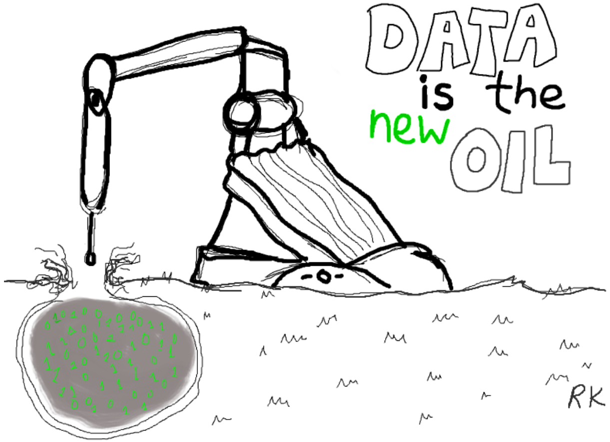
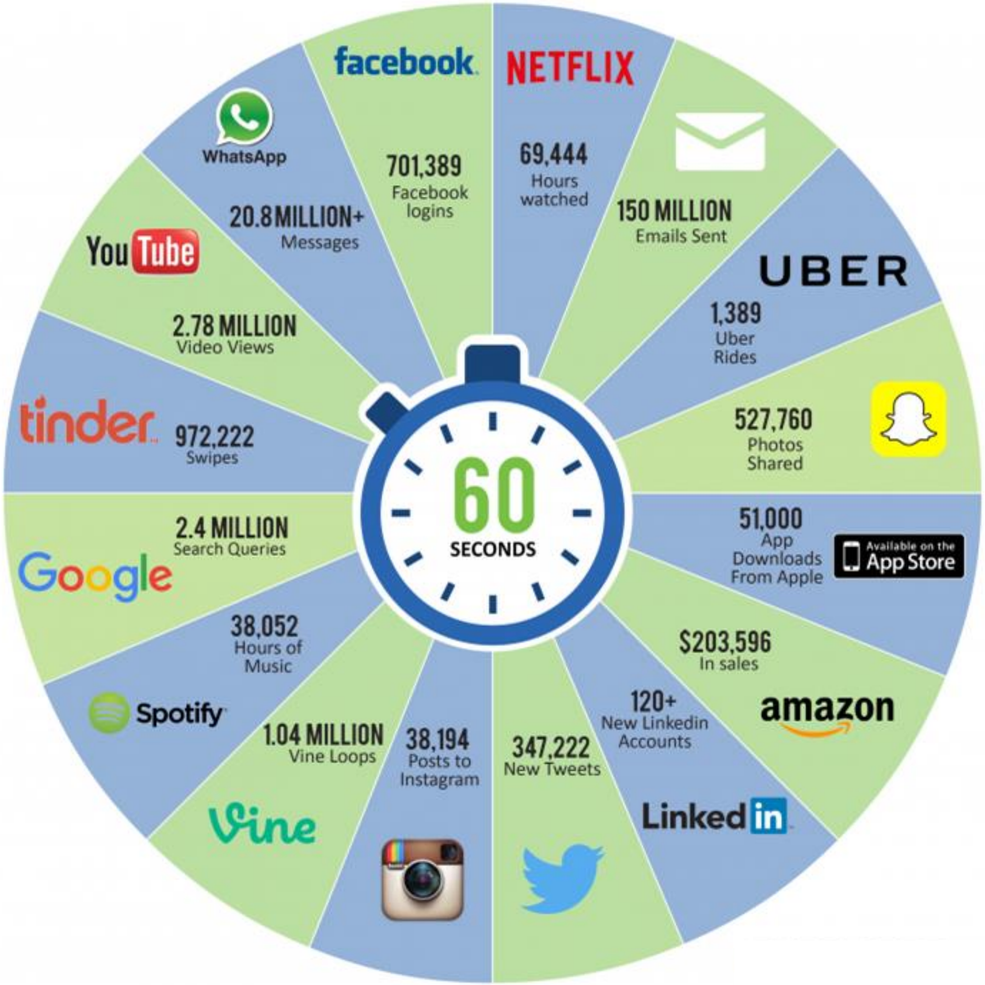
Big Data Science for Recommendation Systems



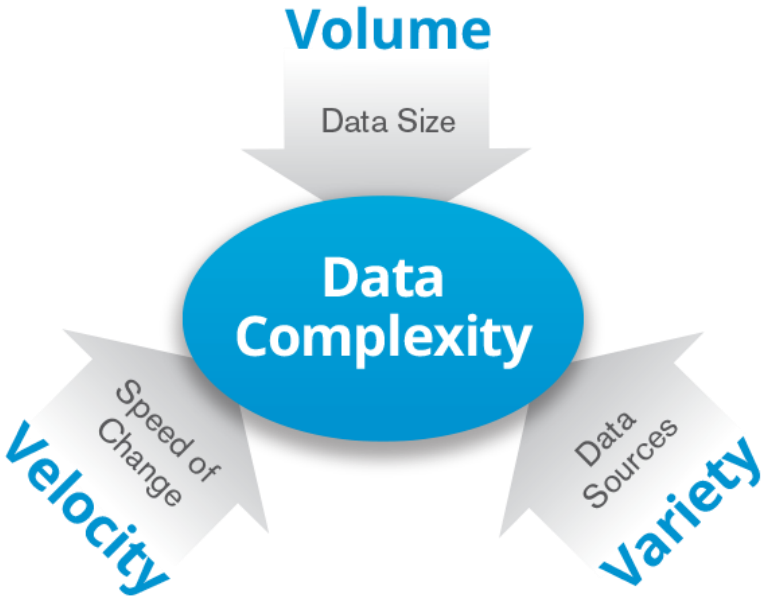
Miguel Costa
Computer Science Researcher, Lead Data Scientist @ Vodafone

