Revise

Course structure

W1. Data Processing with Python

W2. Data Exploration with Python

W3. Data Modeling with Pytyhon

W4. Data Analytics for Timeseries

Student vacations

W5. Holiday

W6-7. Data Analytics for Texts

W8. Data Analytics for Images

W9. Data Analytics for Graphs

W10-11. Data Analytics for Other Data

W12. **Revision**

W1 Recap: Data Science pipeline

1. Ask an interesting question:

- What is the goal?
- What would you do if you had all the data?
- What do you want to predict or estimate?

2. Get the data:

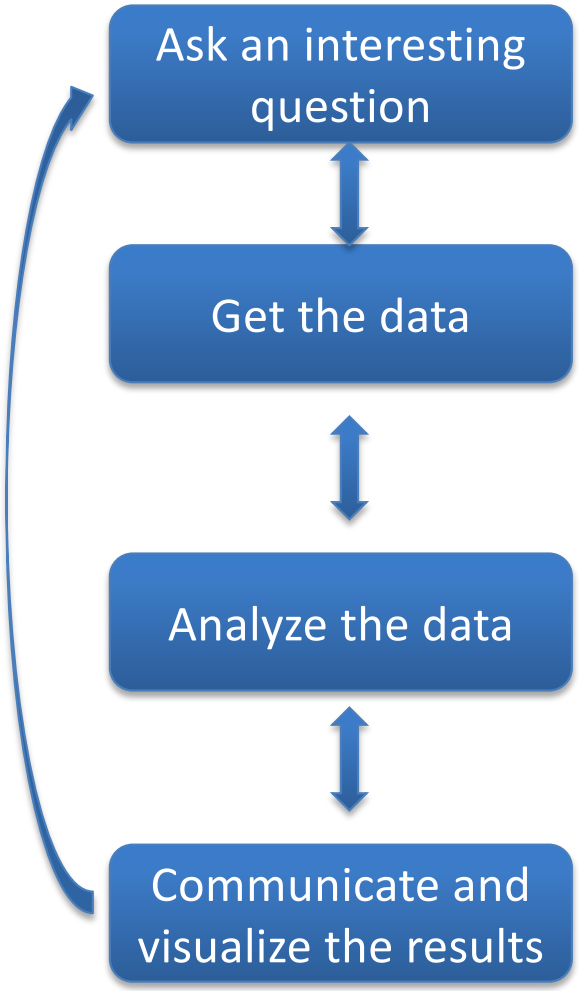
- How were the data sampled?
- Which data are relevant?
- Are there privacy issues?

3. Analyze the data:

- Are there anomalies?
- Are there patterns?
- Are there trends?

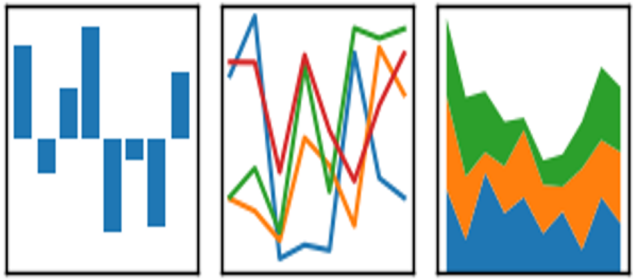
4. Communicate and visualize the results

- What did we learn?
- Do the results make sense?
- Can we tell a story?



W1 Recap: Python for Data Analytics

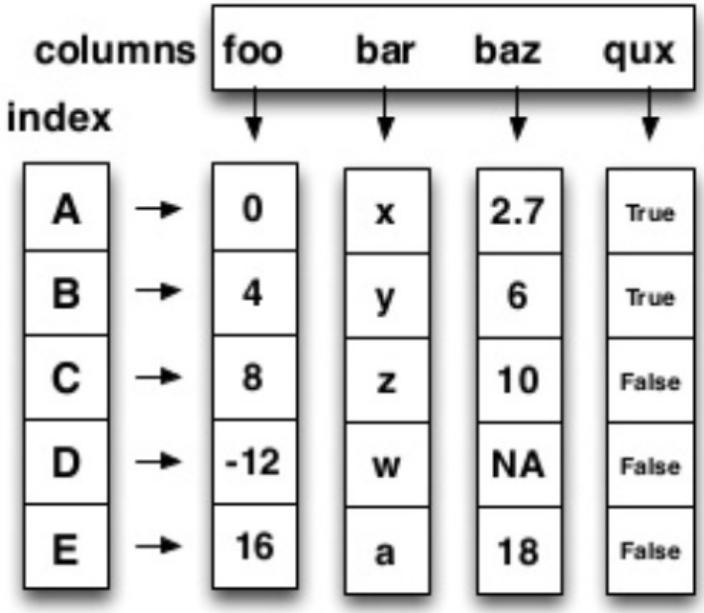
https://seaborn.pydata.org/



- Numpy: great for handling numbers, vectors, matrices
- Scipy: great for numerical optimizations
- Pandas: great for handling tabular/relational data
- Scikit Learn: great for data analytics techniques

W1 Recap: Data Storage with Pandas

- DataFrame: a table with named columns (like in the relational model)
  - Represented as a dictionary
    - columnName -> series
  - Each Series object represents a column
  - Each column can have a different type
  - Row and column indices
  - Size mutable: insert and delete columns



- Why Use DataFrames?
  - better for series with multiples attributes of different types.
  - Easy and efficient search elements by index.

W1 Recap: Clean “dirty” data

- Bad formats
- Missing data
- Erroneous data
- Irrelevant data
- Inconsistent data
- Malicious data
- Outliers

Week 2 Recap: Exploring Two Variables

		Response	
		Categorical	**Quantitative**
Explanatory	**Categorical**	**C→C**	**C→Q**
	Quantitative	**Q→C**	**Q→Q**

- C→Q: use box-and-whisker plots
- C→C: use two-way tables
- Q→Q: use scatter-plot
- Q→C: use other methods (e.g. statistic tests)

Week 2 Recap: From Data Visualization to Visual Analytics

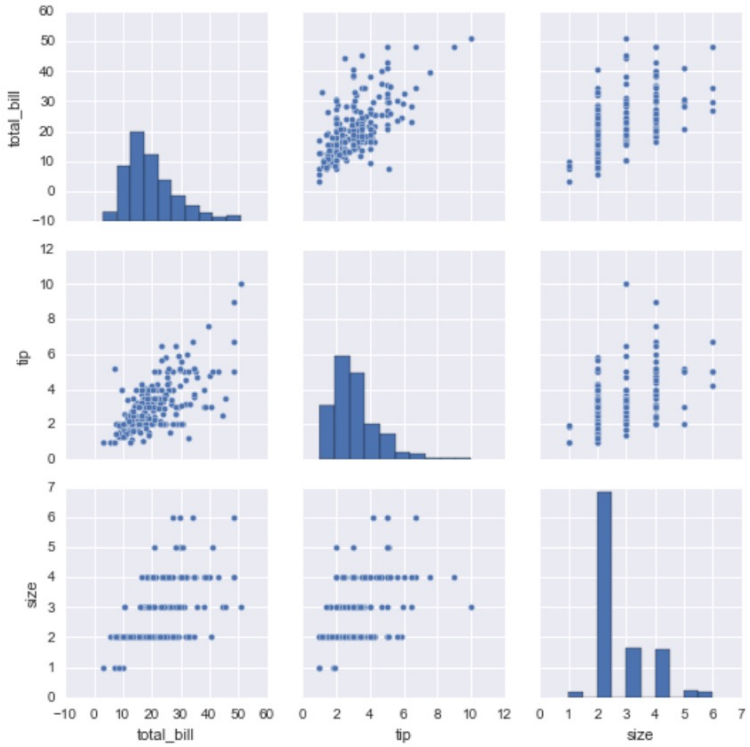
- Interactive viz = Old-fashioned viz + Interaction Scheme
  - Enable **visual analytics** via interactive and reproducible results
  - Easy and fast to develop and customize

- Old-fashioned viz
  - Great for data exploration, developed throughout the last few centuries
  - Rapid data exploration
  - Focus on most important details

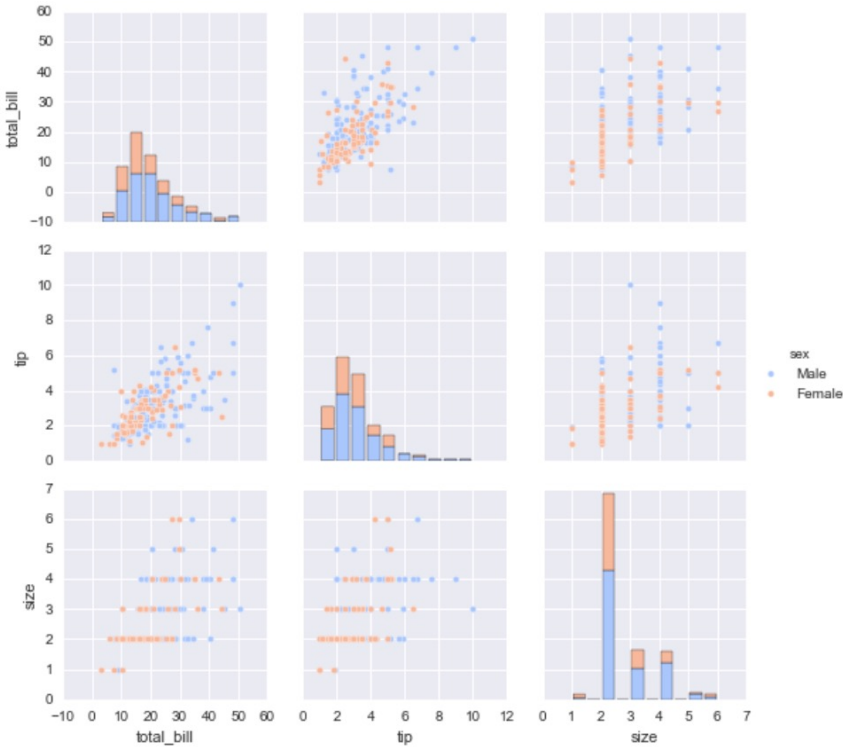
- Interactive viz
  - More and more common nowadays. New frameworks are the key enabler.
  - Support multiple analyses
  - Focus on more dimensions

Week 2 Recap: Exploratory Data Analysis with Python

- Pairplot automatically analyze pairwise relationships



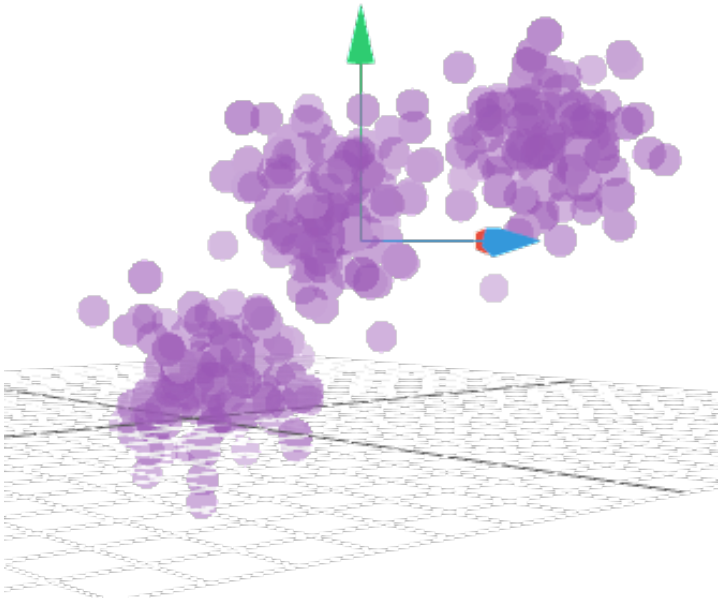
sns.pairplot(tips)

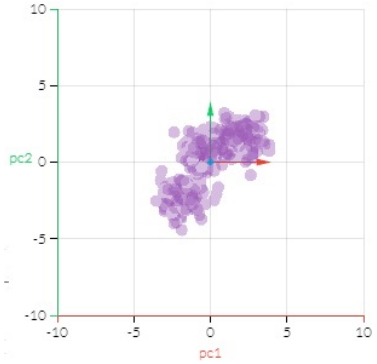


sns.pairplot(tips,hue='sex',palette='coolwarm')

