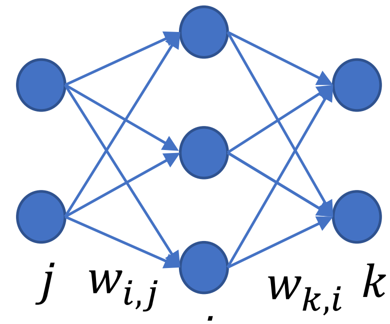
L13: Unsupervised Learning and Clustering

Shan Wang

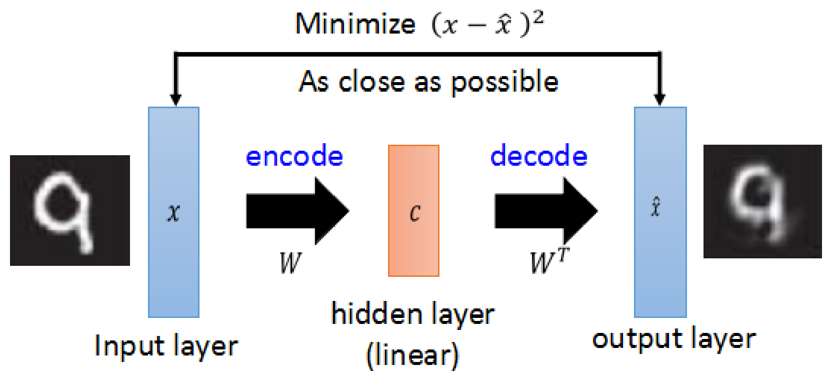
Lingnan College, Sun Yat-sen University

2020 Data Mining and Machine Learning LN3119
https://wangshan731.github.io/DM-ML/

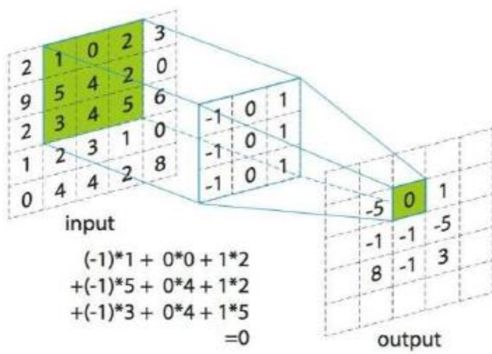
Last lecture

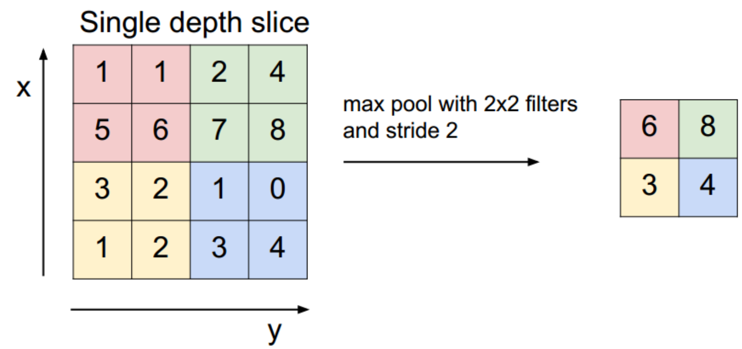


- Multi-layer Perceptron
- Auto-Encoder
  - Encode, decode
  - Stacked Auto-Encoder

