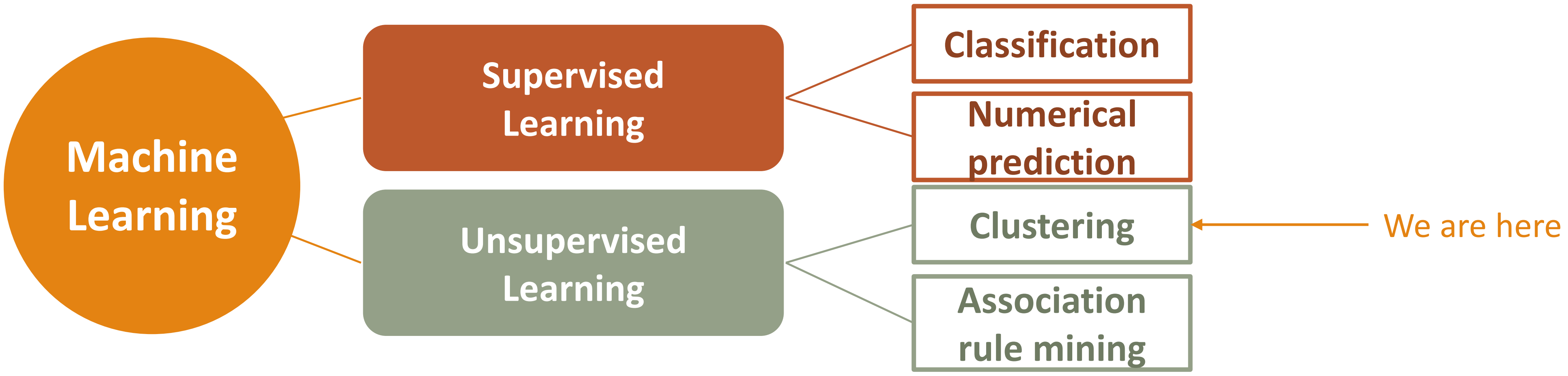
Lecture 9:Unsupervised Learning: Clustering

Outline

- Unsupervised learning
- Clustering
  - K Nearest Neighbor Algorithm and Clustering
  - Understanding Clustering
  - The k-means clustering algorithm
  - Elbow Method
  - Classification with Clustering

Data Mining Tasks



Data Mining Tasks

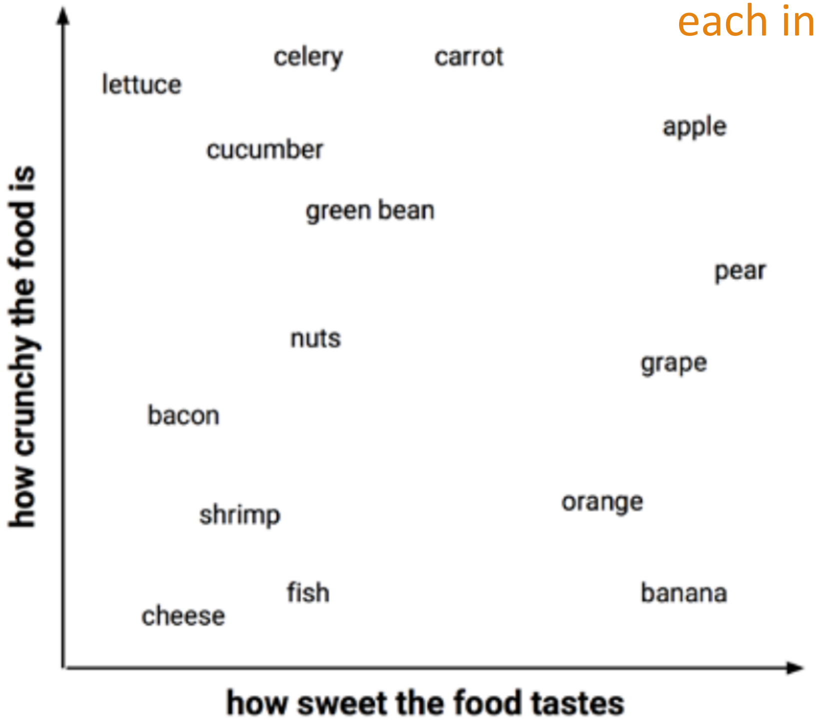
- **Supervised learning** is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.
  - Y=f(X); $X \xrightarrow{\text{predict}} Y$
  - The supervision does not refer to human involvement, but rather to the fact that the target values provide a way for the learner to know how well it has learned the desired task.
  - A **predictive model** is used for tasks that involve, as the name implies, the prediction of one value using other values in the dataset.

Data Mining Tasks

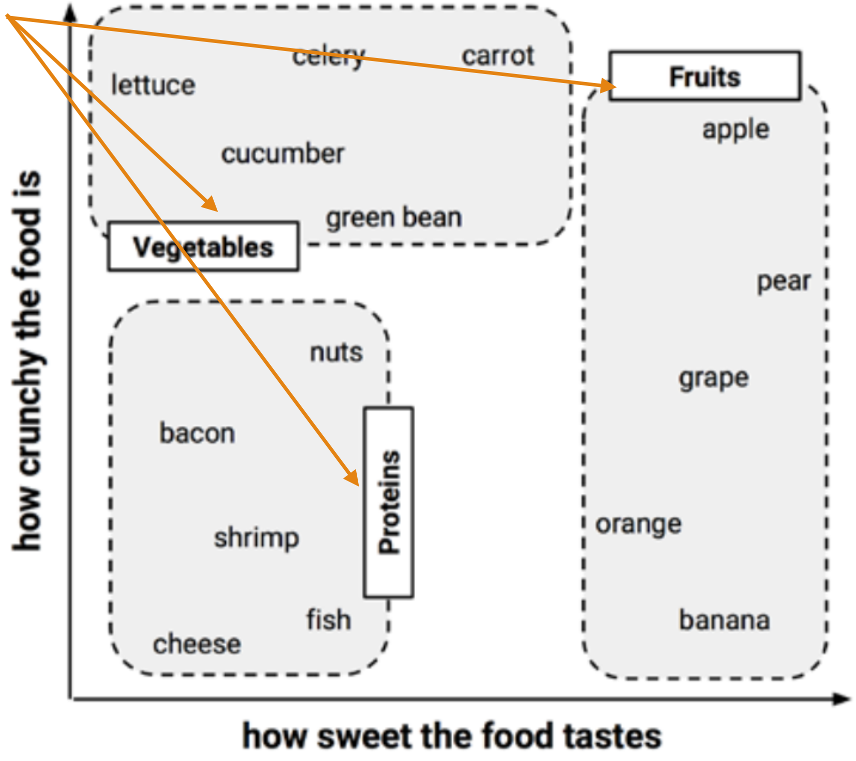
- **Unsupervised learning** is where you only have input data (X) and no corresponding output variables.
  - A **descriptive model** is used for tasks that would benefit from the insight gained from **summarizing data in new and interesting ways**.
  - There is no target to learn, the process of training a descriptive model is called unsupervised learning.

