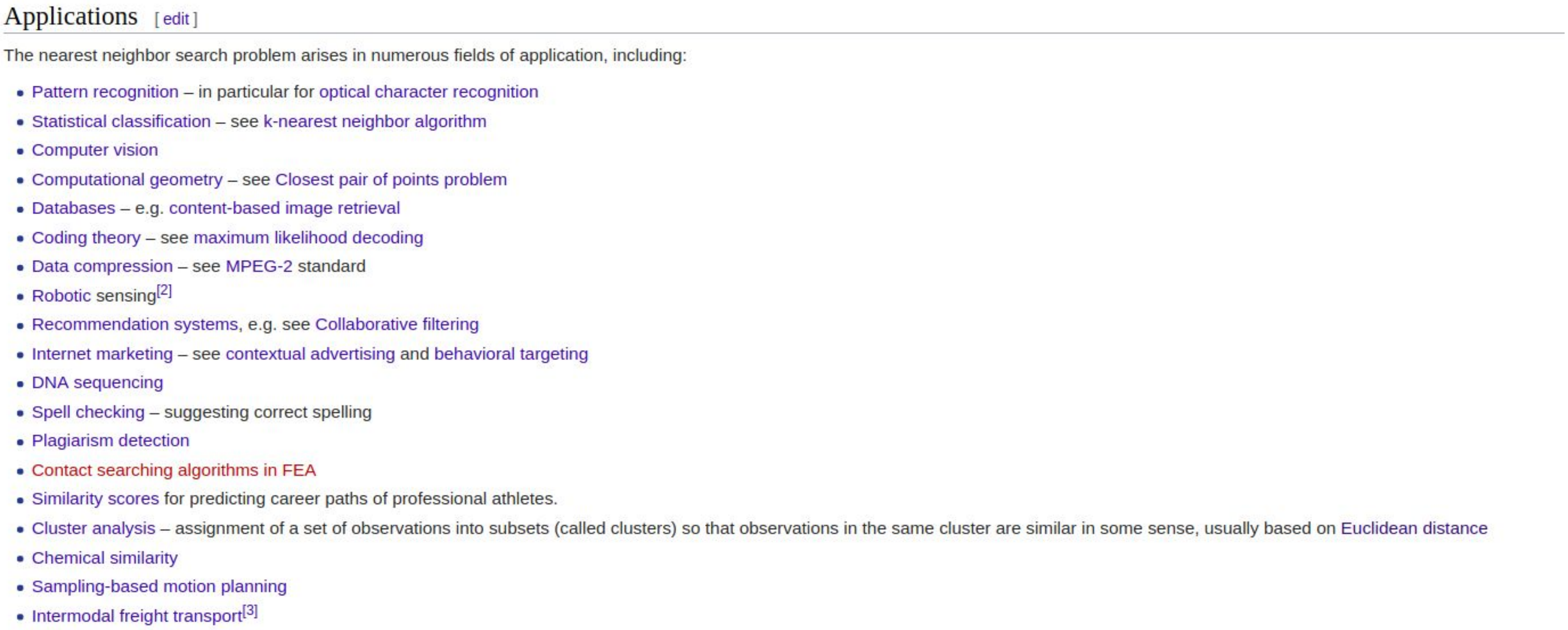
Nearest neighbor search

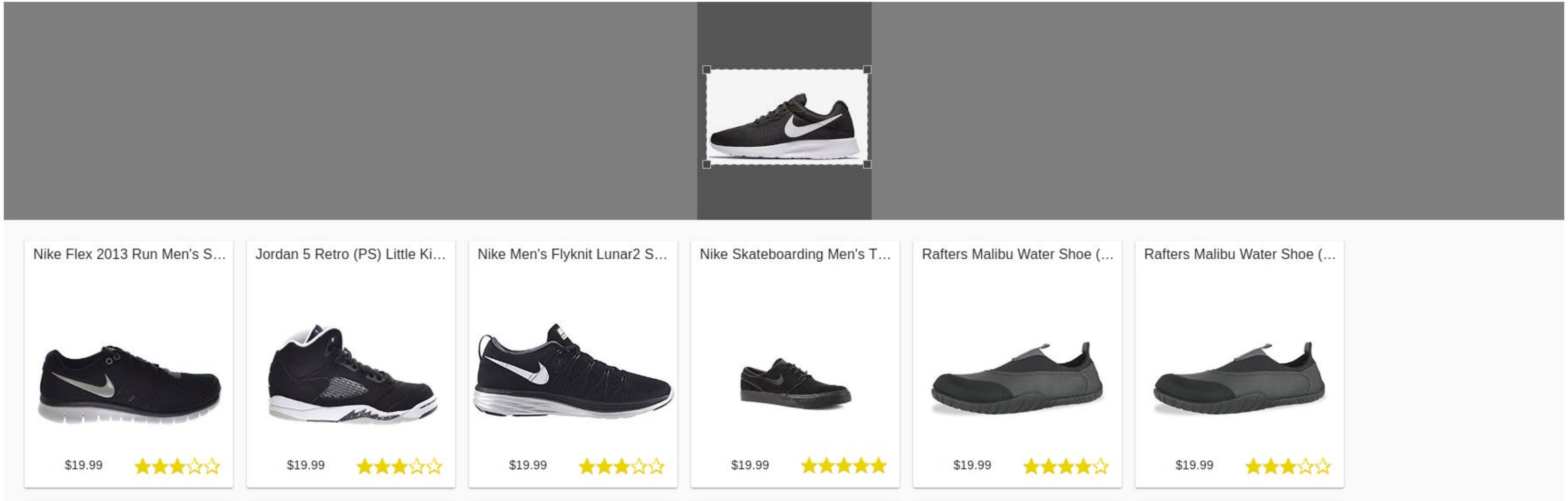


Applications [edit]

The nearest neighbor search problem arises in numerous fields of application, including:

- Pattern recognition – in particular for optical character recognition
- Statistical classification – see k-nearest neighbor algorithm
- Computer vision
- Computational geometry – see Closest pair of points problem
- Databases – e.g. content-based image retrieval
- Coding theory – see maximum likelihood decoding
- Data compression – see MPEG-2 standard
- Robotic sensing[2]
- Recommendation systems, e.g. see Collaborative filtering