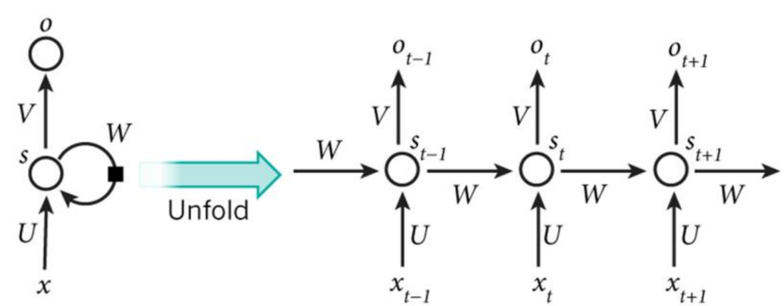


- CNN
  - Convolution
    - Kernel
  - Pooling

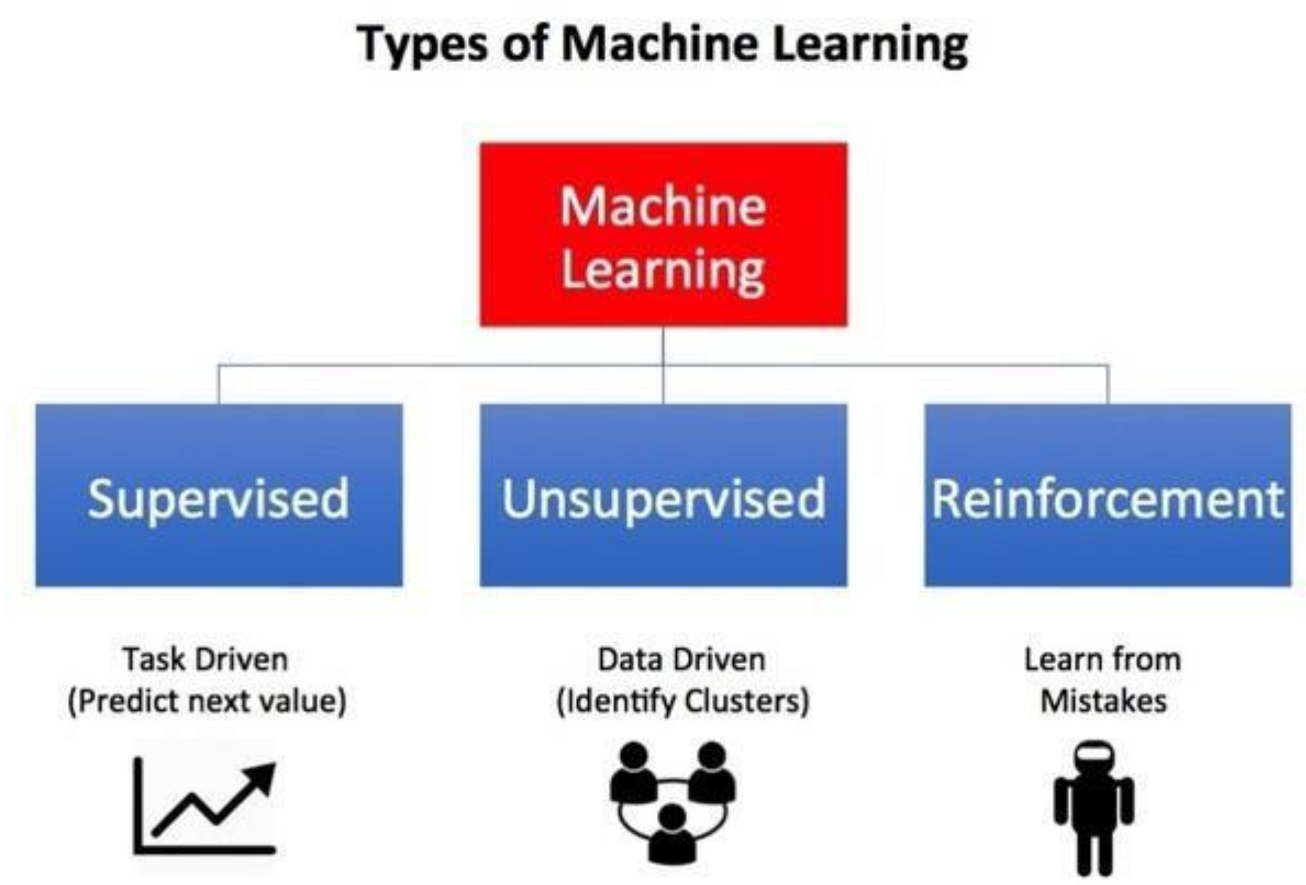




- RNN
  - Time series inputs/outputs
  - Share weights
  - Memory

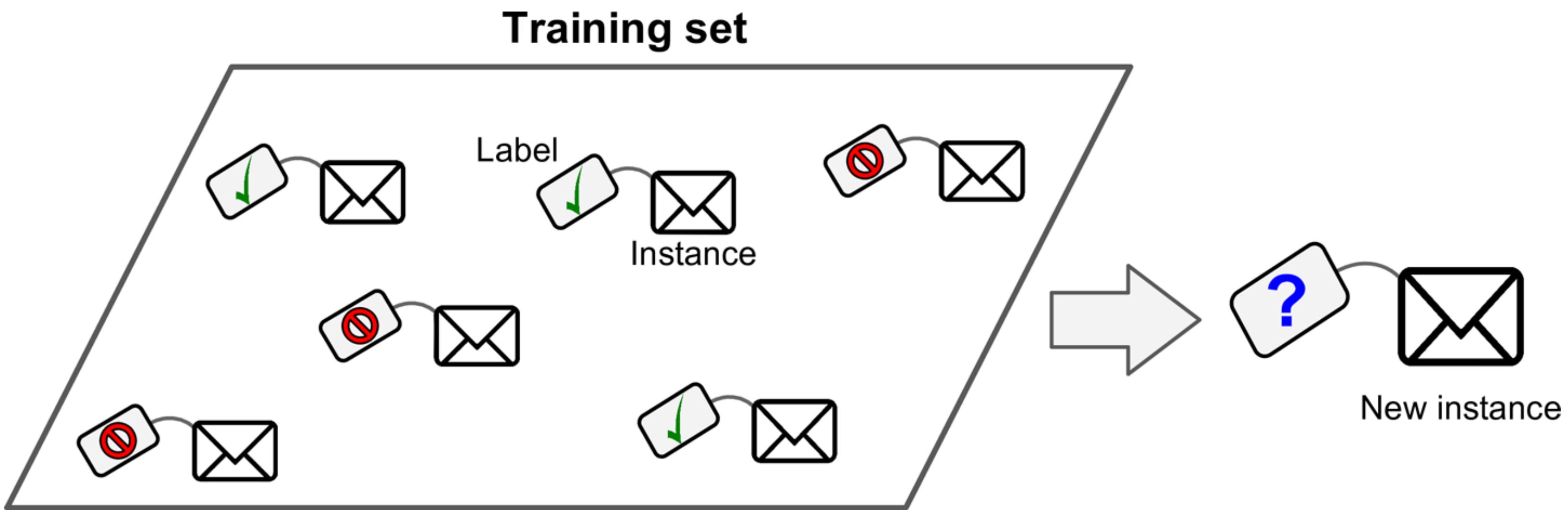


Classification of ML



Supervised Learning Revisit

- Learning a function that maps an input to an output based on example input-output pairs



Can we learn some information about the data without the labels/targets? How?

Course Outline

- Supervised learning
  - Linear regression
  - Logistic regression
  - SVM and kernel
  - Tree models
- Unsupervised learning
  - Clustering
  - PCA
  - EM
- Deep learning
  - Neural networks
  - Convolutional NN
  - Recurrent NN
- Reinforcement learning
  - MDP
  - ADP
  - Deep Q-Network

This lecture

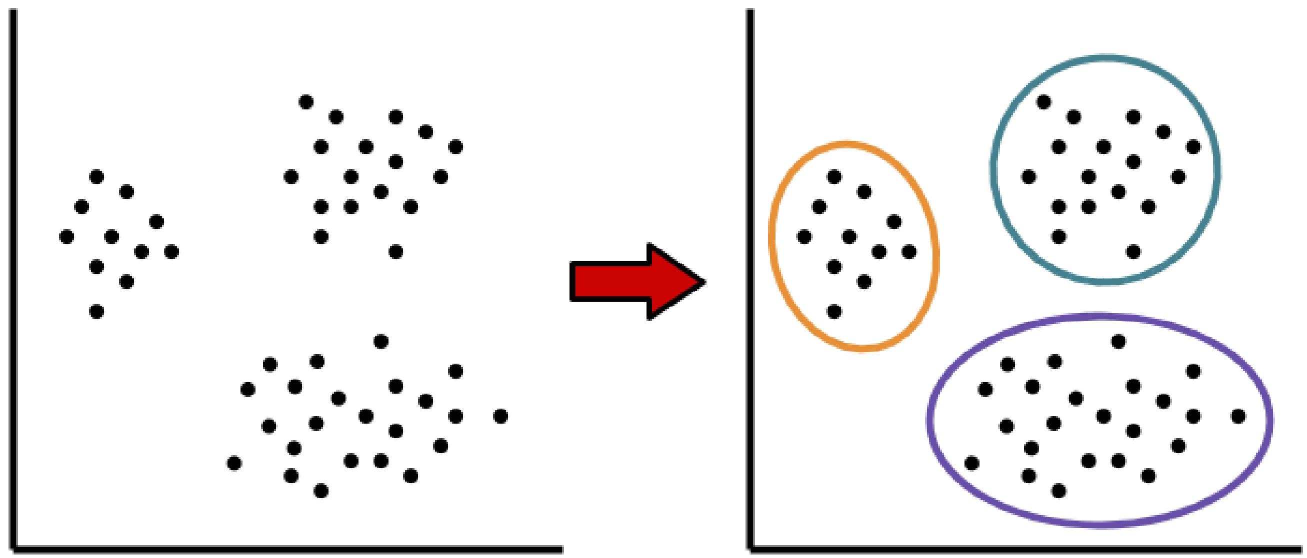
- Unsupervised Learning
- Clustering
  - Hierarchical clustering
  - k-means clustering
- Applications: Netflix

Reference: VE 445, Shuai LI (SJTU); OPIM 326 Daniel Zheng (SMU)

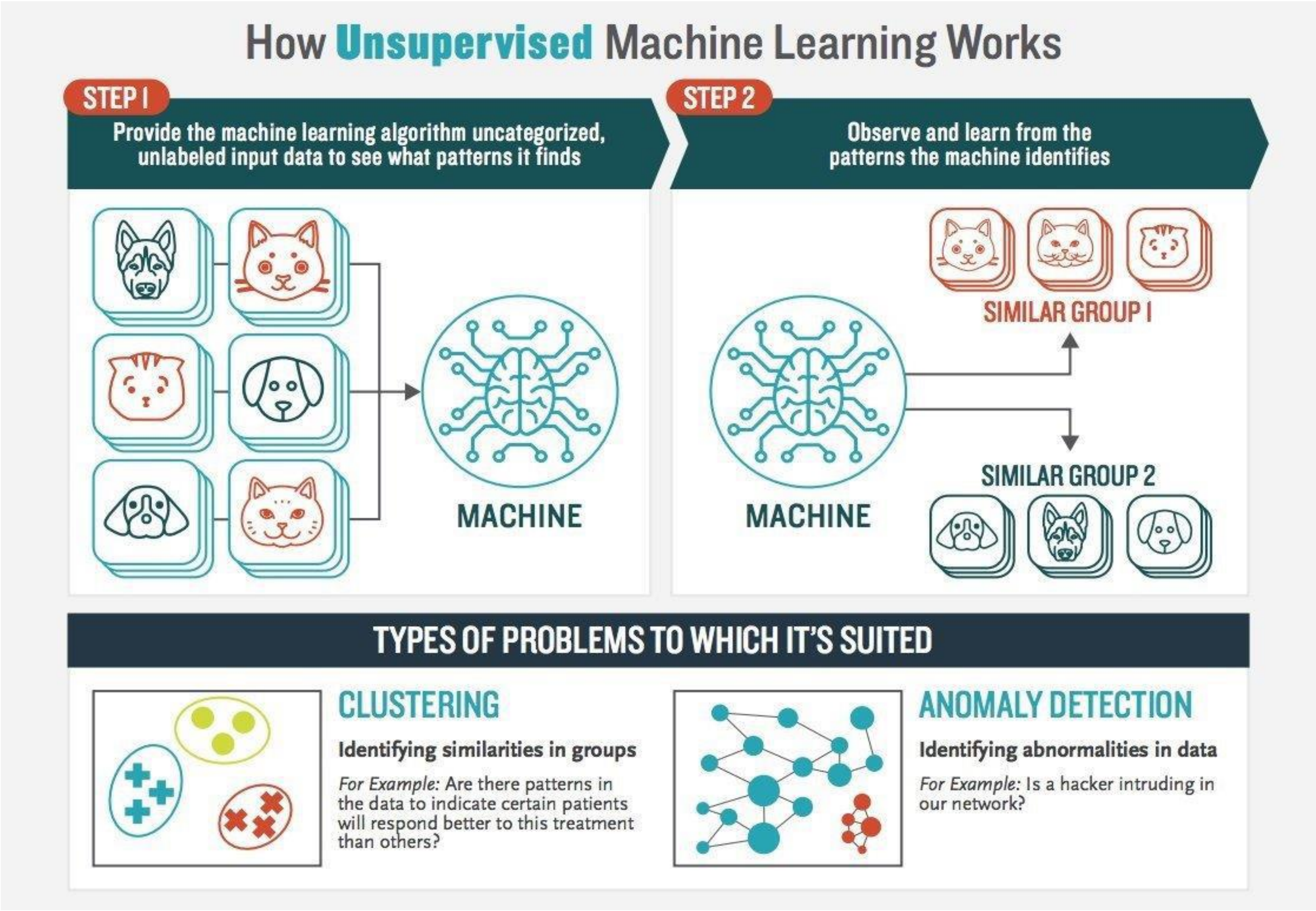
Unsupervised Learning

Unsupervised Learning Example

- Finding previously unknown patterns in data set without pre-existing labels



Unsupervised Learning Process



Compared with Supervised Learning

	Supervised Learning	Unsupervised Learning
Data format		
Training data		
Goals		
Functions		

Generally, unsupervised learning is used to analysis, recognize, discover some patterns in the data.