K Nearest Neighbor Algorithm and Clustering

- Similar types of food tend to be grouped closely together

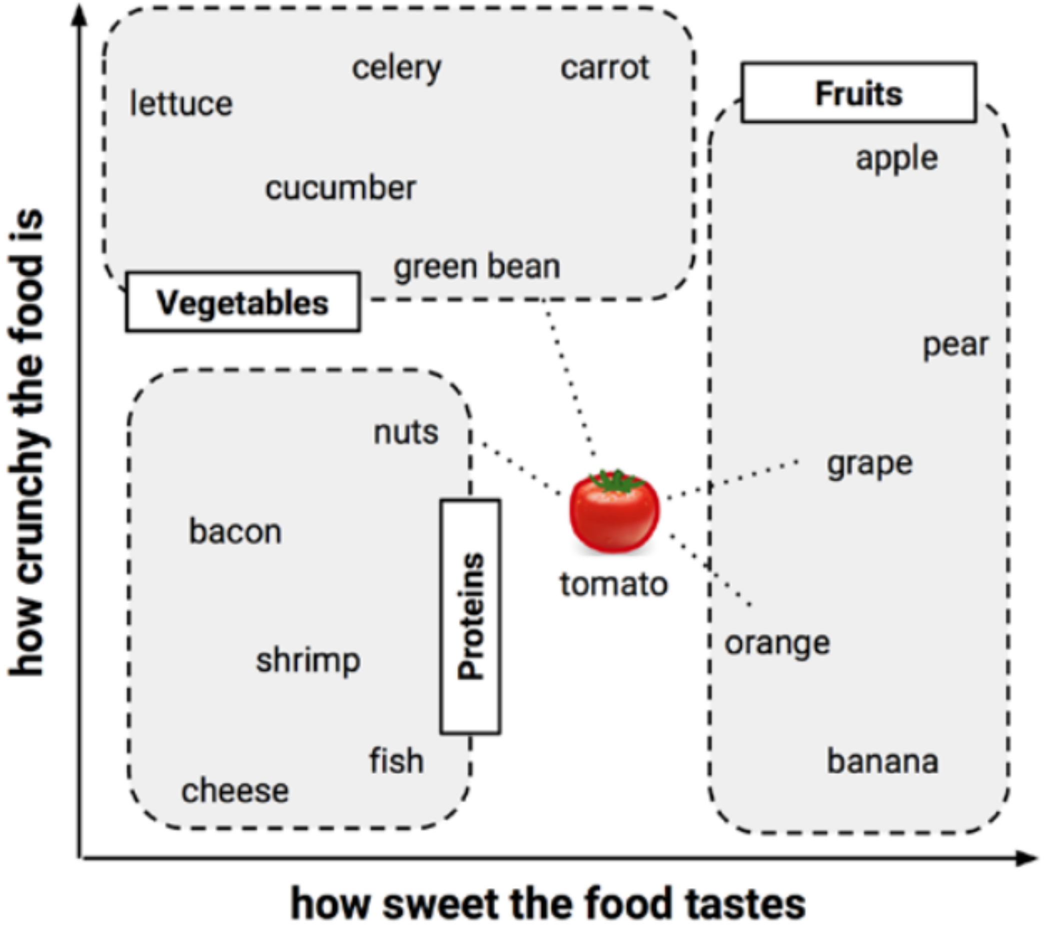


In KNN, the food type of each ingredient is given



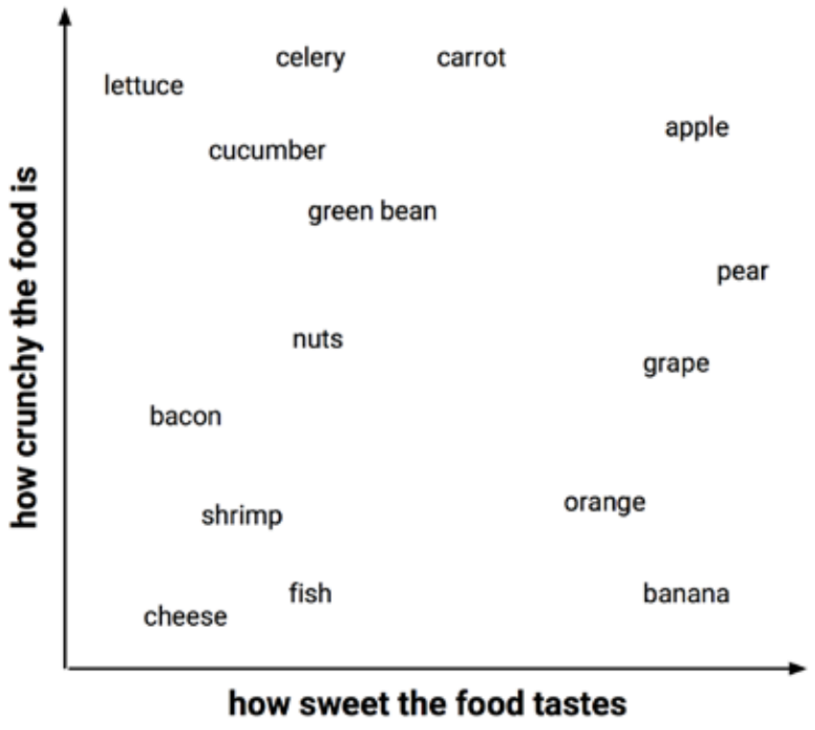
K Nearest Neighbor Algorithm and Clustering

- Suppose you are given a tomato: protein, fruit or vegetable?
- We can use the nearest neighbor approach to determine which class is a better fit.
- We learn the food type of tomato based on other ingredients (that we already their types).



K Nearest Neighbor Algorithm and Clustering

- What if we have no food type information (no labels) about all the ingredients?
  - No training and testing data
  - No prediction task
- We can group ingredients based on crunchiness and sweetness



Understanding Clustering

- **Clustering** is an unsupervised machine learning task that automatically divides the data into **clusters,** or groups of similar items.
  - Without knowing how the groups should look
  - **Knowledge discovery** rather than prediction
  - Guided by the principle that items inside a cluster should be very similar to each other, but very different from those outside

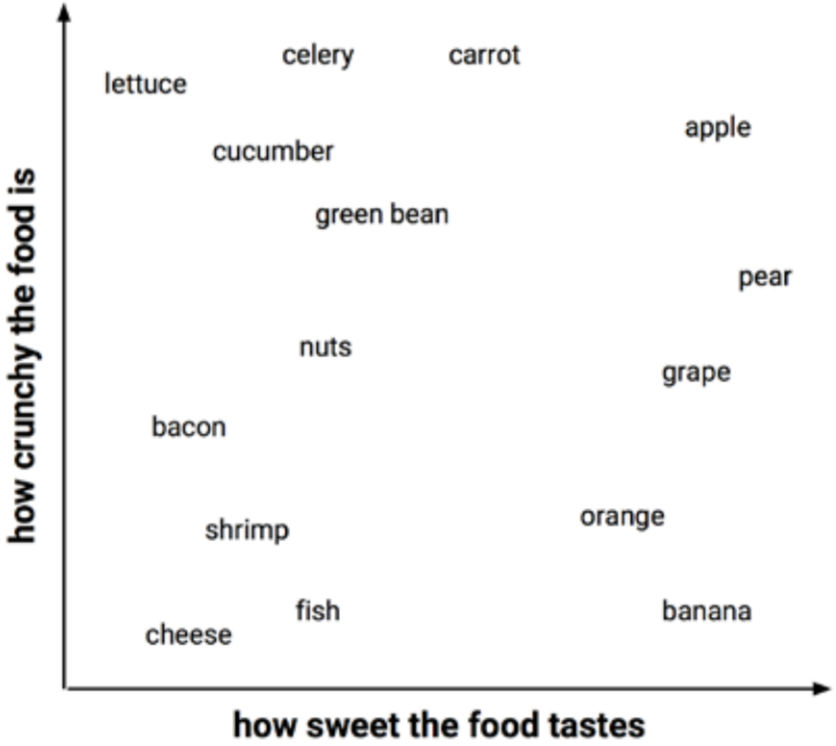
Understanding Clustering

- The goal of clustering: group the data so that the related instances are placed together.

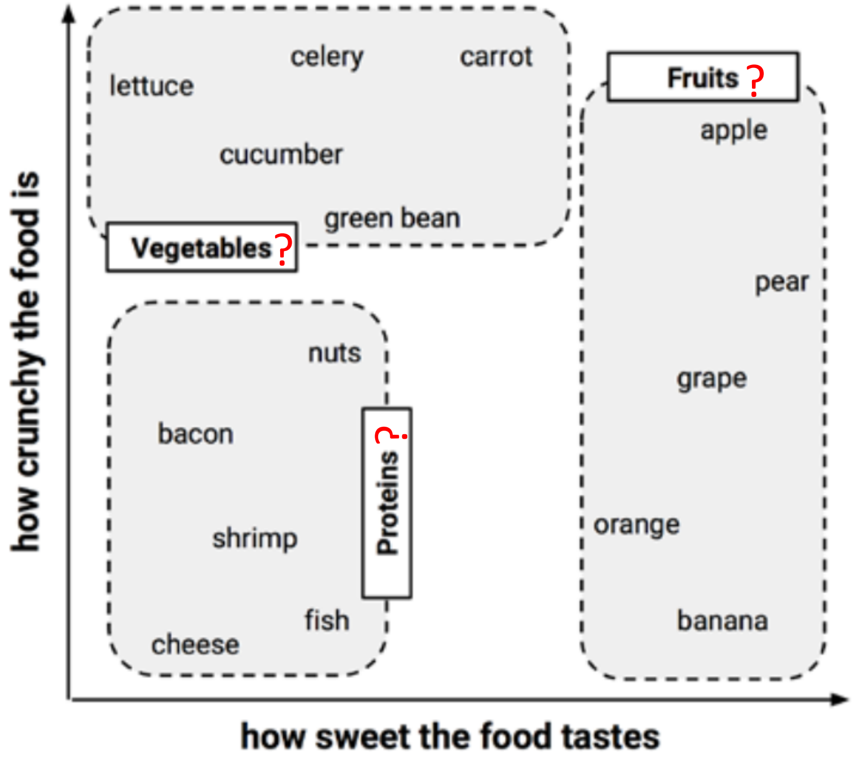
- Applications of clustering
  - Segmenting customers into groups with similar demographics or buying patterns for targeted marketing campaigns
  - Detecting anomalous behavior, such as unauthorized network intrusions, by identifying patterns of use falling outside the known clusters

Understanding Clustering

- For each ingredient we know the crunchiness and sweetness.