- Internet marketing – see contextual advertising and behavioral targeting
- DNA sequencing
- Spell checking – suggesting correct spelling
- Plagiarism detection
- Contact searching algorithms in FEA
- Similarity scores for predicting career paths of professional athletes.
- Cluster analysis – assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense, usually based on Euclidean distance
- Chemical similarity
- Sampling-based motion planning
- Intermodal freight transport[3]

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



Nike Flex 2013 Run Men's S…
$19.99 ★★★☆☆

Jordan 5 Retro (PS) Little Ki…
$19.99 ★★★☆☆

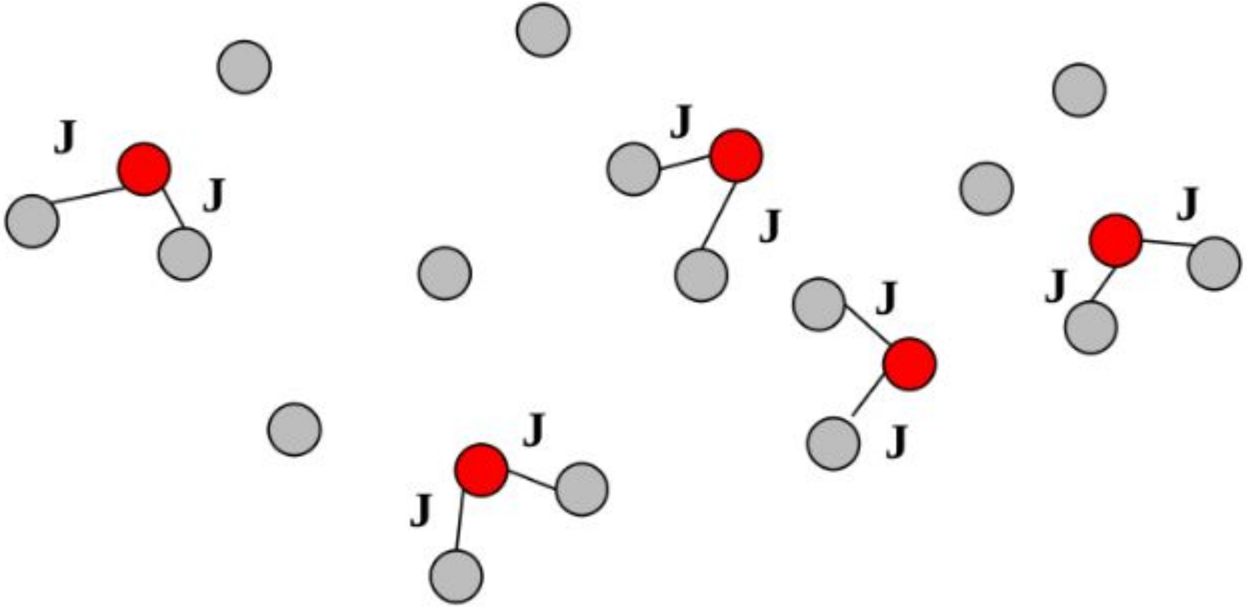
Nike Men's Flyknit Lunar2 S…
$19.99 ★★★☆☆