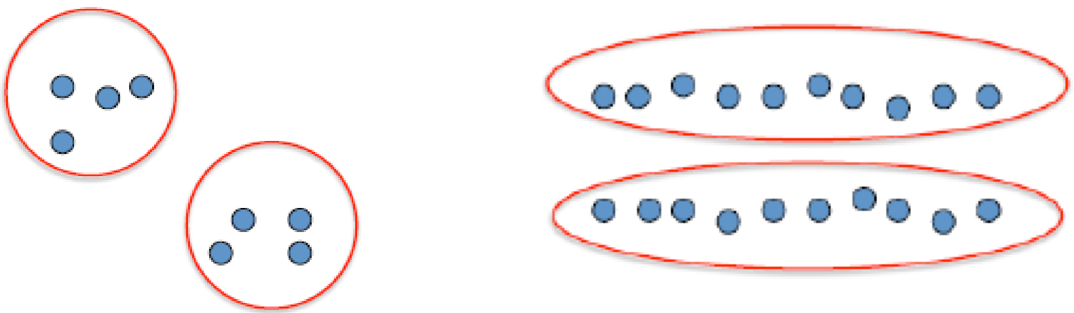
Clustering

Clustering

- Unsupervised learning
- Goal: segment the data into similar groups
- Require: data without labels
- Applications:
  - Cluster customers according to purchase histories
  - Cluster genes according to expression profile
  - Cluster search results according to topic
  - Cluster Facebook users according to interests
  - Cluster a museum catalog according to image similarity
- Can also cluster data into “similar” groups and then build a predictive model for each group
  - Cluster-then-predict

Intuitions for Clustering

- Basic Idea: group similar instances together
- Example:

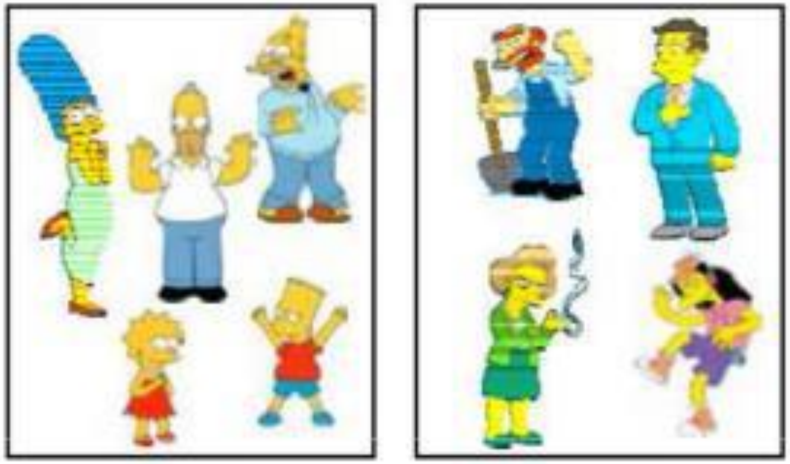


- Key issues:
  - What makes a cluster?
  - How to find them?

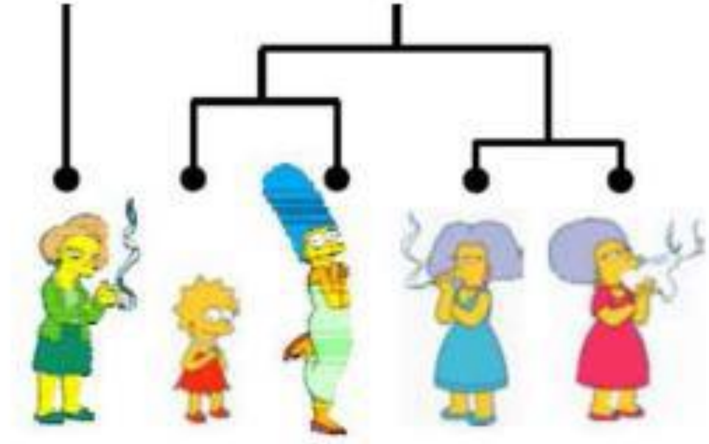
Different Algorithms

Two Types of Algorithms

- Partition algorithm
  - k-means
  - Mixture of Gaussian
  - Spectral Clustering



- Hierarchical algorithm
  - Agglomerative (bottom up)
  - Divisive (top down)



Define Similarity

- What makes a cluster?
  - Similar instances!
- What could “similar” mean?
  - Small distance
- A natural choice is “Euclidean distance”
  - Distance between points $\boldsymbol{x}$ and $\boldsymbol{y}$ is

$$d_{xy} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_k - y_k)^2}$$

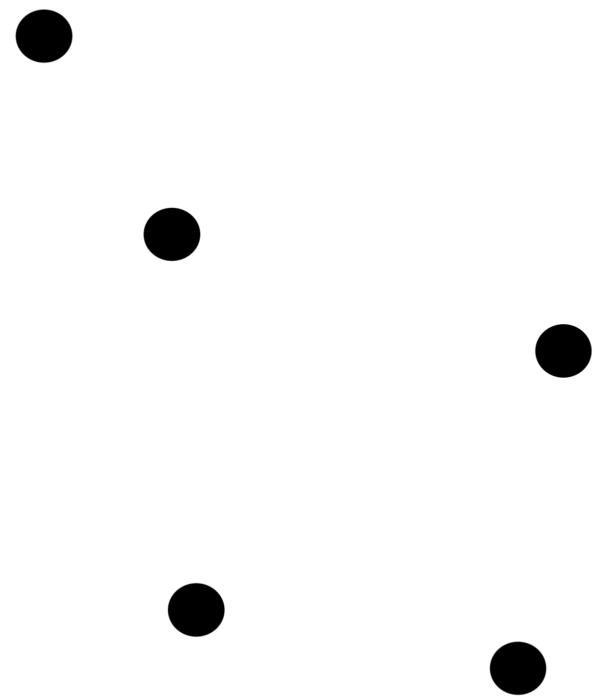
where k is the number of independent variables

K-means clustering

k-Means Clustering

K-Means Clustering Algorithm

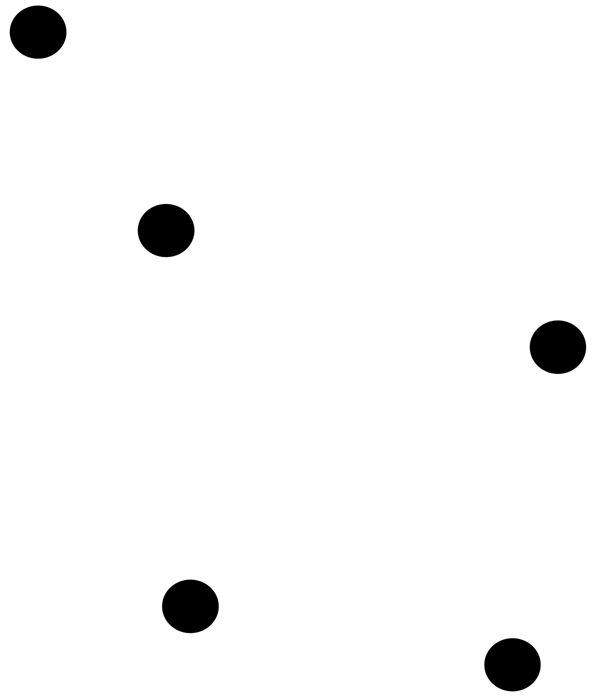
1. Specify desired number of clusters K



$K = 2$

K-Means Clustering

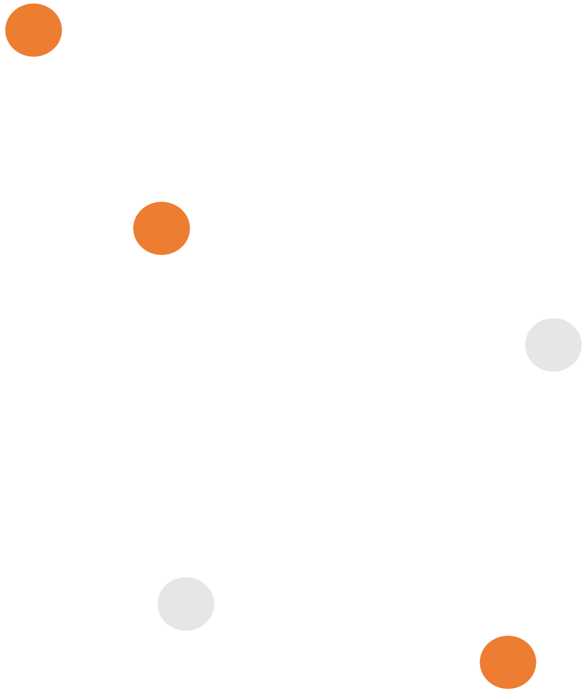
K-Means Clustering Algorithm
1. Specify desired number of clusters K
2. Randomly assign each data point to a cluster



K-Means Clustering

K-Means Clustering Algorithm

1. Specify desired number of clusters K
2. Randomly assign each data point to a cluster



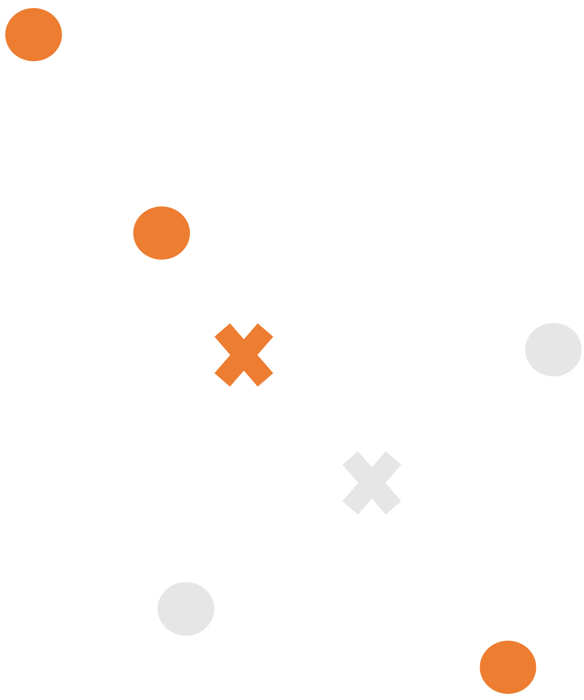
K-Means Clustering

K-Means Clustering Algorithm
1. Specify desired number of clusters K
2. Randomly assign each data point to a cluster
3. Compute cluster centroids

