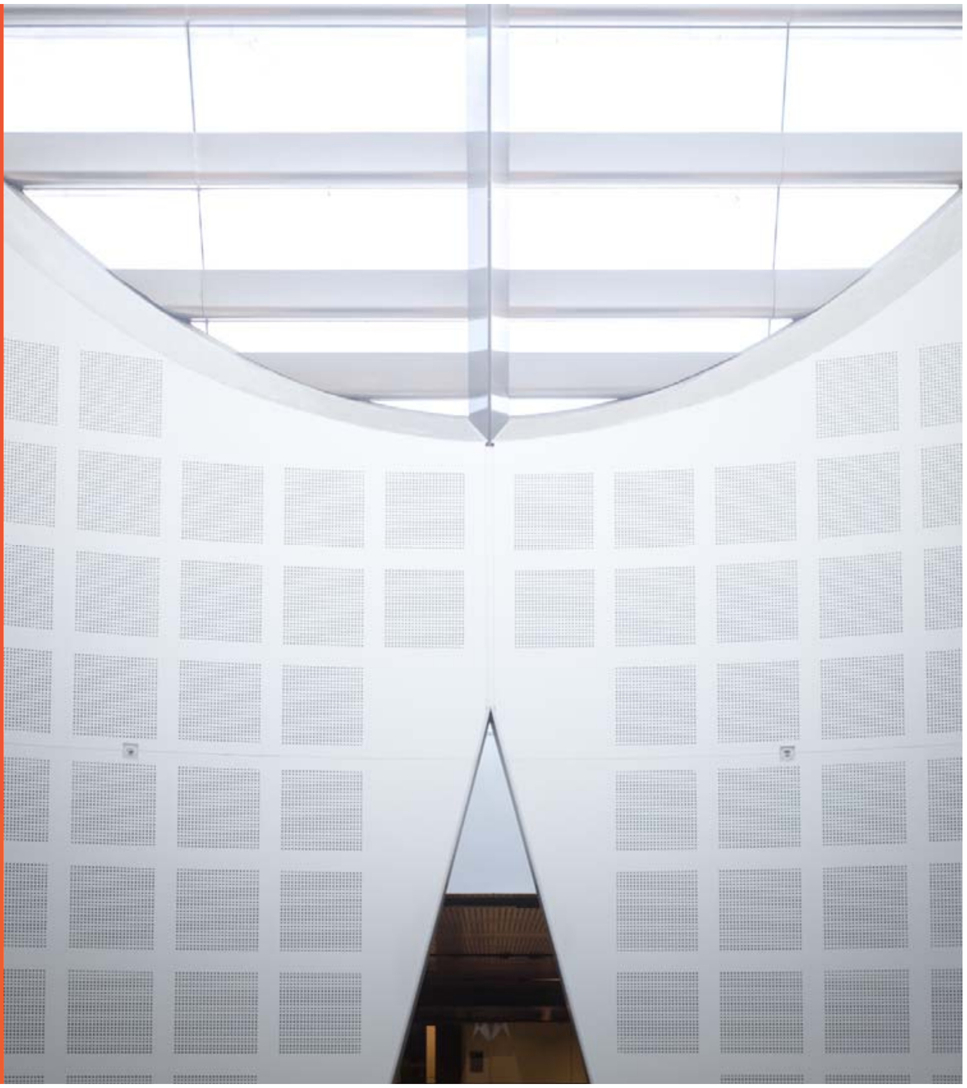
DATA2001

W 6: Text data processing: feature extraction and analysis

Presented by: Dr. Matloob Khushi



Lecture plan

- W1: Introductions
- W2: Data Analysis with Python
- W3: Accessing data in relational databases; introduction to SQL
- W4: Declarative data analysis with SQL
- W5: Scalable Data Analytics: The role of indexes and data partitioning
- **W6: Text data processing**
- W7: No Lecture (ANZAC Day)
- W8: **QUIZ (1st May)**
- W9:
- W10:
- W11:
- W12:
- W13: Review
- ***Exam***

Today: Text data processing

Objective

Learn machine learning tools in Python for text categorisation and forecasting.

