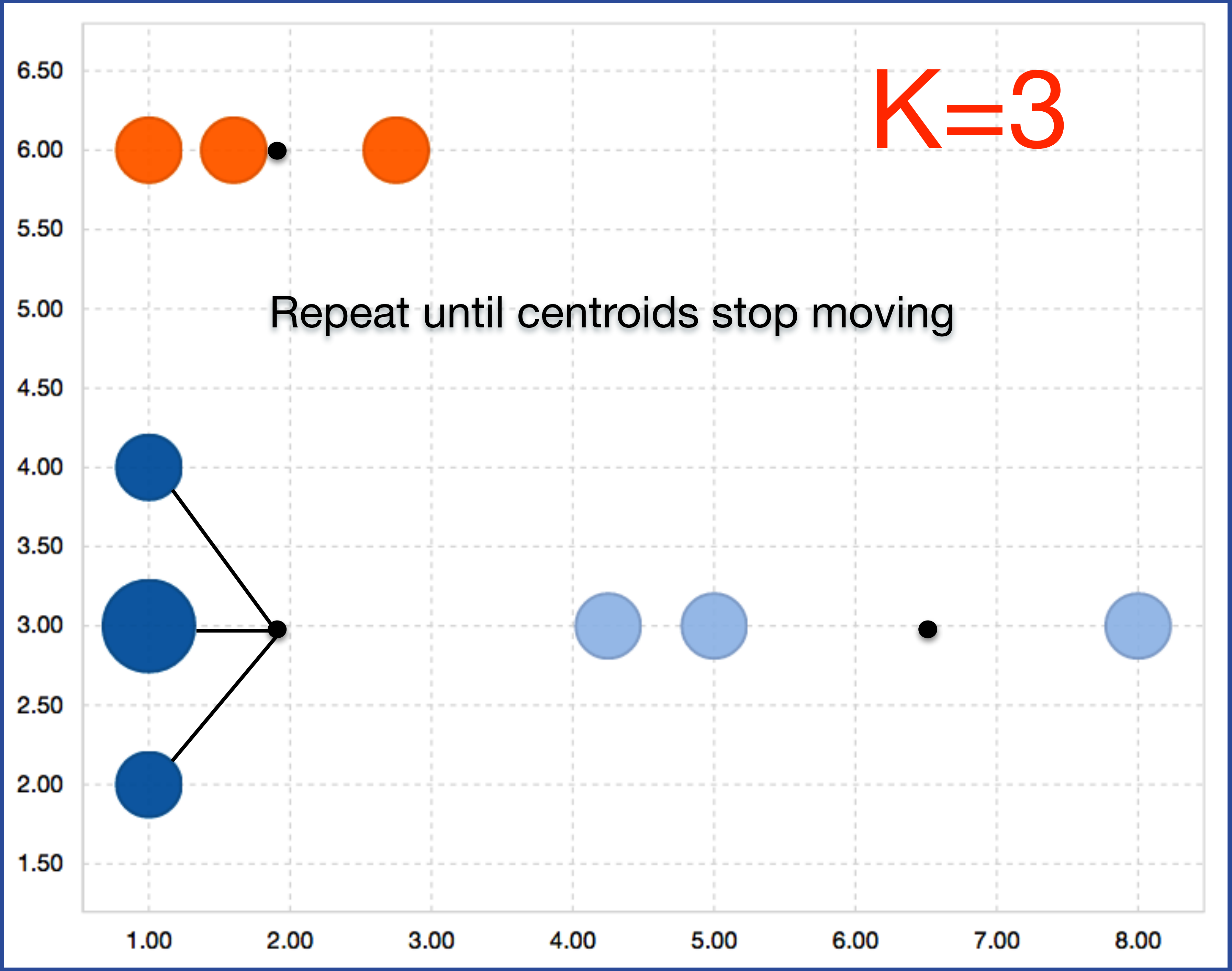
# Plot by Features



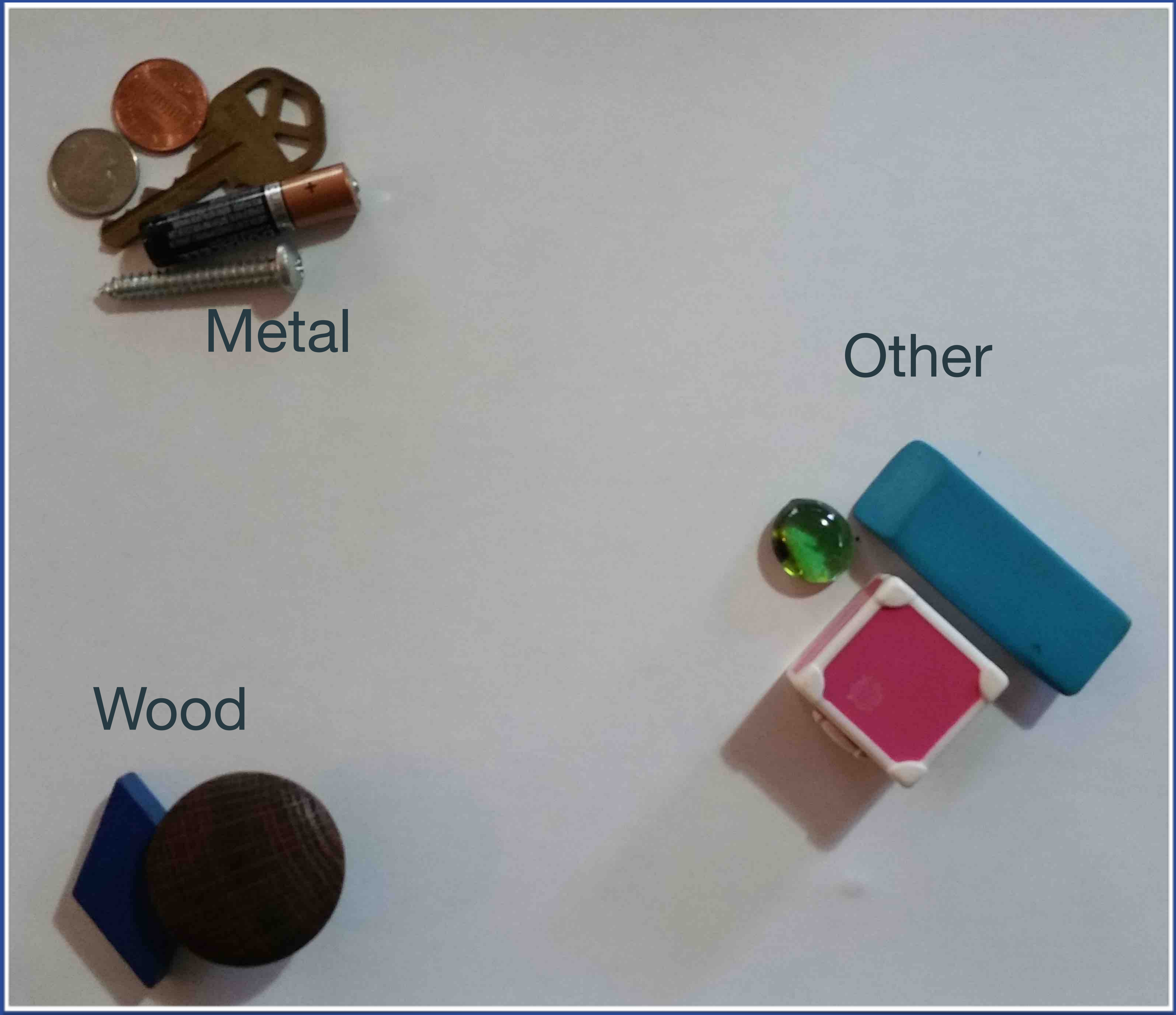
# K-Means Algorithm



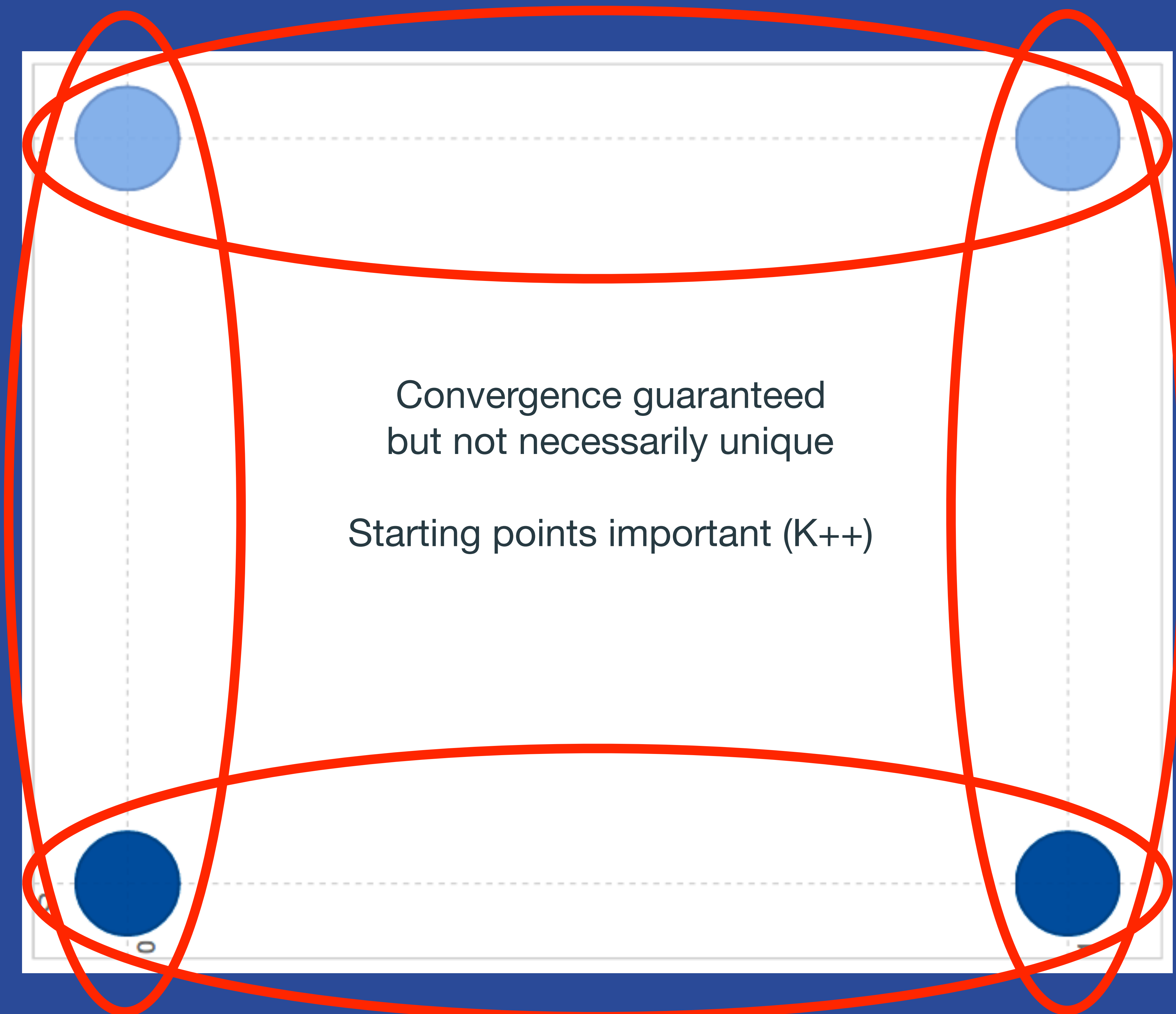
# K-Means Algorithm



# Features