Identify clusters as closely grouped data points



Understanding Clustering

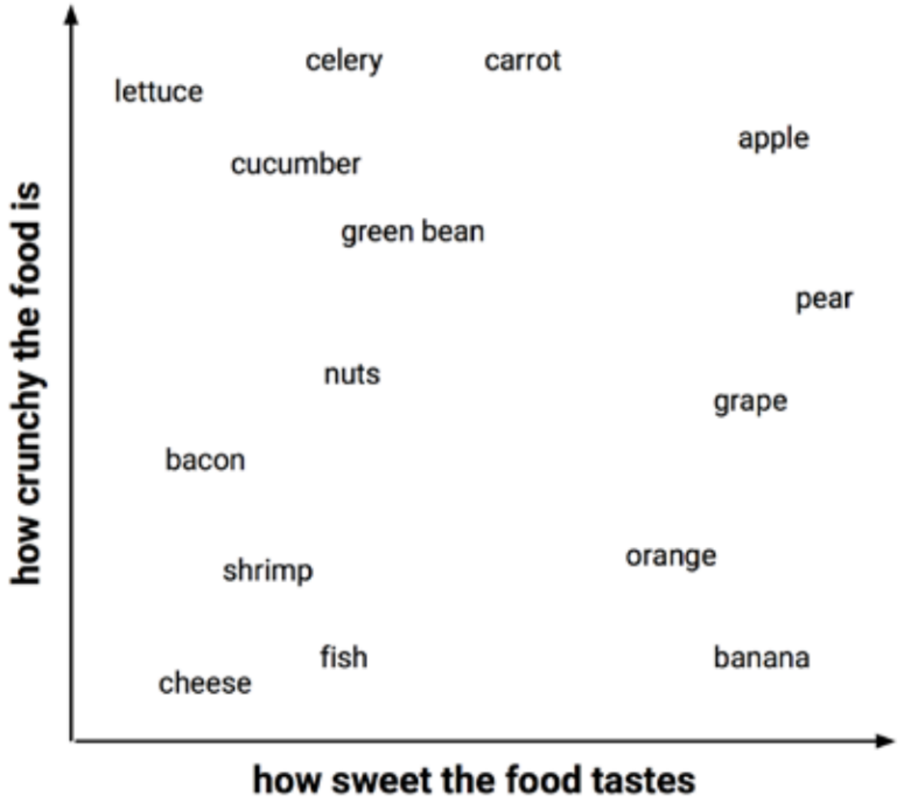
- Rather than defining the group boundaries subjectively, it would be nice to use machine learning to define them objectively.
  - **The k-means clustering algorithm**

The k-means Clustering Algorithm

- The k-means algorithm
  - *k* is the number of clusters
  - Assigns each of the *n* examples to one of the *k* clusters (clusters are not overlapped)
  - The goal is to minimize the differences within each cluster and maximize the differences between the clusters

The k-means Clustering Algorithm

- Using distance to assign and update clusters

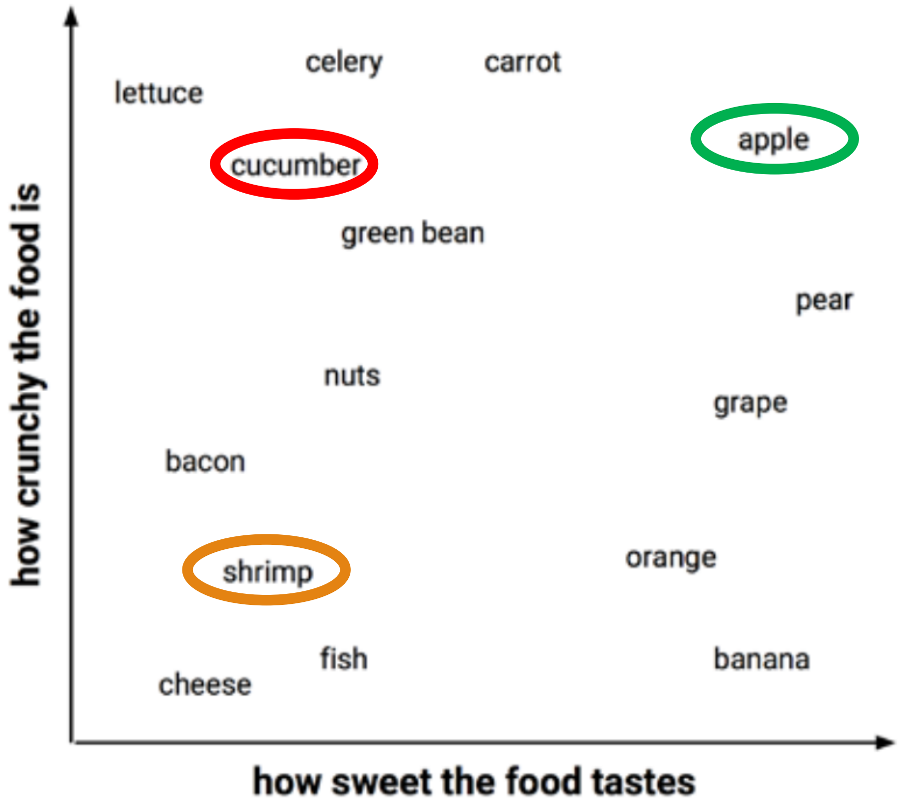


As with k-NN, k-means treats feature values as coordinates in a multidimensional feature space.

- Represent the feature space as a two-dimensional scatterplot

The k-means Clustering Algorithm

- Step 1: choosing *k* points in the feature space to serve as the cluster centers.



k = 3: 3 data points are randomly selected

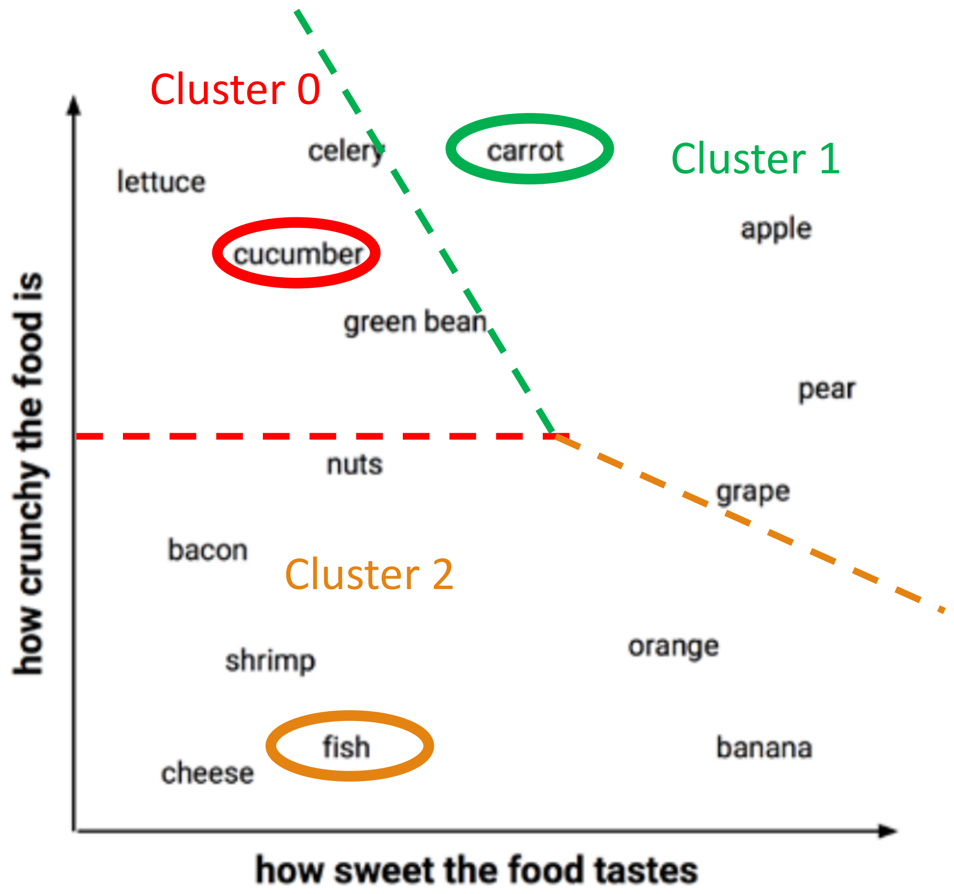
The k-means Clustering Algorithm

- Step 2: Assigning the remaining examples to the cluster center that is nearest according to the distance.
  - Euclidean distance between example *x* and example *y*

$$\mathrm{dist}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

  - Using this distance function, we find the distance between each example and each cluster center. The example is then assigned to the nearest cluster center.