Calculate centroids:

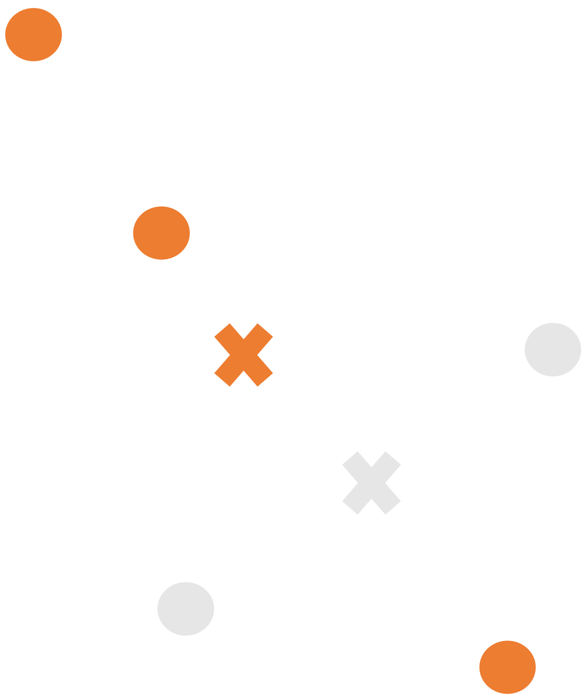
$$\overline{\boldsymbol{x}}^k = \frac{1}{|C_k|} \sum_{\boldsymbol{x} \in C_k} \boldsymbol{x}$$

C_k is the set of data which currently belongs to cluster k



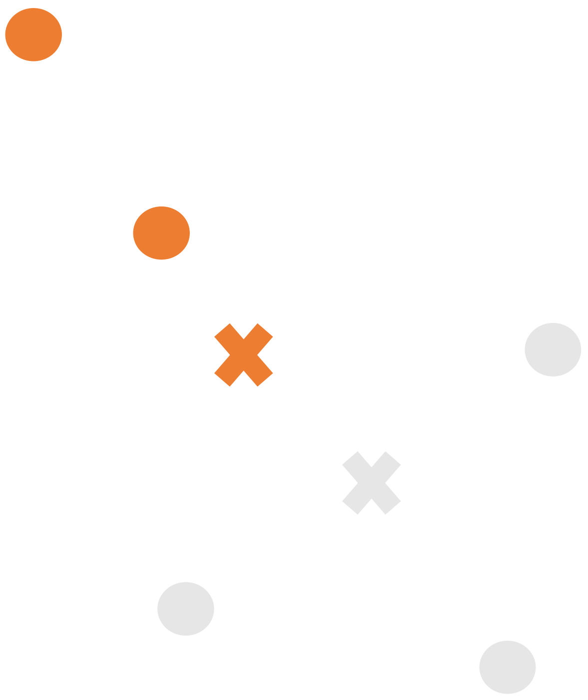
K-Means Clustering

K-Means Clustering Algorithm
1. Specify desired number of clusters K
2. Randomly assign each data point to a cluster
3. Compute cluster centroids
4. Re-assign each point to the closest cluster centroid



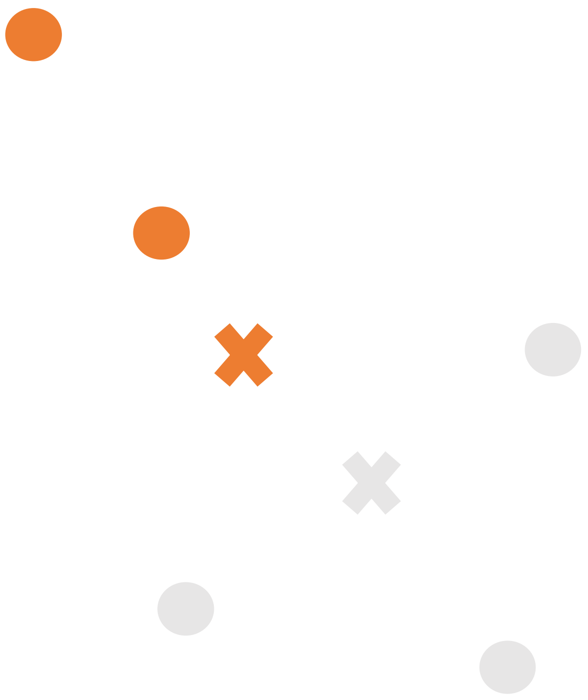
K-Means Clustering

***K*-Means Clustering Algorithm**
1. Specify desired number of clusters K
2. Randomly assign each data point to a cluster
3. Compute cluster centroids
4. Re-assign each point to the closest cluster centroid



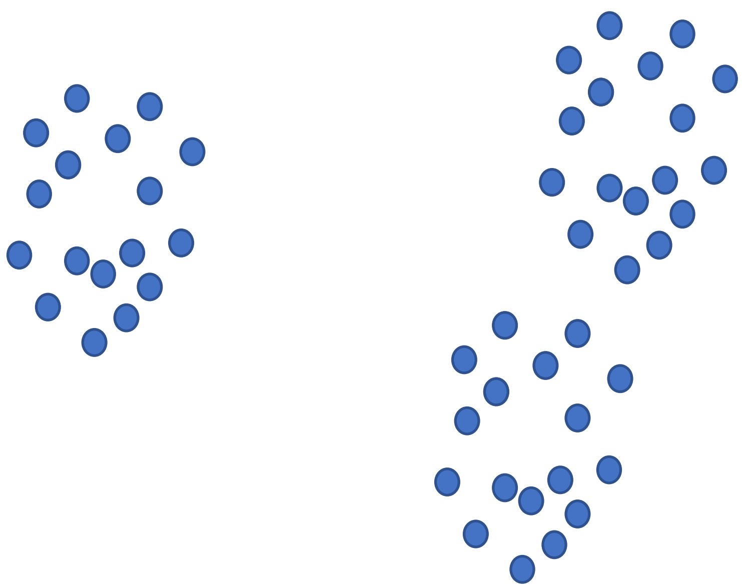
K-Means Clustering

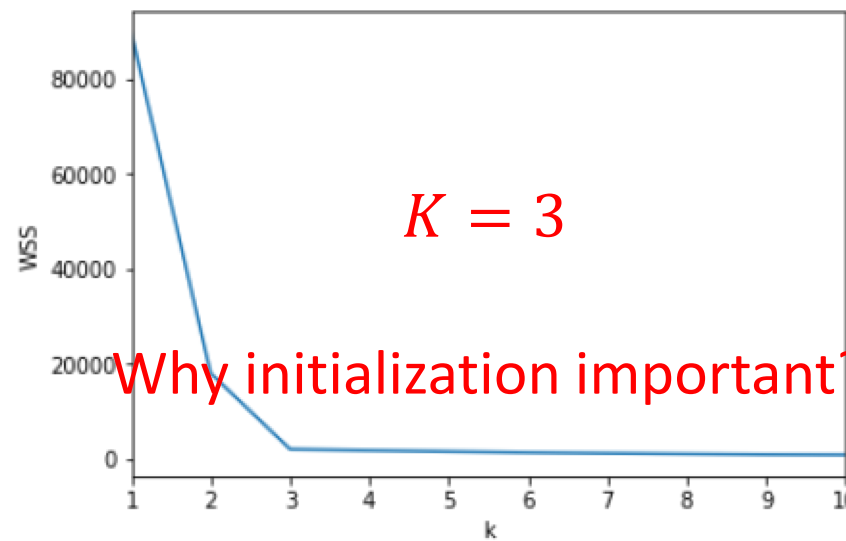
K-Means Clustering Algorithm
1. Specify desired number of clusters K
2. Randomly assign each data point to a cluster
3. Compute cluster centroids
4. Re-assign each point to the closest cluster centroid
5. Re-compute cluster centroids
6. Repeat 4 and 5 until no improvement is made



Important Considerations

- How to choose the number of clusters K
  - Previous knowledge
  - Experimenting for different value of K
    - Calculate the Within-Cluster-Sum of Squared Errors (WSS)
    - Choose the K for which WSS stops dropping significantly
- How to do initialization?