Matter



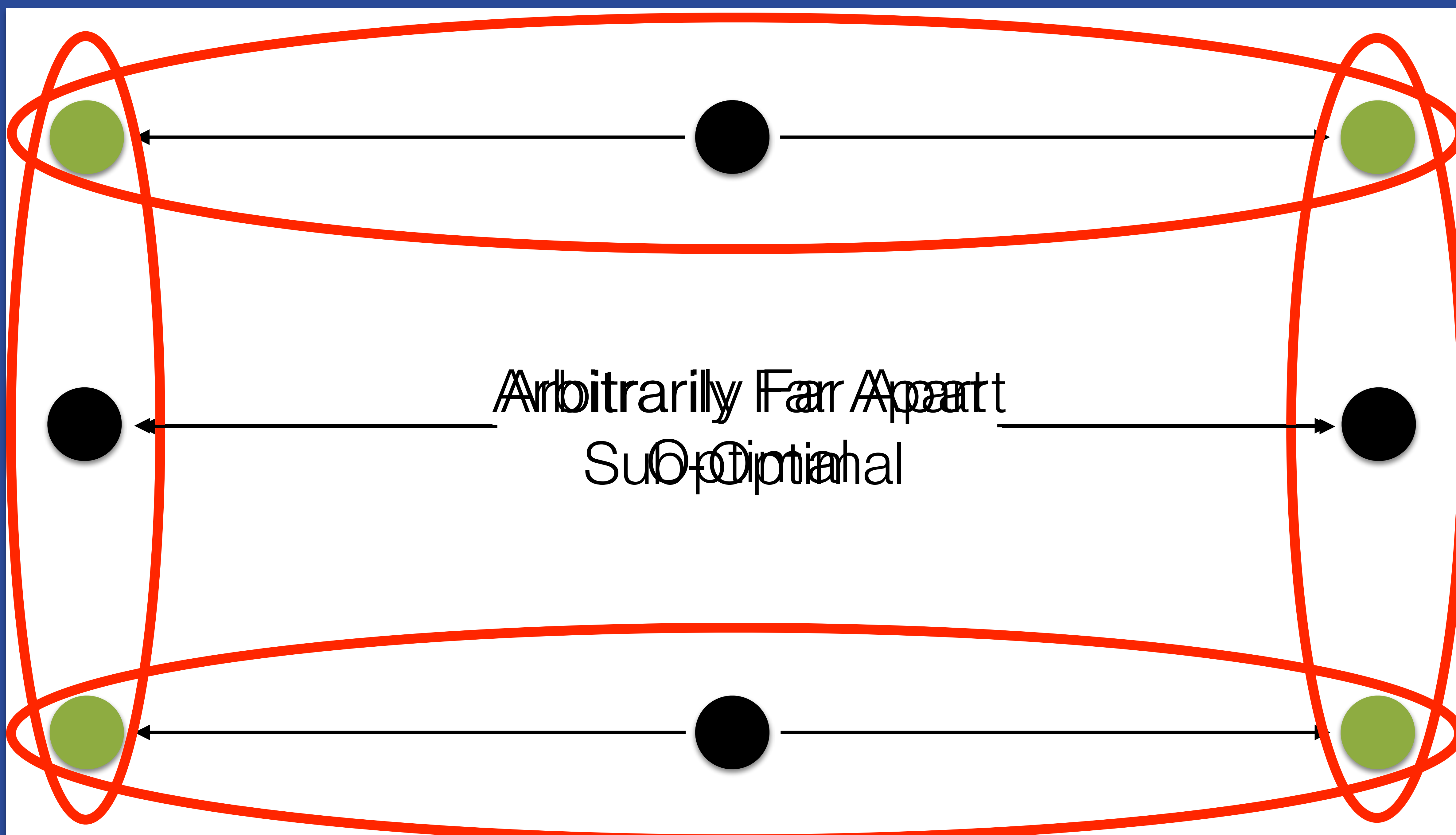
# Convergence



# Starting Points

- Random points or instances in n-dimensional space
    - Might start "too close"
    - Risk of sub-optimal convergence