Lecture

- Text Classification by ML
- Text-driven forecasting
- Structured prediction

Readings

- Data Science from Scratch, Ch. 13
- Doing Data Science, Ch. 4

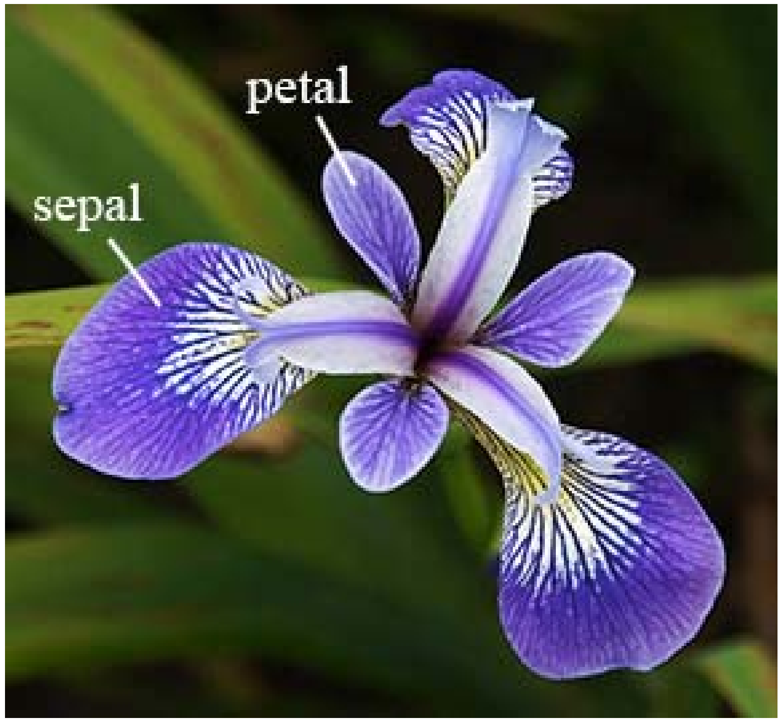
Exercises

- Spam detection
- Predicting box office returns
- Information extraction

Text data usually does not have a pre-defined data model, is unstructured and is typically text-heavy, but may contain dates, numbers and facts as well.

This results in ambiguities that make it more difficult to understand than data in structured databases.

Structured data



- Fielded data
- Stored in databases
- E.g.:
  - Sensor data
  - Financial data
  - Click streams
  - Measurements

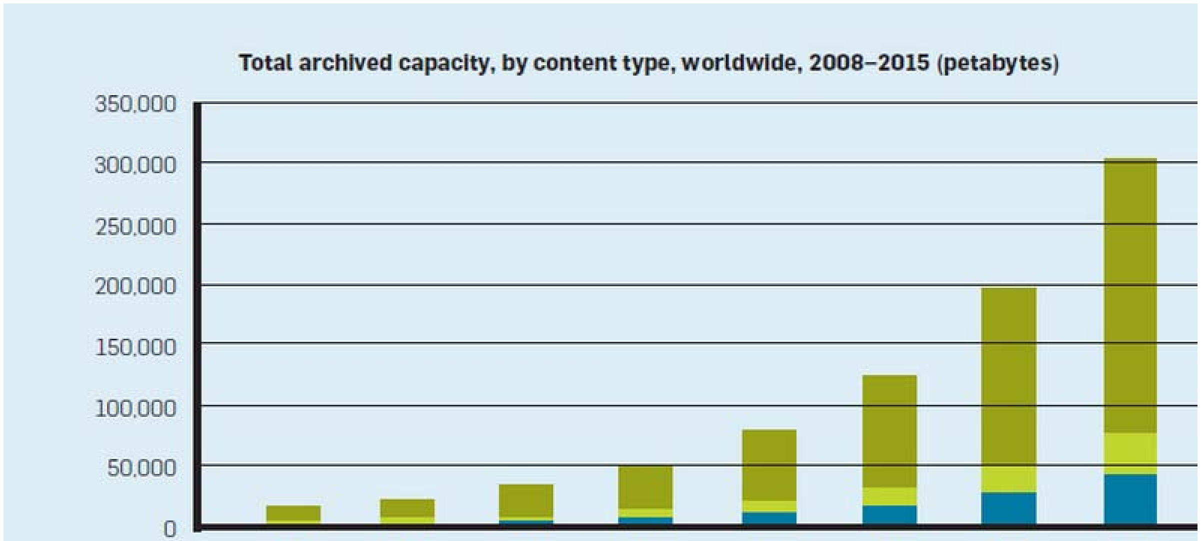
Text as unstructured data

"In the history of cinematic mustaches, few have been as disgusting as that of Rye Gerhardt (Kieran Culkin), the youngest scion of North Dakota's reigning crime family and the stray spark that sets off the powder-keg second season of Fargo."

(From a Slate review of Fargo Season 2)

- 80-90% of all potentially usable business information
- E.g.:
  - Images
  - Video
  - Email
  - Social media