Nike Skateboarding Men's T…
$19.99 ★★★★★

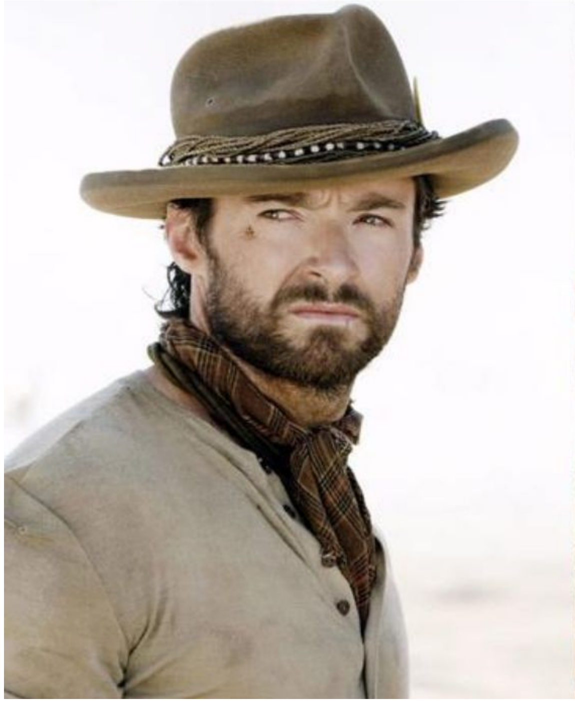
Rafters Malibu Water Shoe (…
$19.99 ★★★★☆

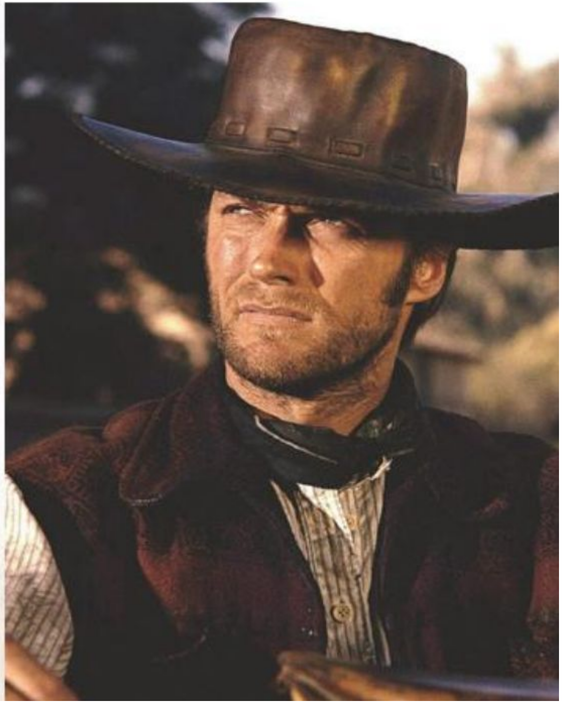
Rafters Malibu Water Shoe (…
$19.99 ★★★☆☆

https://thomasdelteil.github.io/VisualSearch_MXNet/


J
J
J
J
J
J
J
J
J
J

Look aLike





What is profile advertising?