Data Science in (Astro)Particle Physics and the bridge to industry
LIP - Laboratory of Instrumentation and Experimental Particle Physics
March 15, 2018

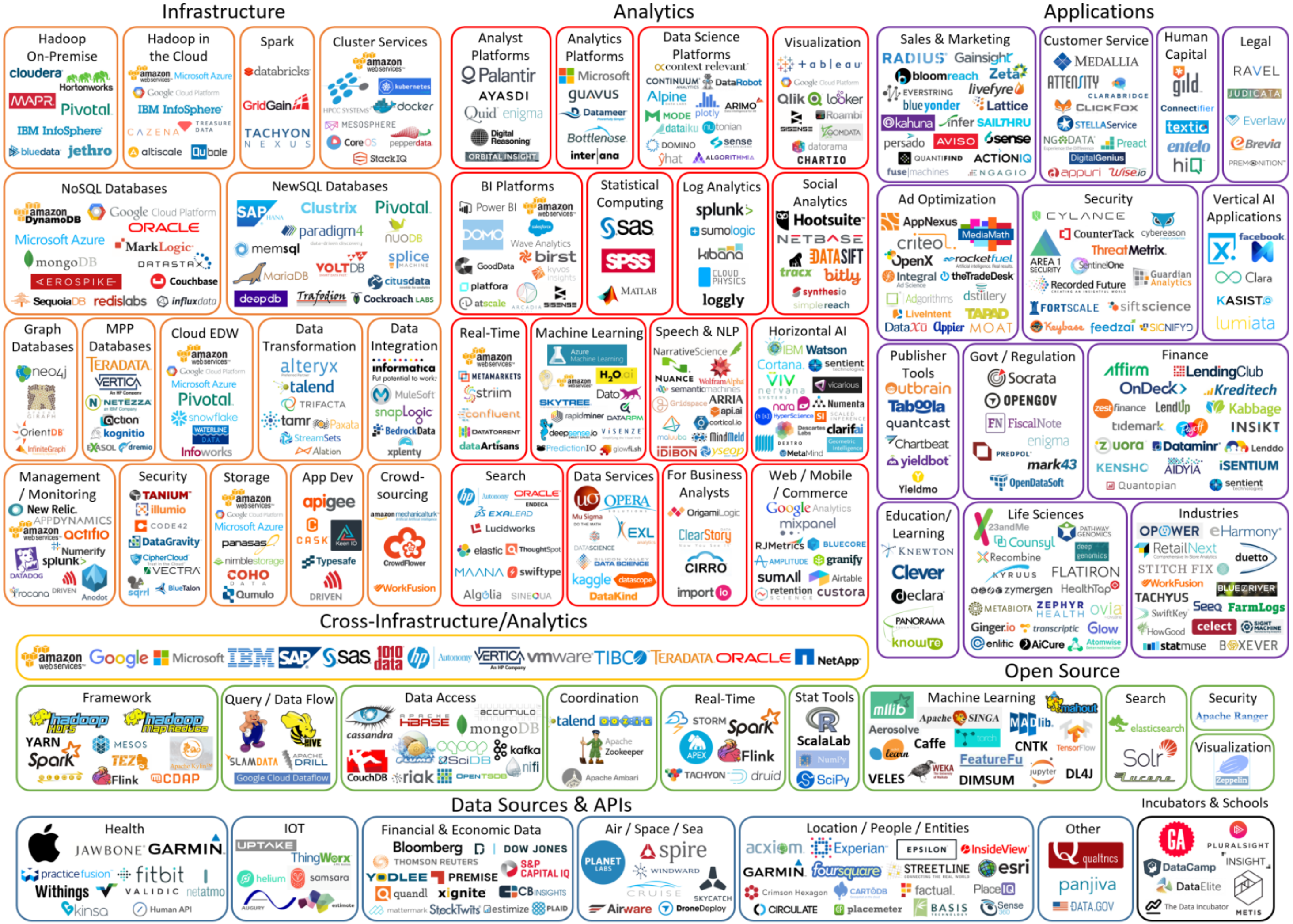
Big Data



"Big data is a term for data sets that are so large or complex that **traditional data processing applications are inadequate** to deal with them."
- Wikipedia

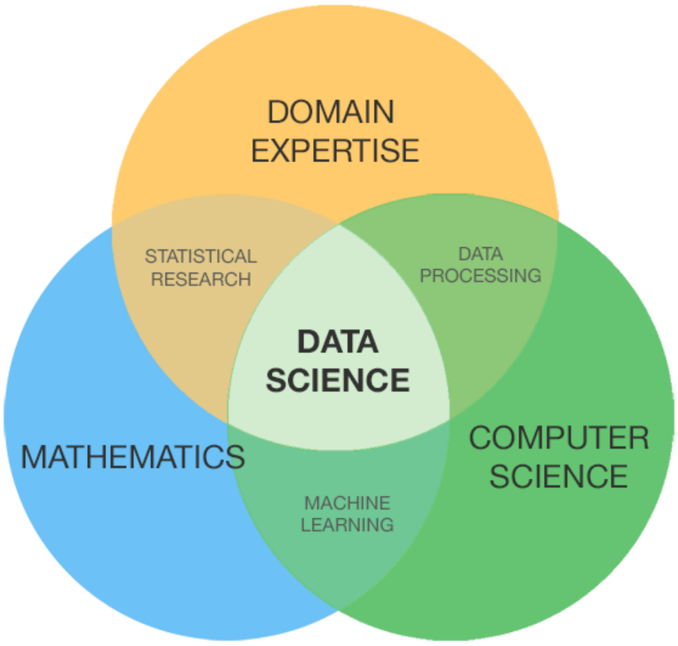


Big Data Tools



What is Data Science?

