Spark NLP+ML for Sentiment Analysis

Galvanize, Seattle

Spark NLP+ML for Sentiment Analysis

Galvanize, Seattle

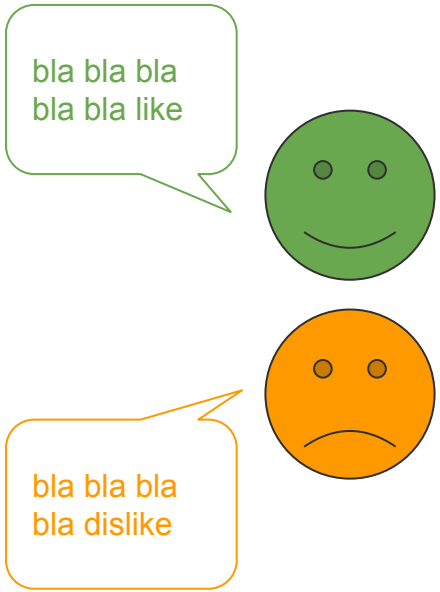
OBJECTIVES

- **Describe** the process for sentiment analysis
- **Define** bag-of-words representation
- **Use** natural language processing to extract BoW representation from raw documents
- **Implement** Machine Learning to analyse sentiments in product reviews

Sentiment Analysis

Finding the **opinions of authors** / users
about specifics entities (company, brand, product, content…)

Based on the **analysis of their comments** / reviews



Example **use cases**:

- Finding the features people like/dislike in a product
- Monitor the reputation of abrand
- Measure the emotional pulse of a nation

Machine Learning: different types of learning

[Samuel, 1959] : Machine learning is the “field of study that gives computers the ability to learn without being explicitly programmed”

Supervised learning:

- The model is derived from observations of input/output pairs
- You have data samples with labelled output (quantitative / qualitative)

Unsupervised learning:

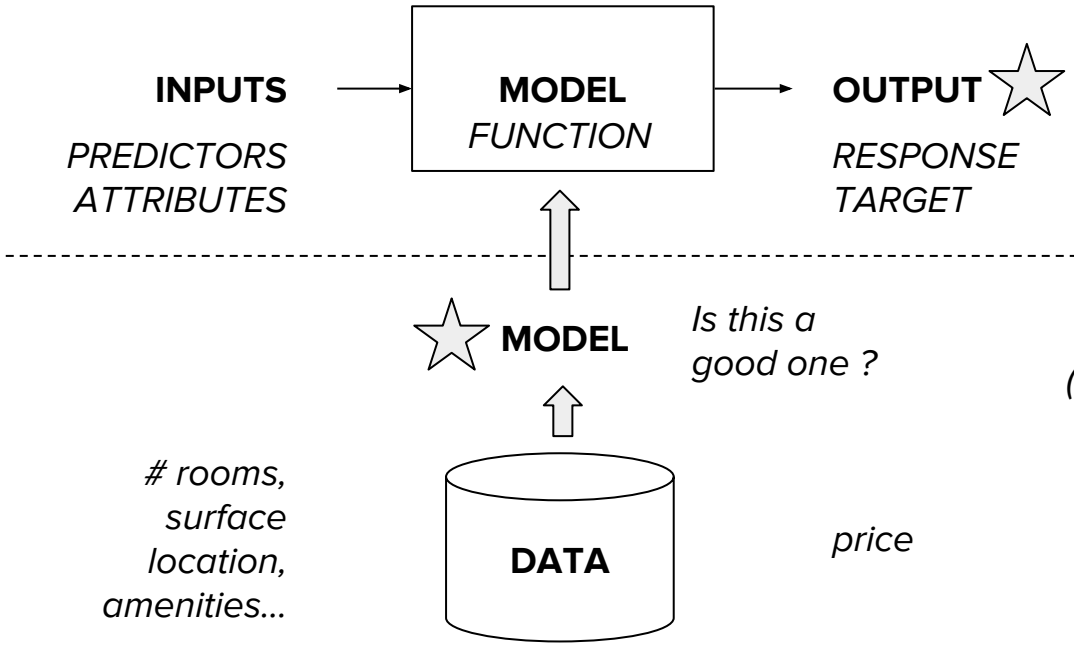
- The model is derived from the confrontation of a meta-model with observations
- You have data samples without no output class, and you want to explain or describe them (but you have an idea of what you’re looking for)