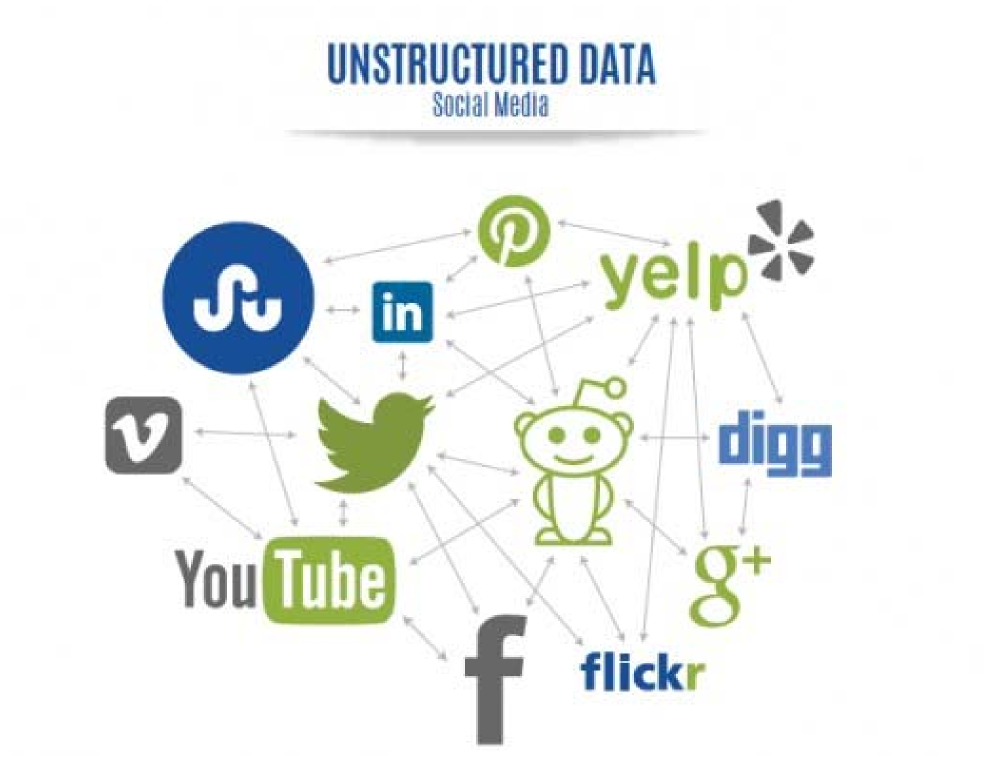
Information overload



	2008	2009	2010	2011	2012	2013	2014	2015
🟩 Unstructured	11,430	16,737	25,127	39,237	59,600	92,536	147,885	226,716
🟩 Database	1,952	2,782	4,065	6,179	9,140	13,824	21,532	32,188
🟦 Email	1,652	2,552	4,025	6,575	10,411	16,796	27,817	44,091

http://bhavnaober.blogspot.com.au/2015_02_01_archive.html

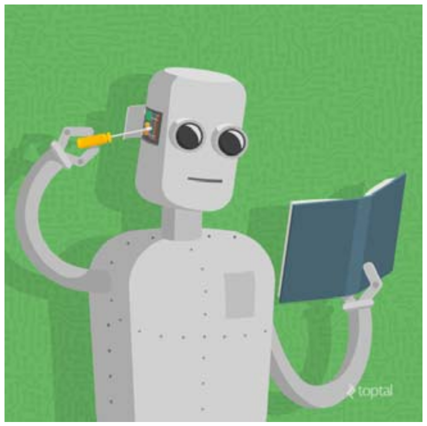
Social media data



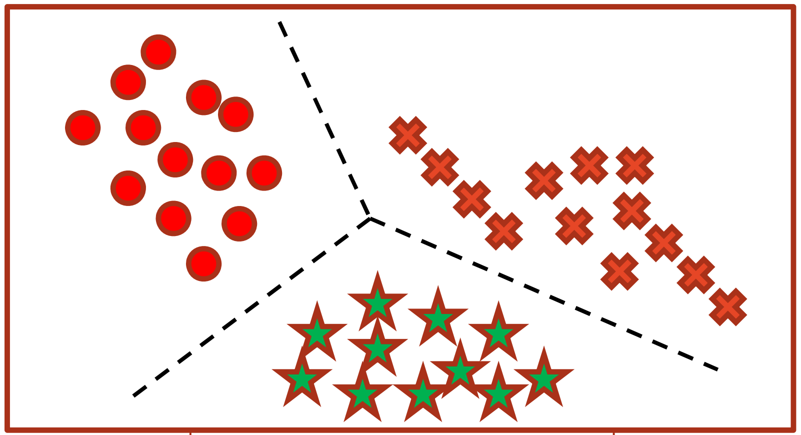
http://hubpages.com/technology/Big-Data-Understanding-New-Insights#

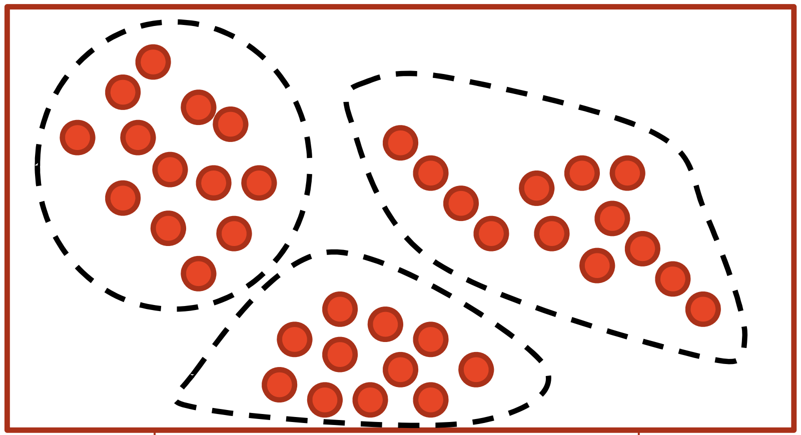
Machine Learning

- **Supervised learning**
  - Prediction
  - Classification (discrete labels), Regression (real values)
- **Unsupervised learning**
  - Clustering
  - Probability distribution estimation
  - Finding association (in features)
  - Dimension reduction
- **Semi-supervised learning**
- **Reinforcement learning**
  - Decision making (robot, chess machine)