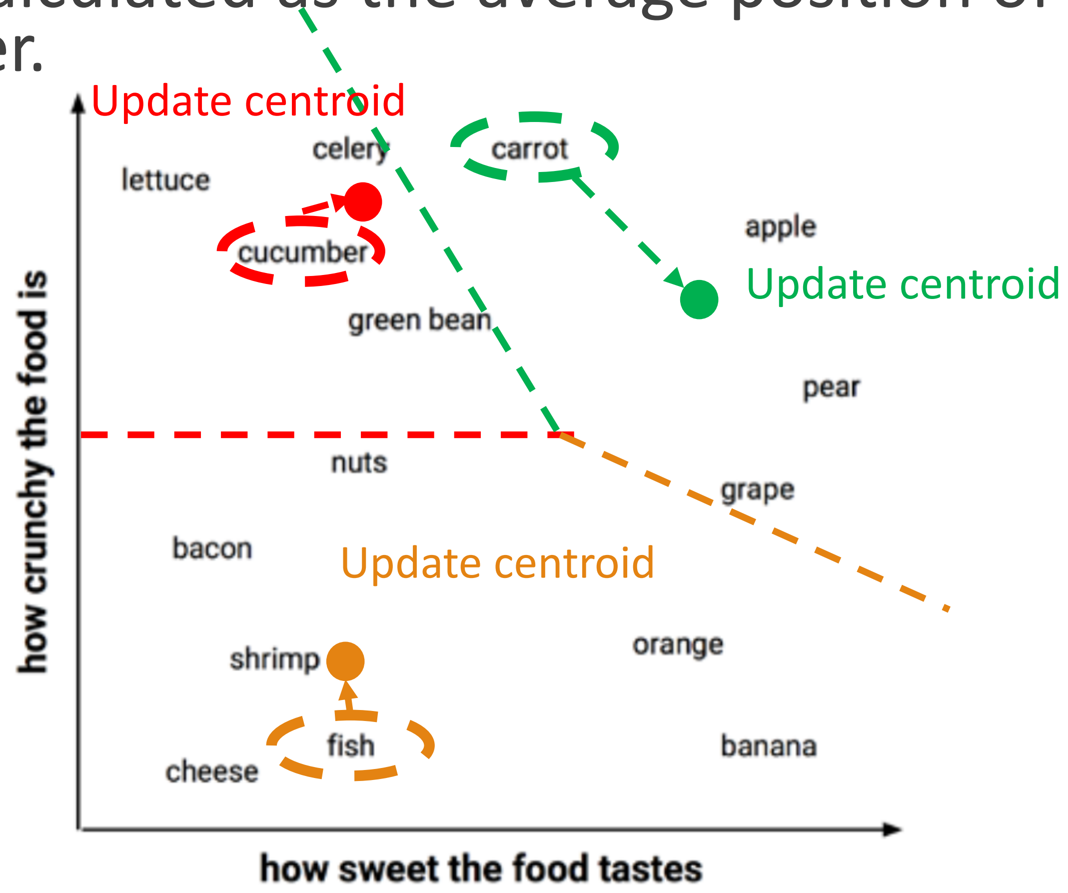
The k-means Clustering Algorithm



Keep in mind that as we are using distance calculations, all the features need to be numeric, and the values should be normalized to a standard range ahead of time.

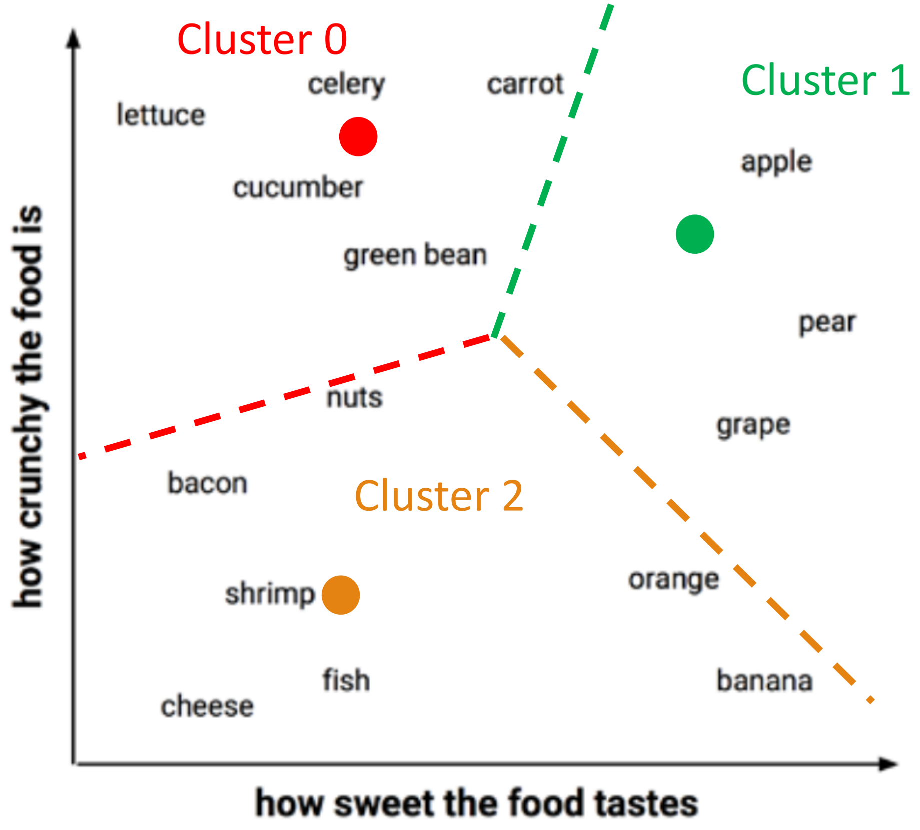
The k-means Clustering Algorithm

- Step 3: shifting the initial centers to a new location, known as the **centroid**, which is calculated as the average position of the points currently assigned to that cluster.