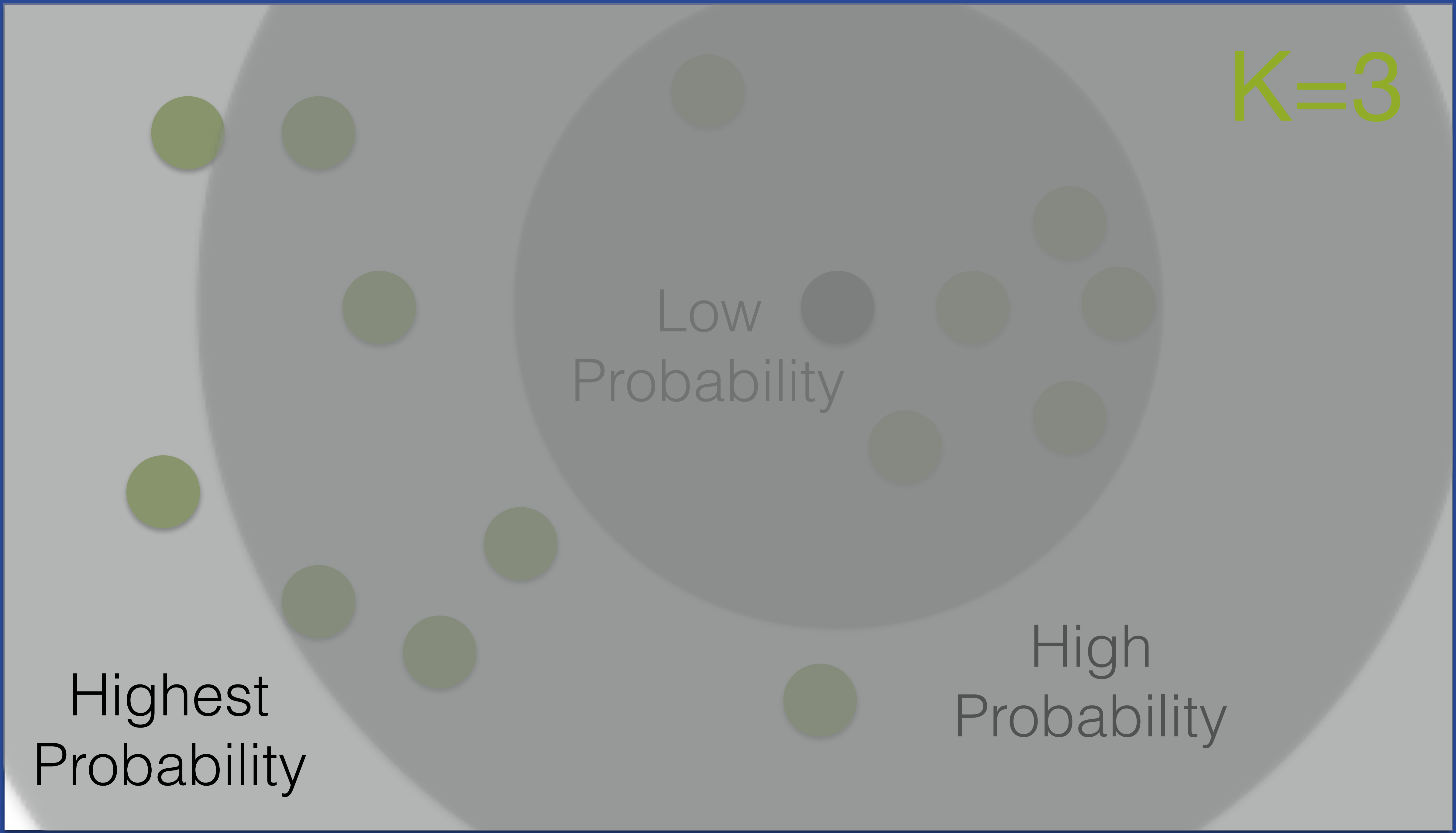
# Sub-Optimal Converge



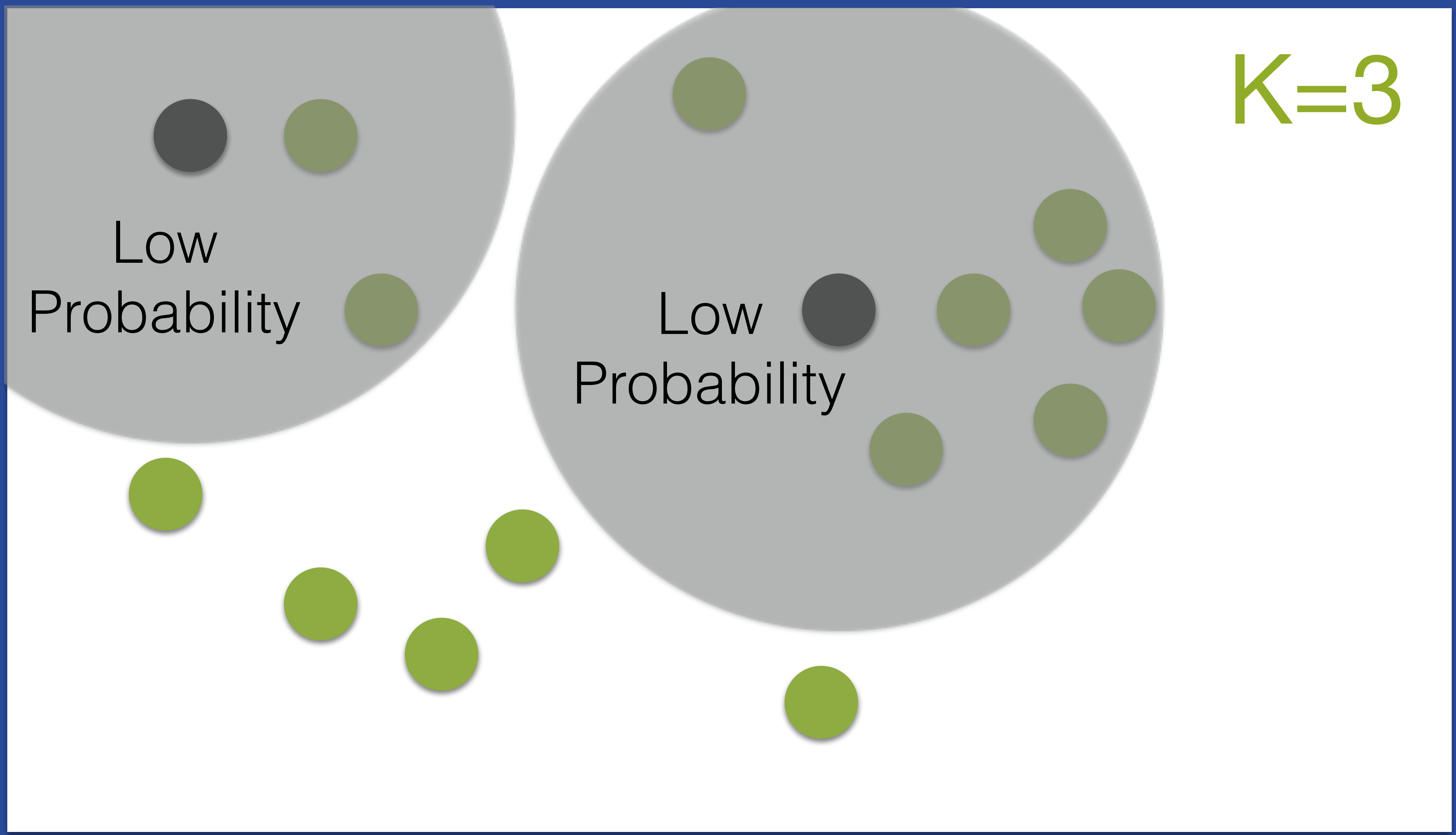
# Starting Points

- Random points or instances in n-dimensional space
    - Might start "too close"
    - Risk of sub-optimal convergence
- Chose points “farthest” away from each other
    - but this is sensitive to outliers
- k++
    - the first point is chosen randomly from instances
    - each subsequent point is chosen from the remaining instances with a probability proportional to the squared distance from the point's closest existing cluster center