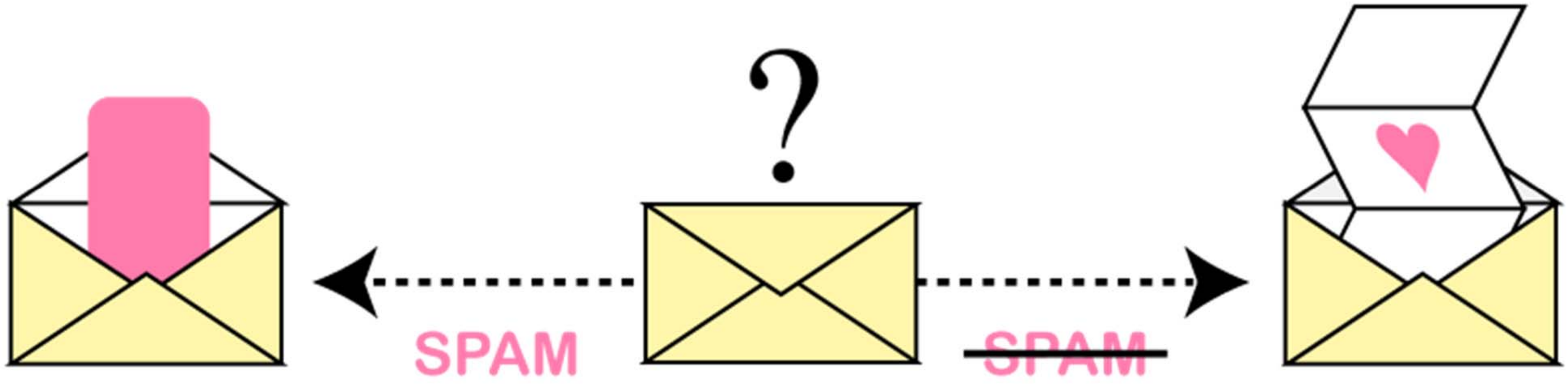
Supervised vs Unsupervised Learning


Supervised learning


Unsupervised learning

Text categorisation

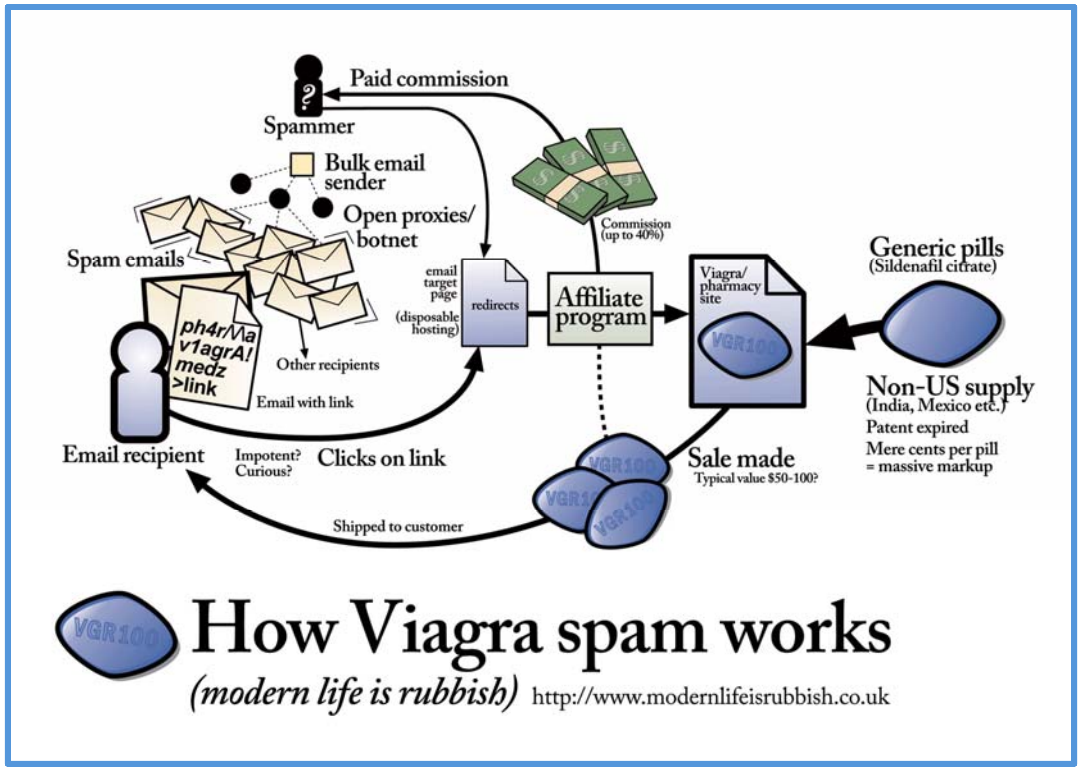
Task: Spam/ham detection



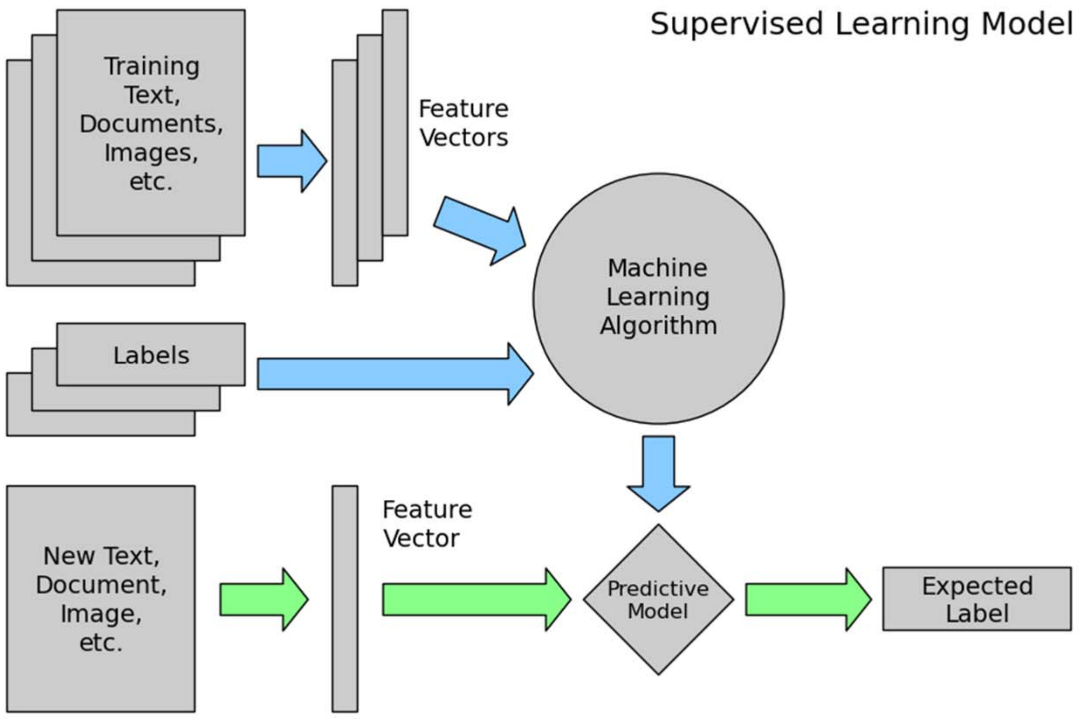
http://blog.dato.com/how-to-evaluate-machine-learning-models-part-2a-classification-metrics

Modelling spam detection

- Input:
  - Emails
  - SMS messages
  - Facebook pages
- Predict:
  - 1 (spam)
  - 0 (ham)



Spam detection as supervised classification


Supervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Labels
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

http://www.astroml.org/sklearn_tutorial/general_concepts.html#supervised-learning-model-fit-x-y

Feature vectors from text

http://www.python-course.eu/text_classification_python.php

– Represent document as a multiset of words
– Keep frequency information
– Disregard grammar and word order

Tokenisation

“Friends, Romans, Romans, countrymen”

⬇

[“Friends”,
“Romans”,
“Romans”,
“countrymen”]

– Split a string (document) into pieces called tokens
– Possibly remove some characters, e.g., punctuation