Data Science Tools

Reinforcement learning:

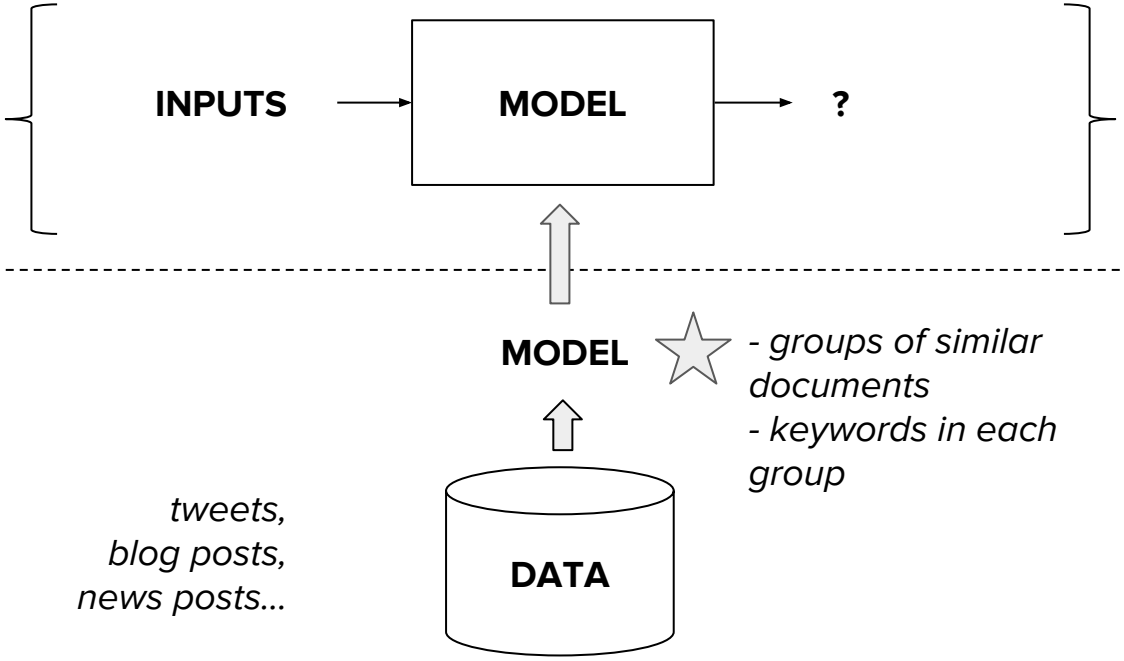
- The model is derived from interactions with an external agent or environment

Supervised Learning



Example: based on what we see on the market, how to compute the price of an apartment depending on the number of rooms, surface, location, amenities?

Unsupervised Learning



Example: what are things people talk about related to a given brand?

Machine Learning for sentiment analysis

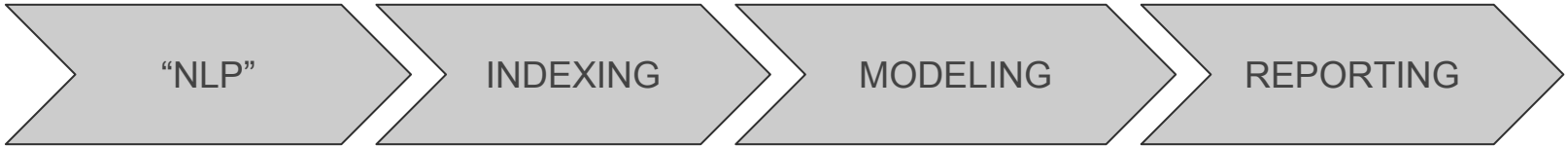
Supervised learning:

- Model what makes the difference between positively and negatively labelled reviews.