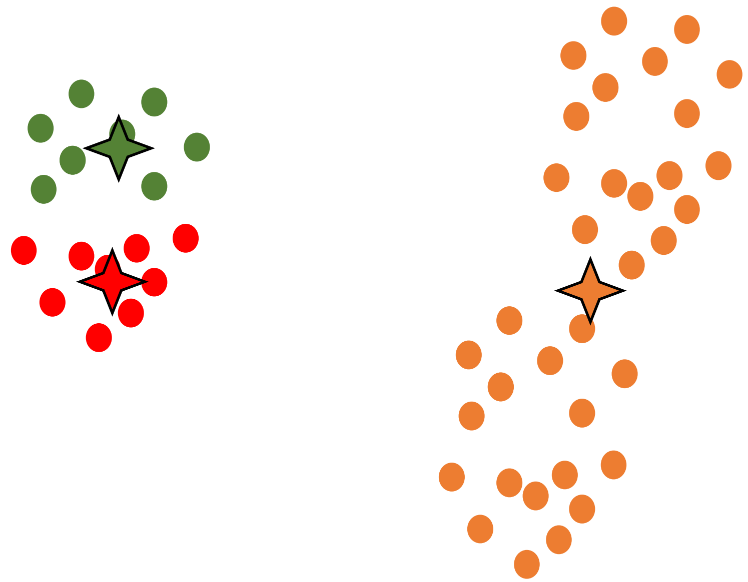


$K = 3$

Why initialization important?

Important Considerations

- How to choose the number of clusters K
  - Previous knowledge
  - Experimenting for different value of K
    - Calculate the Within-Cluster-Sum of Squared Errors (WSS)
    - Choose the K for which WSS stops dropping significantly
- How to do initialization?



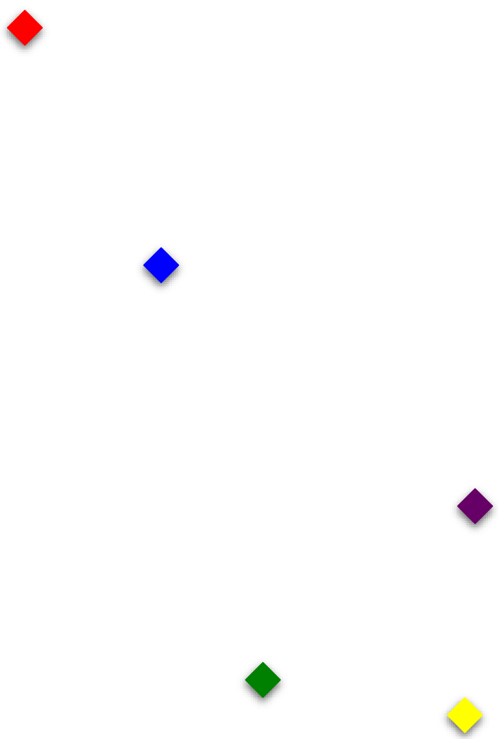
Important Considerations

- How to choose the number of clusters K
  - Previous knowledge
  - Experimenting for different value of K
    - Calculate the Within-Cluster-Sum of Squared Errors (WSS)
    - Choose the K for which WSS stops dropping significantly
- How to do initialization?
  - Previous knowledge
  - Experimenting with different initializations
    - Calculate the Within-Cluster-Sum of Squared Errors (WSS)
    - Choose the one with more balanced WSS

Agglomerative clustering

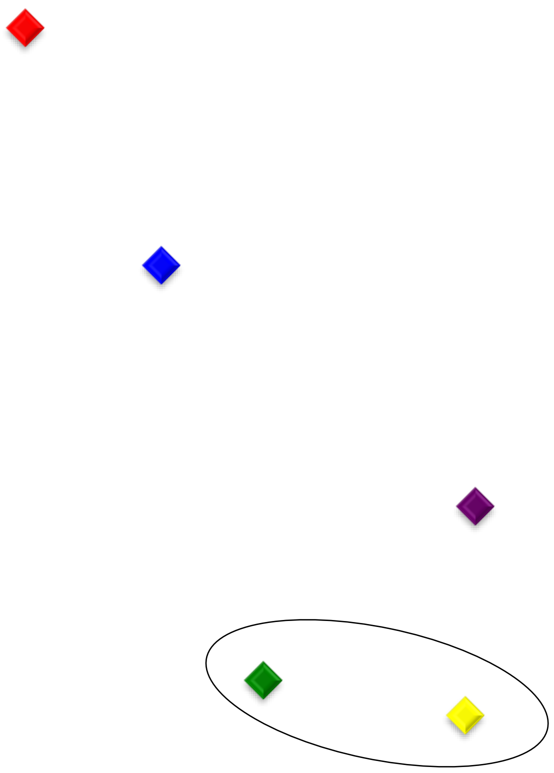
Hierarchical Clustering

- Start with each data point in its own cluster



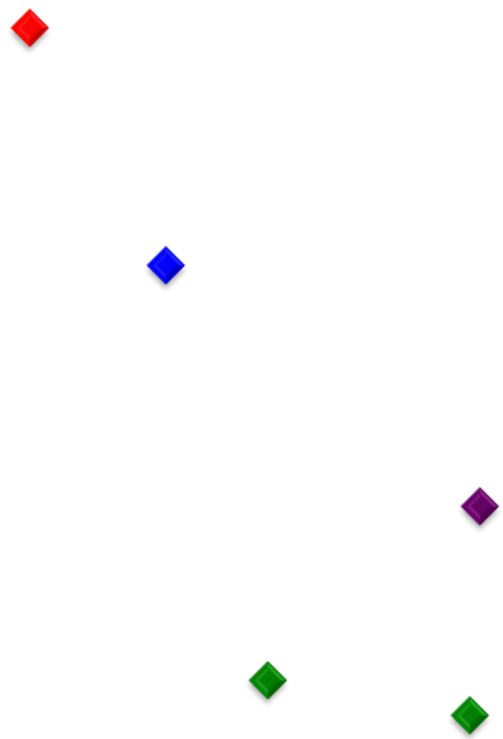
Hierarchical Clustering

- Combine two nearest clusters (Euclidean, Centroid)



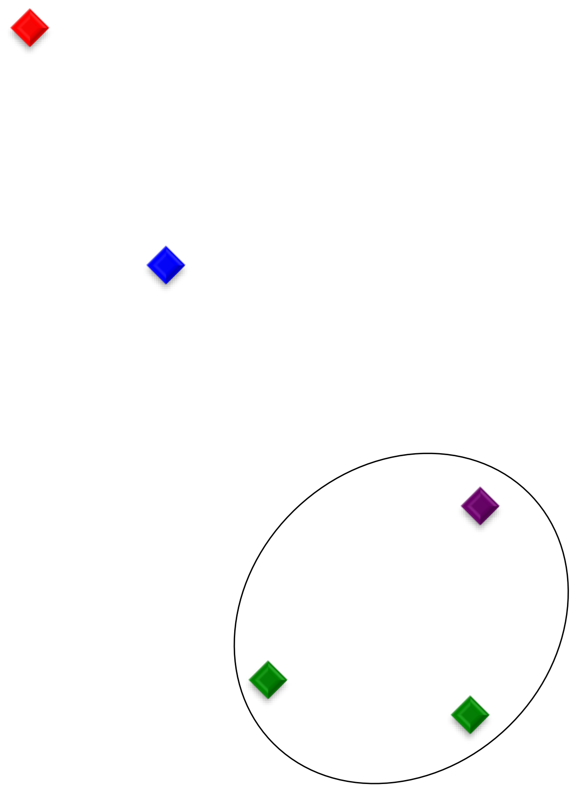
Hierarchical Clustering

- Combine two nearest clusters (Euclidean, Centroid)



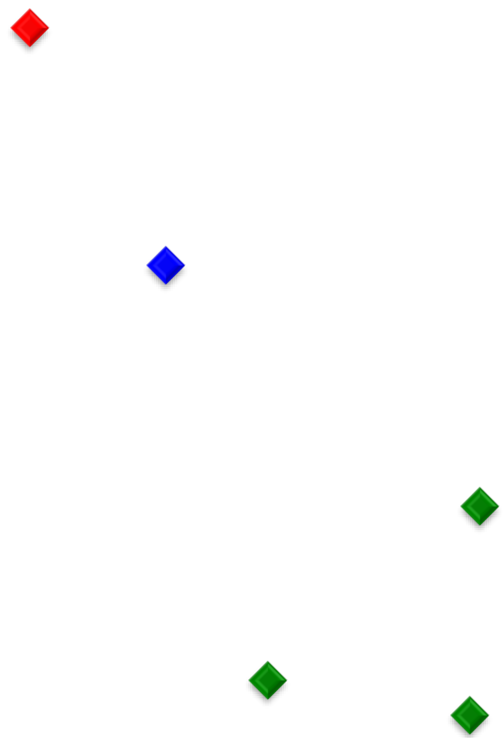
Hierarchical Clustering

- Combine two nearest clusters (Euclidean, Centroid)



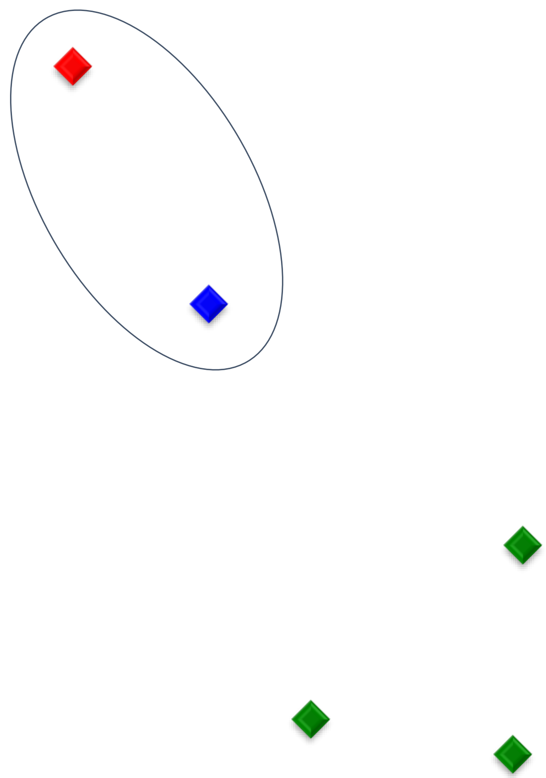
Hierarchical Clustering

- Combine two nearest clusters (Euclidean, Centroid)