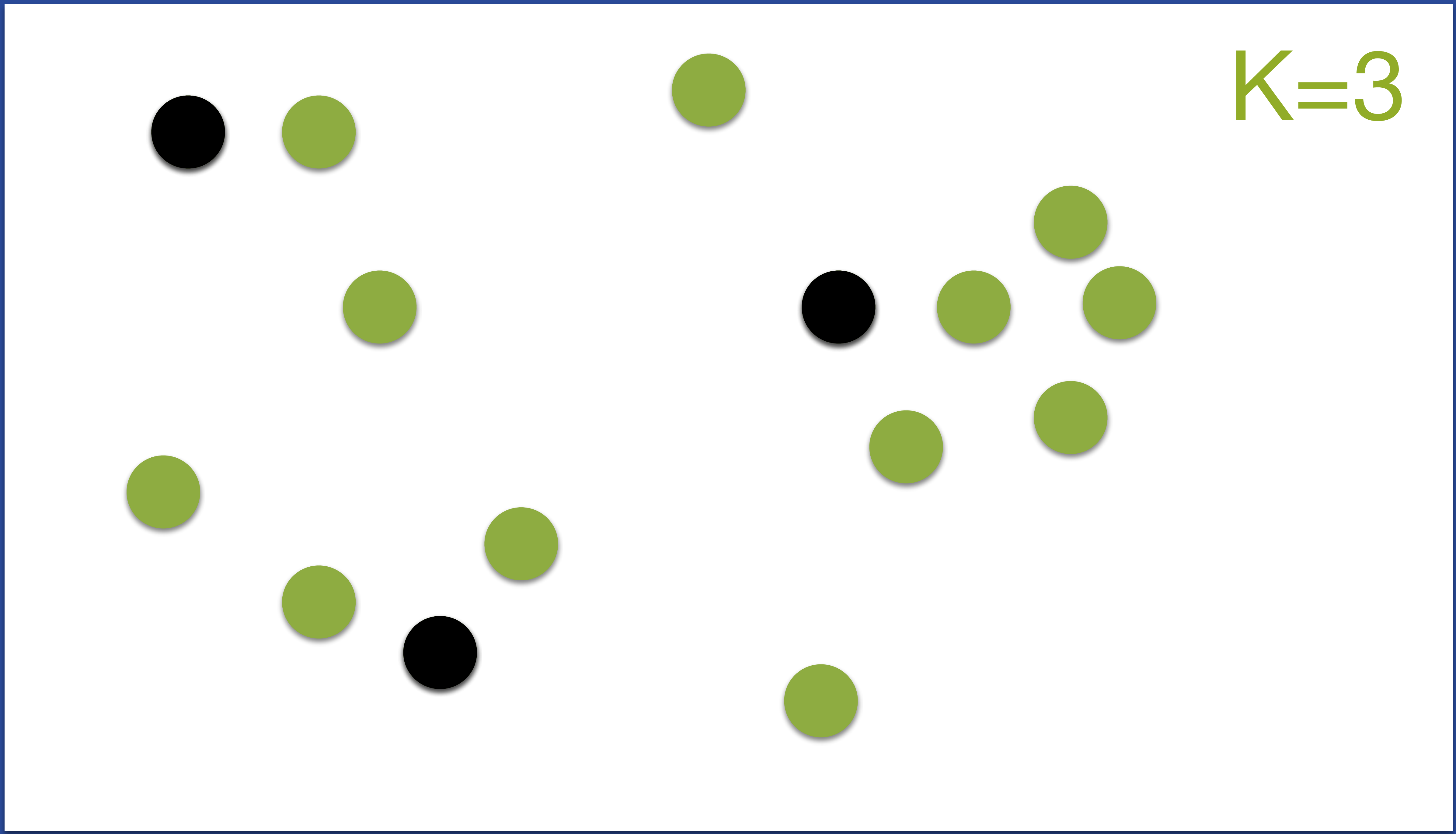
# K++ Initial Centers



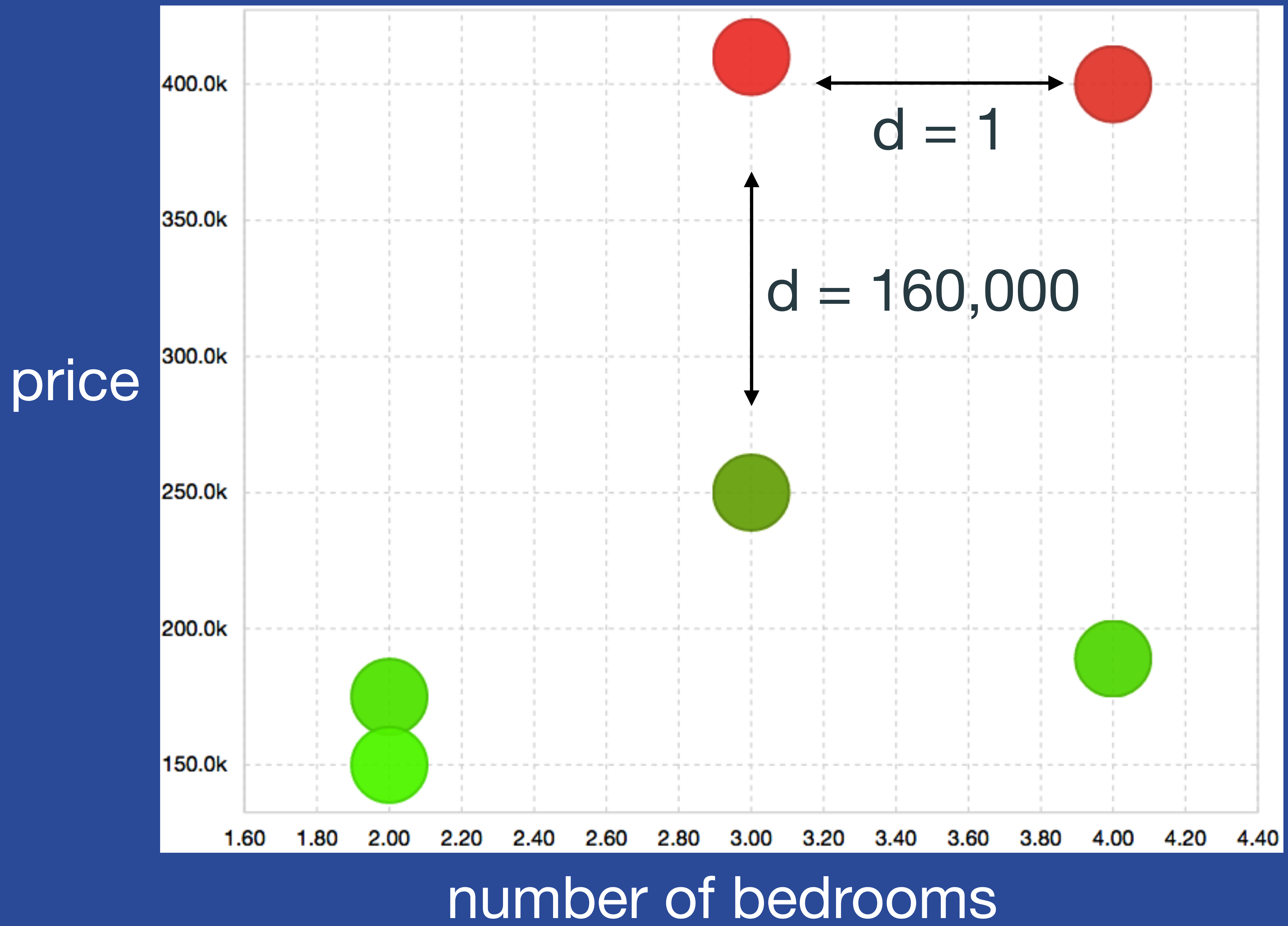
# K++ Initial Centers



# K++ Initial Centers



# Scaling Matters



# Internal Tricks

- What is the distance to a “missing value”?
- What is the distance between categorical values?
    - How far is “red” from “green”?
- What is the distance between text features?
- Does it have to be Euclidean distance?
- Unknown ideal number of clusters, “K”?

# Distance to Missing?

- Nonsense! Try replacing missing values with:
    - Maximum
    - Mean
    - Median
    - Minimum
    - Zero
- Ignore instances with missing values

# Distance to Categorical?

Then compute Euclidean distance between vectors

| animal | favorite toy | toy color | |
|---|---|---|---|
| cat | ball | red | |
| cat | ball | green | |
| d=0 | d=0 | d=1 | → **D = 1** |

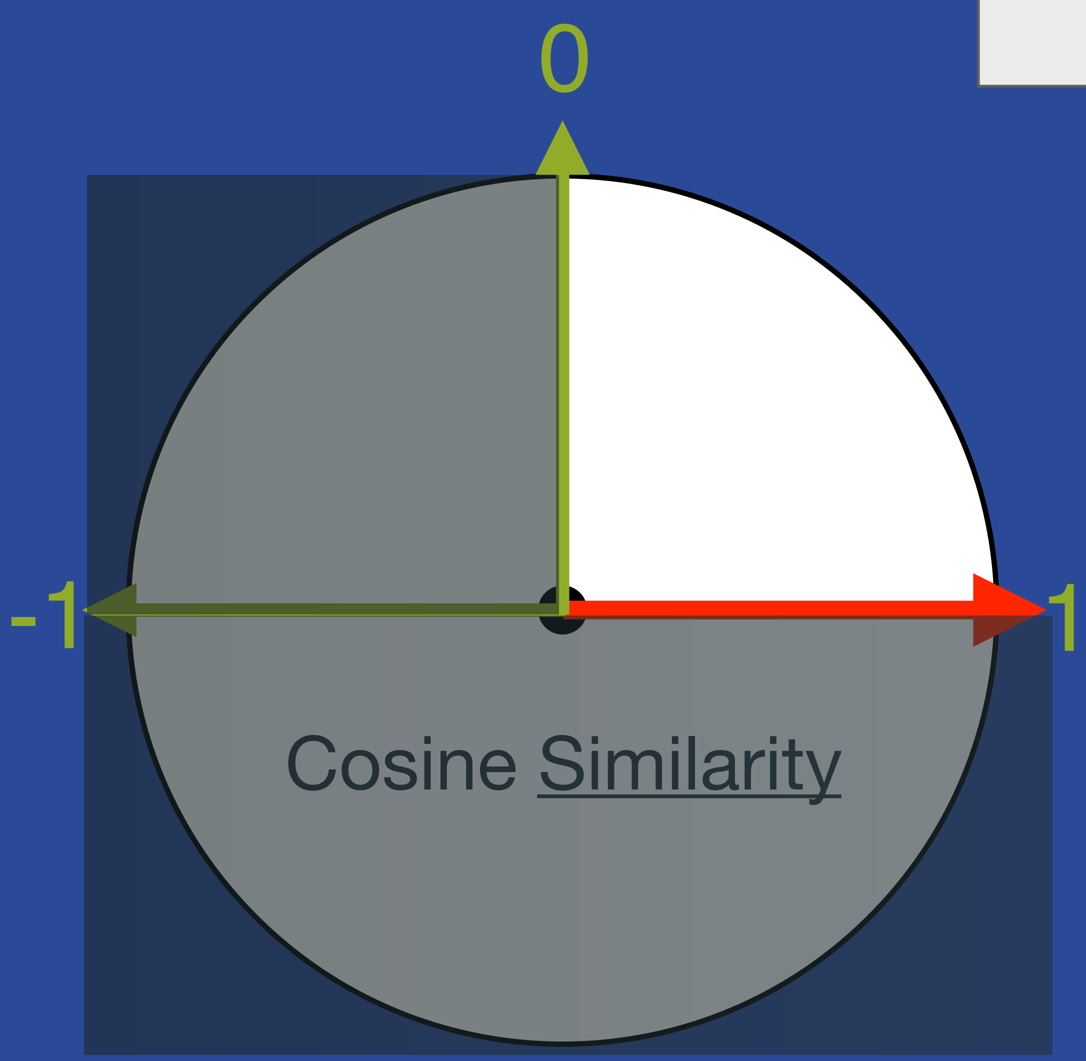
| | | | |
|---|---|---|---|
| cat | laser | red | |
| dog | squeaky | red | |
| d=1 | d=1 | d=0 | → **D = $\sqrt{2}$** |