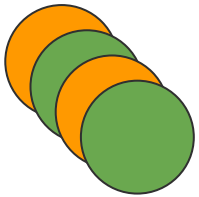
Unsupervised learning:

- Reveal relations between reviews, or between words appearing in these reviews

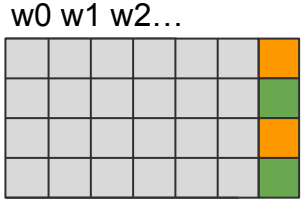
NLP+ML Pipeline for Sentiment Analysis



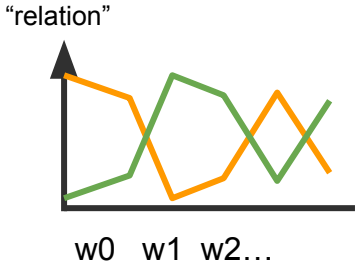
raw text +labels



bags of words +labels



vectors +labels

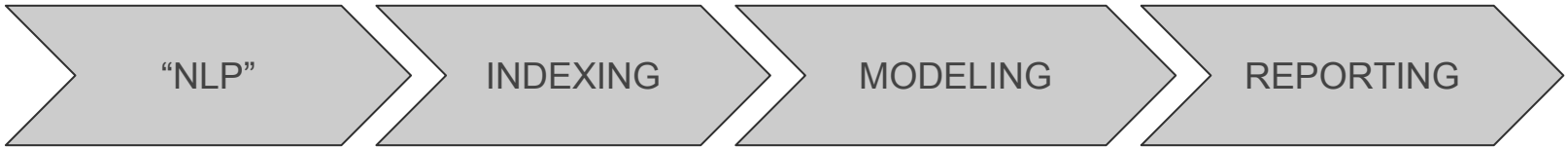


model

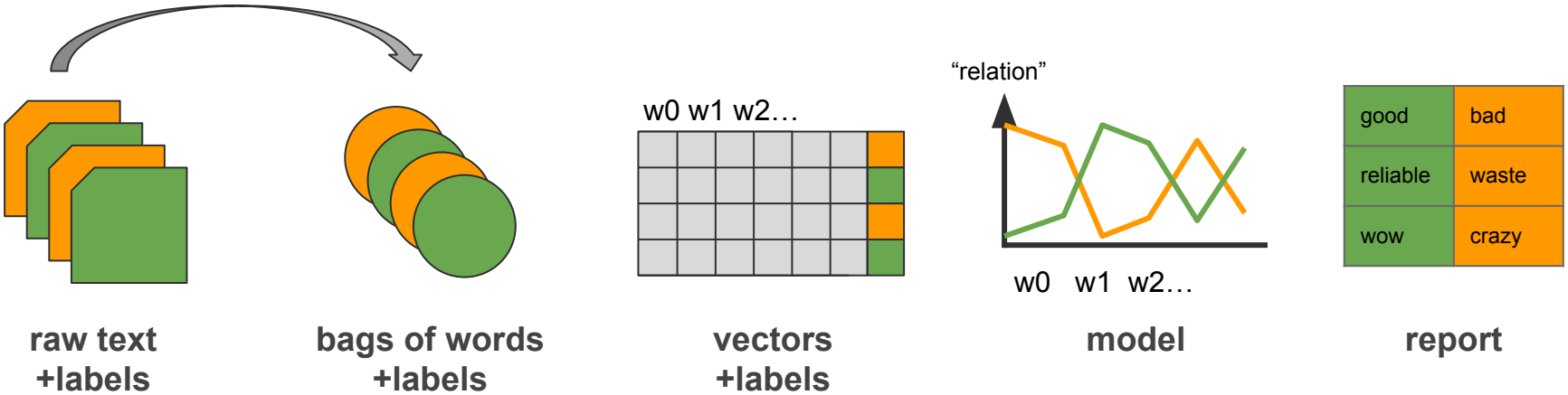
good	bad
reliable	waste
wow	crazy

report

NLP+ML Pipeline for Sentiment Analysis



walkthrough



raw text +labels

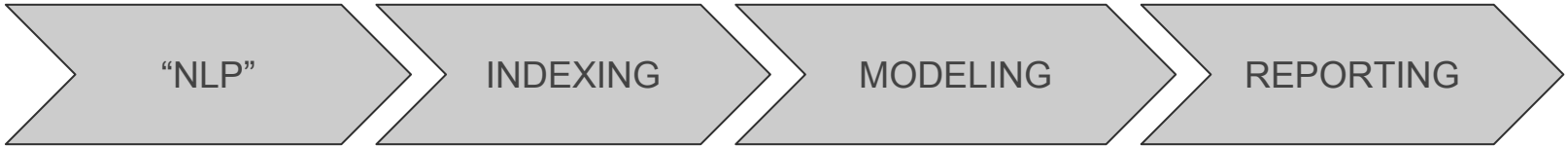
bags of words +labels

vectors +labels

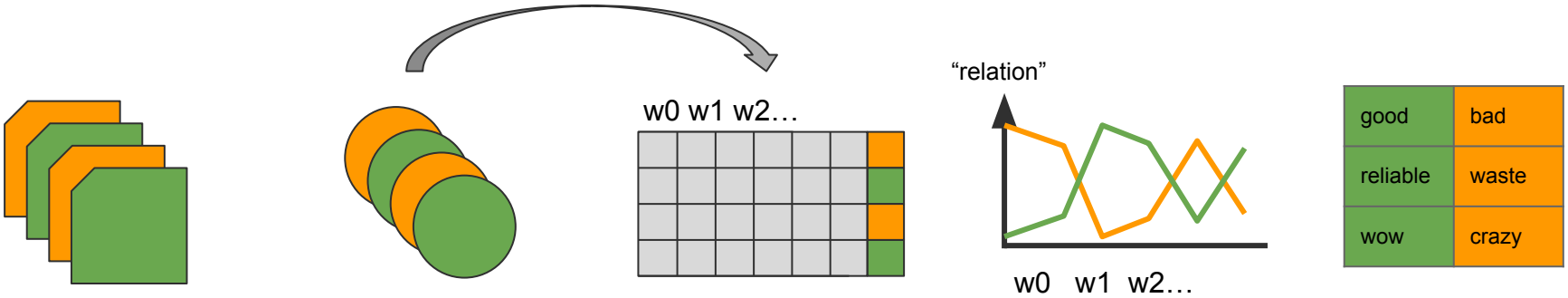
model