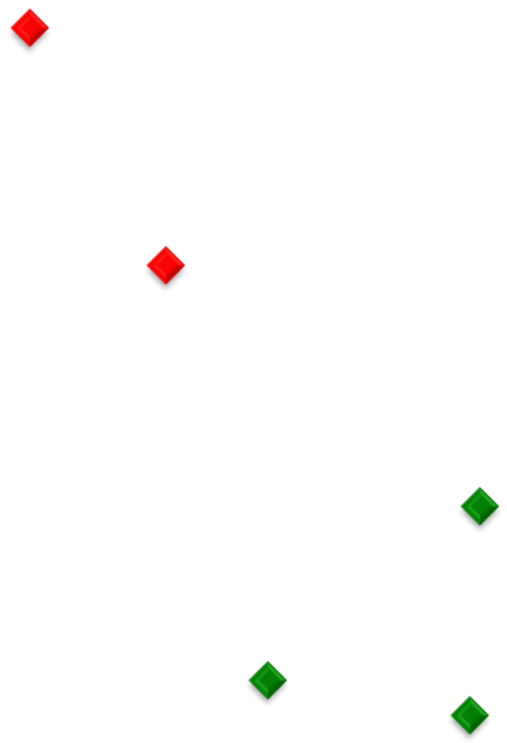
Hierarchical Clustering

- Combine two nearest clusters (Euclidean, Centroid)



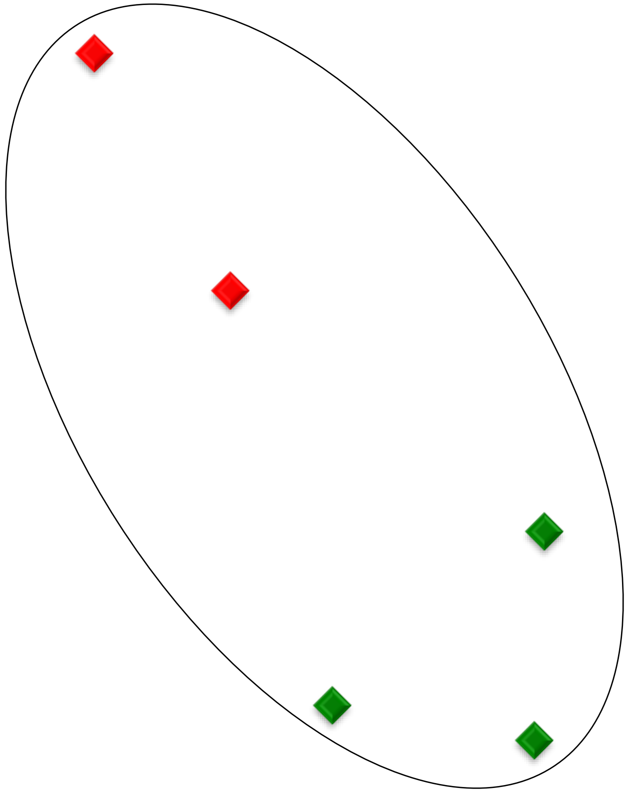
Hierarchical Clustering

- Combine two nearest clusters (Euclidean, Centroid)



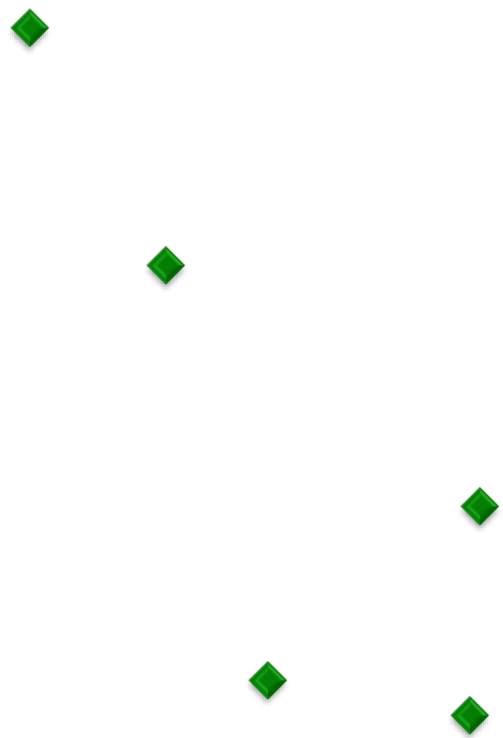
Hierarchical Clustering

- Combine two nearest clusters (Euclidean, Centroid)



Hierarchical Clustering

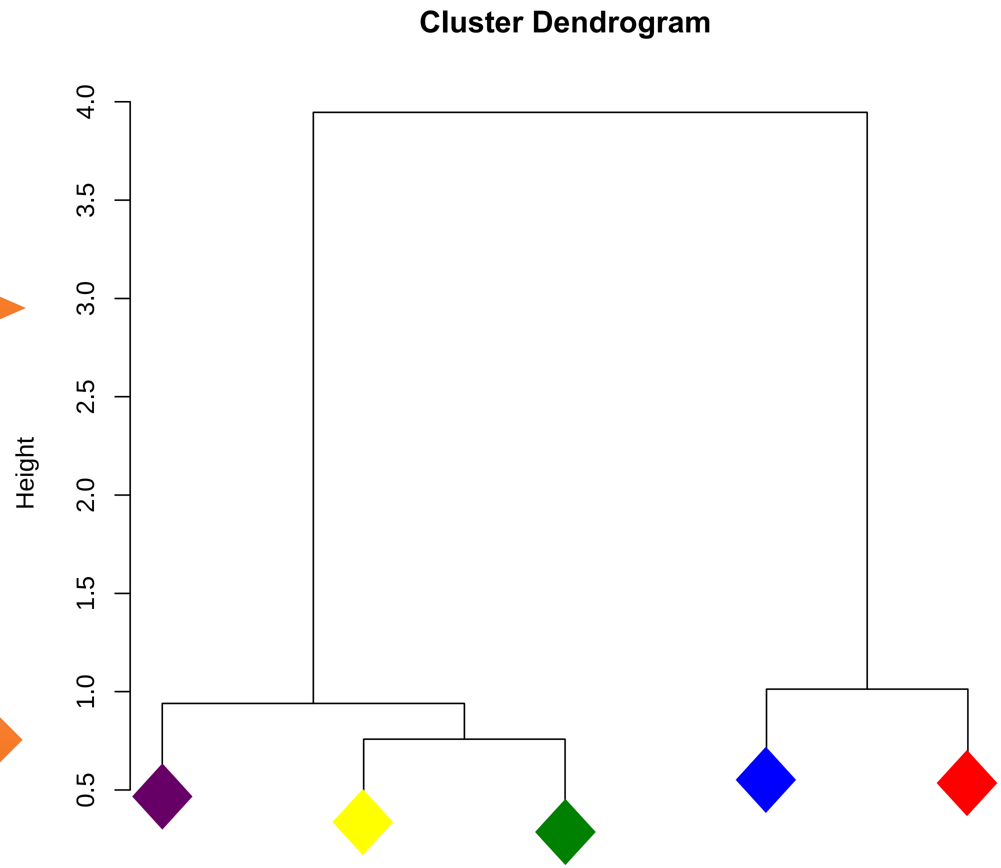
- Combine two nearest clusters (Euclidean, Centroid)