It is an interdisciplinary field about processes and systems to **extract knowledge or insights** from data in various forms … - *Wikipedia*



Source: Palmer, Shelly. Data Science for the C-Suite. New York: Digital Living Press, 2015. Print.

- Discovering what we don’t know from **data**
- Obtaining predictive and actionable insights from **data**
- Creating **data** products that have business impact
- Communicating relevant business stories from **data**
- Building confidence in decisions that drive business value based on **data**

Big Data + Data Science = Big Data Science

Search engines

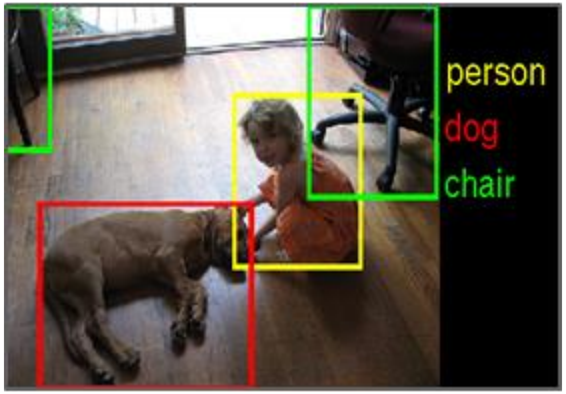
bing YAHOO! Google

Recommendation systems

NETFLIX

Personal assistants

Computer Vision

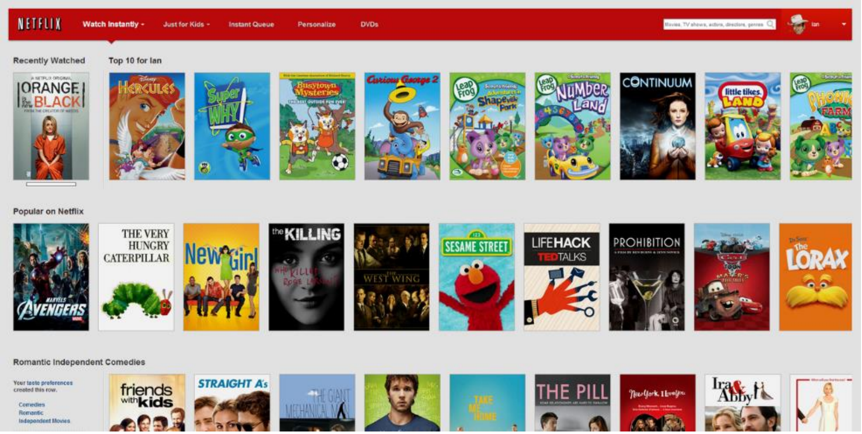


Speech translators

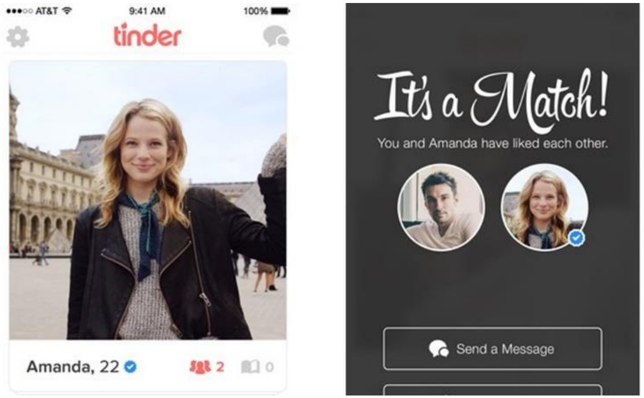
Beating humans …

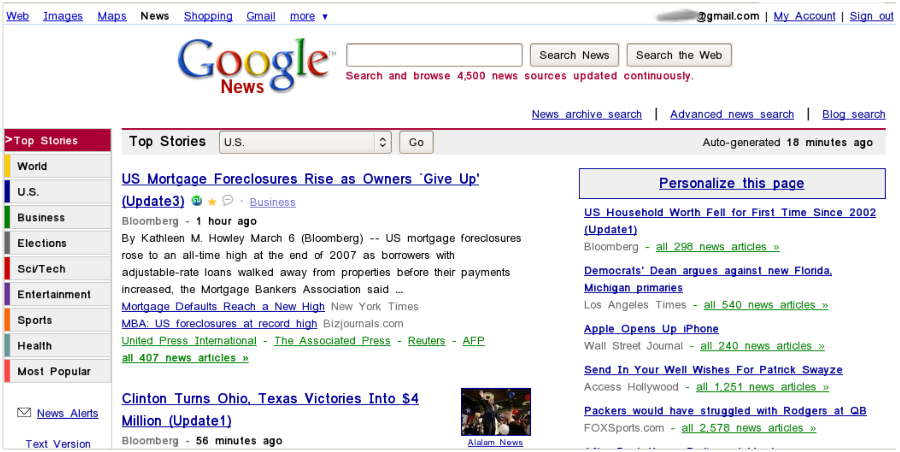
Examples of Recommendation Systems

35% of sales come from recommendations



2/3 of the movies watched are recommended





38% more clicks due to recommendations

Recommendation Systems for Video Content (& other items)

Problem

- Information overflow
  - too many video contents from which to choose
  - too much time exploring video contents

(thousands of programs broadcast in hundreds of TV channels, plus thousands of movies & series on VOD)

> If a typical subscriber doesn't find something to watch in about **60 to 90 seconds**, they could lose interest and move on to something else.
>
> Source: The Netflix Recommender System: Algorithms, Business Value, and Innovation, 2016

- Impact
  - dissatisfaction
  - change to other systems with recommendations (e.g. Netflix, Youtube)
  - less visualization time
  - less revenue
  - churn

Solution Approach (5 steps)