The k-means Clustering Algorithm

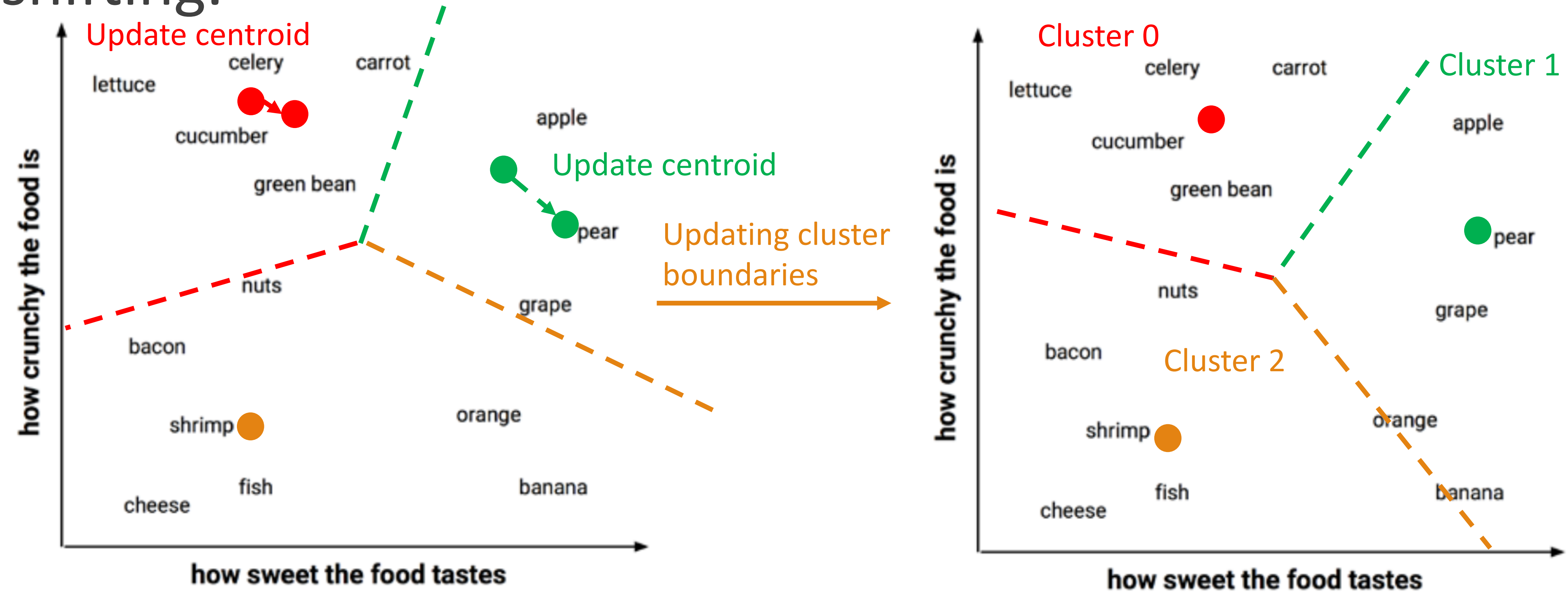
- Step 4: Updating the cluster boundaries, and reassigning points into new clusters.



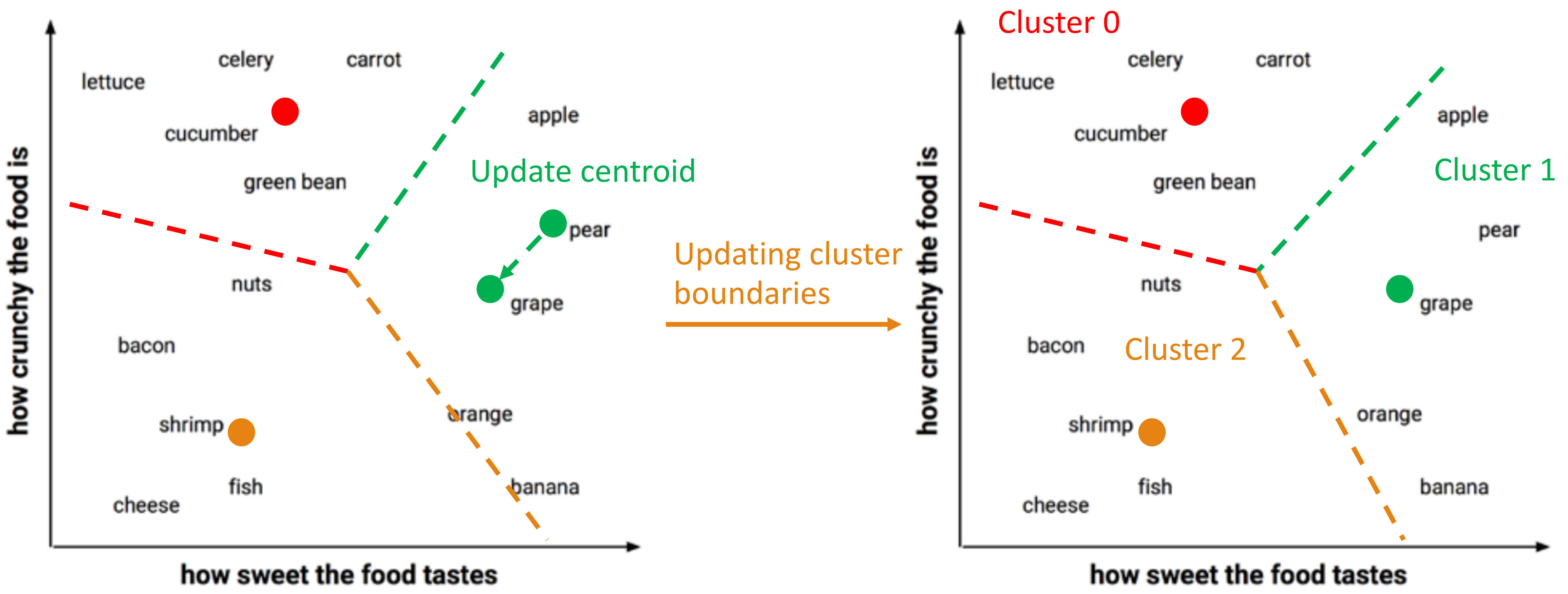
Update clusters

The k-means Clustering Algorithm

- Step 5: Repeat step 3 and step 4 until the centroids stop shifting.



The k-means Clustering Algorithm



The k-means Clustering Algorithm

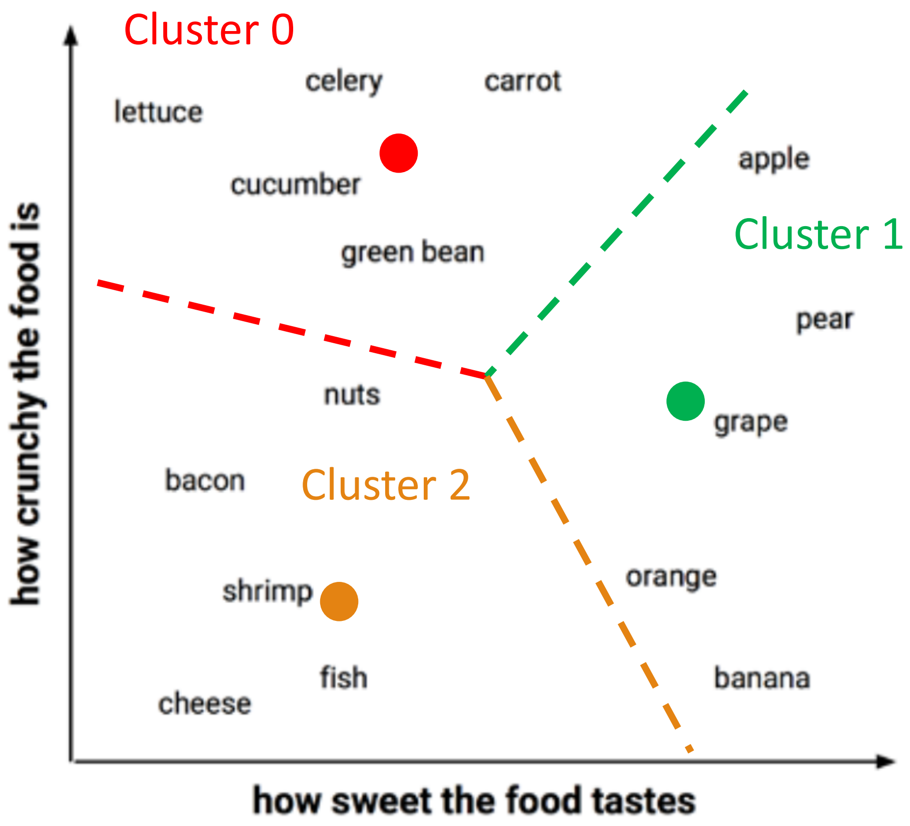
- Step 1: choosing *k* points in the feature space to serve as the cluster centers.
- Step 2: Assigning the remaining examples to the cluster center that is nearest according to the distance.
- Step 3: shifting the initial centers to a new location, known as the **centroid**, which is calculated as the average position of the examples currently assigned to that cluster.
- Step 4: Updating the cluster boundaries, and reassigning examples into new clusters.
- Step 5: Repeat step 3 and step 4 until the centroids stop shifting.

The k-means Clustering Algorithm

- Clustering results
  - The cluster assignments of each example (size of cluster)
  - Cluster centroids (cluster center)

- There is no training and testing process for unsupervised learning.

The k-means Clustering Algorithm



- Cluster 0
  - Instances: lettuce, celery, carrot, cucumber, green bean
  - Centroid: crunchiness-high, sweetness-low
- Cluster 1
  - Instances: apple, pear, grape, orange, banana
  - Centroid: crunchiness-medium，sweetness-high
- Cluster 2
  - Instances: nuts, bacon, shrimp, fish, cheese
  - Centroid: crunchiness-low, sweetness-low

The k-means Clustering Algorithm (Disadvantage)

- k-means algorithm is highly sensitive to the starting position of the cluster centers.
  - Random chance may have a substantial impact on the final set of clusters.
  - k-means++ proposes an alternative method for selecting the initial cluster centers.