Note: the centroid is assigned the most common category of the member instances

# Text Vectors

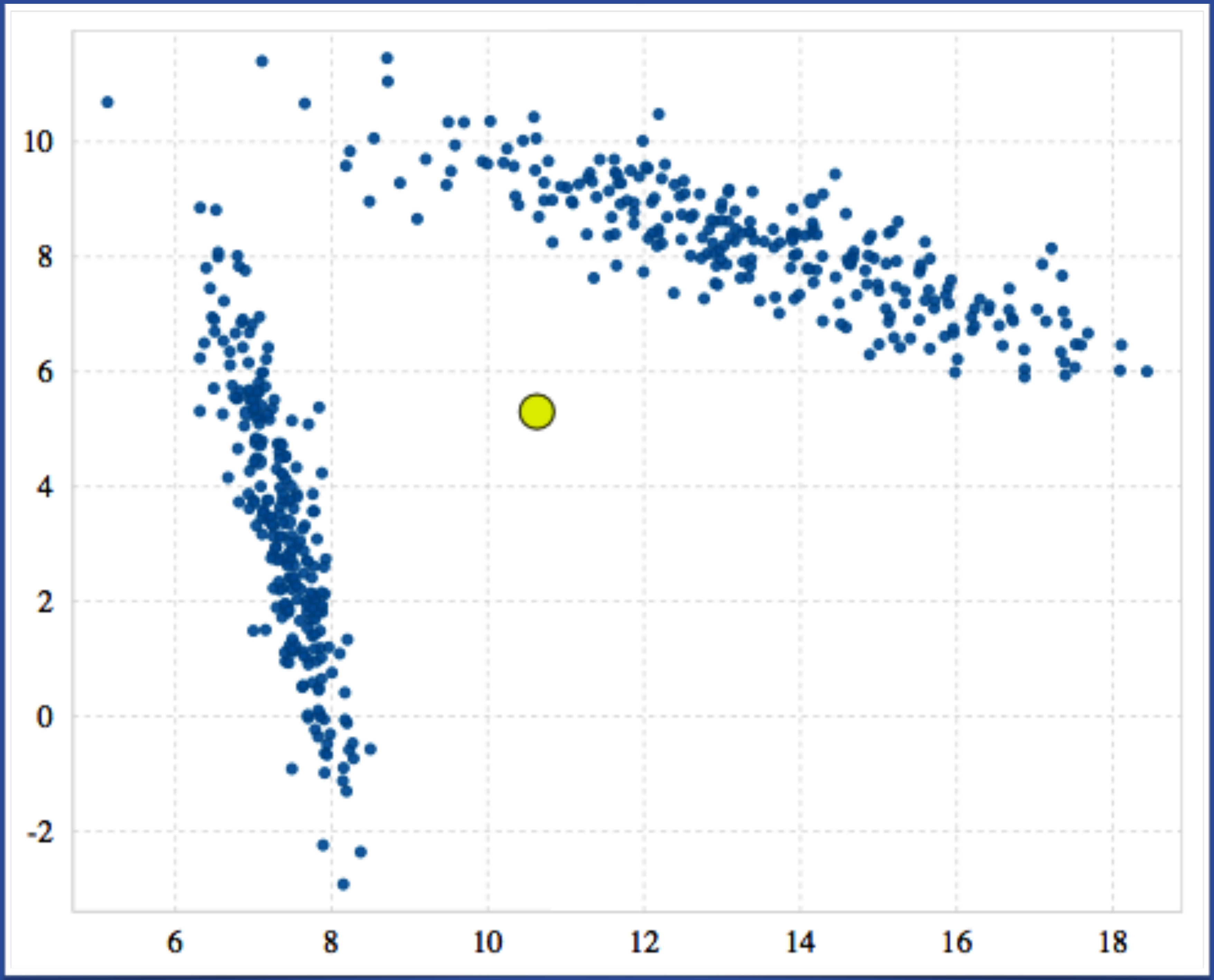
Features *(thousands)*

| | "hippo" | "safari" | "zebra" | .... |
|---|---|---|---|---|
| Text Field #1 | 1 | 0 | 1 | ... |
| Text Field #2 | 1 | 1 | 0 | ... |
| | 0 | 1 | 1 | ... |