– What about “O’Neill”? “Aren’t”?

Normalisation

[“Friends”,
“Romans”,
“Romans”,
“countrymen”]

⬇

[“friend”,
“roman”,
“roman”,
“countrymen”]

- Map similar words to the same token
- Stemming/lemmatisation
  - Avoid grammatical and derivational sparseness
  - E.g., “was” => “be”
- Lower casing, encoding
  - E.g., “Naïve” => “naive”

Indicator features

[“friend”,
“roman”,
“roman”,
“countrymen”]

⬇

{“friend”: 1,
“roman”: 1,
“countrymen”: 1}

- Binary indicator feature for each word in a document
- Ignore frequencies

Term frequency weighting

[“friend”,
“roman”,
“roman”,
“countrymen”]

⬇

{“friend”: 1,
“roman”: 2,
“countryman”: 1}

- Term frequency
  - Give more weight to terms that are common in document
  - *TF = |occurrences of term in doc|*
- Damping
  - Sometimes want to reduce impact of high counts
  - *TF = log(|occurrences of term in doc|)*

TFIDF Weighting

[“friend”,
“roman”,
“countrymen”]

⬇

{“friend”: 0.1,
“roman”: 0.8,
“countrymen”: 0.2}

- Inverse document frequency
  - Give less weight to terms that are common across documents
  - *IDF = log(|total docs|/|docs containing term|)*
- TFIDF
  - *TFIDF = TF * IDF*