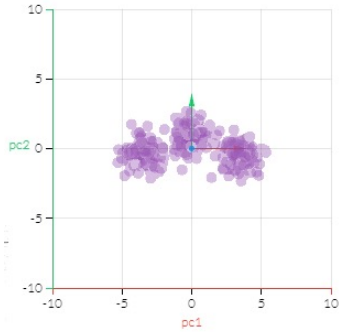
Week 2 Recap: Dimensionality Reduction

- http://setosa.io/ev/principal-component-analysis/



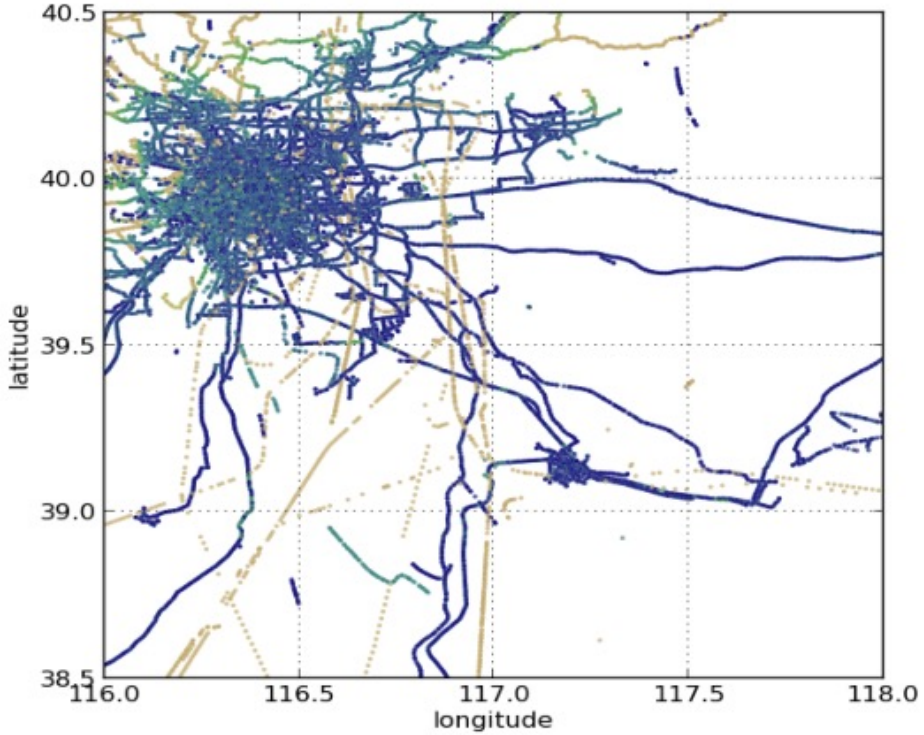


Bad feature reduction

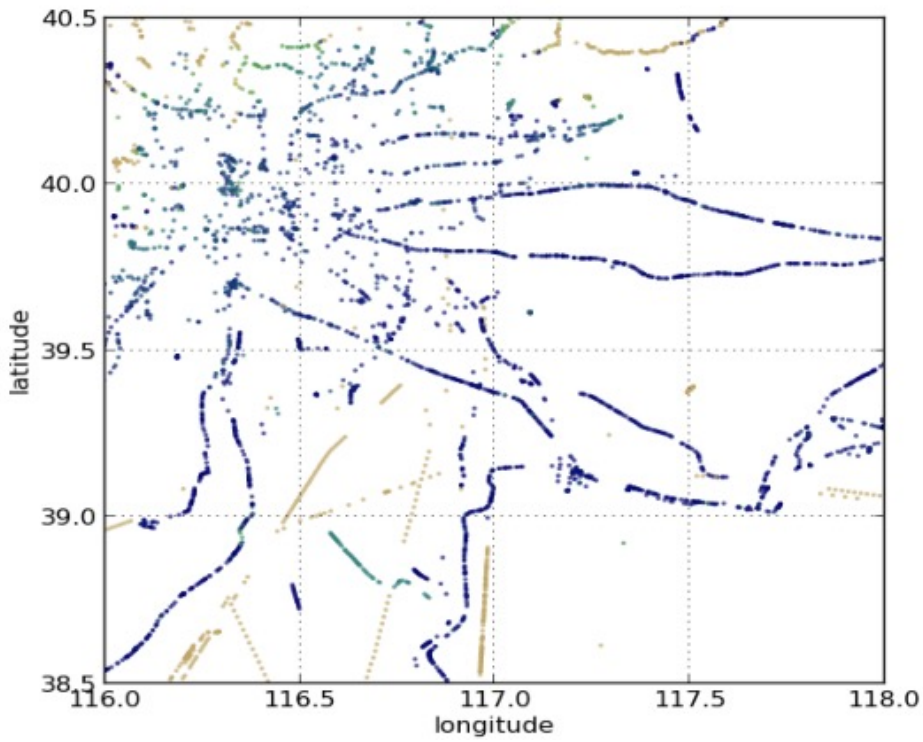


Good feature reduction

Week 2 Recap: Data Sampling



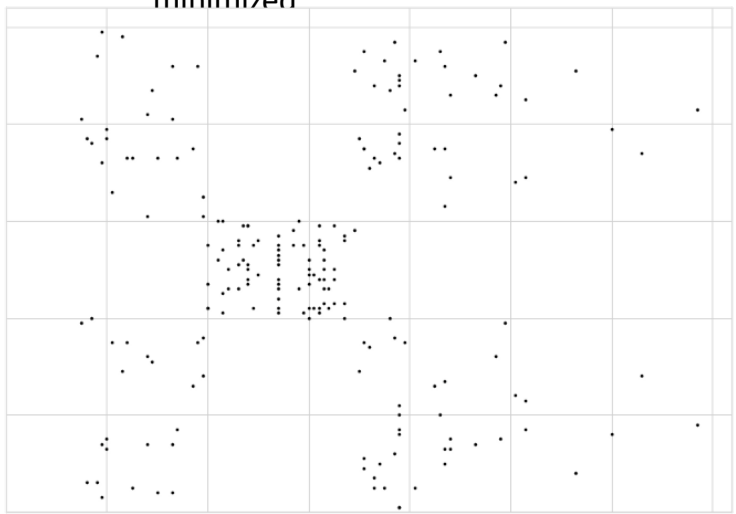
original data



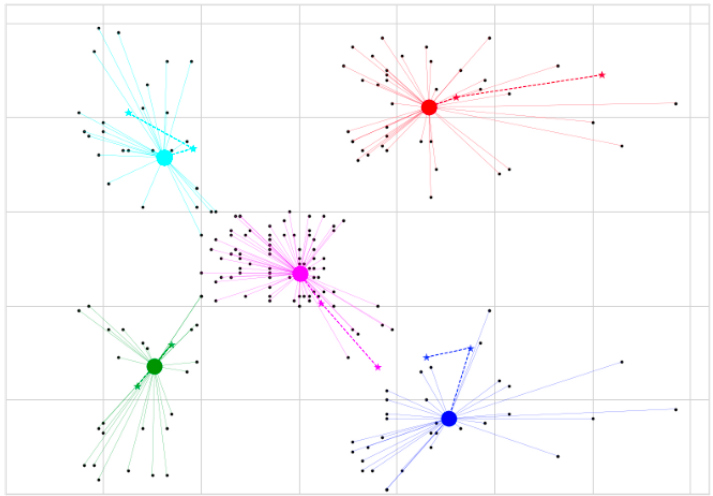
a good sample

Week 3 Recap: K-means clustering

- A simple greedy algorithm (usually called Lloyd's algorithm):
  - Divide data into K clusters, each of which has a center (centroid)
  - Each data point belongs to a single cluster only
  - K centroids are chosen such that distance (usually Euclidean) from data points to their centroids is locally minimized



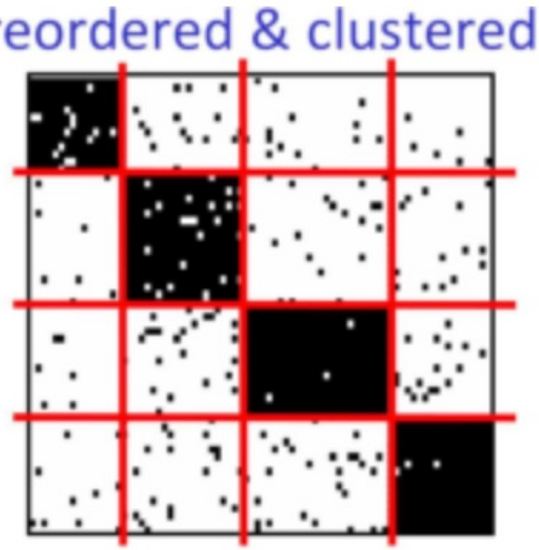
Input



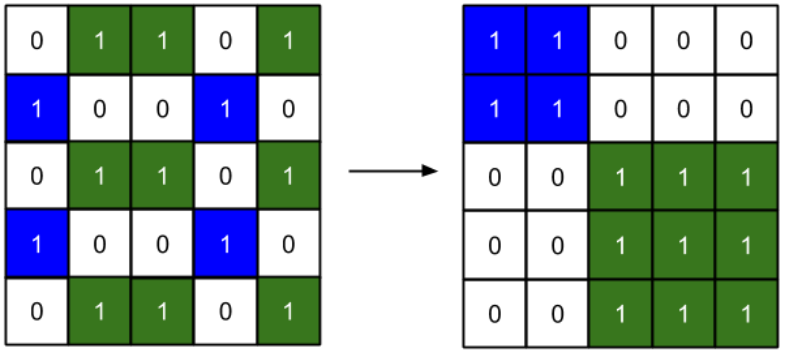
Output

Week 3 Recap: Co-clustering

- Clustering two variables simultaneously



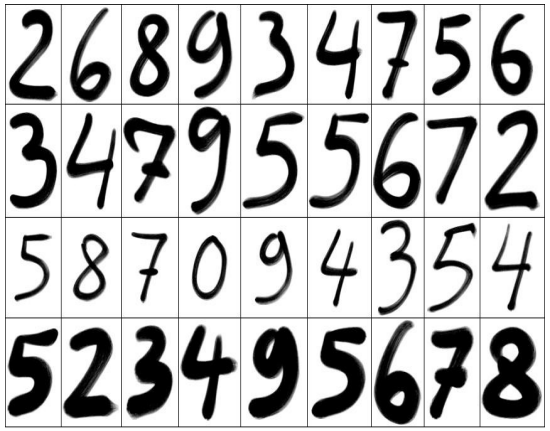
A simple case: similarity values are only 1s and 0s.



Vlachos, Michail et al. "Improving Co-Cluster Quality with Application to Product Recommendations." CIKM (2014).

Week 3 Recap: Classification

- Handwritten digit recognition:



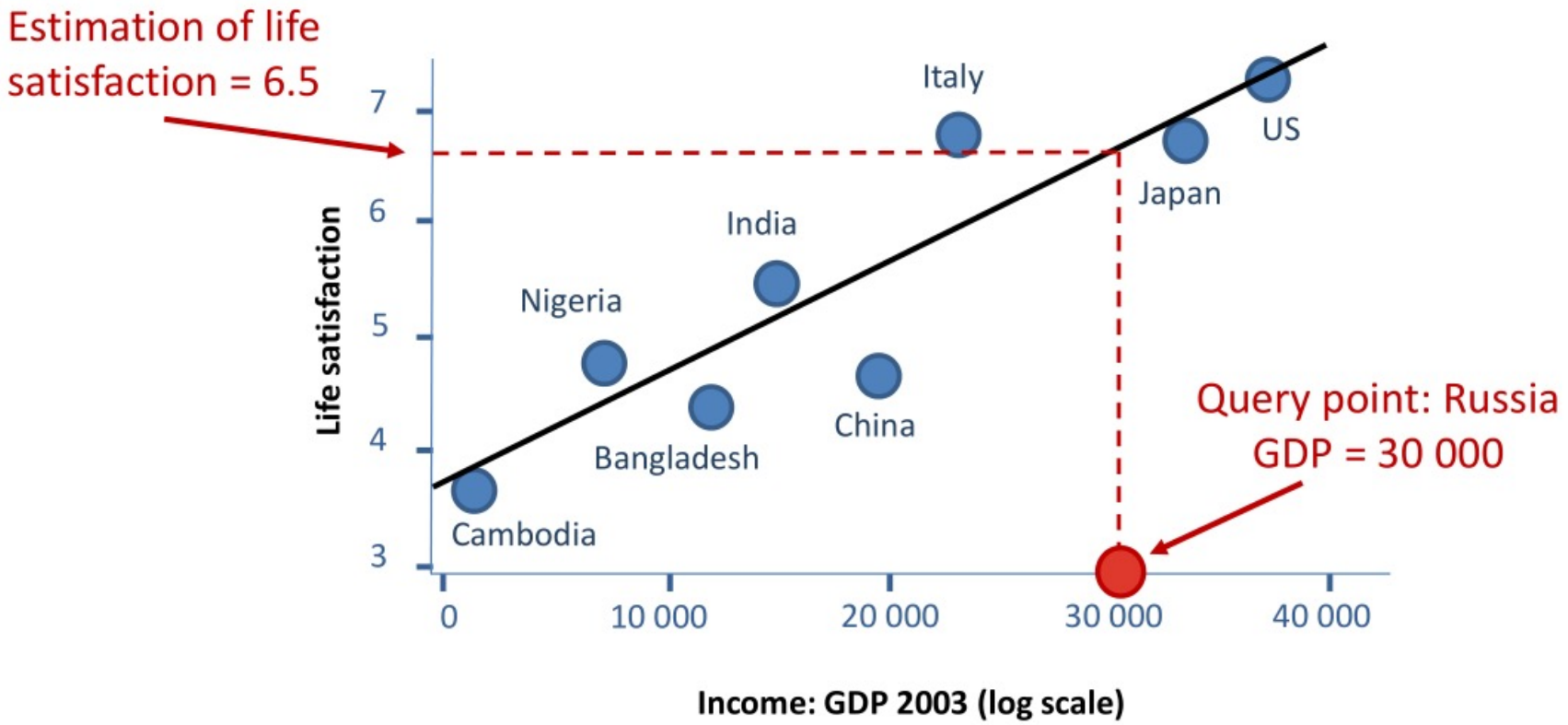
Classifier

2	6	8	9	3	4	7	5	6
3	4	7	9	5	5	6	7	2
5	8	7	0	9	4	3	5	4
5	2	3	4	9	5	6	7	8

- Email spam recognition:

Week 3 Recap: Regression

- Maps N-dimensional input $x \in R^N$ to **continuous** values $y \in R$
- e.g. x = [Income (GDP)], y = Continuous value of life satisfaction

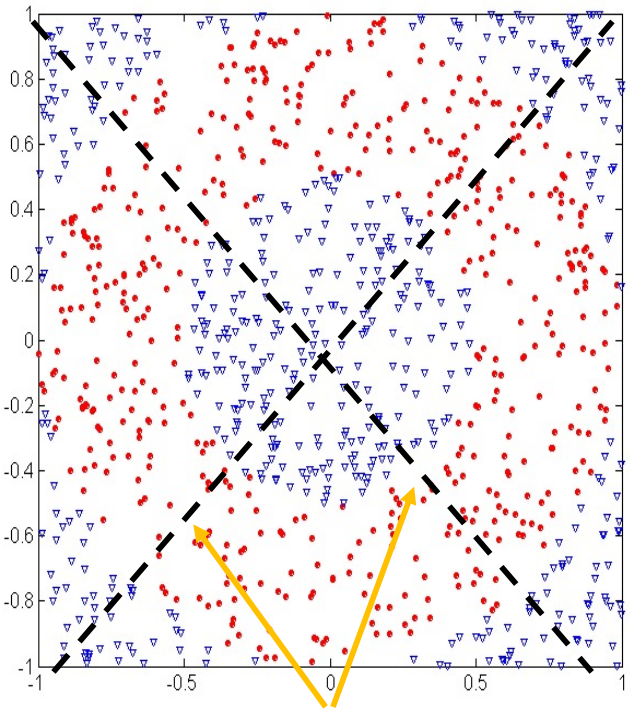


Week 3 Recap: Model Evaluation

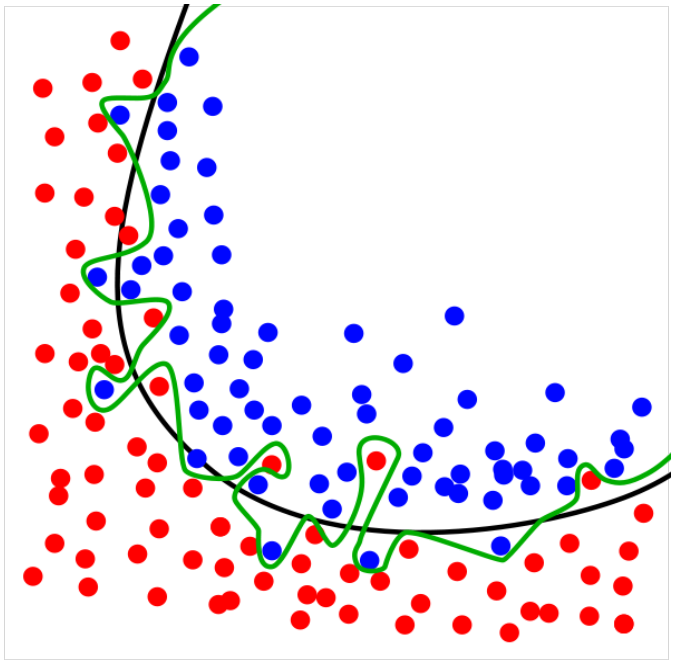
Underfitting

E.g. Regression

Overfitting



Models are too simple!



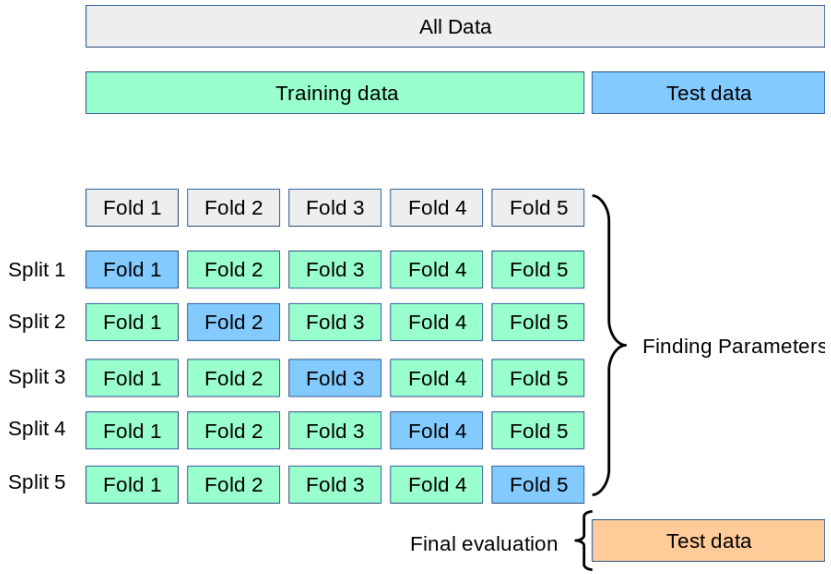
- Green line is overfitting (tailored too much to the training data)
- Black line is what we want

Low variance, high bias

High variance, low bias

Week 3 Recap: K-Fold Cross validation

- Prevent you from over-fitting a single train/test split
- General process:
  - Split your training data into K randomly-assigned folds
  - Reserve one segment as your validation data
  - Train on each of the remaining K-1 folds to find model parameters
  - Take the best parameters based on validation accuracy
- You can do the same process for different train/test splits and take the average result

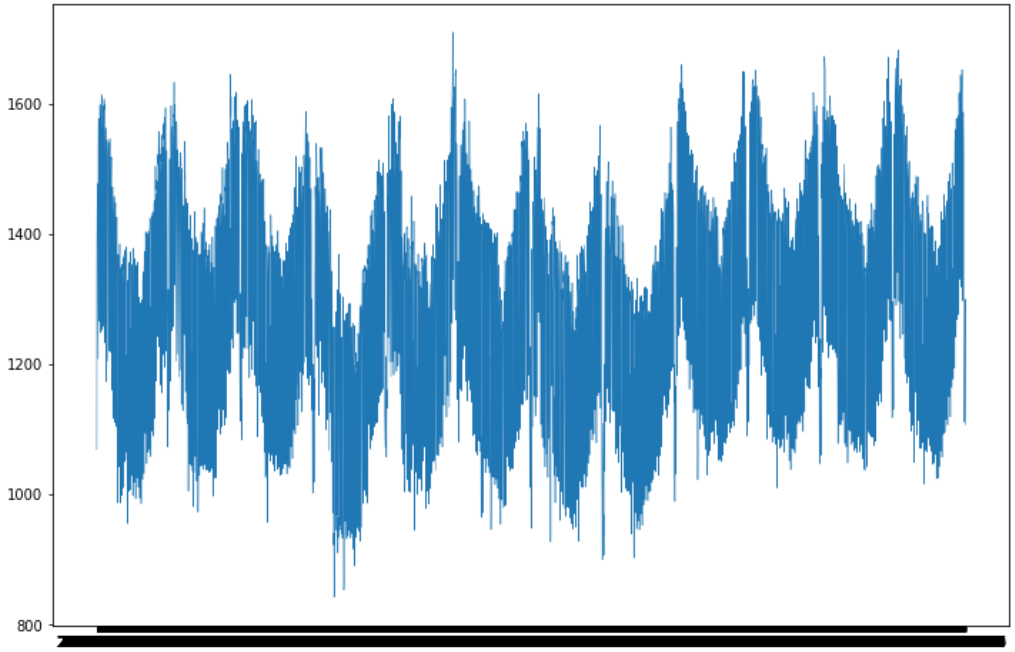


Week 4 Recap: Time Series Analysis

- **Exploratory data analysis:** examine a time series with a line chart or statistics.
- **Segmentation:** Splitting a time-series into a sequence of segments. Represent a time-series as a sequence of individual segments, each with its own characteristic properties.
- **Classification:** Assigning time series pattern to a specific category → Natural phenomena, astronomical phenomena, animal movement.
- **Forecasting:** is the use of a model to predict future values based on previously observed values.
- **Decomposition:** deals with complex timeseries to help forecasting easier
- **Anomaly Detection:** Finding outlier data points relative to some standard or usual pattern.

Recap: EDA of Time Series

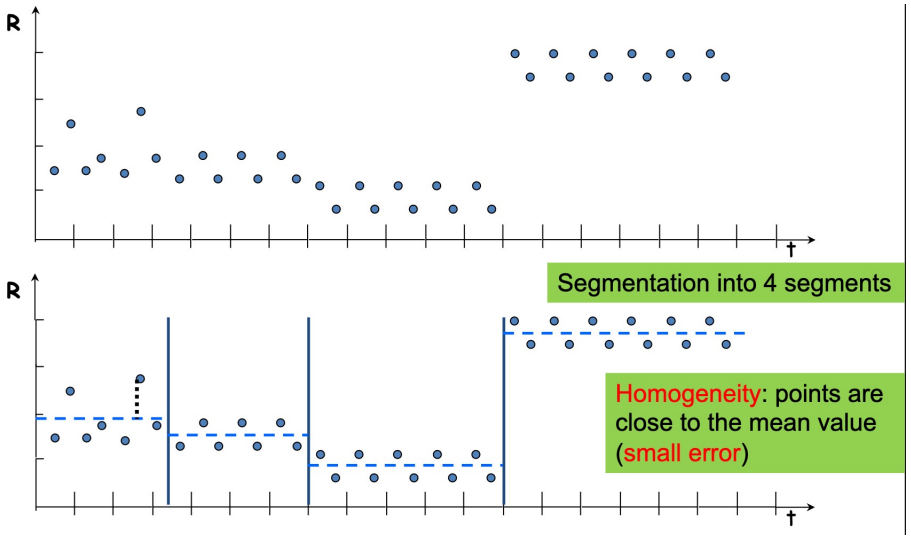
- Examine a time-series with statistics:
  - **Pros:** Summarize the values
  - **Cons:** do not consider the timestamps



count	4383.000000
mean	1338.675836
std	165.775710
min	842.395000
25%	1217.859000
50%	1367.123000
75%	1457.761000
max	1709.568000

Recap: Time Series Segmentation

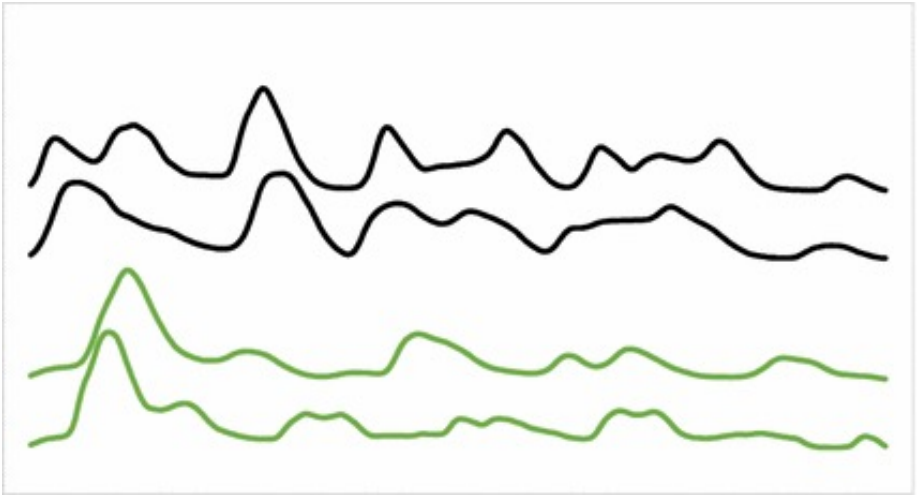
- **Goal:** discover structure in the time series and provide a concise summary
  - Useful for really long time series
  - Divide and conquer: Make it easier to analyze
- **How:** given a time series S, segment it into K disjoint segments (or partitions) that are as homogeneous as possible
  - Data points in the same segment are ``similar''
  - Similar to clustering but only allow grouping along the time dimension



Recap: Time Series Classification

- Assigning time series pattern to a specific category.
  - Given a time series $\mathrm{Y} = (y_1, y_2, \dots, y_n)$ and a list of categories $C = (c_1, c_2, \dots, c_k)$, we want to assign Y to it the best matching category c_i.
  - Needs a similarity/distance measure.