1. Extraction of implicit or explicit feedback for each pair <user, content>
   – Get preferences of what users like to watch
2. Feature engineering
   – Get signals that quantify how much a user likes a TV content
3. Creation of a large-scale dataset for learning & evaluation
   – Compile all examples with signals and preferences

4. Creation of a recommendation model
   – Learn a model using the large-scale dataset
5. Evaluation (offline & online)
   – Quantify how good are the recommendations provided by the model

Step 1

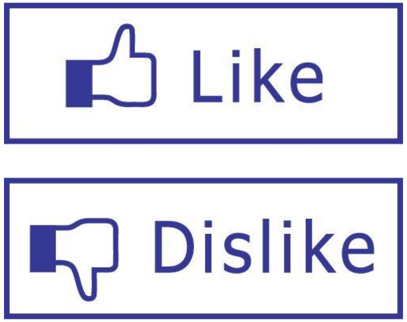
Extraction of explicit & implicit feedback
(get user preferences)

We assume users like/dislike a TV content if they:

- explicitly rate the content
- implicitly watch the content more than x% or more than y minutes
- implicitly record the content
- implicitly rewind and watch the content from the beginning

Explicit feedback provides more trustable data.

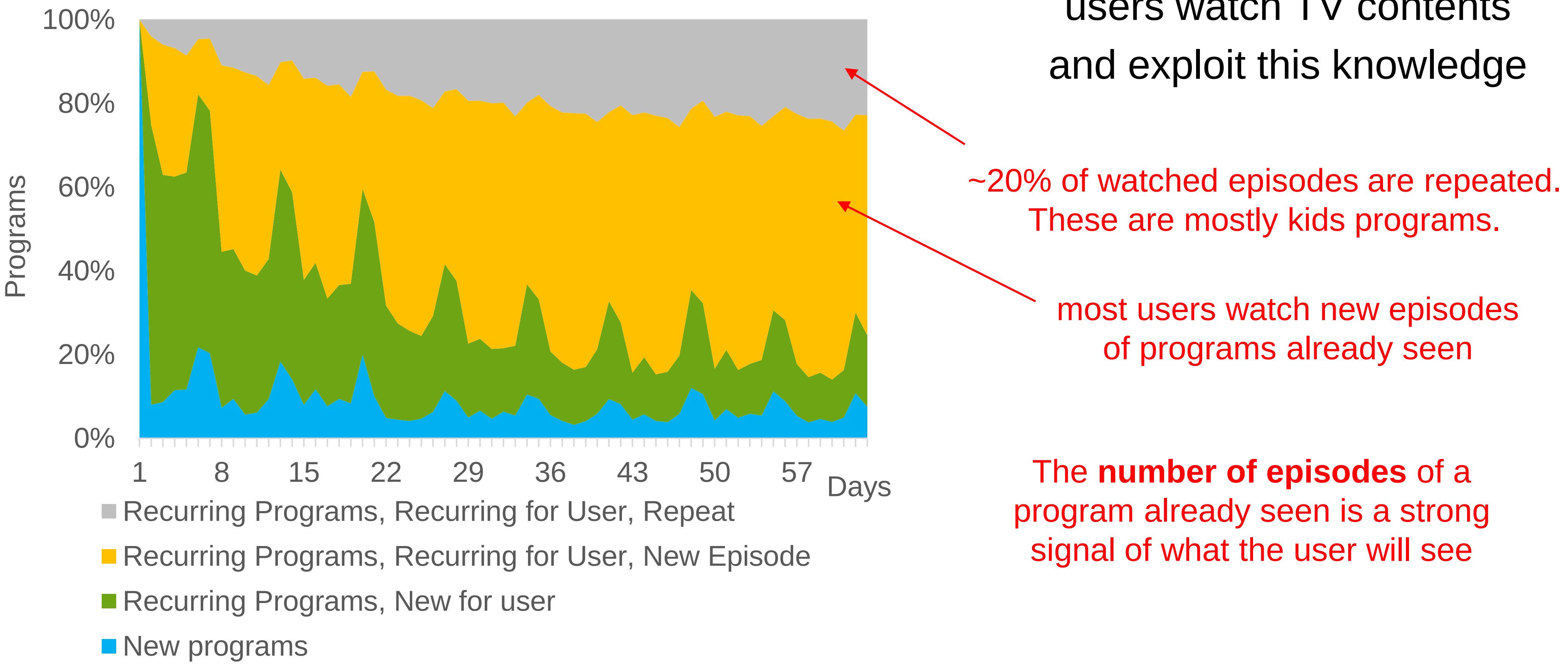
Implicit feedback provides much more data.