Clustering on BART Riders

- You have been given a data file by the San Francisco Bay Area Rapid Transit (BART), which identifies a set of demographics for residents in a local area. We will use this file to determine residents segmentations so that we can use it to develop marketing plans accordingly.

Clustering on BART Riders

```
> summary(BartRider)
      Age          DistToWork     DualInc    Education       Gender        Income         Language      NbrInHouseHold
 Min.   :1.000   Min.   : 3.00   N:4153   Min.   :1.000   F:2958   Min.   :1.000   English:5025   Min.   :1.000
 1st Qu.:2.000   1st Qu.:10.00   Y:1340   1st Qu.:3.000   M:2535   1st Qu.:2.000   Other  : 164   1st Qu.:2.000
 Median :3.000   Median :11.00            Median :4.000            Median :6.000   Spanish: 304   Median :3.000
 Mean   :3.484   Mean   :11.48            Mean   :3.872            Mean   :5.161                  Mean   :2.905
 3rd Qu.:5.000   3rd Qu.:13.00            3rd Qu.:5.000            3rd Qu.:8.000                  3rd Qu.:4.000
 Max.   :7.000   Max.   :20.00            Max.   :6.000            Max.   :9.000                  Max.   :9.000
 NbrInHouseholdUnder18   OwnRent        YrsInArea       Rider
 Min.   :0.0000        Own   :2391   Min.   :1.000   No :3139
 1st Qu.:0.0000        Parent:1403   1st Qu.:4.000   Yes:2354
 Median :0.0000        Rent  :1699   Median :5.000
 Mean   :0.7073                      Mean   :4.292
 3rd Qu.:1.0000                      3rd Qu.:5.000
 Max.   :9.0000                      Max.   :5.000
```

Clustering on BART Riders

- Create dummy variables

```
> summary(BartRider)
      Age          DistToWork       DualInc          Education         Gender            Income        NbrInHouseHold
 Min.   :1.000   Min.   : 3.00   Min.   :0.0000   Min.   :1.000   Min.   :0.0000   Min.   :1.000   Min.   :1.000  
 1st Qu.:2.000   1st Qu.:10.00   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:2.000  
 Median :3.000   Median :11.00   Median :0.0000   Median :4.000   Median :1.0000   Median :6.000   Median :3.000  
 Mean   :3.484   Mean   :11.48   Mean   :0.2439   Mean   :3.872   Mean   :0.5385   Mean   :5.161   Mean   :2.905  
 3rd Qu.:5.000   3rd Qu.:13.00   3rd Qu.:0.0000   3rd Qu.:5.000   3rd Qu.:1.0000   3rd Qu.:8.000   3rd Qu.:4.000  
 Max.   :7.000   Max.   :20.00   Max.   :1.0000   Max.   :6.000   Max.   :1.0000   Max.   :9.000   Max.   :9.000  
 NbrInHouseholdUnder18   YrsInArea          Rider         Language_English Language_Spanish   OwnRent_own     OwnRent_Parent
 Min.   :0.0000        Min.   :1.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:0.0000        1st Qu.:4.000   1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000  
 Median :0.0000        Median :5.000   Median :0.0000   Median :1.0000   Median :0.00000   Median :0.0000   Median :0.0000  
 Mean   :0.7073        Mean   :4.292   Mean   :0.4285   Mean   :0.9148   Mean   :0.05534   Mean   :0.4353   Mean   :0.2554  
 3rd Qu.:1.0000        3rd Qu.:5.000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1.0000  
 Max.   :9.0000        Max.   :5.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000  
```

Clustering on BART Riders

BartRider_clustering <- SimpleKMeans(BartRider, Weka_control(N=2))

```
Number of iterations: 12
Within cluster sum of squared errors: 6618.976450318834

Initial starting points (random):

Cluster 0: 6,9,0,6,0,9,4,0,5,0,1,0,1,0
Cluster 1: 3,16,0,3,1,5,2,1,5,0,1,0,0,0
```

Within-cluster sum of squared errors: the squared errors from the mean (centroid) of the cluster of all the observations belonging to that cluster. The Within-cluster sum of squared errors $W(C_k)$ of a cluster C_k is defined as $\sum_{x_i \in C_k}(x_i - \overline{x_k})^2$, where $\overline{x_k}$ is the mean (centroid) of cluster C_k.

Clustering on BART Riders

- k=2

Final cluster centroids:

Attribute	Full Data	Cluster# 0	1
	(5493.0)	(2678.0)	(2815.0)
Age	3.4837	4.5732	2.4472
DistToWork	11.4828	11.5362	11.432
DualInc	0.2439	0.4671	0.0316
Education	3.8724	4.4145	3.3567
Gender	0.5385	0.6247	0.4565
Income	5.1606	6.9712	3.438
NbrInHouseHold	2.9053	2.7054	3.0956
NbrInHouseholdUnder18	0.7073	0.5963	0.8128
YrsInArea	4.2922	4.5063	4.0885
Rider	0.4285	0.1617	0.6824
Language_English	0.9148	0.9563	0.8753
Language_Spanish	0.0553	0.0295	0.0799
OwnRent_own	0.4353	0.8794	0.0128
OwnRent_Parent	0.2554	0	0.4984

Cluster size

Centroids

Clustering on BART Riders

- k=3

Final cluster centroids:

Attribute	Full Data (5493.0)	Cluster# 0 (2368.0)	1 (1210.0)	2 (1915.0)
Age	3.4837	4.6917	3.2603	2.1311
DistToWork	11.4828	11.5051	11.5231	11.4298
DualInc	0.2439	0.4472	0.1694	0.0397
Education	3.8724	4.4054	4.1347	3.0475
Gender	0.5385	0.5904	0.4893	0.5055
Income	5.1606	6.9949	5.6868	2.5598
NbrInHouseHold	2.9053	2.7758	2.2653	3.47
NbrInHouseholdUnder18	0.7073	0.6326	0.3702	1.0125
YrsInArea	4.2922	4.5743	3.9256	4.1749
Rider	0.4285	0.1858	0.0149	0.9901
Language_English	0.9148	0.951	0.9612	0.8407
Language_Spanish	0.0553	0.0329	0.0256	0.1018
OwnRent_own	0.4353	0.9996	0	0.0125
OwnRent_Parent	0.2554	0	0.1157	0.6595

Based on the cluster size and centroids, which k (k=2 or k=3) you will use, and why?

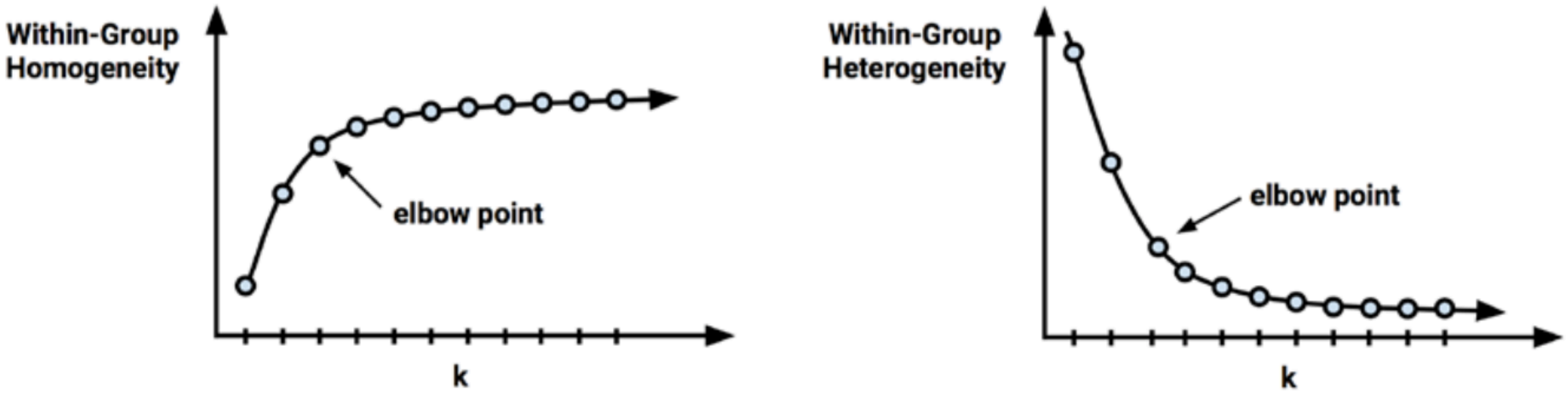
If we segment residents into 3 clusters, what marketing plans you can use to target each cluster?