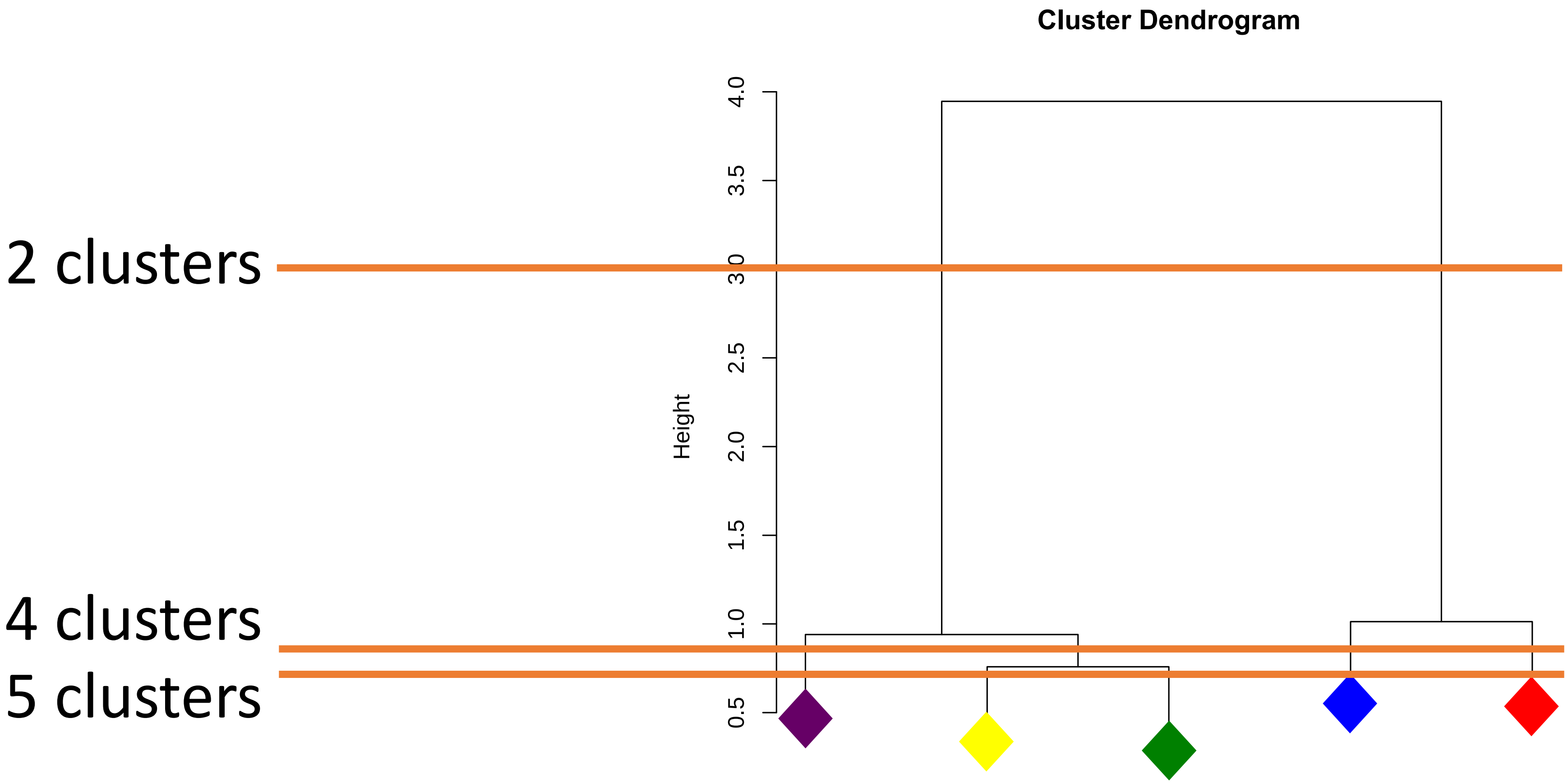
Display Clustering Process

Height of vertical lines represents distance between points or clusters

Data points listed along bottom



Select Clusters



Important Considerations

- Meaningful clusters?
  - Look at statistics (mean, min, max, …) for each cluster and each variable
  - See if the clusters have a feature in common that was not used in the clustering (like an outcome)

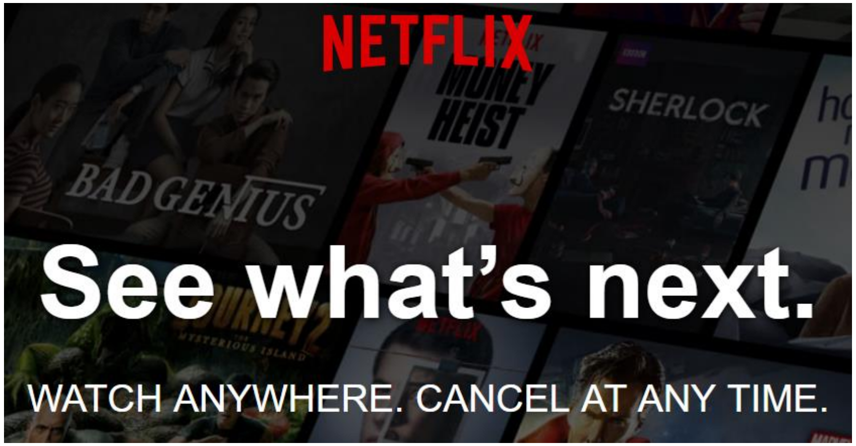
- Drawbacks
  - Need to compute pairwise distances
  - Computational time and memory consuming
  - Prohibitive for large data

Applications

Netflix

Netflix

- Online DVD rental and streaming video service
- More than 139 million subscribers worldwide
- Over 40 countries
- $15.8 billion in revenue



- Key aspect is being able to offer customers accurate movie recommendations based on preferences and viewing history

The Netflix Prize

- From 2006 – 2009 Netflix ran a contest asking the public to submit algorithms to predict user ratings for movies
- Training data set of ~100,000,000 ratings and test data set of ~3,000,000 ratings were provided
- Offered a grand prize of $1,000,000 USD to the team who could beat Netflix’s own algorithm, Cinematch, by more than 10%, measured in RMSE

Contest Rules

- If the grand prize was not yet reached, progress prizes of $50,000 USD per year would be awarded for the best result so far, as long as it had >1% improvement over the previous year.
- Teams must submit code and a description of the algorithm to be awarded any prizes
- If any team met the 10% improvement goal, last call would be issued and 30 days would remain for all teams to submit their best algorithm.

Initial Results

- The contest went live on October 2, 2006
- By October 8, a team submitted an algorithm that beat Cinematch
- By October 15, there were three teams with algorithms beating Cinematch
- One of these solutions beat Cinematch by >1%, qualifying for a progress prize

Progress During the Contest

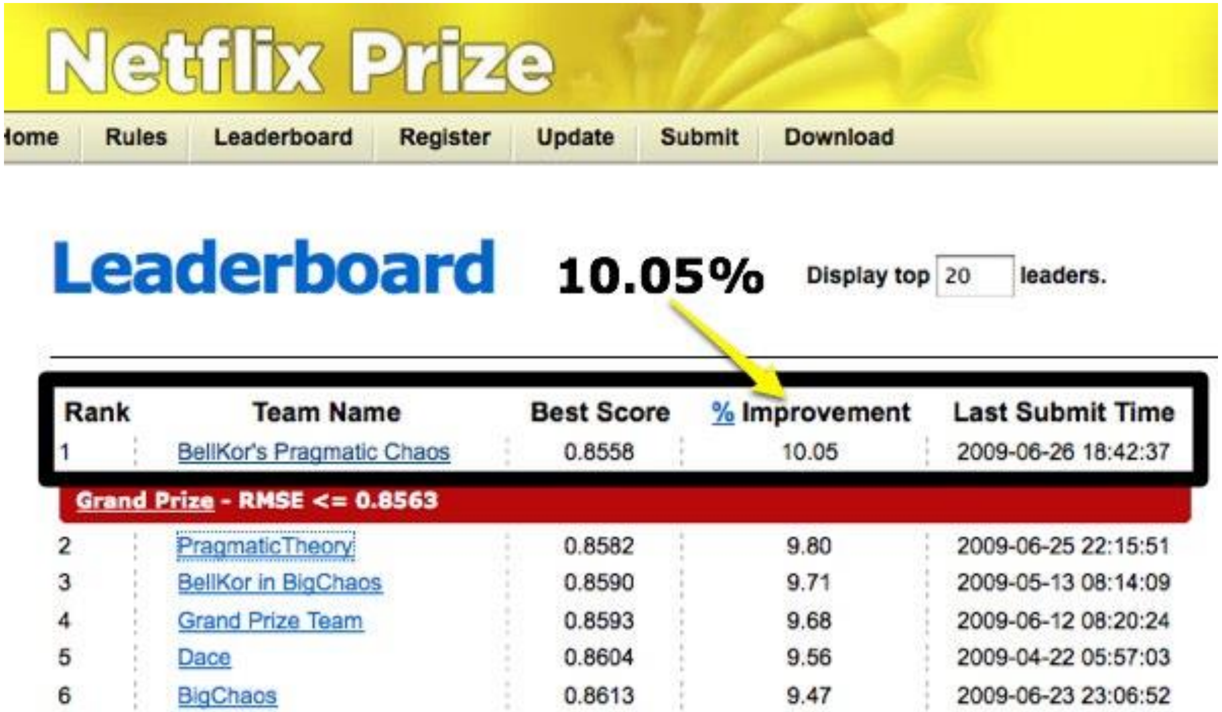
- By June 2007, over 20,000 teams had registered from over 150 countries
- The 2007 progress prize went to team BellKor—formed by researchers from AT&T Labs—with an 8.43% improvement on Cinematch
- In the following year, several teams from across the world joined forces

Competition Intensifies

- The 2008 progress prize went to team BellKor which contained researchers from the original BellKor team as well as the team BigChaos
- This was the last progress prize because another 1% improvement would reach the grand prize goal of 10%

Last Call Announced

- On June 26, 2009, the team BellKor's Pragmatic Chaos submitted a 10.05% improvement over Cinematch



Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	BellKor's Pragmatic Chaos	0.8558	10.05	2009-06-26 18:42:37
Grand Prize - RMSE <= 0.8563				
2	PragmaticTheory	0.8582	9.80	2009-06-25 22:15:51
3	BellKor in BigChaos	0.8590	9.71	2009-05-13 08:14:09
4	Grand Prize Team	0.8593	9.68	2009-06-12 08:20:24
5	Dace	0.8604	9.56	2009-04-22 05:57:03
6	BigChaos	0.8613	9.47	2009-06-23 23:06:52

The Netflix Prize: The Final 30 Days

- 29 days after last call was announced, the team The Ensemble submitted a 10.09% improvement
- When Netflix stopped accepting submissions:
  - BellKor's Pragmatic Chaos - 10.09% improvement
  - The Ensemble - 10.10% improvement
- Netflix would now test the algorithms on a private test set and announce the winners

Winners are Declared!