Step 2

Feature engineering: business context

Distribution of program types watched by users

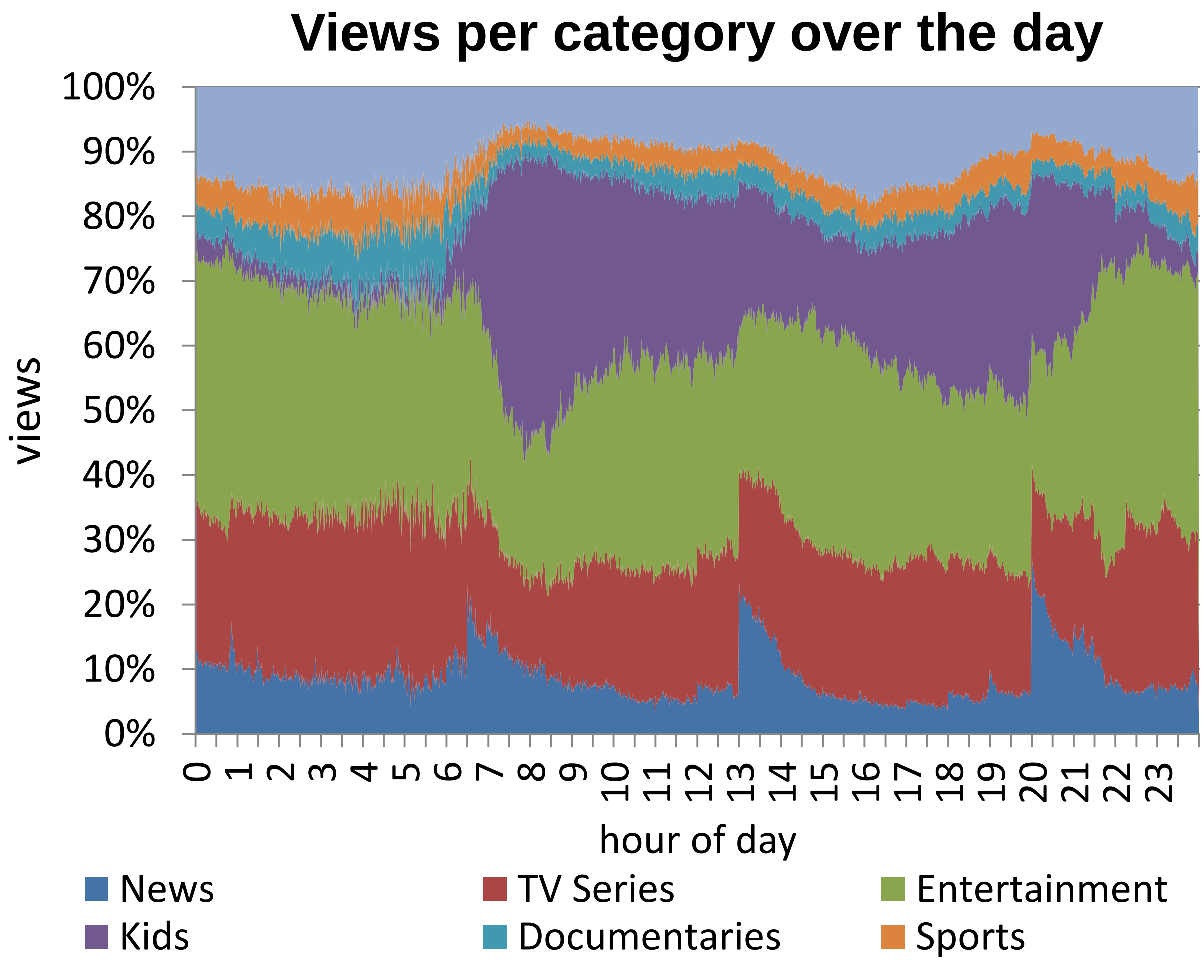


Understand how users watch TV contents and exploit this knowledge

The **number of episodes** of a program already seen is a strong signal of what the user will see

Step 2

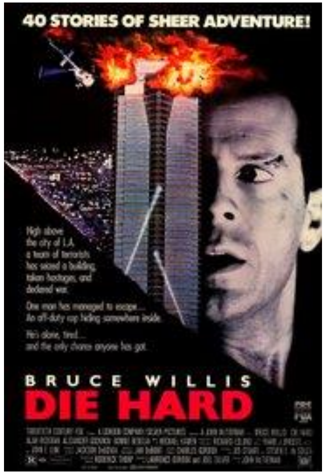
Feature engineering: business context



users watch different categories in different hours

The **hour of day** is a strong signal of what the user will see

Feature engineering: content-based filtering



Die Hard
2h11min | Action, Thriller | 1988
Director:
John McTiernan
Writers:
Roderick Thorp (novel), Jeb Stuart (screenplay)
Stars:
Bruce Willis, Alan Rickman, Bonnie Bedelia

Recommend contents similar to the contents that the user liked in the past

The Last Boy Scout
1h45min | Action, Thriller | 1991
Director:
Tony Scott
Writers:
Shane Black (story), Greg Hicks (story)
Stars:
Bruce Willis, Damon Wayans, Chelsea Field

Metadata similarity:

Close years

Same category

Same star

Step 2

Feature engineering: content-based filtering

Textual similarity:

Sentence 1: William Wallace begins a revolt against King Edward I of England.

Sentence 2: Braveheart fought against Edward Longshanks.

$$\mathrm{Jaccard}(\mathcal{S}_1, \mathcal{S}_2) = \frac{|\mathcal{S}_1 \cap \mathcal{S}_2|}{|\mathcal{S}_1 \cup \mathcal{S}_2|}$$

$$\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \times \log\left(\frac{N}{\mathrm{df}_i}\right)$$

$\mathrm{tf}_{i,j}$ = # of occurences of i in j

$\mathrm{df}_{i,}$ = # of sentences with i

N = # sentences

Semantic similarity:

King Edward I = Edward Longshanks

William Wallace = Braveheart

Step 2

Feature engineering: collaborative filtering

User-based: recommends contents that similar users liked

- People who agreed in the past are likely to agree again

Item-based: recommends similar contents that the user liked

- A user is likely to have the same opinion for similar items

	Item 1	Item 2	Item 3	Item 4
User 1	5	3	5	1
User 2	2	1	2	1
User 3	4	2	?	1
User 4	1	4	2	3

How to measure similarity?

similar users

similar items

Feature engineering: social context

We are likely to share interests and preferences with our friends (homophily)

&

Users can be easily influenced by the friends they trust

	Item 1	Item 2	Item 3	Item 4
User 1	5	3	5	1
User 2	2	1	2	1
User 3	4	2	?	1
User 4	1	4	2	3

friends

similar users

Step 2

Feature engineering: matrix factorization (MF)

	Item 1	Item 2	Item 3	Item 4
User 1	?	?	5	?
User 2	2	?	2	1
User 3	?	2	?	1
User 4	1	?	?	?