SMS spam detection

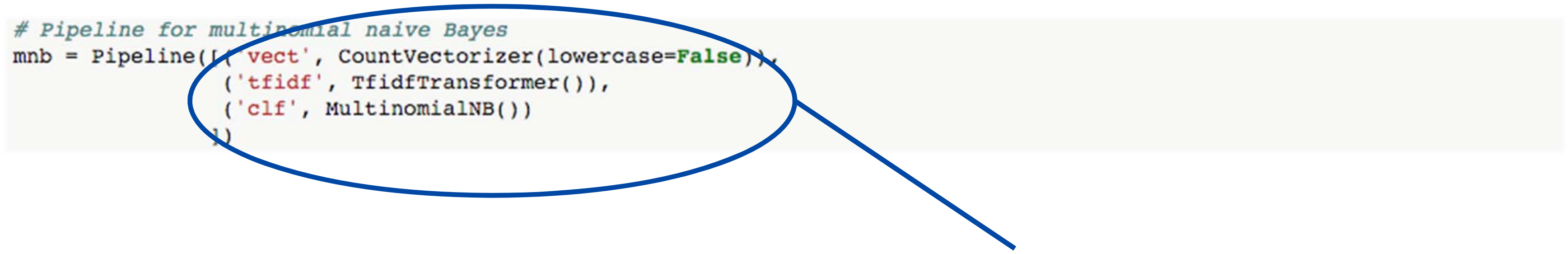
Label	Message snippet
Ham	Go until jurong point, crazy.. Available only …
Spam	Ok lar… Joking wif u oni...
Ham	U dun say so early hor… U c already then say...
Spam	FreeMsg Hey there darling it's been 3 week's n…

– 425 SMS spam messages from UK Grumbletext web forum
– 3,375 ham randomly chosen from NUS SMS corpus (students)
– 450 ham from somebody's PhD thesis
– 322 spam and 1,002 ham from SMS Spam Corpus

https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

Use scikit-learn Pipeline to manage cross validation

Scikit-learn Pipelines provide a mechanism for fitting and predicting a sequence of components. This is good practice to avoid data leakage.



```
# Pipeline for multinomial naive Bayes
mnb = Pipeline([('vect', CountVectorizer(lowercase=False)),
                ('tfidf', TfidfTransformer()),
                ('clf', MultinomialNB())
               ])
```

1. **Convert string to a bag-of-words token vector.**
2. **Transform vector counts using TFIDF weighting.**
3. **Train/predict using multinomial naïve Bayes.**

The curse of dimensionality

- From a theoretical point of view, increasing the number of features should lead to better performance.
- In practice, the inclusion of more features leads to worse performance (i.e., curse of dimensionality).
- The number of training examples required increases exponentially with dimensionality.
- Dimensionality reduction can improve the prediction accuracy.

Exercise: SMS spam filtering

- Build a spam filter
  - ⏯ code cell under “Download data from UCI ML data repo”
  - ⏯ code cell under “Read and profile data using Pandas”
  - ⏯ code cell under “Text feature extraction”
  - ⏯ code cell under “Build pipelines and choose parameters”
- Exercises
  - Which is better: support vector or multinomial naïve Bayes classifier?
  - Handling class imbalance in support vector classifier
  - Generalisation, data size, features

Text-driven forecasting

Text-driven forecasting

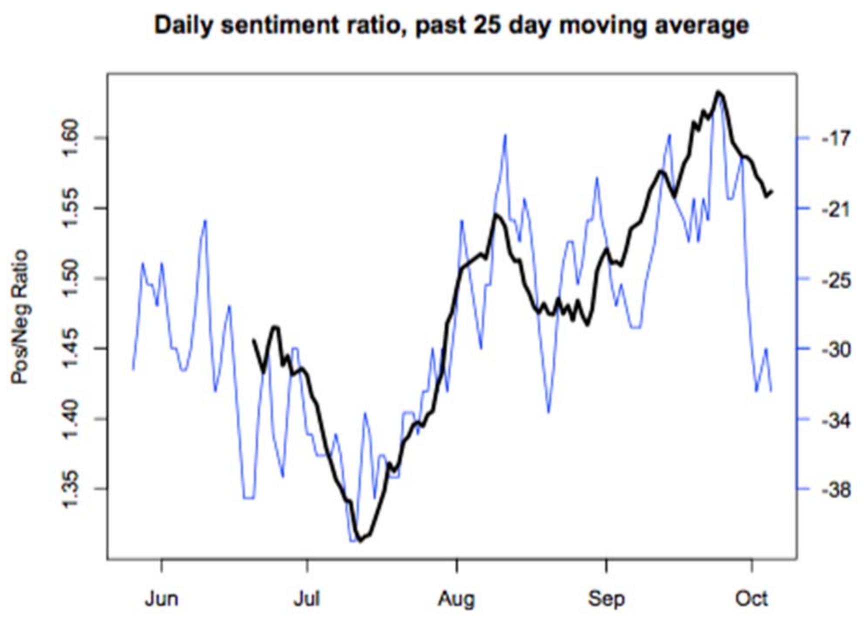
Given a body of text *T* pertinent to a social phenomenon, make a concrete prediction about a measurement *M* of that phenomenon, obtainable only in the future.

http://www.cs.cmu.edu/~nasmith/papers/smith.whitepaper10.pdf

Some text-driven forecasting tasks

- Predict box office gross for films
  - T: description, script, reviews, etc
  - M: how much the film earns at the box office
- Predict volatility of a stock
  - T: annual report, etc
  - M: volatility over the following year
- Predict blog reader behaviour
  - T: political blog posts, etc
  - M: number of reader comments