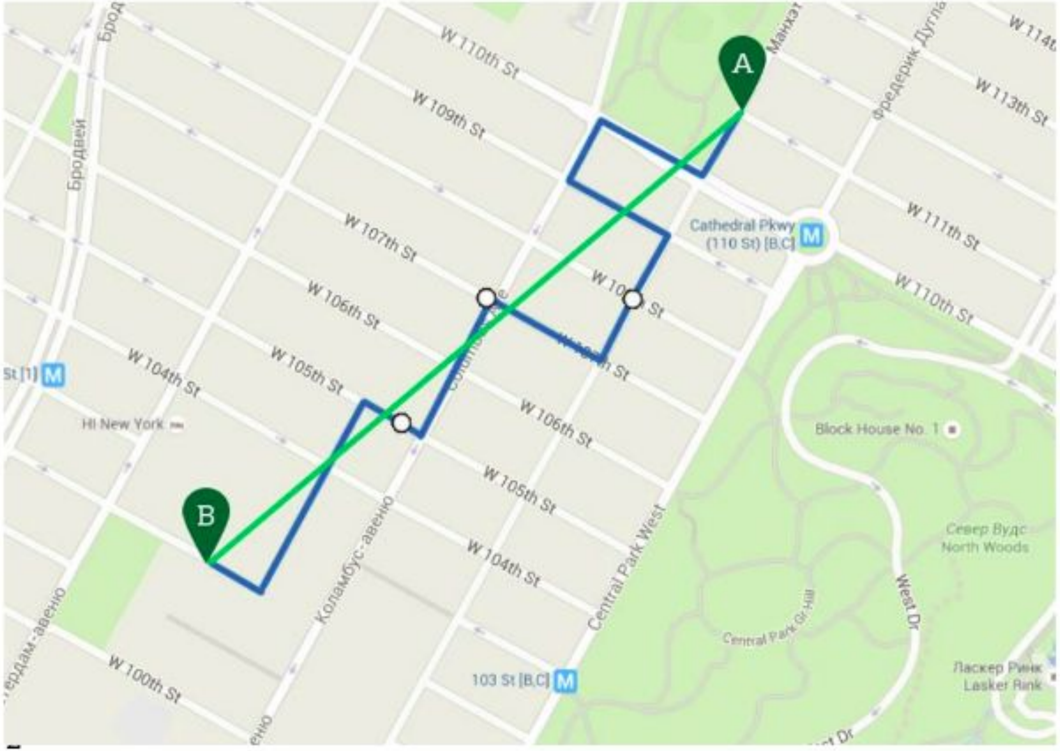
kaggle.com

westerns.ru

guns.ru

habrahabr.ru

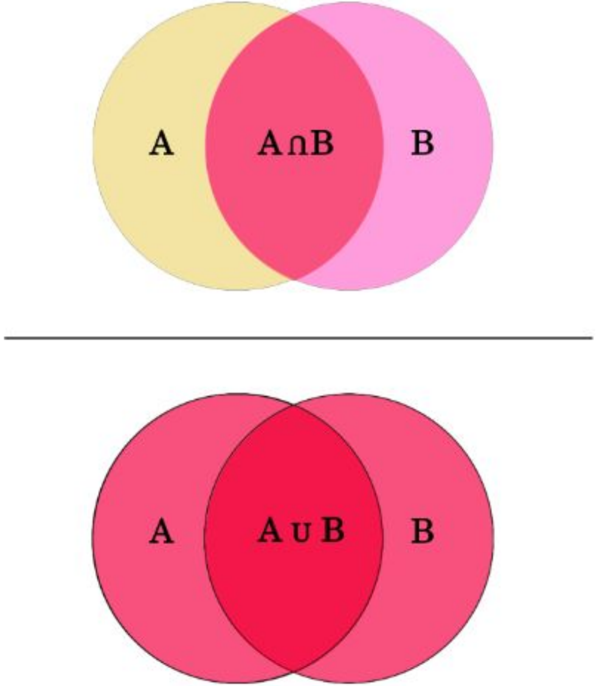
Manhattan and Euclidean distance



$$d_1(\mathbf{p},\mathbf{q}) = \|\mathbf{p}-\mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