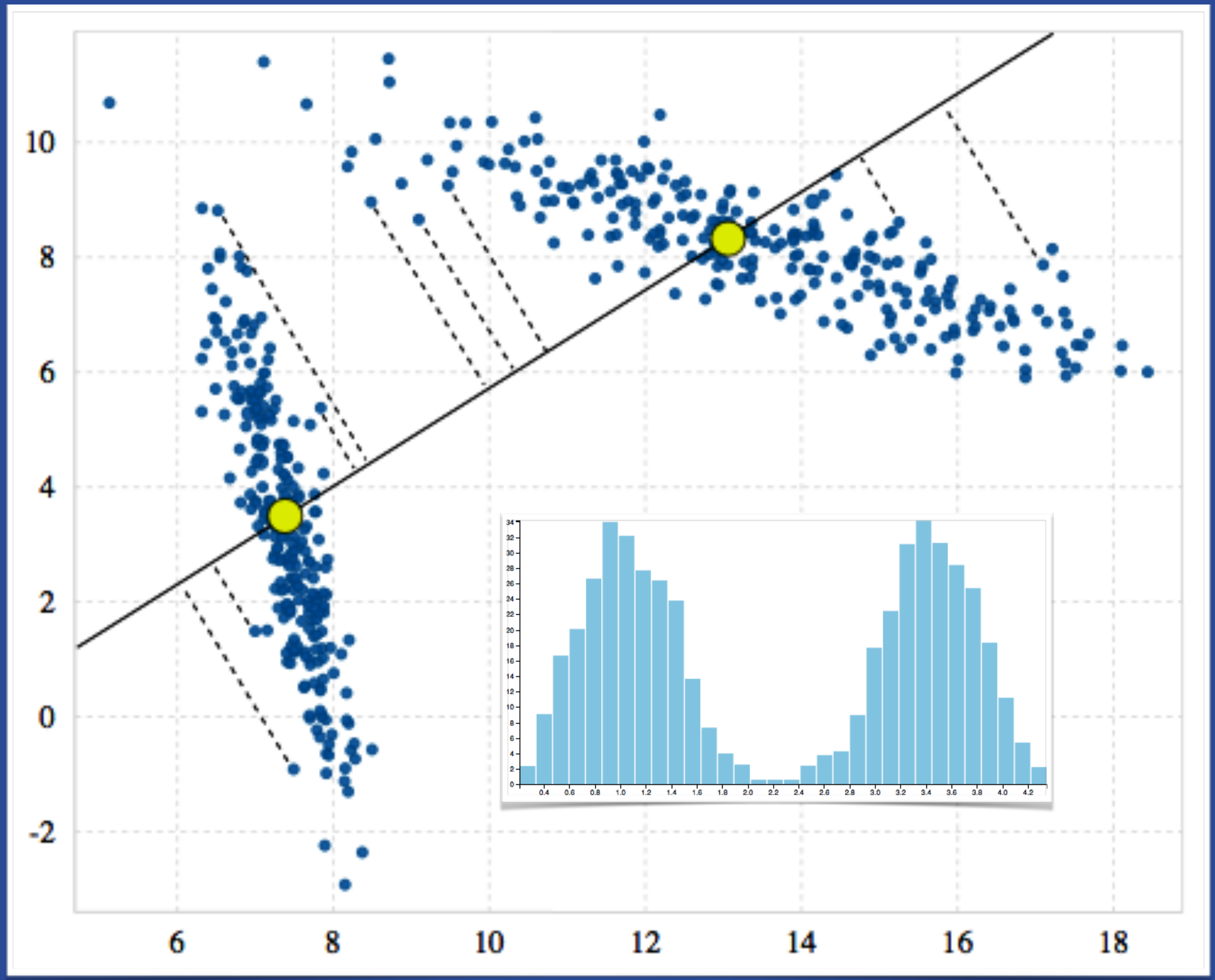



- Cosine Similarity
  - cos() between two vectors
  - 1 if collinear, 0 if orthogonal
  - only positive vectors: $0 \leq CS \leq 1$
- Cosine Distance = 1 − Cosine Similarity
- CD(TF1, TF2) = 0.5