- **Applications**:
  - Identify a word based on series of hand movements in sign language.
  - Handwriting classification.
  - Moving pattern similarity.



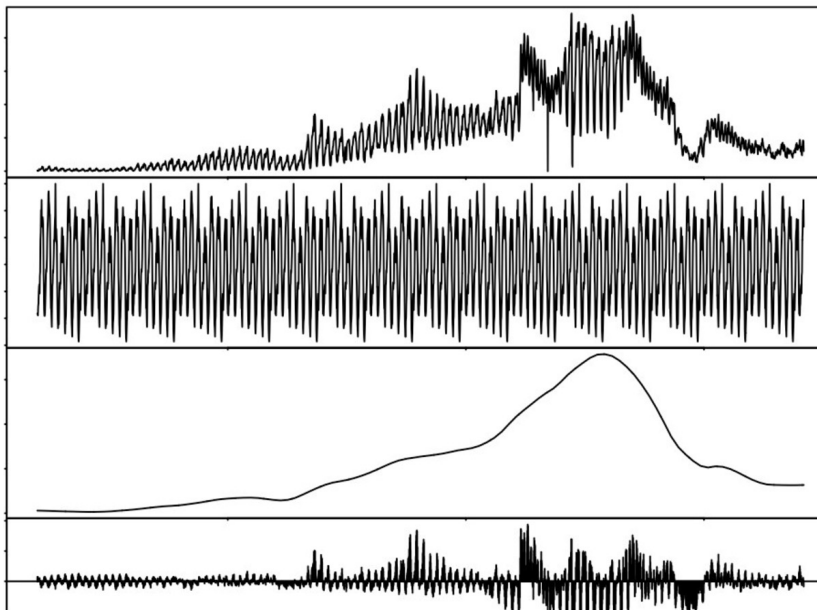
Recap: Forecasting with Seasonality

- Step 1. Calculate the average demand for each year
- Step 2. Calculate seasonal indexes
- Step 3. Average the indexes
- Step 4. Forecast demand for the next year
- Step 5. Multiple next year's average seasonal demand by each average seasonal index

Quarter	Year 1	Seasonal Index	Year 2	Seasonal Index	Avg. Index	Year3
Fall	24000	**1.2**	26000	**1.24**	**1.22**	**26840**
Winter	23000	**1.15**	22000	**1.05**	…	**…**
Spring	19000	**0.95**	19000	…		
Summer	14000	**0.7**	17000	…		
Average	**20000**		**21000**			**22000**

Recap: Time Series Decomposition

- Sometimes, a time series is too complex for segmentation, classification, or forecasting
  - It is better to understand short-term, long-term and recurring patterns first
  - **Approach**: decompose a time series into several components, each representing one of the underlying patterns.



Original time series =

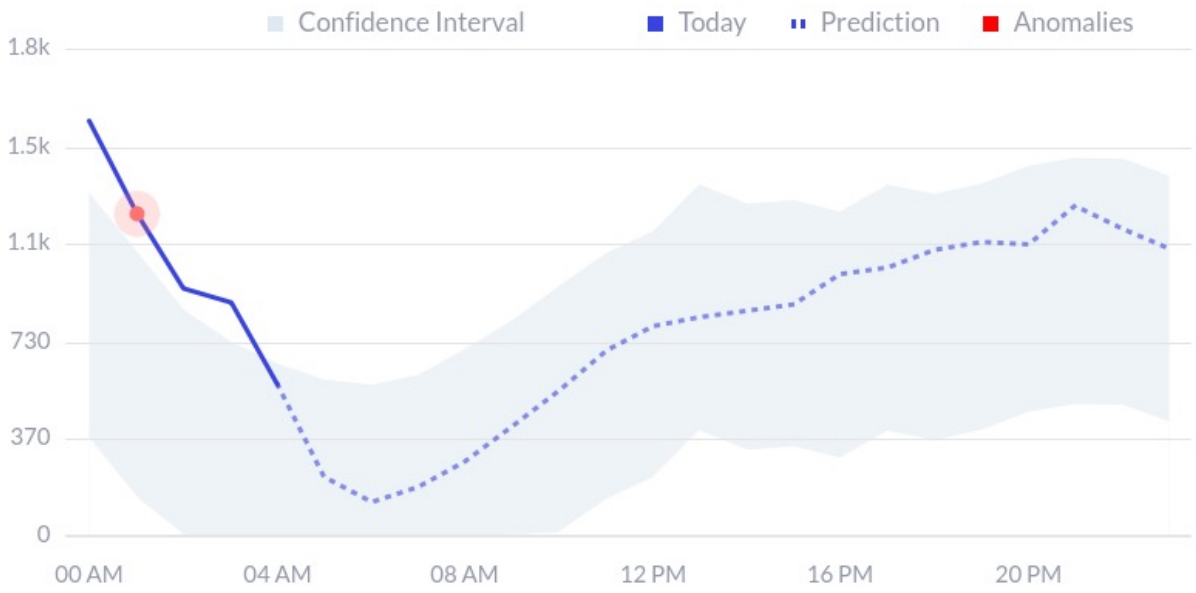
Seasonality component +

Trend component +

Residue component

Recap: Time Series Anomaly Detection

- Finding **outlier** data points relative to some standard or usual pattern.
  - Such as unexpected spikes, drops, trend changes and level shifts.
  - Basically, an anomaly detection algorithm should either **label** each time point with *anomaly/not anomaly,* or **forecast** a signal for some point and test if this point value varies from the forecasted enough to deem it as an anomaly.



Examples:

- Growth of users in a short period of time that looks like a spike.
- When your server goes down and you see zero or a really low number of users for some short period of time.

Week 5+6 Recap:
Text Analytics Application

Social Media Analytics

1. **Brand Reputation Monitoring**
   - Social Media, Blogs, News sites
2. **Advertising Performance Metrics**
   - Social Media, Blogs

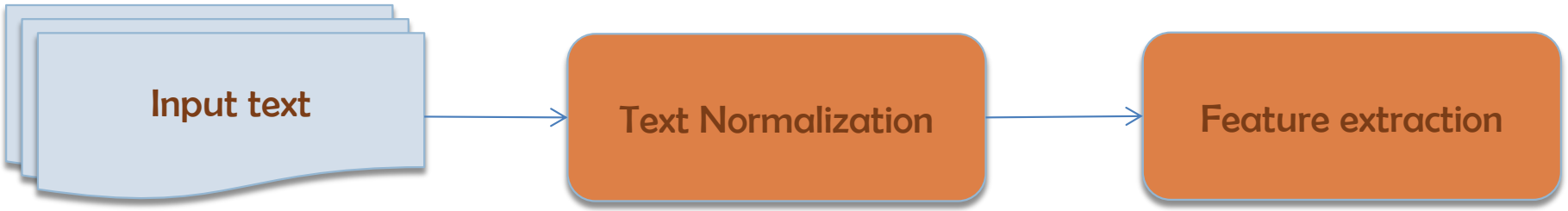
1. **Complaint Tracking**
   - Social Media, Blogs, News
2. **Call Center Analytics**
   - Call Center Transcripts
3. **Competitive Analysis**
   - Communication, Surveys
4. **Market Research**
   - Surveys, Feedback

CRM

Predictive Analytics

1. **Prediction of Stocks**
   - Financial News, Newspapers
2. **Prediction of Election Results**
   - Social Media
3. **Movie Intake**
   - Twitter

Recap: Text Syntactical Analysis



Strings | Bag-of-words model | Vector representation

Example:

Document 1

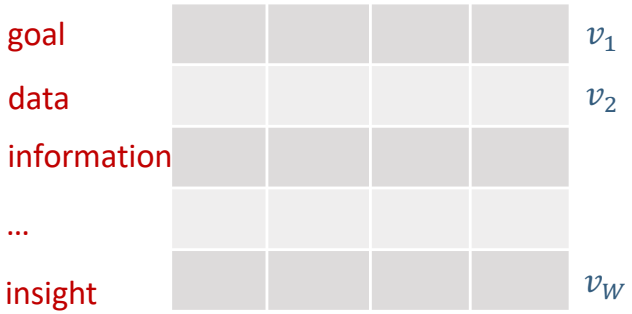
"The goal is to turn data into information, and information into insight"

Carly Fiorina

Document 1

"The goal is to turn data into information, and information into insight"

Carly Fiorina



Recap: TF-IDF example

term frequency (tf)

Terms	goal	data	information	insight	you
Doc1	1	1	2	1	0
Doc2	0	2	2	0	1

Document 1

"The goal is to turn data into information, and information into insight"

Carly Fiorina

Document 2

"You can have data without information, but you cannot have information without data."

Daniel Keys Moran

document frequency (df)

Terms	goal	data	information	insight	you
df	1	2	2	1	1

inverse document frequency (idf)

Terms	goal	data	information	insight	you
idf	0.69	0	0	0.69	0.69

$\log\frac{2}{1}$ $\log\frac{2}{2}$

tfidf

Terms	goal	data	information	insight	you
Doc1	0.69	0	0	0.69	0
Doc2	0	0	0	0	0.69

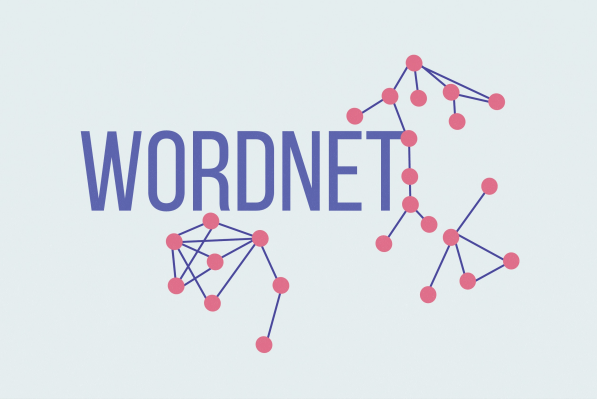
$\frac{0.69}{\sqrt{0.69^2 + 0.69^2}}$

tfidf (l2 normalized)

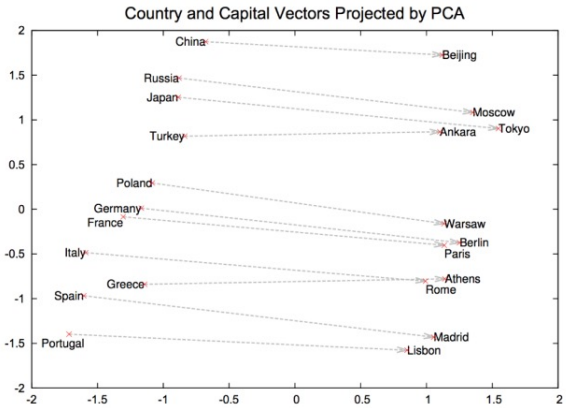
Terms	goal	data	information	insight	you
Doc1	0.71	0	0	0.71	0
Doc2	0	0	0	0	0.69

Recap: Text Representation Learning

- Words in a document are not independent, but stand in a semantic relation to one another.



- Word embedding: neural embedding and vector representation of words
  - Similar words will stay closer
  - State-of-the-art: word2vec



word2vec

[Mikolov et al. 2013]

Recap: Sentiment Analysis

- Computational study of opinions, sentiments, evaluations, attitudes, appraisal, affects, views, emotions, subjectivity, etc., expressed in text.
  - E.g. extract from text how people feel about different products (Reviews, blogs, discussions, news, comments, feedback, …)
- Is a review positive or negative toward the movie?