Sparse matrix with millions of users and items

N

	Factor 1	Factor 2
User 1		
User 2		
User 3		
User 4		

*

	Item 1	Item 2	Item 3	Item 4
Factor 1				
Factor 2				

N

MF uncover the most relevant latent dimensions

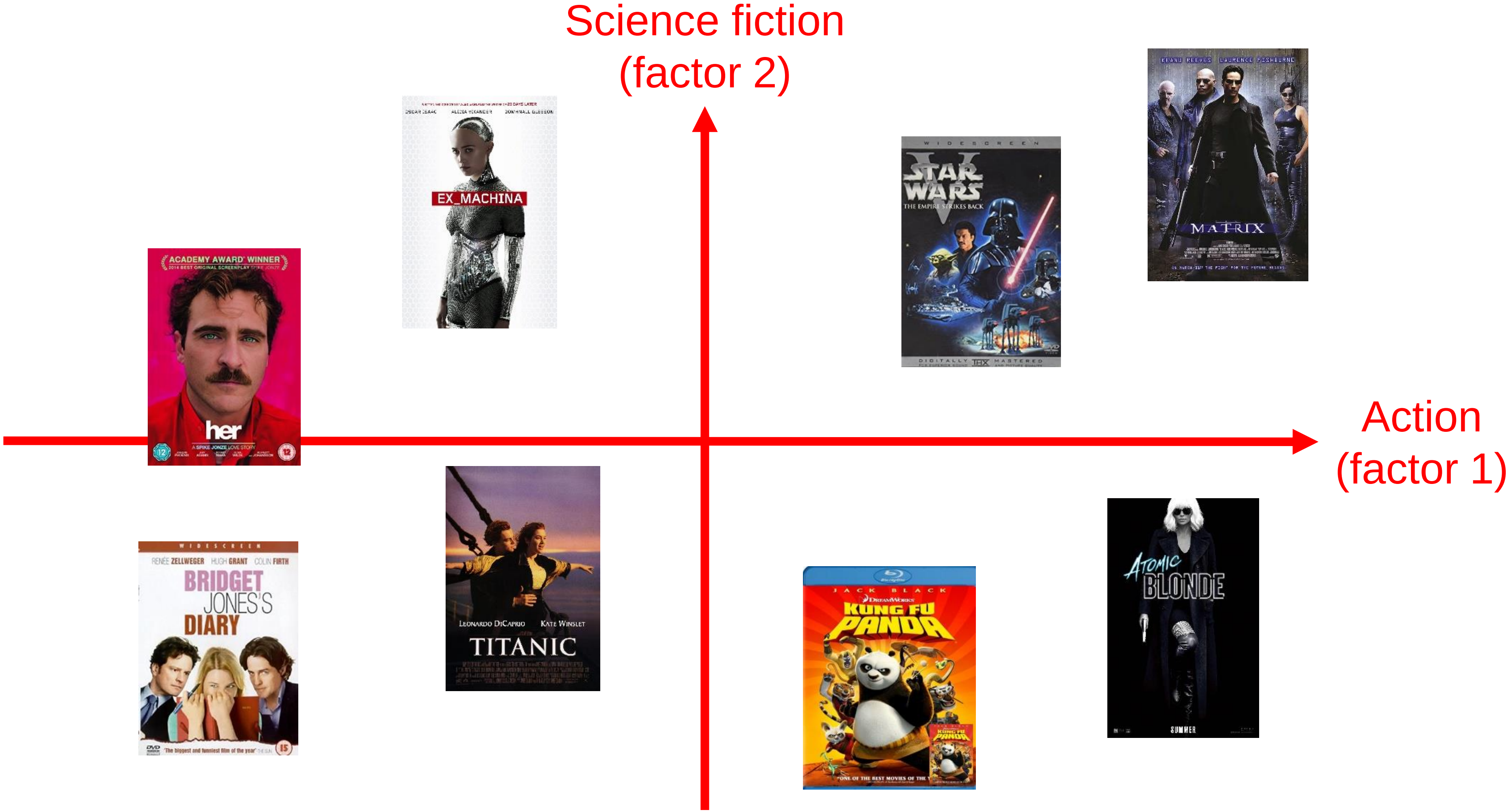
item latent vector

user latent vector

$$rating_{u,I} = user_u^T * item_i$$

Step 2

Feature engineering: latent factors example

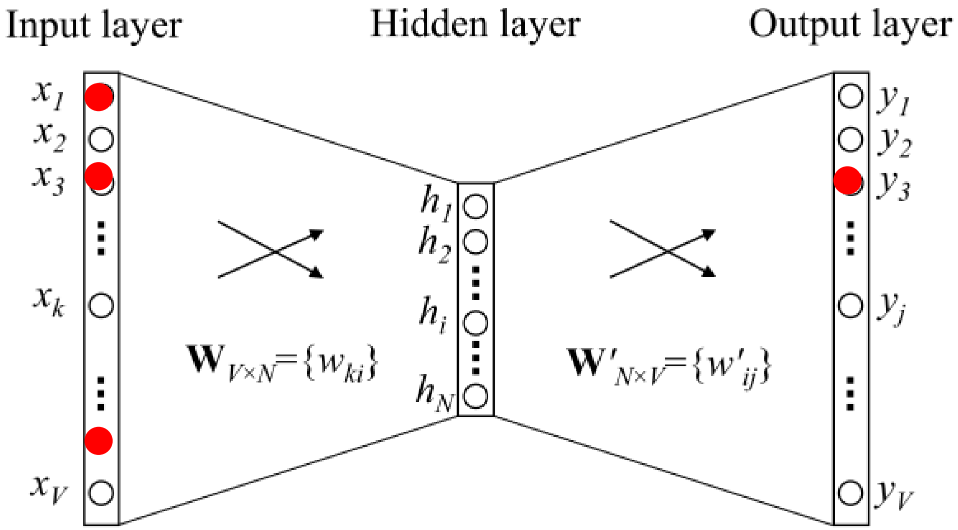


Step 2

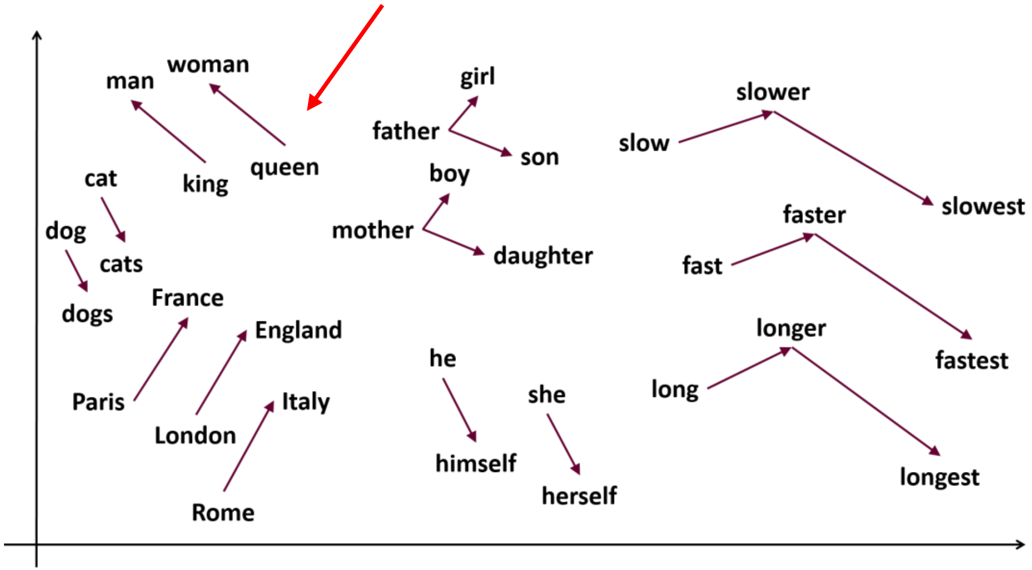
Feature engineering: word2vec & item2vec

The context (e.g. adjacent words/items) is used to create **embeddings** that can be used to measure similarity and infer semantic relations.