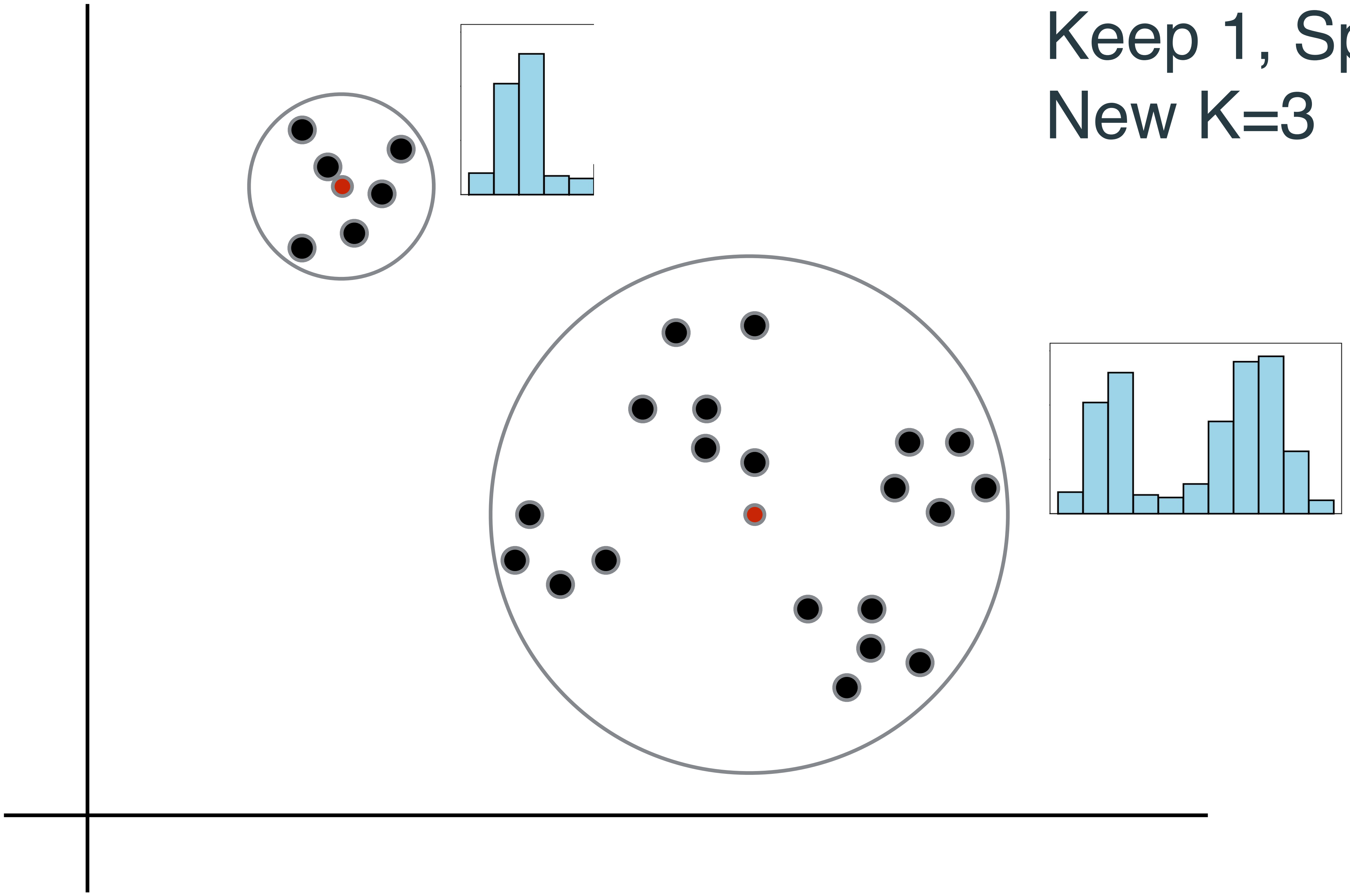
# Finding K: G-Means



# Finding K: G-Means



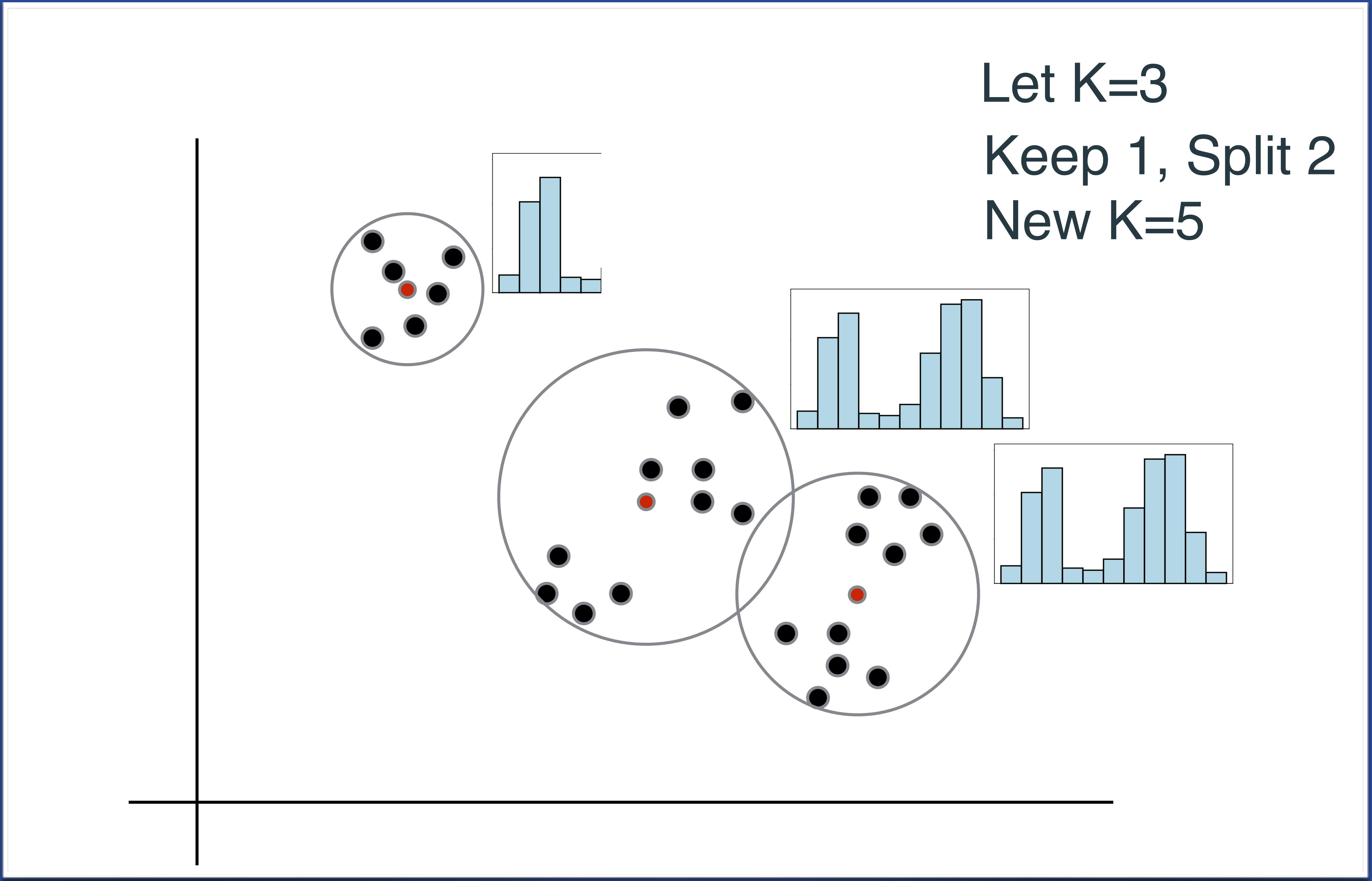
# Finding K: G-Means



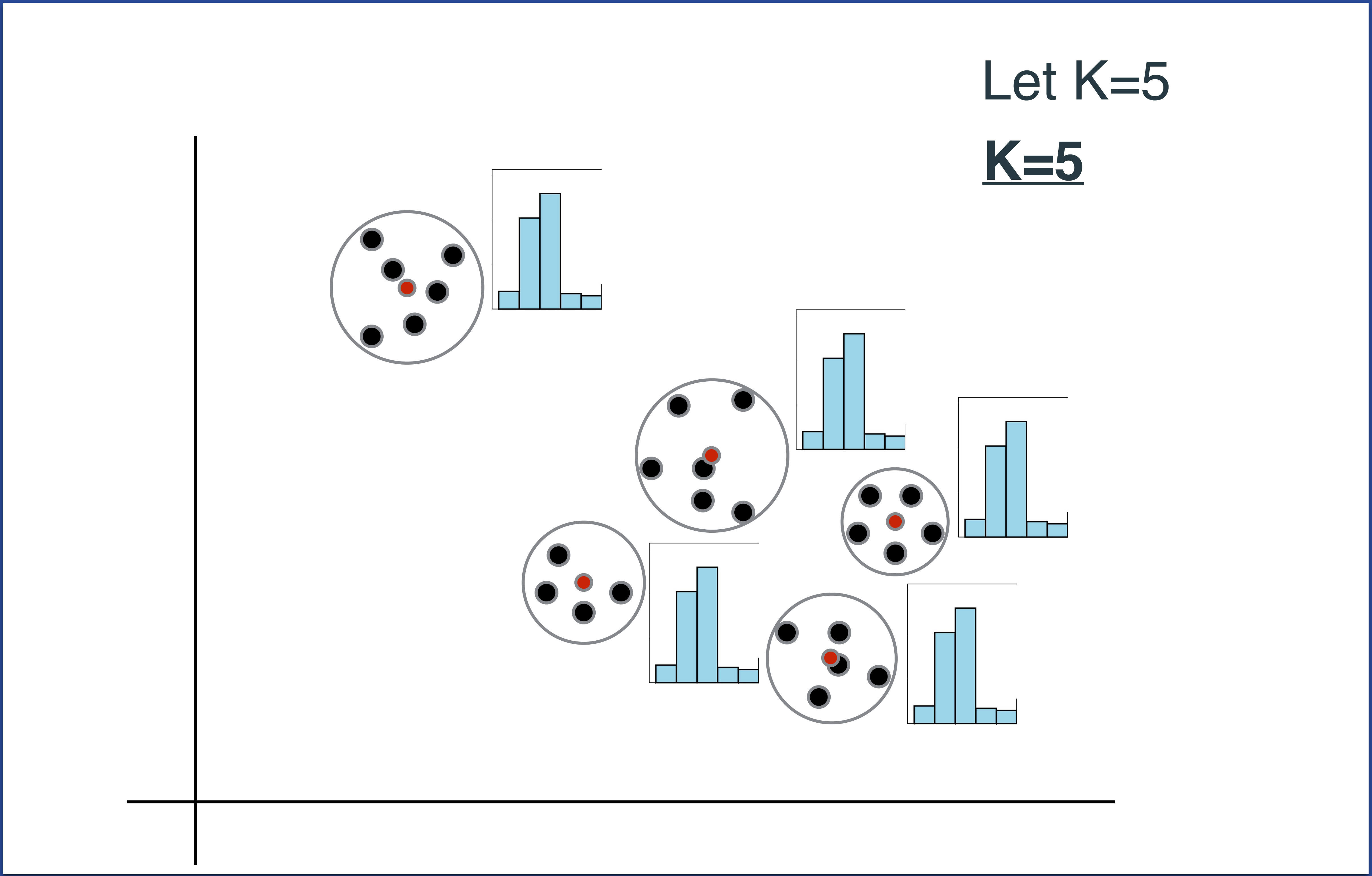
Let K=2

Keep 1, Split 1
New K=3

# Finding K: G-Means



# Finding K: G-Means



# Clusters Demo #2

# Your Turn!

- Create a 1-click (g-means) cluster for Diabetes
- Be certain to use the **entire dataset** (not the 80/20)
    - How many groups does it find?
    - How many groups have diabetes=True?
- Create a 1-click (g-means) cluster for PDX Homes
    - How many groups does it find?
    - How many total houses are in the groups and does this match the dataset?
    - How could you increase the number of groups?

bigml®