  - "Unbelievably disappointing"

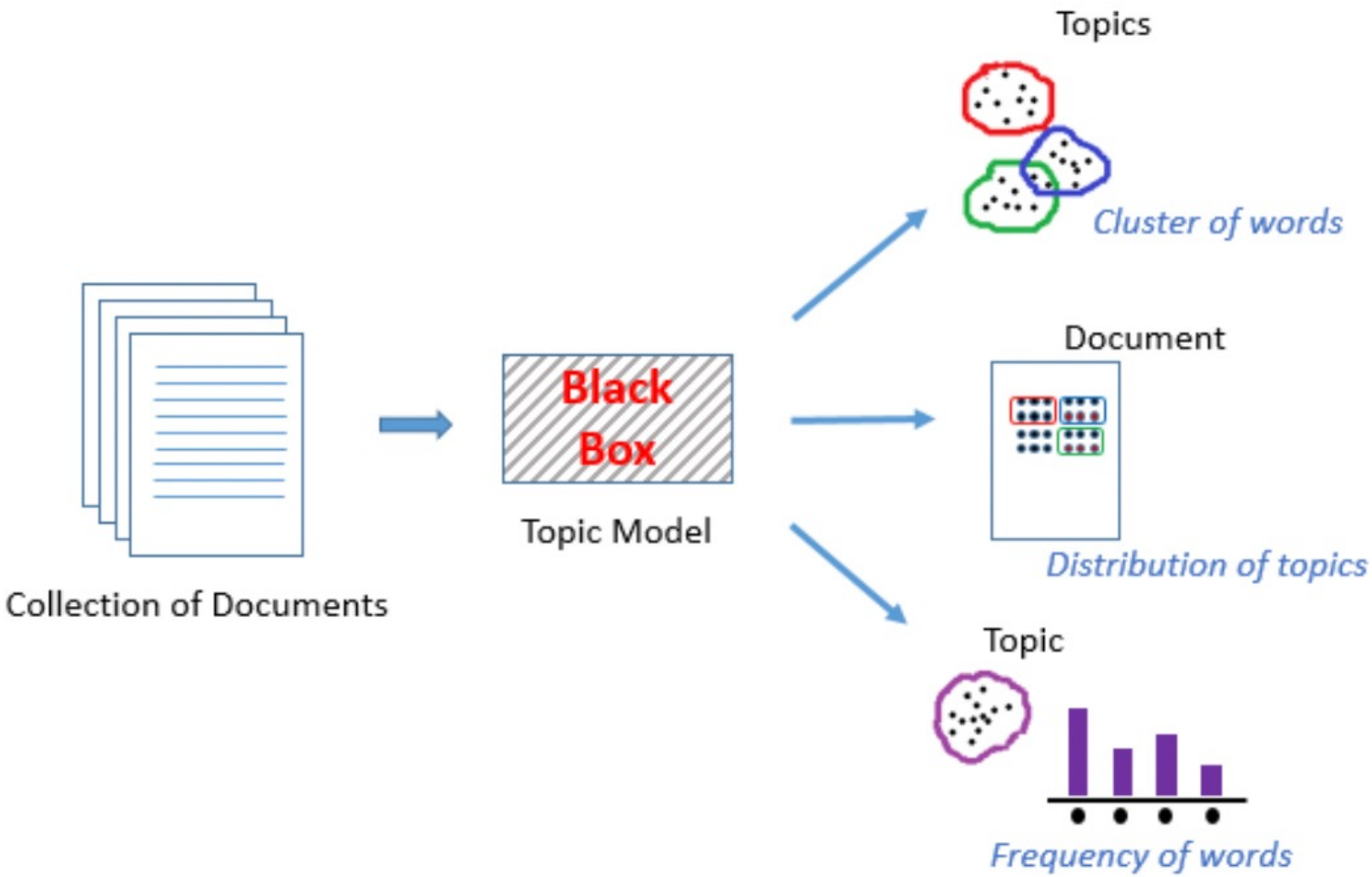

  - "Full of zany characters and richly applied satire, and some great plot twists"


  - "This is the greatest screwball comedy ever filmed"


  - "It was pathetic. The worst part about it was the boxing scenes"

http://nlp.stanford.edu:8080/sentiment/rntnDemo.html

W7 Recap: Topic Modeling



Recap: Latent Semantic Indexing

- Represent Matrix M as M = K.S.D
  - Such a decomposition always exists and is unique

M = K x S x D

Words	Documents 0	1	2	3	4
banana	2	0	4	2	0
kiwi	1	0	5	0	0
apple	1	1	7	4	0
computer	0	1	0	0	4
screen	0	1	0	0	1

=

Words	Topics A	B	C
banana	0.5	0	0.1
kiwi	0.3	0	0.2
apple	0.2	0.2	0.3
computer	0	0.4	0.2
screen	0	0.4	0.2

x

a	0	0
0	b	0
0	0	c

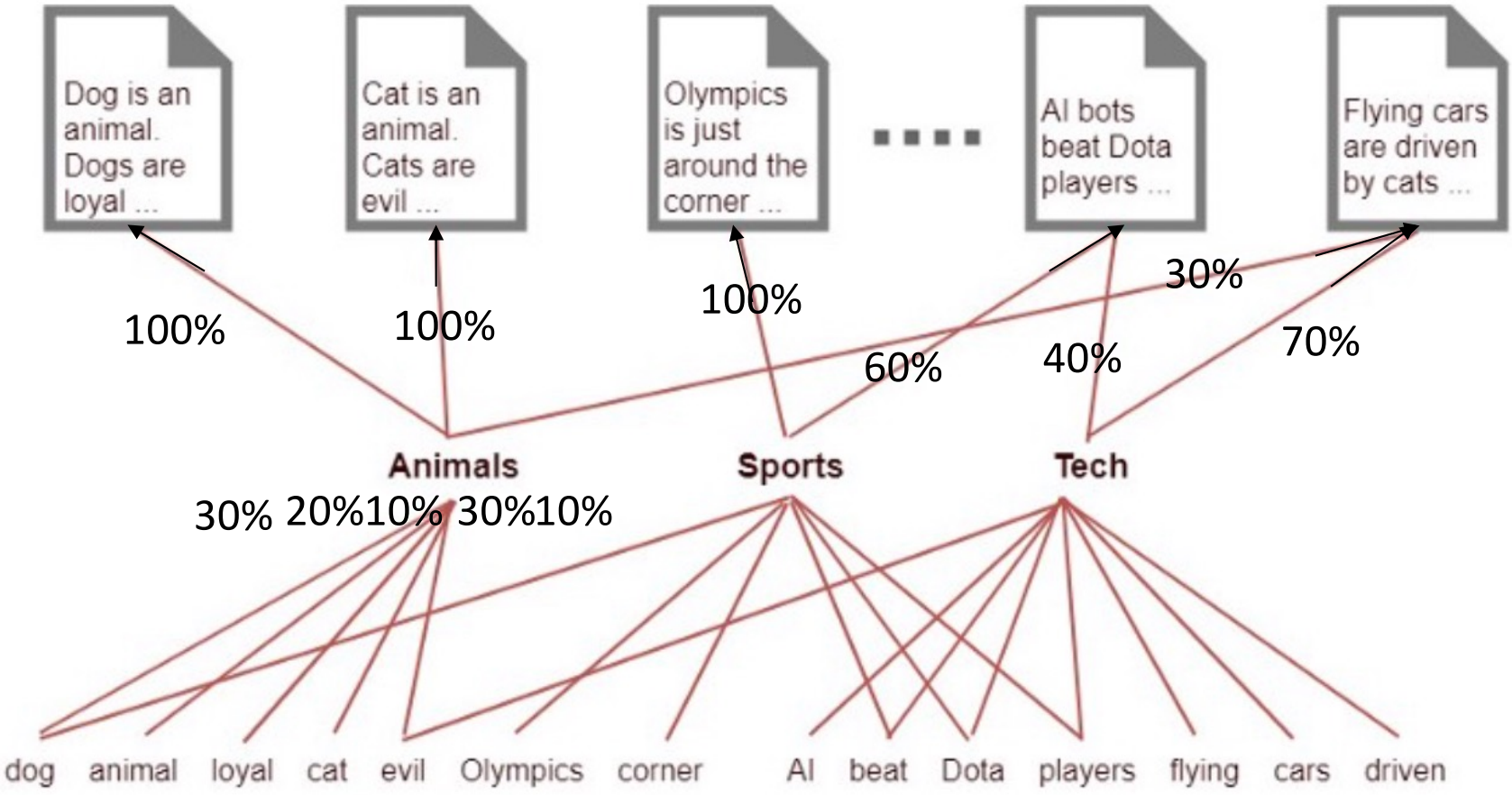
x

Topics	Documents 0	1	2	3	4
A	0.8	0	1	0.7	0
B	0	0.9	0	0	1
C	0.2	0.1	0	0.3	0

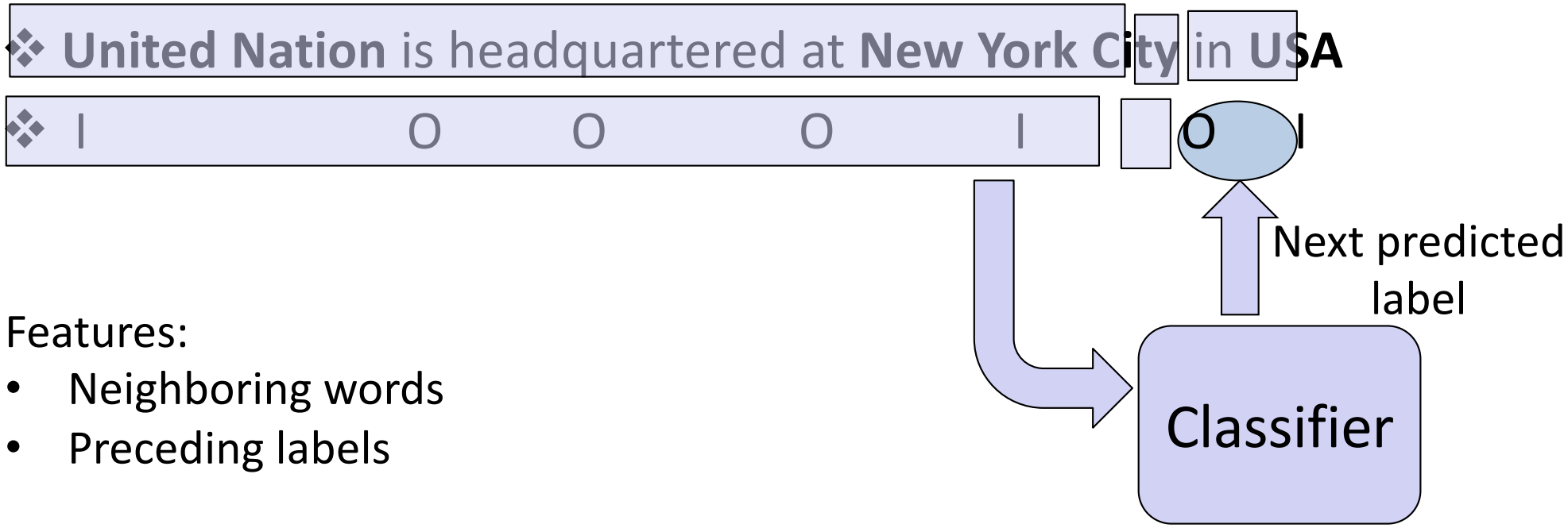
- S is a diagonal matrix of singular values in decreasing order: each value represents the weight of the corresponding topic
- K is the term-topic matrix
- D is the document-topic matrix

Recap: Latent Dirichlet Allocation

- **Idea**: assume a document collection is (randomly) generated from a known set of topics (probabilistic generative model)



Recap: Named Entity Recognition



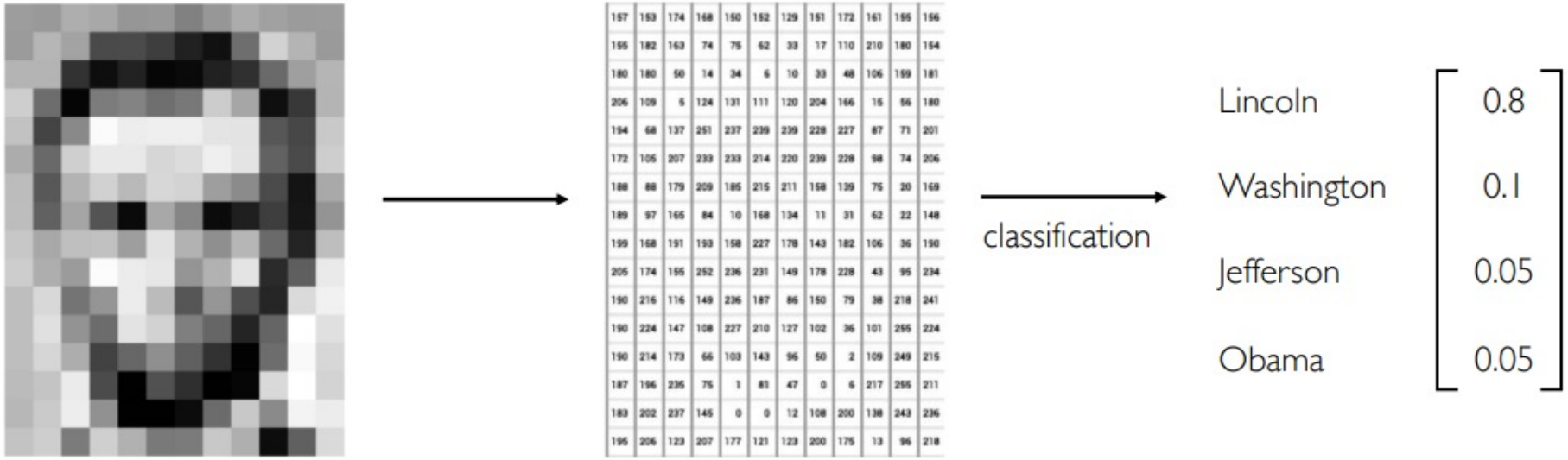
Features:

- Neighboring words
- Preceding labels

Naïve Bayes, HMM, MEMM, CRF, …

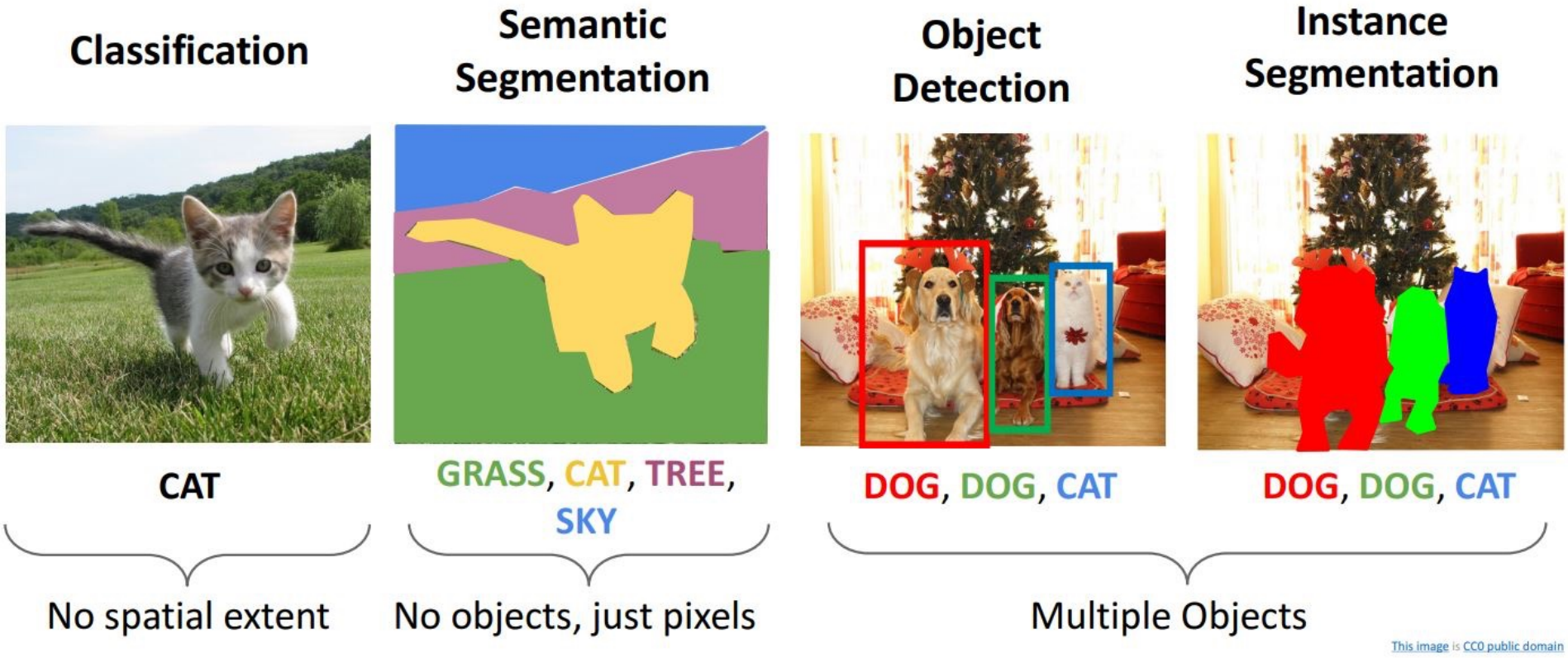
Week 8 Recap

- In computers, images are just numbers
  - We can still apply traditional analysis methods



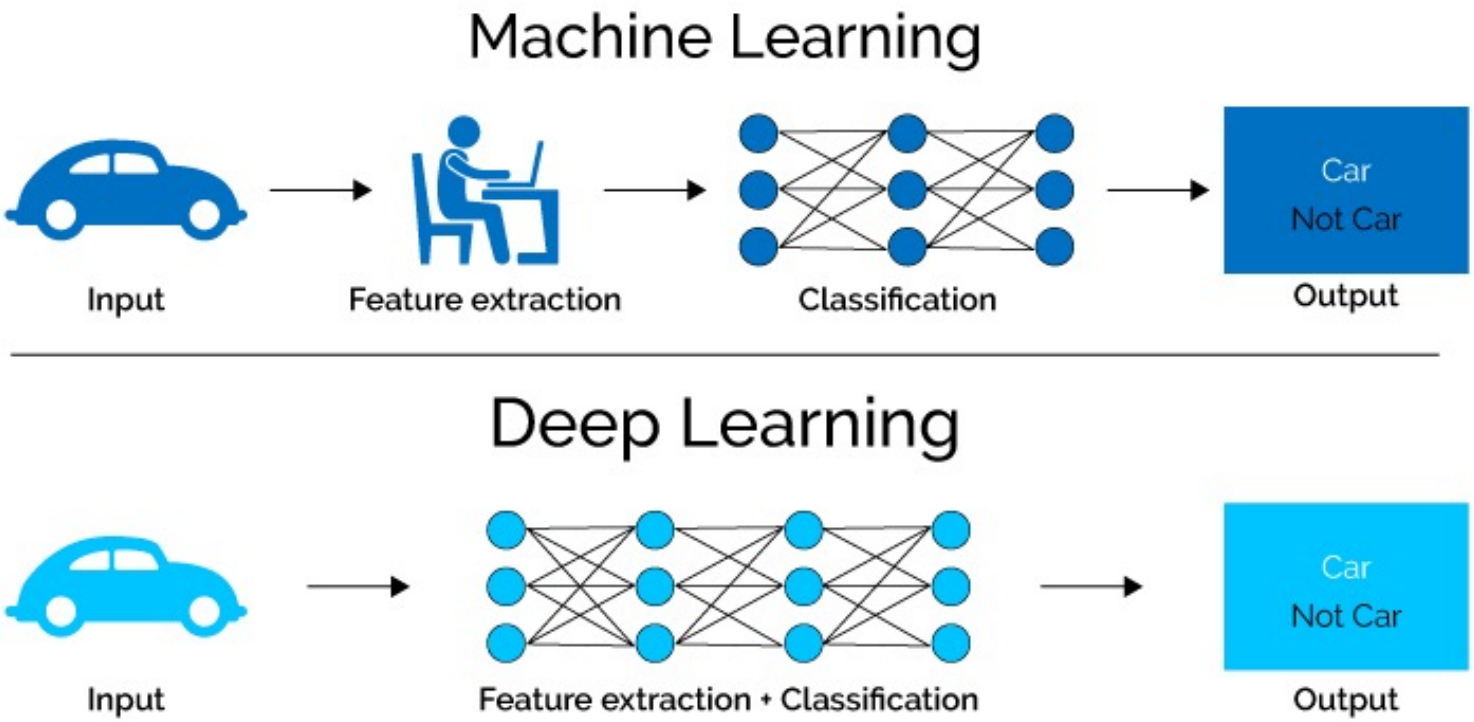
- **Regression**: output variable takes continuous value
- **Classification**: output variable takes class label. Can produce probability of belonging to a particular class

Recap: Computer Vision Applications



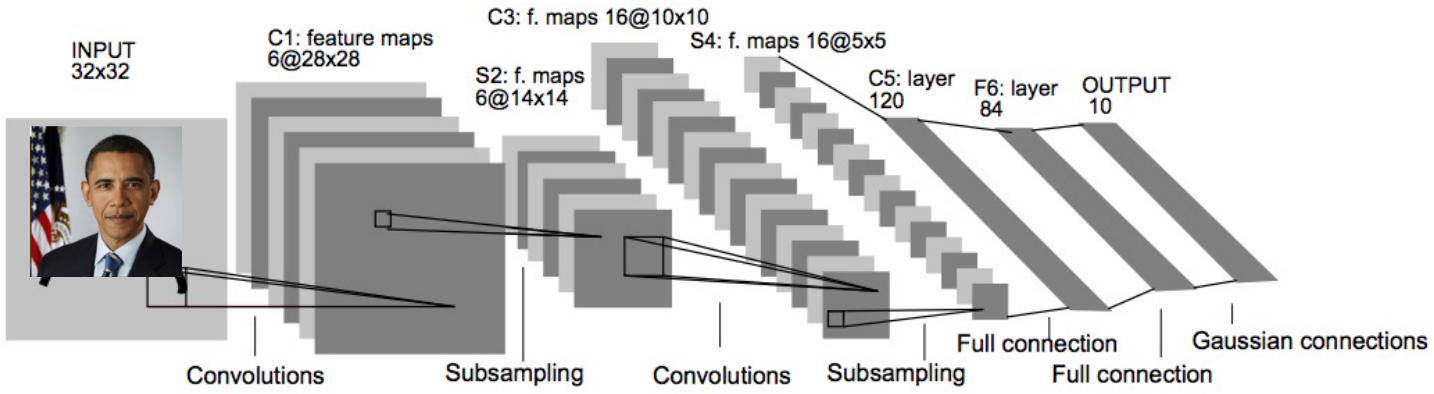
Recap: Deep Learning to the Rescue

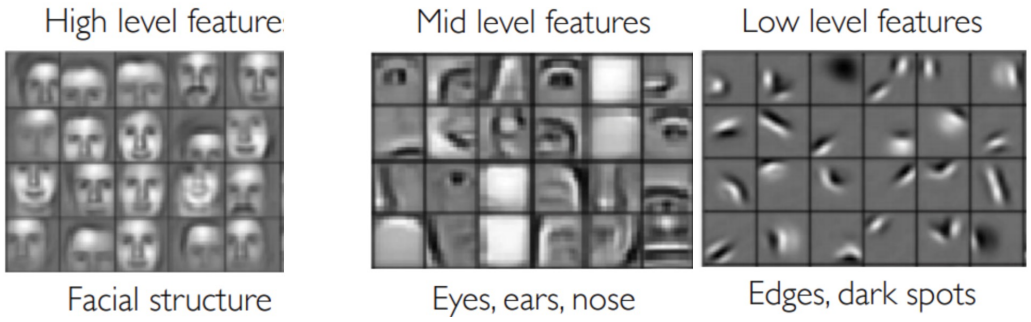
- Features are learnt automatically from the data
- Can be applied or reused to different applications or domains



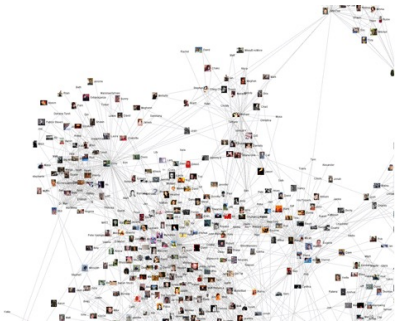
Recap: Secret of Deep Learning

- History:
  - CNNs were not always famous because of hardware limitation.
  - CNNs exploded with AlexNet in 2012, a CNN that made a breakthrough in the Computer Vision field thanks to deep architecture and GPU.
- Intuition:
  - Look at the big region then look at smaller regions