report

NLP+ML Pipeline for Sentiment Analysis



walkthrough



raw text +labels

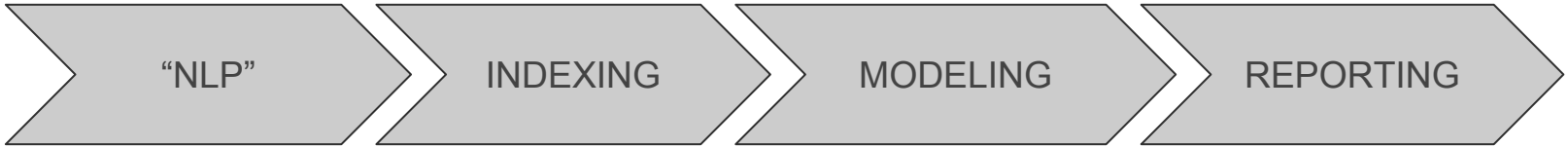
bags of words +labels

vectors +labels

model

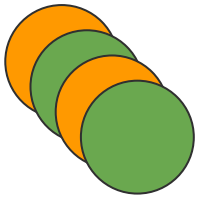
report

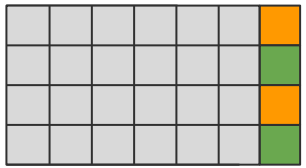
NLP+ML Pipeline for Sentiment Analysis

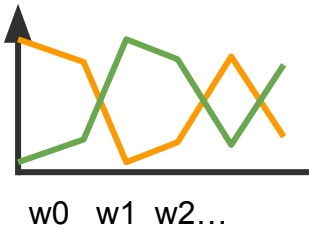


lecture+assignment

“relation”







good	bad
reliable	waste
wow	crazy

raw text +labels

bags of words +labels

vectors +labels

model

report

Naive Bayes (classification)

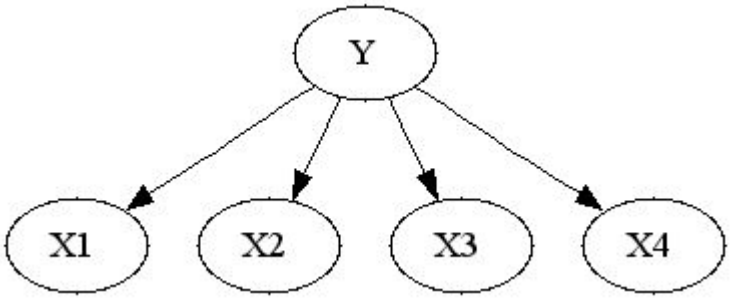
With

- w: word
- y: the **class** of a document
  (0 for neg, 1 for pos)

Given a dataset of labelled documents.