Elbow Method

- Choosing the appropriate number of clusters
  - k-means is sensitive to the number of clusters
    - *What if k=1*
    - *What if k=n*, where *n* is the number of examples (instances)
    - Ideally, you will have *a priori* knowledge (a prior belief) about the true groupings and you can apply this information to choosing the number of clusters.
      - Sometimes the number of clusters is dictated by business requirements or the motivation for the analysis.

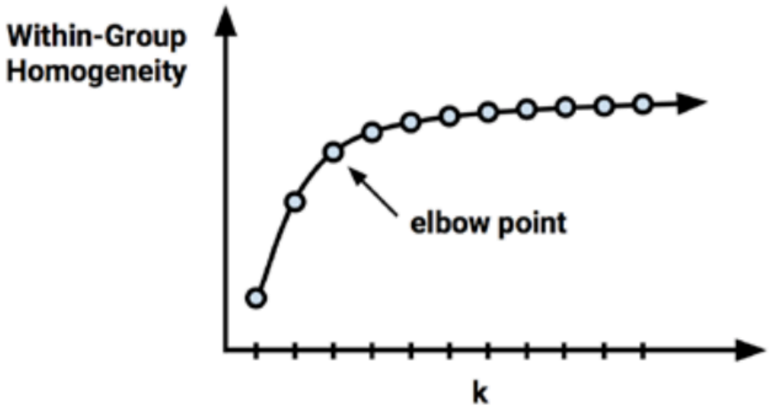
Elbow Method

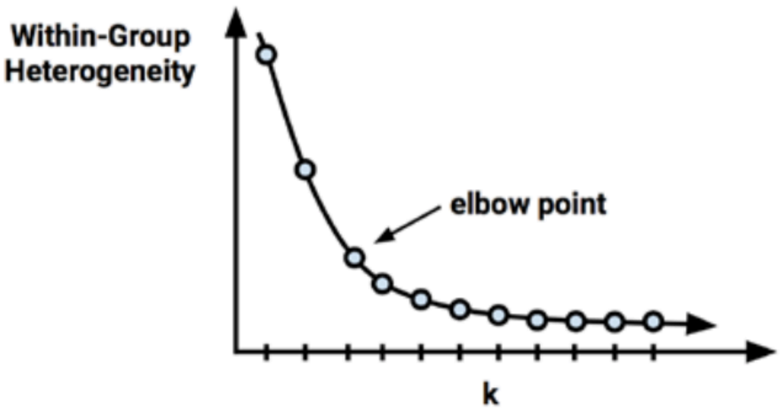
- Choosing the appropriate number of clusters: **elbow method**
  - **Elbow method** attempts to gauge how the homogeneity or heterogeneity within the clusters changes for various values of *k*.
    - The homogeneity within clusters is expected to increase as additional clusters are added; heterogeneity will continue to decrease with more clusters.



Elbow Method

- Choosing the appropriate number of clusters: **elbow method**
  - You could continue to see improvements until each example is in its own cluster, the goal is not to maximize homogeneity or minimize heterogeneity, but rather to find *k* so that there are diminishing returns beyond that point. This value of *k* is known as the **elbow point** because it looks like an elbow .

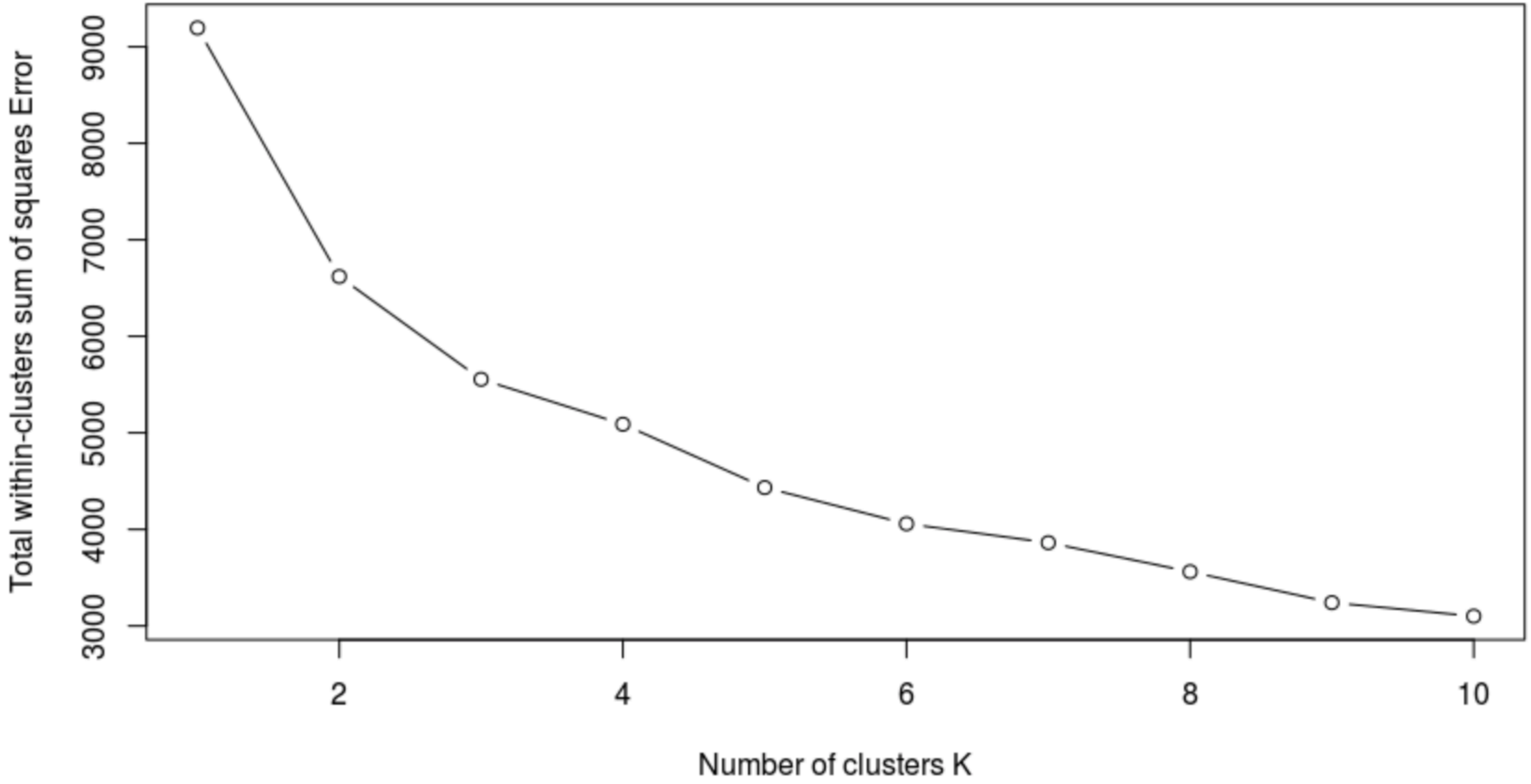




Elbow Method

- Choosing the appropriate number of clusters: **elbow method**
  - Measure homogeneity or heterogeneity within the clusters
    - Within-cluster sum of squared errors: the squared errors from the mean (centroid) of the cluster of all the observations belonging to that cluster. The Within-cluster sum of squared errors $W(C_k)$ of a cluster C_k is defined as $\sum_{x_i \in C_k} (x_i - \overline{x_k})^2$, where $\overline{x_k}$ is the mean (centroid) of cluster C_k.