Predicting public opinion from tweets



- T: tweets mentioning the word "economy"
- M: Gallup's economic confidence index (blue)
- Predictions (black) closely track Gallup's polling data

https://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1536/1842

Collect data from DBpedia

- DBPedia converts Wikipedia pages into structured semantic web data.
- Provides public SPARQL endpoint to query data at http://dbpedia.org/sparql

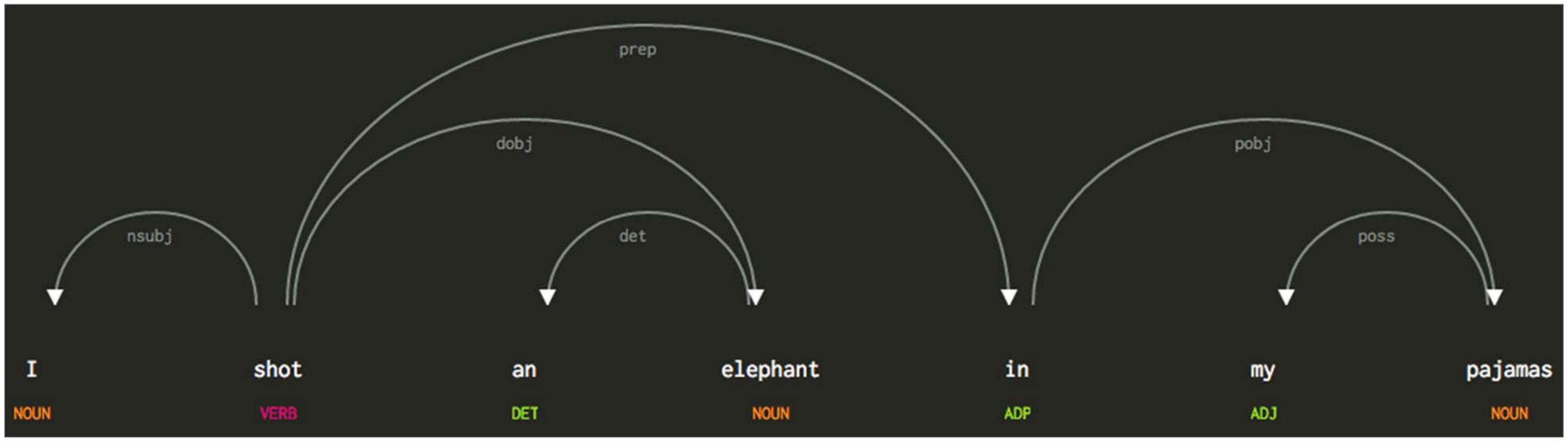
Exercise: Predicting box office gross

- Build model to forecast gross
  - ⏭ code cell under “Download reviews and movie gross data”
  - ⏭ code cell under “Select parameters for support vector regression”
- Exercises
  - Which parameters are best?
  - Is the model a good fit?
  - How could we improve the experimental setup?

Information extraction

THE UNIVERSITY OF SYDNEY

Parsing natural language



https://spacy.io/demos/displacy?share=2473569563126265042

Natural language processing

"Interdisciplinary field concerned with modelling natural language from a computational perspective."

https://en.wikipedia.org/wiki/Computational_linguistics

- Understanding
  - Tokenisation
  - POS tagging
  - Parsing
- Generation
- Summarisation

Information extraction

"Task of automatically extracting structured information from unstructured and/or semi-structured documents."

https://en.wikipedia.org/wiki/Information_extraction

- Named entity recognition
- Entity disambiguation
- Relation extraction

Knowledge base population (KBP)

- Aim is to build structured knowledge bases from massive unstructured text corpora
- Two subtasks:
  - Entity linking: identify mentions of entities, link to KB or NIL
  - Slot filling: extract and populate facts for given entity

Entity linking

John Williams

Richard Kaufman goes a long way back with **John Williams**. Trained as a classical violinist, Californian Kaufman started doing session work in the Hollywood studios in the 1970s. One of his movies was Jaws, with **Williams** conducting his score in recording sessions in 1975...

Michael Phelps

Debbie Phelps, the mother of swimming star **Michael Phelps**, who won a record eight gold medals in Beijing, is the author of a new memoir, ...

Michael Phelps is the scientist most often identified as the inventor of PET, a technique that permits the imaging of biological processes in the organ systems of living individuals. **Phelps** has ...

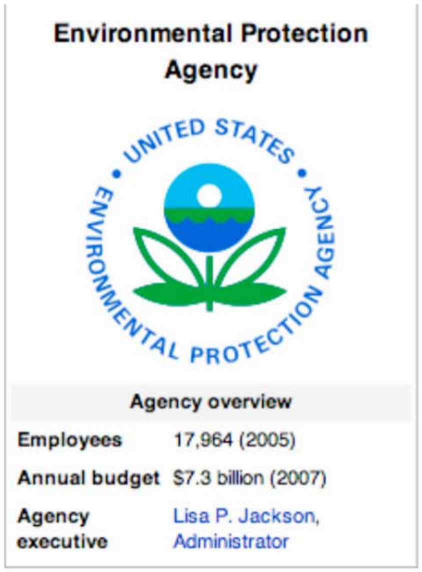
John Williams	author	1922-1994
J. Lloyd Williams	botanist	1854-1945
John Williams	politician	1955-
John J. Williams	US Senator	1904-1988
John Williams	Archbishop	1582-1650
John Williams	composer	1932-
Jonathan Williams	poet	1929-