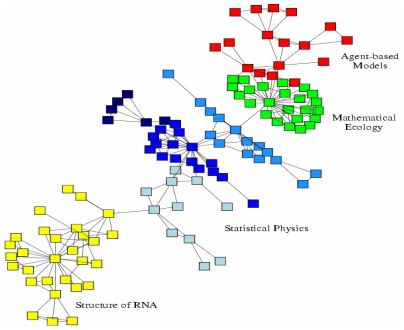




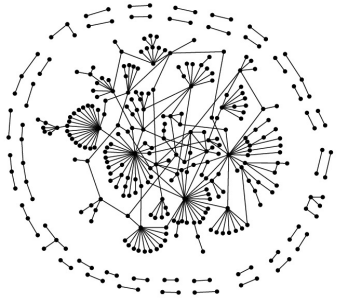
Week 9 Recap:
Many Data are Networks



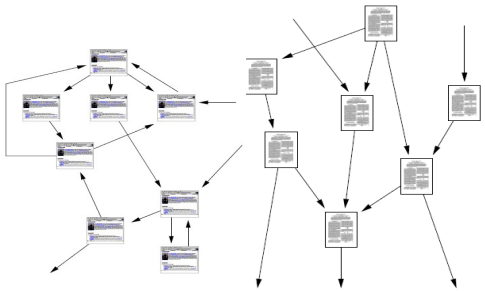
Social networks



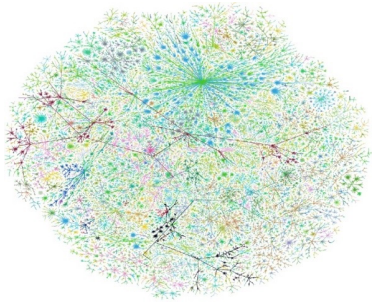
Economic networks



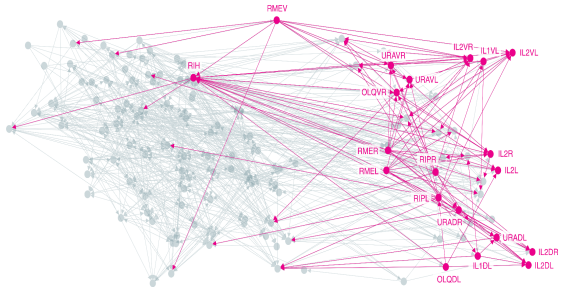
Biomedical networks



Information networks:
Web & citations



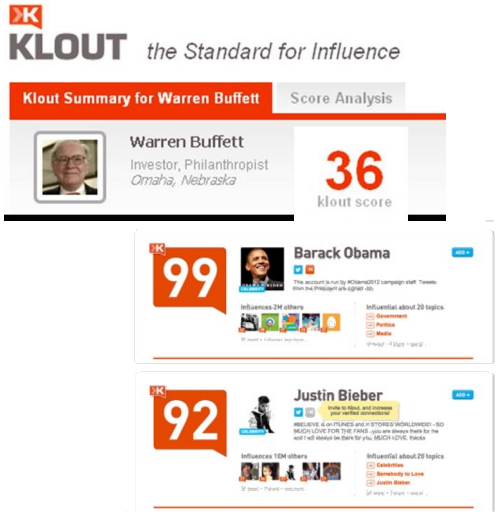
Internet



Networks of neurons

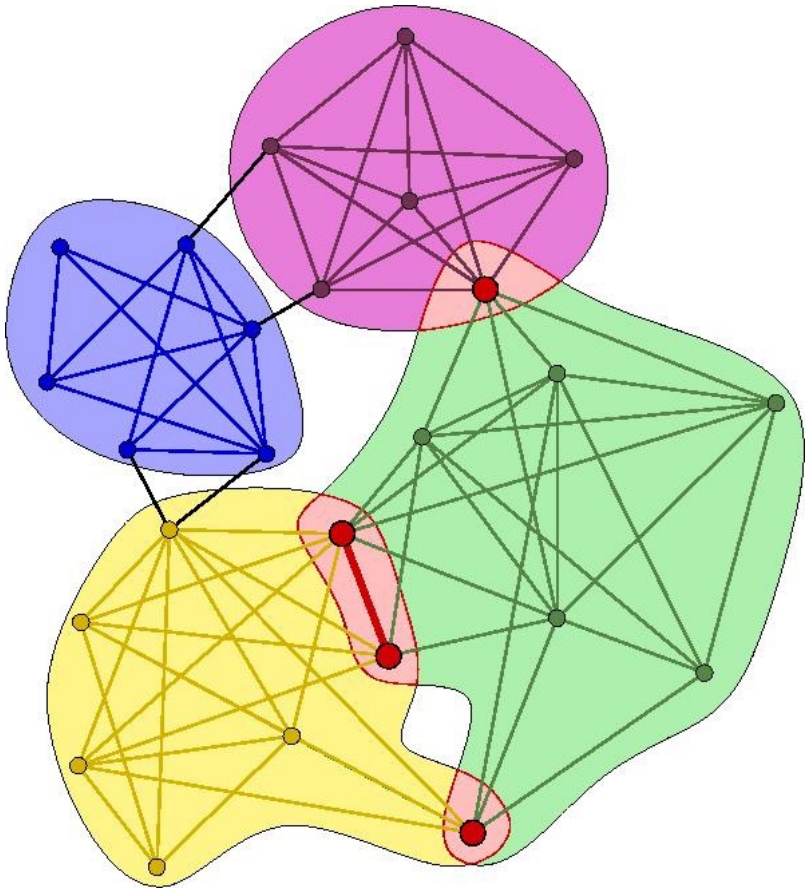
W9 Recap: Centrality

- What is centrality?
  - Centrality defines how important an actor is within a network

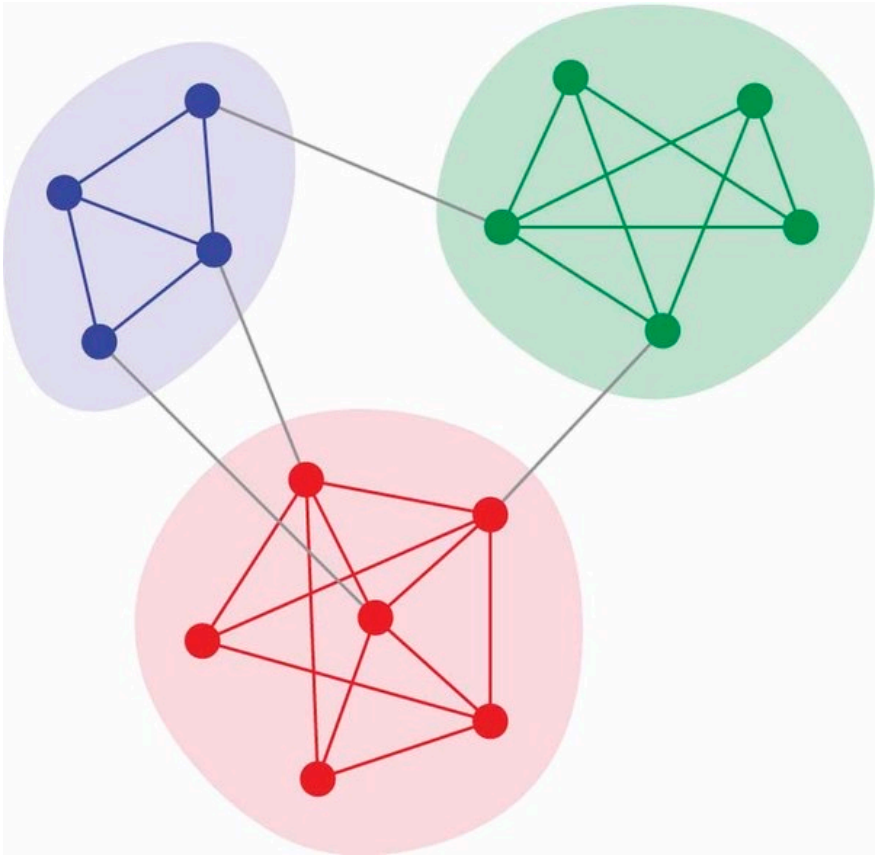


- Why centrality? a measure of influence
  - The act or power of producing an effect without apparent exertion of force or direct exercise of command

W9 Recap: Community



Overlapping Communities



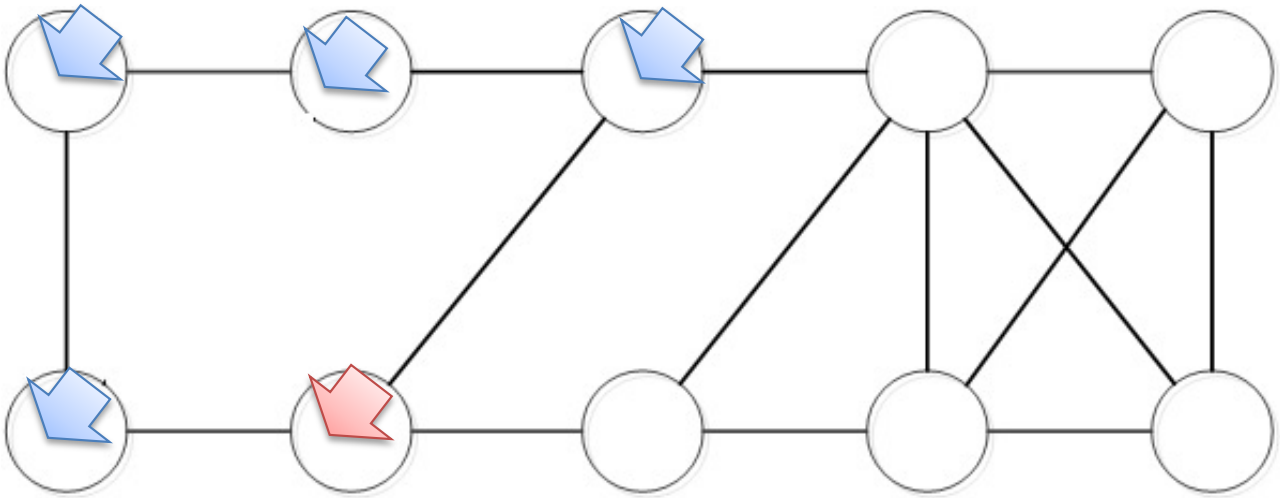
Disjoint Communities

W9 Recap: Information Diffusion

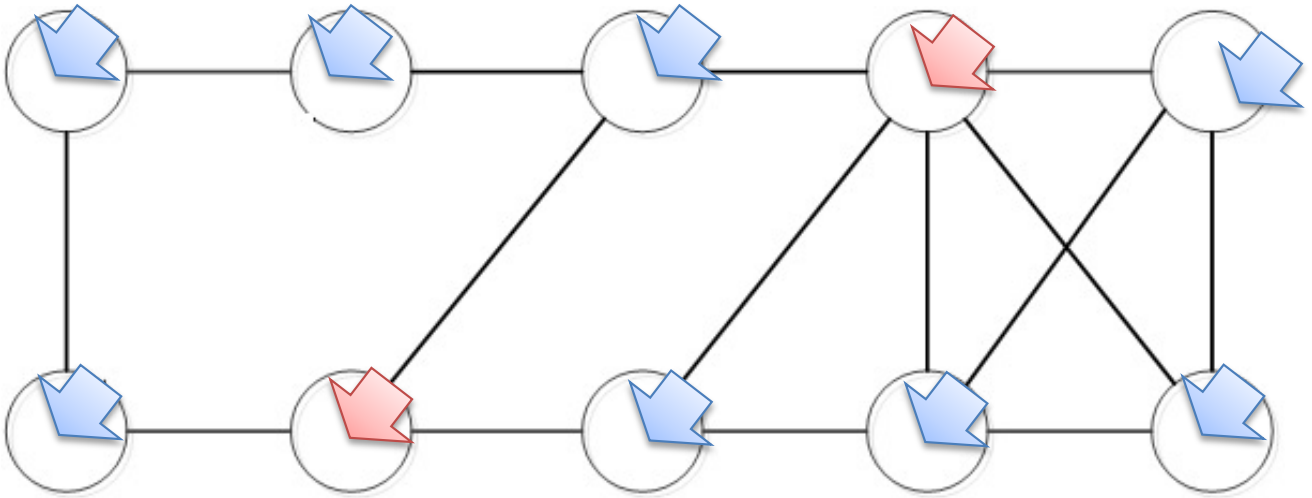
seed

activated nodes

k=1 seed



k=2 seeds



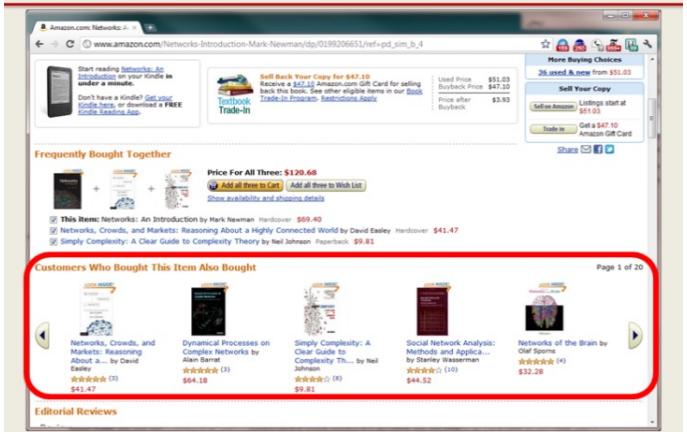
Week 10 Recap:
What is Recommendation?

- Which mobile phone should I buy?
- Where should I visit for my business trip?
- Whom should I follow on Twitter?
- Where should I invest my money?

- Which tour is the best for **our class**?