Probability $P(y \mid w_1, w_2, \ldots, w_p) = P(y)\frac{\prod_i P(w_i|y)}{\prod_i P(w_i)}$

Hint: use bayes rule ^^;



Keywords: Naive Bayes, Bayes rule, spam classification
Recommended reading: Naive Bayes explained

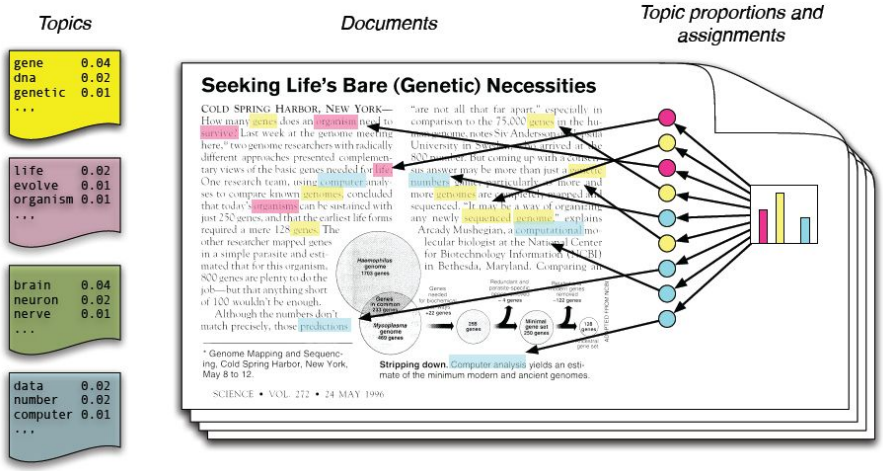
LDA : Latent Dirichlet Allocation (clustering)

With

- w: word
- d: document
- D: the corpus of docs
- k: a given number of topics

A **topic** is a distribution over words.
Each **document** is a mixture of corpus-wide topics.
Each **word** is drawn from these topics.

Find the k distributions that would likely “generate” every document d in D.



Keywords: LDA, Latent Semantic Analysis
Recommended reading: [Topic modeling for Humanists, a guided tour]

Word2Vec (dimensionality reduction)

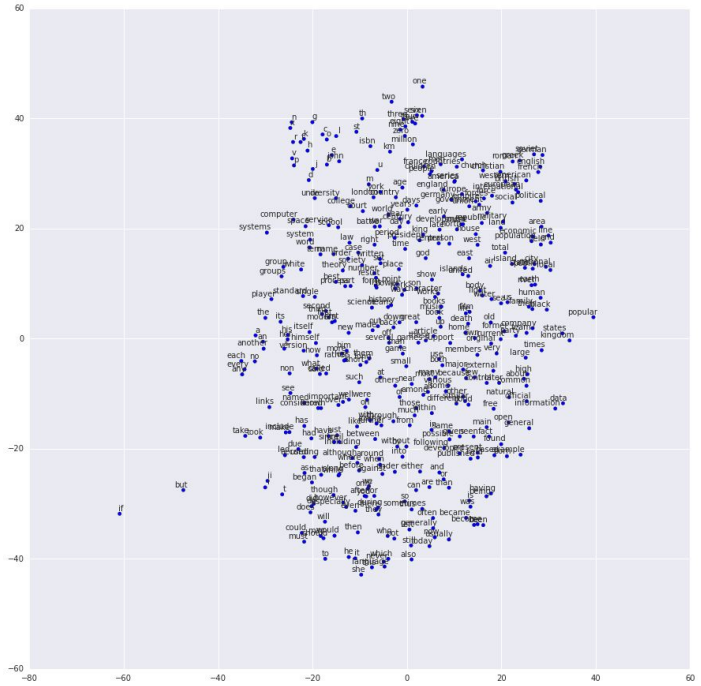
With

- w: word
- **c:context** (as a window surrounding w)
- D the corpus of docs

Model the **probability** $P(D = 1|w, c)$
that w in context c has been observed in D

Train on **positive examples** (from the observed corpus) and **negative examples** (made up)

Somehow analogous to the **matrix factorization** for finding latent features for words.



Keywords: Word2Vec, Skip-Gram with Negative Sampling (SGNS), Pointwise Mutual Information
Recommended reading: Neural Word Embeddings as Implicit Matrix Factorization. Omer Levy and Yoav Goldberg. NIPS 2014. [pdf]