Michael Phelps	swimmer	1985-
Michael Phelps	biophysicist	1939-

Identify matching entry, or determine that entity is missing from KB

Slot filling

Target: EPA
(plus 1 document)



Environmental Protection Agency

Agency overview	
Employees	17,964 (2005)
Annual budget	$7.3 billion (2007)
Agency executive	Lisa P. Jackson, Administrator

Generic Entity Classes

Person, Organization, GPE

Missing information to mine from text:

- **Date formed: 12/2/1970**
- **Website: http://www.epa.gov/**
- **Headquarters: Washington, DC**
- **Nicknames: EPA, USEPA**
- **Type: federal agency**
- **Address: 1200 Pennsylvania Avenue NW**

Optional: Also want to link some learned values within the KB:

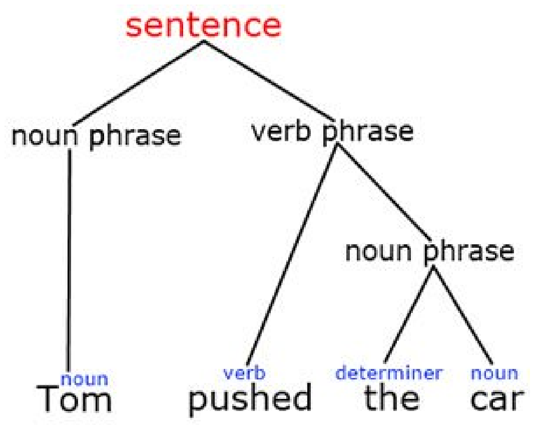
- **Headquarters: Washington, DC (kbid: 735)**

Overview of wikification

– Roth and Ji. Wikification and beyond (slides 4-17)
  http://nlp.cs.rpi.edu/paper/wikificationtutorial.pdf

Sparsity

- spaCy is an open-source software library for advanced Natural Language Processing, written in the programming languages Python and Cython.
- Developers claim to offer the fastest syntactic parser in the world.



Exercise: Natural language parsing with SpaCy

- Go parse Alice
  - ⏭ code cell under “Load English parser”
  - ⏭ code cell under “View sentences and tokens”
  - ⏭ code cell under “View detailed linguistic analysis”
- Exercises
  - Visualise parses using https://spacy.io/demos/displacy
  - Discuss how parses could be used for relation extraction
  - Discuss (and implement) linguistic features for text classification

Review

Review: Unstructured data

Objective

Learn machine learning tools in Python for text categorisation and forecasting.

Lecture

- Text as unstructured data
- Text classifier
- Text-driven forecasting
- Structured prediction

Readings

- Data Science from Scratch, Ch. 13
- Doing Data Science, Ch. 4

Exercises

- Spam detection
- Predicting box office returns
- Information extraction

COMP5046: Statistical natural language processing

- "This unit introduces computational linguistics and the statistical techniques and algorithms used to automatically process natural languages (such as English or Chinese). It will refresh the core statistics and information theory, and the basic linguistics required to process language."
- http://sydney.edu.au/engineering/it/courses/comp5046/

Text-driven forecasting (not examinable)

– CMU seminar on text-driven forecasting.
  http://www.cs.cmu.edu/~nasmith/TDF/
– Smith. Text-driven forecasting (whitepaper).
  https://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1536/1842
– Henry. Predicting with words (blog post).
  http://harmony-institute.org/latest/2012/10/19/forecasting-the-influence-of-entertainment/

Information extraction (not examinable)

- Ji and Grisham. KBP: successful approaches and challenges. http://www.aclweb.org/anthology/P11-1115.pdf
- Roth and Ji. Wikipedia and beyond (tutorial). http://nlp.cs.rpi.edu/paper/wikificationtutorial.pdf
- Bordes and Gabrilovich. Web-scale knowledge graphs. http://www.cs.technion.ac.il/~gabr/publications/papers/KDD14-T2-Bordes-Gabrilovich.pdf

Natural language processing (not examinable)

- Manning and Schutze. Foundations of statistical NLP.
  http://nlp.stanford.edu/fsnlp/
  Jurafsky and Martin. Speech and language processing.
  https://web.stanford.edu/~jurafsky/slp3/
- Bird et al. Natural language processing with Python.
  http://www.nltk.org/book/

Structured prediction for NLP (not examinable)

– Smith. Linguistic structure prediction.
  http://www.cs.cmu.edu/~nasmith/LSP/
– Smith. Structured prediction for NLP (tutorial).
  http://www.cs.cmu.edu/~nasmith/slides/sp4nlp.icml09.pdf

Information retrieval and analytics (not examinable)

- Manning et al. Introduction to information retrieval
  http://nlp.stanford.edu/IR-book/
- Kibana user guide
  https://www.youtube.com/watch?v=Kqs7UcCJquM
- Setting up Elasticsearch and Kibana for analytics
  https://www.youtube.com/watch?v=wHWb1d_VGp8