Elbow Method



The k-means Clustering Algorithm

Strengths	**Weaknesses**
• Uses simple principles that can be explained in non-statistical terms • Highly flexible, and can be adapted with simple adjustments to address nearly all of its shortcomings • Performs well enough under many real-world use cases	• Not as sophisticated as more modern clustering algorithms • Because it uses an element of random chance, it is not guaranteed to find the optimal set of clusters • Requires a reasonable guess as to how many clusters naturally exist in the data • Not ideal for non-spherical clusters or clusters of widely varying density

Classification with Clustering

- Classification with Clustering
  - Step 1: clustering
  - Step 2: build a classification model for each cluster

Classification with Clustering

- Testing performance with cross validation

```
> df <- BartRider
> target <- 10
> nFolds <- 10
> seedVal <- 1
> prediction_method <- C5.0
> metrics_list <- c("ACC","PRECISION","TPR","F1")
> cv_function_test(df, target, nFolds, seedVal, prediction_method, metrics_list)
|       |   ACC| PRECISION1| PRECISION2|  TPR1|  TPR2|   F11|   F12|
|:------|-----:|----------:|----------:|-----:|-----:|-----:|-----:|
|Fold01 | 89.44|      88.79|      90.41| 93.31| 84.26| 90.99| 87.22|
|Fold02 | 87.45|      90.97|      83.27| 86.62| 88.56| 88.74| 85.83|
|Fold03 | 90.53|      93.38|      87.04| 89.81| 91.49| 91.56| 89.21|
|Fold04 | 87.27|      86.09|      89.15| 92.68| 80.08| 89.26| 84.38|
|Fold05 | 90.16|      92.48|      87.24| 90.13| 90.21| 91.29| 88.70|
|Fold06 | 88.52|      90.88|      85.54| 88.85| 88.09| 89.86| 86.79|
|Fold07 | 88.34|      90.32|      85.77| 89.17| 87.23| 89.74| 86.50|
|Fold08 | 89.80|      89.78|      89.82| 92.65| 86.02| 91.19| 87.88|
|Fold09 | 88.16|      88.07|      88.29| 91.72| 83.40| 89.86| 85.78|
|Fold10 | 92.36|      95.33|      88.80| 91.08| 94.07| 93.16| 91.36|
|Mean   | 89.20|      90.61|      87.53| 90.60| 87.34| 90.57| 87.36|
|Sd     |  1.57|       2.68|       2.21|  2.08|  4.11|  1.31|  2.01|
```

No clustering: use the whole dataset to perform 10-fold cross validation

Classification with Clustering

- Perform clustering without target variable

Final cluster centroids:

Attribute	Full Data (5493.0)	Cluster# 0 (2389.0)	1 (1404.0)	2 (1700.0)
Age	3.4837	4.6735	1.7521	3.2418
DistToWork	11.4828	11.5036	11.4309	11.4965
DualInc	0.2439	0.4429	0.0064	0.1606
Education	3.8724	4.3951	2.76	4.0565
Gender	0.5385	0.5873	0.5726	0.4418
Income	5.1606	6.9481	2.9915	4.44
NbrInHouseHold	2.9053	2.7848	3.9972	2.1729
NbrInHouseholdUnder18	0.7073	0.6342	1.2778	0.3388
YrsInArea	4.2922	4.5701	4.4295	3.7882
Language_English	0.9148	0.9473	0.8504	0.9224
Language_Spanish	0.0553	0.036	0.0933	0.0512
OwnRent_own	0.4353	1	0	0.0012
OwnRent_Parent	0.2554	0	0.9993	0

Classification with Clustering

- Create three clusters based on the clustering results

```
BartRider$class_ids = BartRider_clustering3$class_ids
BartRider1 = BartRider[which(BartRider$class_ids==0),]
BartRider2 = BartRider[which(BartRider$class_ids==1),]
BartRider3 = BartRider[which(BartRider$class_ids==2),]
```

Classification with Clustering

```
'data.frame':   2389 obs. of  15 variables:
 $ Age    'data.frame':   1404 obs. of  15 variables:
 $ DistTo $ Age  'data.frame':   1700 obs. of  15 variables:
 $ DualIn $ Dist $ Age                     : int  7 3 2 3 2 4 6 3 7 3 ...
 $ Educat $ Dual $ DistToWork              : int  14 9 10 12 13 11 13 10 10 9 ...
 $ Gender $ Educ $ DualInc                 : num  0 0 0 0 0 0 1 0 0 1 ...
 $ Income $ Gend $ Education               : int  3 3 4 4 3 3 1 5 4 4 ...
 $ NbrInH $ Inco $ Gender                  : num  1 0 0 1 0 1 1 1 1 1 ...
 $ NbrInH $ NbrI $ Income                  : int  3 1 3 4 3 3 2 4 1 6 ...
 $ YrsInA $ NbrI $ NbrInHouseHold          : int  1 1 4 4 1 1 4 2 1 2 ...
 $ Rider  $ YrsI $ NbrInHouseholdUnder18: int  0 0 2 1 0 0 1 0 0 0 ...
 $ Langua $ Ride $ YrsInArea               : int  5 5 2 2 5 4 3 3 5 1 ...        2 ...
 $ Langua $ Lang $ Rider                   : Factor w/ 2 levels "0","1": 2 2 2 2 2
 $ OwnRen $ Lang $ Language_English        : num  1 1 1 1 1 1 0 1 1 1 ...
 $ OwnRen $ OwnR $ Language_Spanish        : num  0 0 0 0 0 0 1 0 0 0 ...
 $ class_ $ OwnR $ OwnRent_own             : num  0 0 0 0 0 0 0 0 0 0 ...
          $ clas $ OwnRent_Parent          : num  0 0 0 0 0 0 0 0 0 0 ...
                 $ class_ids               : int  2 2 2 2 2 2 2 2 2 2 ...
```

Classification with Clustering

- Performance on cluster 0

	ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
Fold01	92.47	94.87	81.82	95.85	78.26	95.36	80.00
Fold02	93.31	92.75	96.88	99.48	67.39	96.00	79.49
Fold03	94.56	95.45	90.24	97.93	80.43	96.68	85.06
Fold04	91.21	93.43	80.49	95.85	71.74	94.63	75.86
Fold05	92.47	93.53	86.84	97.41	71.74	95.43	78.57
Fold06	92.02	95.77	77.55	94.27	82.61	95.01	80.00
Fold07	91.67	92.61	86.49	97.41	68.09	94.95	76.19
Fold08	94.54	96.86	85.11	96.35	86.96	96.61	86.02
Fold09	92.05	93.50	84.62	96.89	71.74	95.17	77.65
Fold10	91.21	93.88	79.07	95.34	73.91	94.60	76.40
Mean	92.55	94.27	84.91	96.68	75.29	95.44	79.52
Sd	1.22	1.40	5.73	1.48	6.49	0.75	3.53

Use the cluster 0
to perform 10-fold cross validation

Classification with Clustering

- Performance on cluster 1

	ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
Fold01	90.00	75.00	90.44	18.75	99.19	30.00	94.62
Fold02	89.29	60.00	90.37	18.75	98.39	28.57	94.21
Fold03	89.29	66.67	89.78	12.50	99.19	21.05	94.25
Fold04	89.36	55.56	91.67	31.25	96.80	40.00	94.16
Fold05	92.20	100.00	91.91	31.25	100.00	47.62	95.79
Fold06	90.78	66.67	92.42	37.50	97.60	48.00	94.94
Fold07	92.20	100.00	91.91	31.25	100.00	47.62	95.79
Fold08	90.00	66.67	91.04	25.00	98.39	36.36	94.57
Fold09	92.86	100.00	92.54	37.50	100.00	54.55	96.12
Fold10	92.86	100.00	92.54	37.50	100.00	54.55	96.12
Mean	90.88	79.06	91.46	28.12	98.96	40.83	95.06
Sd	1.50	18.70	1.00	8.96	1.14	11.54	0.81

Use the cluster 1
to perform 10-fold cross
validation

Classification with Clustering

- Performance on cluster 2

	ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
Fold01	81.76	82.46	80.36	89.52	69.23	85.84	74.38
Fold02	87.57	88.89	85.25	91.43	81.25	90.14	83.20
Fold03	83.53	86.67	78.46	86.67	78.46	86.67	78.46
Fold04	85.29	90.82	77.78	84.76	86.15	87.68	81.75
Fold05	84.12	86.11	80.65	88.57	76.92	87.32	78.74
Fold06	80.00	84.47	73.13	82.86	75.38	83.65	74.24
Fold07	82.94	88.00	75.71	83.81	81.54	85.85	78.52
Fold08	81.76	83.64	78.33	87.62	72.31	85.58	75.20
Fold09	77.78	84.00	69.01	79.25	75.38	81.55	72.06
Fold10	82.94	82.20	84.62	92.38	67.69	87.00	75.21
Mean	82.77	85.72	78.33	86.69	76.43	86.13	77.18
Sd	2.72	2.87	4.92	4.07	5.73	2.32	3.55

Use the cluster 2
to perform 10-fold cross validation