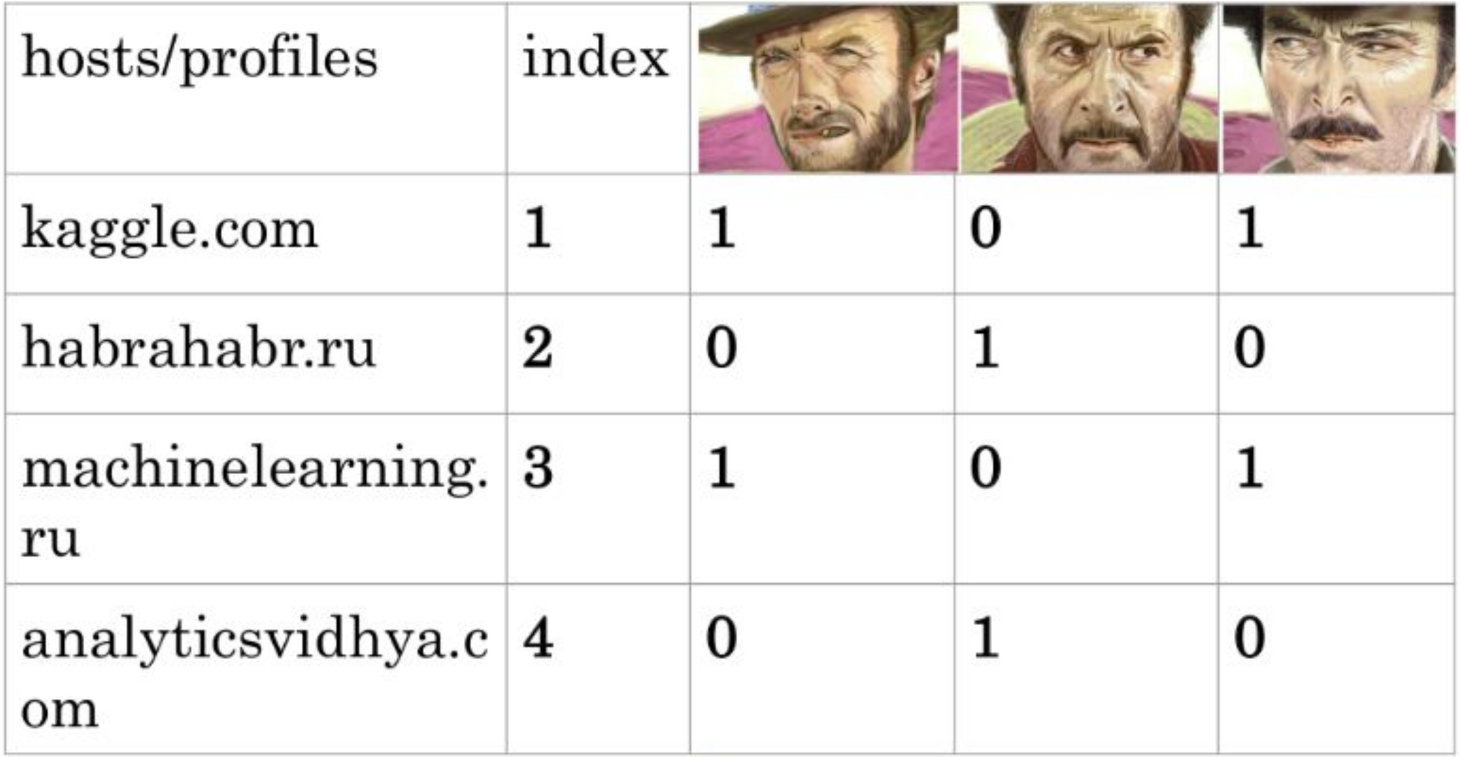
$$\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$