- On September 18, 2009, a winning team was announced
- BellKor's Pragmatic Chaos won the competition and the $1,000,000 grand prize

Winning the Netflix Prize

Predicting the Best User Ratings

- Netflix was willing to pay over $1M for the best user rating algorithm, which shows how critical the recommendation system was to their business
- What data could be used to predict user ratings?
- Every movie in Netflix's database has the ranking from all users who have ranked that movie
- We also know facts about the movie itself
  - Actors, director, genre classifications, year released, etc.

Using Other Users' Rankings

	Men in Black	Apollo 13	Top Gun	Terminator
Amy	5	4	5	3
Bob	3		2	5
Carl		5	4	2
Dan	4	2		

- Consider suggesting to Carl that he watch "Men in Black", since Amy rated it highly and Carl and Amy seem to have similar preferences
- This technique is called **collaborative filtering**

Using Movie Information

- We saw that Amy liked "Men In Black"
  - It was directed by Barry Sonnenfeld
  - Classified in the genres of action, adventure, sci-fi and comedy
  - It stars actor Will Smith

- Consider recommending to Amy:
  - Barry Sonnenfeld's movie "Get Shorty"
  - "Jurassic Park", which is in the genres of action, adventure, and sci-fi
  - Will Smith's movie "Hitch"

This technique is called **content filtering**

Strengths and Weaknesses

- Collaborative filtering
  - Can accurately suggest complex items without understanding the nature of the items
  - Requires a lot of data about the user to make accurate recommendations
  - Millions of items – need lots of computing power
- Content filtering
  - Requires very little data to get started
  - Can be limited in scope

MovieLens Data

- www.movielens.org is a movie recommendation website run by the GroupLens Research Lab at the University of Minnesota
- They collect user preferences about movies and do collaborative filtering to make recommendations
- We will use their movie database to do content filtering using **clustering**

MovieLens Dataset

- Movies in our dataset are categorized as belonging to different genres

• Action	• Adventure	• Animation	• Children’s	• Comedy
• Crime	• Documentary	• Drama	• Fantasy	• Film Noir
• Horror	• Musical	• Mystery	• Romance	• Sci-Fi
• Thriller	• War	• Western		

- Each movie may belong to many genres
- Can we systematically find groups of movies with similar sets of genres?

Distance Example

- The movie “Toy Story” is categorized as Adventure, Animation, Children’s, Comedy, and Fantasy:
  - Toy Story: (0,1,1,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0)

- The movie “Batman Forever” is categorized as Action, Adventure, Comedy, and Crime
  - Batman Forever: (1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0)

Distance Between Points

- Toy Story: (0,1,1,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0)
- Batman Forever: (1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0)

$$d = \sqrt{(0-1)^2 + (1-1)^2 + \cdots + (0-0)^2} = \sqrt{5}$$

- In this application, can be interpreted as the square root of the number of genres in which they differ

Let's get our hands dirty!

Data Analysis

Let's do some basic data analysis using our WHO data.

Hide

```
WHO$Under15
```

```
  [1] 47.42 21.33 27.42 15.20 47.58 25.96 24.42 20.34 18.95 14.51 22.25 21.62 20.16 30.57 18.99 15.10 16.88 34.4
0 42.95 28.53
 [21] 35.23 16.35 33.75 24.56 25.75 13.53 45.66 44.20 31.23 43.08 16.37 30.17 40.07 48.52 21.38 17.95 28.03 42.1
7 42.37 30.61
 [41] 23.94 41.48 14.98 16.58 17.16 14.56 21.98 45.11 17.66 33.72 25.96 30.53 30.29 31.25 30.62 38.95 43.10 15.6
9 43.29 28.88
 [61] 16.42 18.26 38.49 45.90 17.62 13.17 38.59 14.60 36.96 40.00 42.46 41.55 36.77 35.35 35.72 14.62 20.71 29.4
3 29.27 23.68
 [81] 40.51 21.54 27.53 14.04 27.78 13.12 34.13 25.46 42.37 30.10 24.90 30.21 35.61 14.57 21.64 36.75 43.06 29.4
5 15.13 17.46
[101] 42.72 45.44 26.65 29.03 47.14 14.98 30.10 40.22 20.17 29.02 35.81 18.26 27.05 19.01 27.85 45.38 25.28 36.5
9 30.10 35.58
[121] 17.21 20.26 33.37 49.99 44.23 30.61 18.64 24.19 34.31 30.10 28.65 38.37 32.78 29.18 34.53 14.91 14.92 13.2
8 15.25 16.52
[141] 15.05 15.45 43.56 25.96 24.31 25.70 37.88 14.04 41.60 29.69 41.54 16.45 21.95 41.74 16.48 15.00 14.16 40.3
7 47.35 29.53
[161] 42.28 15.20 25.15 41.48 27.83 30.05 16.71 14.79 35.35 35.75 18.47 16.89 46.33 41.89 37.33 20.73 23.22 26.0
0 28.65 30.61
[181] 48.54 14.18 14.41 17.54 44.85 19.63 22.05 28.90 37.37 28.84 22.87 40.72 46.73 40.24
```

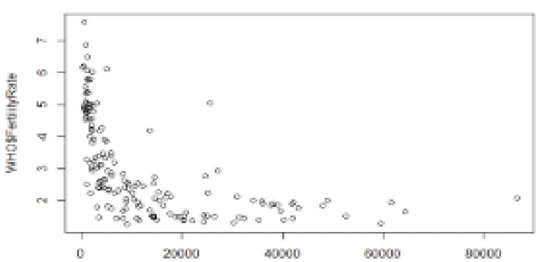
```
WHO$Country[which.min(WHO$Under15)]
```

```
[1] Japan
194 Levels: Afghanistan Albania Algeria Andorra Angola Antigua and Barbuda Argentina Armenia Australia Austria
... Zimbabwe
```

Let's create some plots for exploratory data analysis (EDA). First, Let's create a basic scatterplot of GNI versus FertilityRate.

Hide

```
plot(WHO$GNI, WHO$FertilityRate)
```



Beyond Movies: Mass Personalization

- "If I have 3 million customers on the web, I should have 3 million stores on the web"

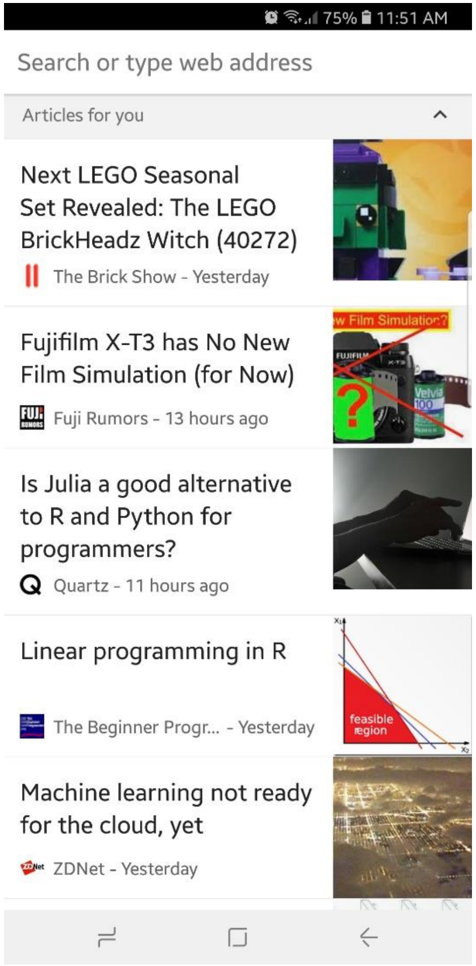
  – Jeff Bezos, CEO of Amazon.com

- Recommendation systems build models about users' preferences to personalize the user experience
- Help users find items they might not have searched for:
  - A new favourite band
  - An old friend who uses the same social media network
  - A book or song they are likely to enjoy

Cornerstone of these Top Businesses

Recommendation Systems at Work

- In today's digital age, businesses often have hundreds of thousands of items to offer their customers
- Excellent recommendation systems can make or break these businesses
- Clustering algorithms, which are tailored to find similar customers or similar items, form the backbone of many of these recommendation systems



Lecture 9 Wrap-up

- ✓ Unsupervised Learning
- ✓ Clustering
  - ✓ Hierarchical clustering
  - ✓ k-means clustering
- ✓ Applications: Netflix

Next Lecture

- Supervised learning
  - Linear regression
  - Logistic regression
  - SVM and kernel
  - Tree models
- Unsupervised learning
  - Clustering
  - PCA (Dimension Reduction)
  - EM
- Deep learning
  - Neural networks
  - Convolutional NN
  - Recurrent NN
- Reinforcement learning
  - MDP
  - ADP
  - Deep Q-Network

Questions?

Shan Wang (王杉)
https://wangshan731.github.io/