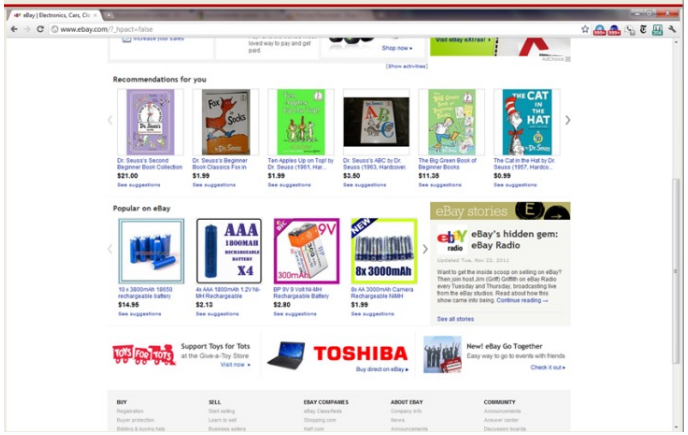
Week 10 Recap: Recommender Systems

Book recommendation in Amazon

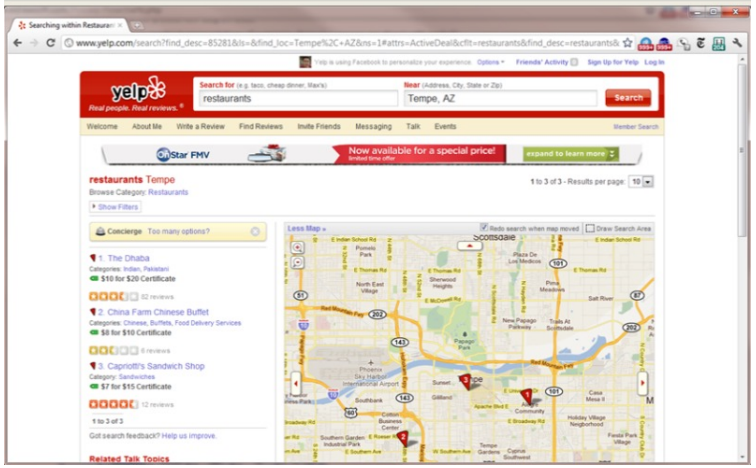


Video clip recommendation in YouTube

Product Recommendation in ebay



Restaurant Recommendation in Yelp



recommendation = personalized prediction

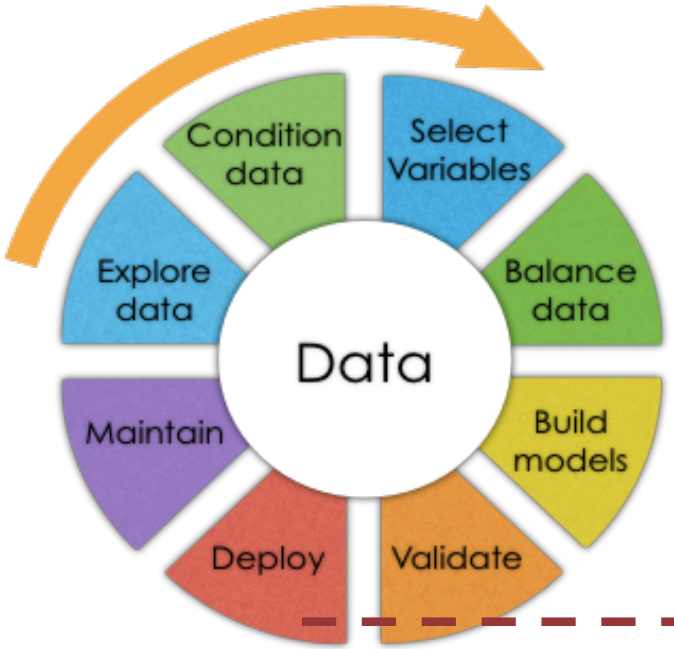
W10 Recap: Types of Recommender Systems

1. Content-based recommendation:
   - Recommend **based on similarity** between user features and item features
2. Rating-based recommendation (Collaborative Filtering)
   - Recommend **based on rating** matrix
3. Hybrid recommendation
   - Use both **contents** and **ratings**
4. Clustering-based recommendation
   - Recommend **based on clusters of rating** matrix

Week10 Recap: RecSys Challenges

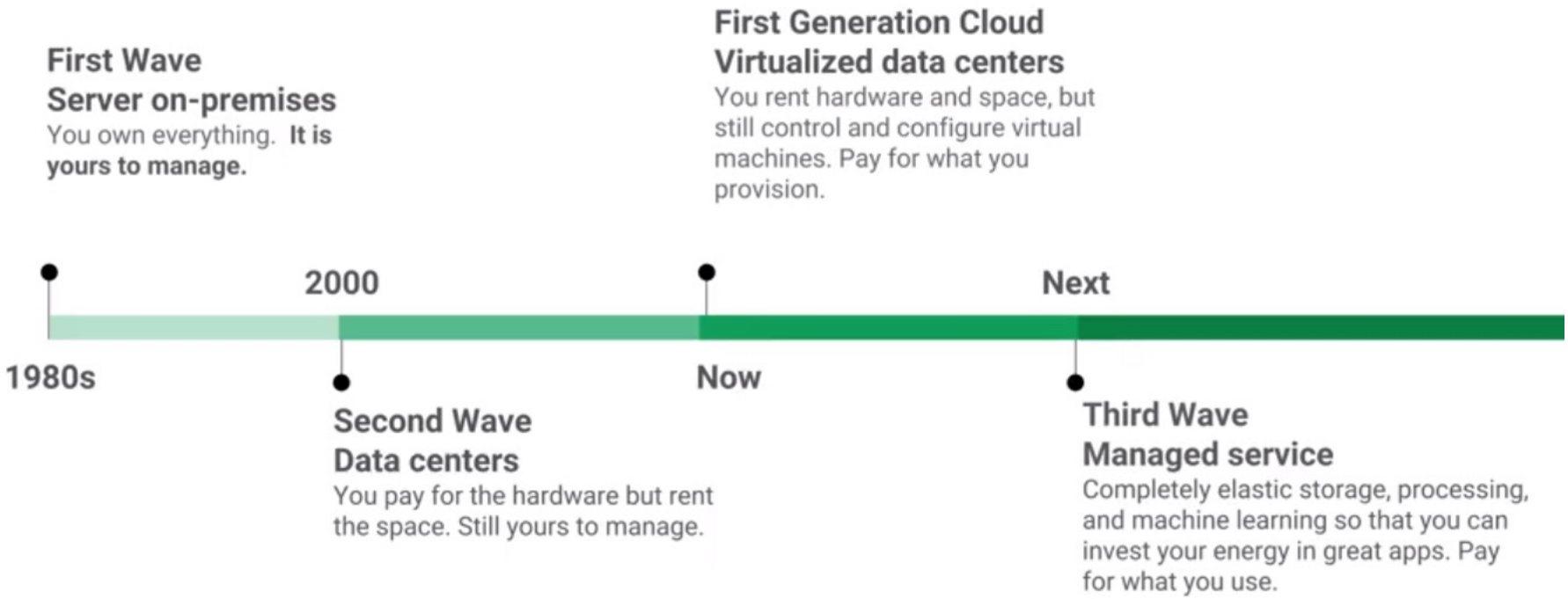
- **Cold-Start Problem**
  - Recommender systems use **historical data** or information provided by the user to recommend items, products, etc.
  - When user join sites, they still haven't bought any product, or they have no history.
  - It is hard to infer what they are going to like when they start on a site.
- **Data Sparsity**
  - When historical or prior information is **insufficient**.
  - Unlike the cold start problem, this is in the system as a whole and is not specific to an individual.
- **Attacks**
  - **Push Attack**: pushing ratings up by making fake users
  - **Nuke attack**: DDoS attacks, stop the whole recommendation systems
- **Explanation**
  - Recommender systems often recommend items with **no explanation** on why these items are recommended

W11 Recap: Data Lifecycle



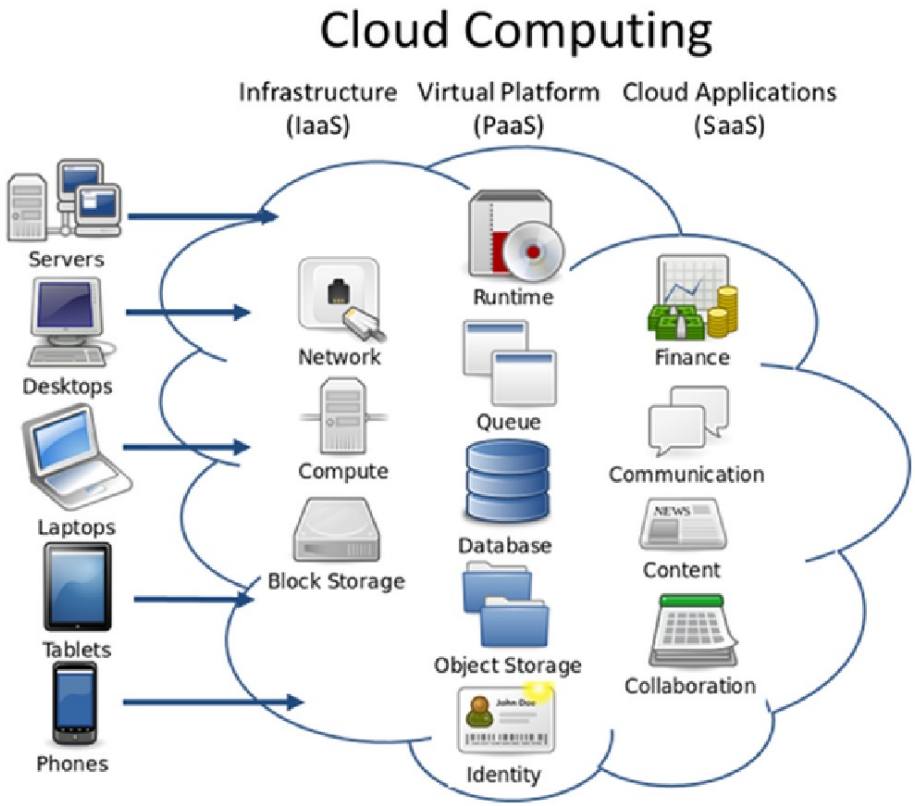
http://www.evolved-analytics.com/sites/default/files/modelLifeCycleSmall.png

W11 Recap: Timeline



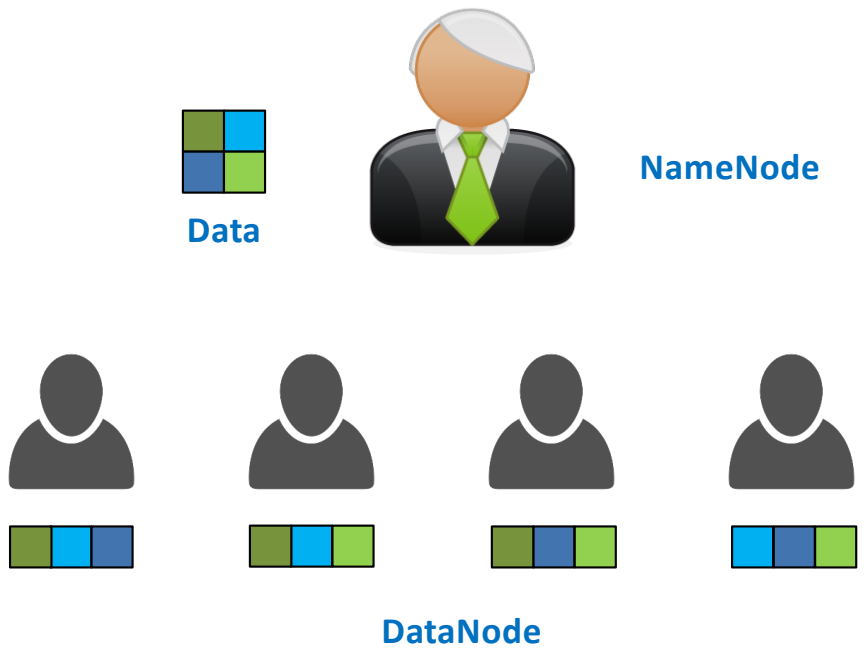
W11 Recap: Service Models

- IaaS
- PaaS
- Saas
- …



W11 Recap: Hadoop - Distributed File System (HDFS)

- Master(NameNode)
  - ✓ Cut the input data into blocks
  - ✓ Decide the destination of the blocks and replicas
  - ✓ Store the metadata of DataNodes
  - ✓ Monitor the DataNodes
- Slave(DataNode)
  - ✓ Store the data blocks
  - ✓ Provide access



W11 Recap: Spark

- Fast version of Hadoop
  - ✓ **In-memory system:** significantly reduce the disk write
  - ✓ Performs better when the memory size is large
- Data Storage
  - ✓ Support HDFS
  - ✓ Other distributed/cloud file system (HBase, Amazon S3, etc.)
- Data Processing
  - ✓ More operations
  - ✓ Pipeline mode based on RDDs(Resilient Distributed Datasets)
- More language supported
  - ✓ Scala
  - ✓ Python
  - ✓ Java
  - ✓ R