Two algorithms: Continous Bag of Words (CBOW) & Skip-gram

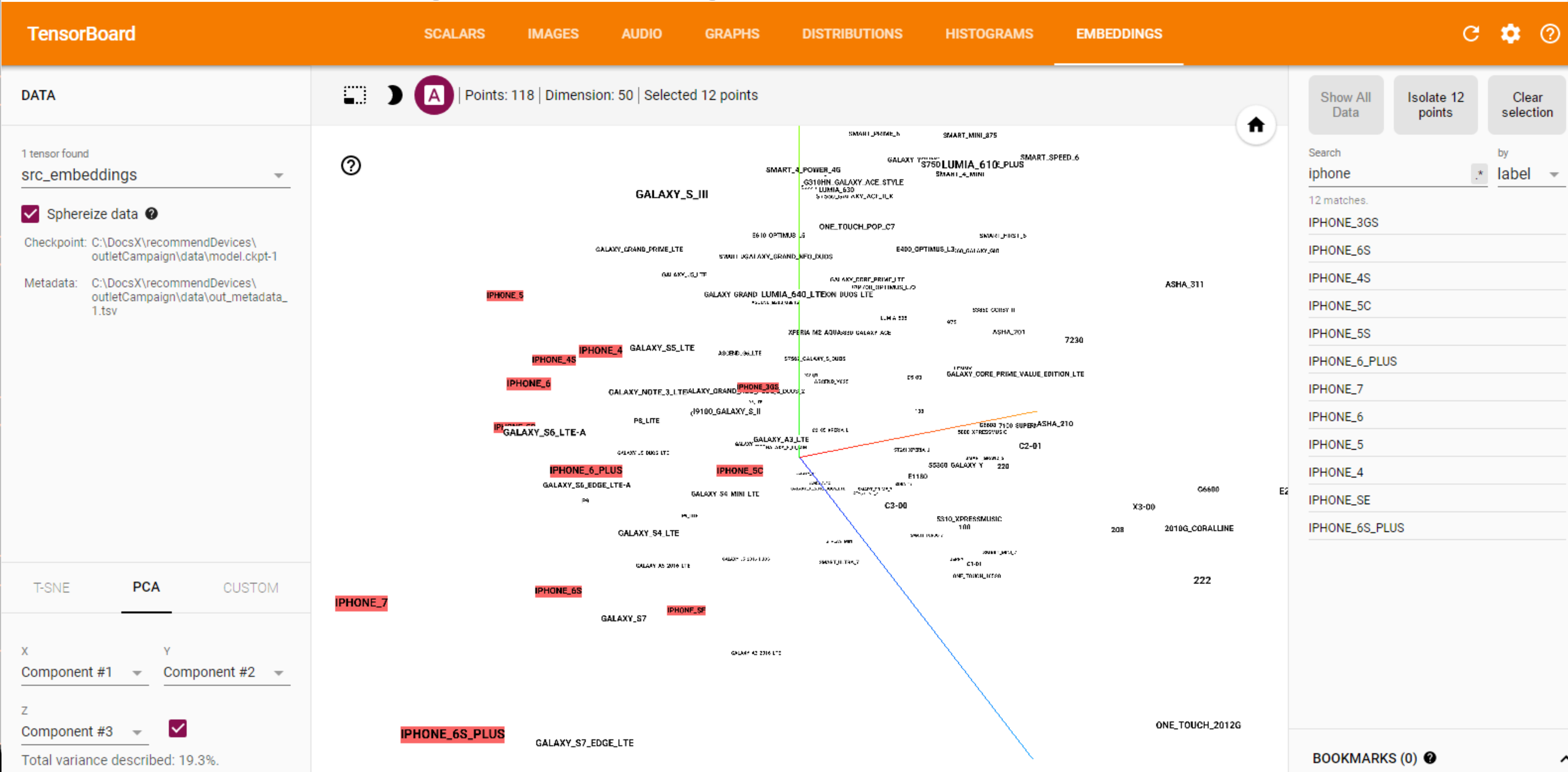


man – king + woman = queen



Step 2

Feature engineering: item2vec example



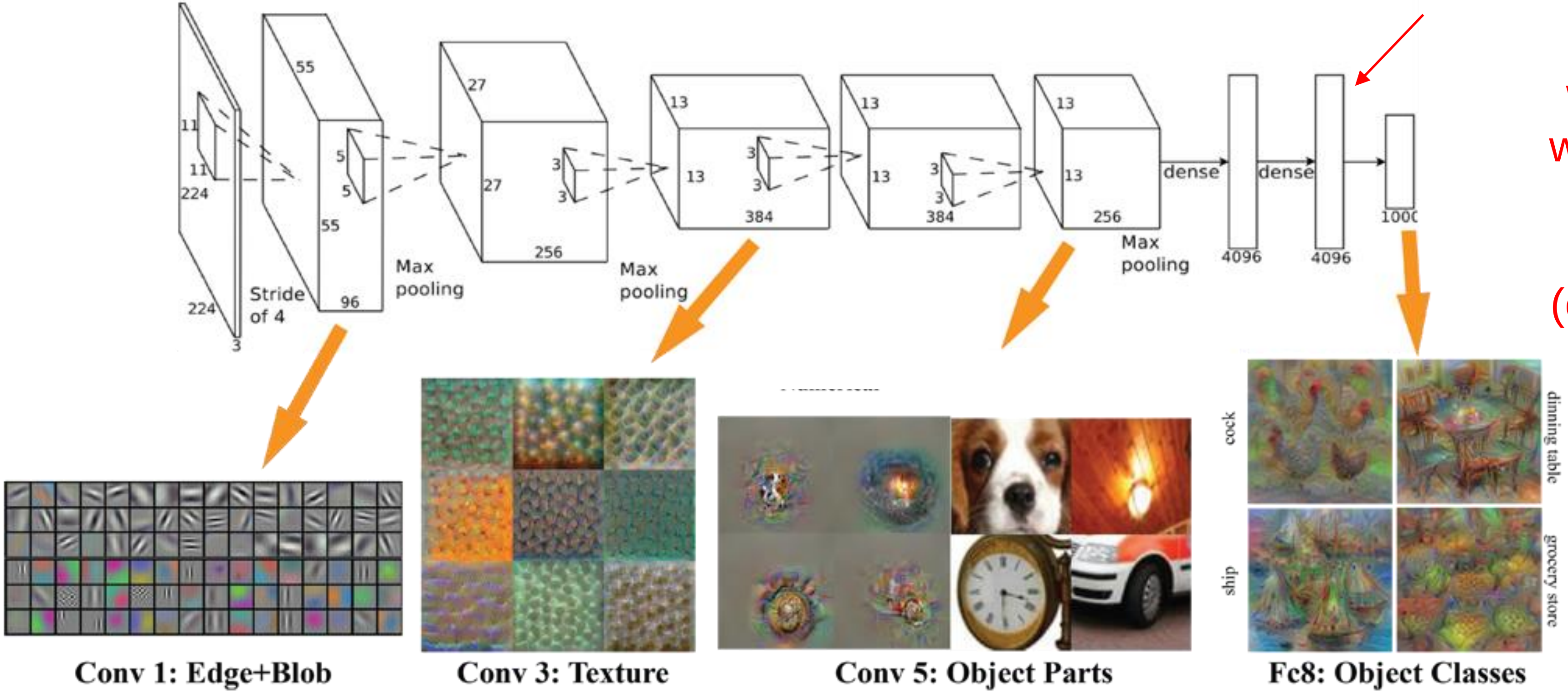
Step 2

Feature engineering: deep learning (CNN)

use one off-the-shelf model (e.g. ResNet-50 pre-trained on ImageNet dataset) and retrain the last-layers with your examples

use this fully connected (dense) layer to extract image embeddings

works better when learning with pairs of images (e.g. Siamese CNN)



AlexNet network learned from ImageNet dataset and visualized with mNeuron.

Step 2

Feature engineering: CNN embeddings example

Women's clothing embeddings extracted from AlexNet network & visualized with t-SNE in 2D

Step 3

Creation of a large-scale dataset

User ID	Item ID	Feature 1	Feature 2	Feature 3	Feature 4	…	Label
1012321	8643244	0.5	0.56	55	4544		LIKE
1232335	1344690	0.3	0.23	3	---		DISLIKE
1023877	0456431	0.1	---	23	345		DISLIKE
2234432	9343990	1.0	0.0	45	0		LIKE

Do not forget to:

- “clean” the data
- handle missing values & outliers
- normalize/standardize data
- anonymize data

Feature domains:

Business Context	Content-based filtering
Social Context	Collaborative filtering
Demographics	Others

Some datasets have **millions** of users and items with **thousands** of features.

Conclusions

Big data science is revolutionizing the world. Join the revolution.
Recommendations are used in numerous systems to improve business.

To build a recommendation system you should follow these steps:

1. Extraction of implicit or explicit feedback for each pair <user, item>
2. Feature engineering
3. Creation of a (large-scale) dataset for learning & evaluation
4. Learning a recommendation model
5. Evaluation

Questions?

Thank you.

We are looking for interns & collaborations.

miguel.costa2@vodafone.com

http://www.vodafone.com/content/bigdata/