Jaccard similarity

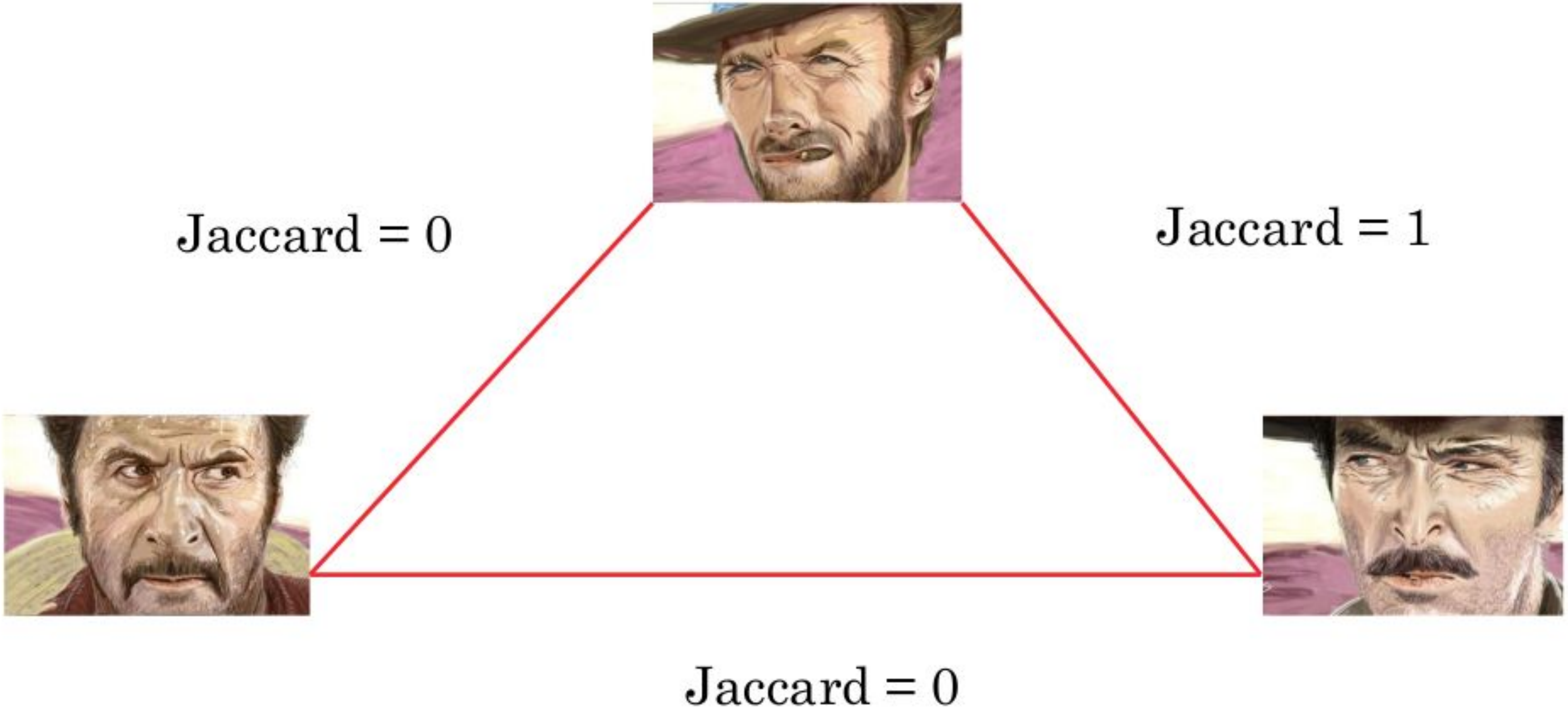


Minhash

Profile representation



hosts/profiles	index			
kaggle.com	1	1	0	1
habrahabr.ru	2	0	1	0
machinelearning.ru	3	1	0	1
analyticsvidhya.com	4	0	1	0

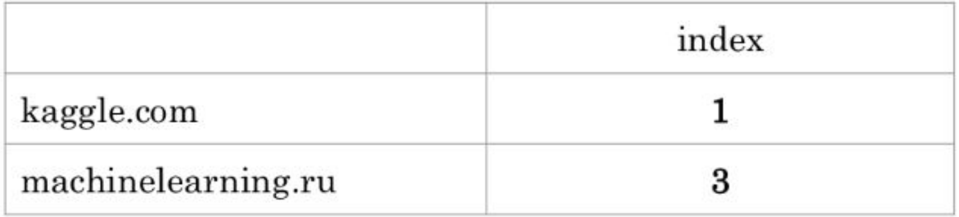


Jaccard = 0

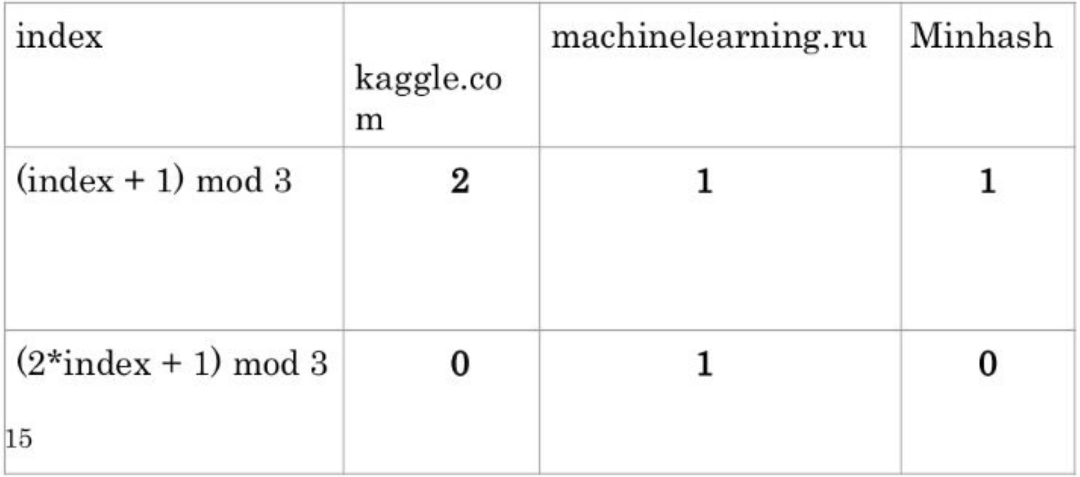
Jaccard = 1

Jaccard = 0

Hash functions

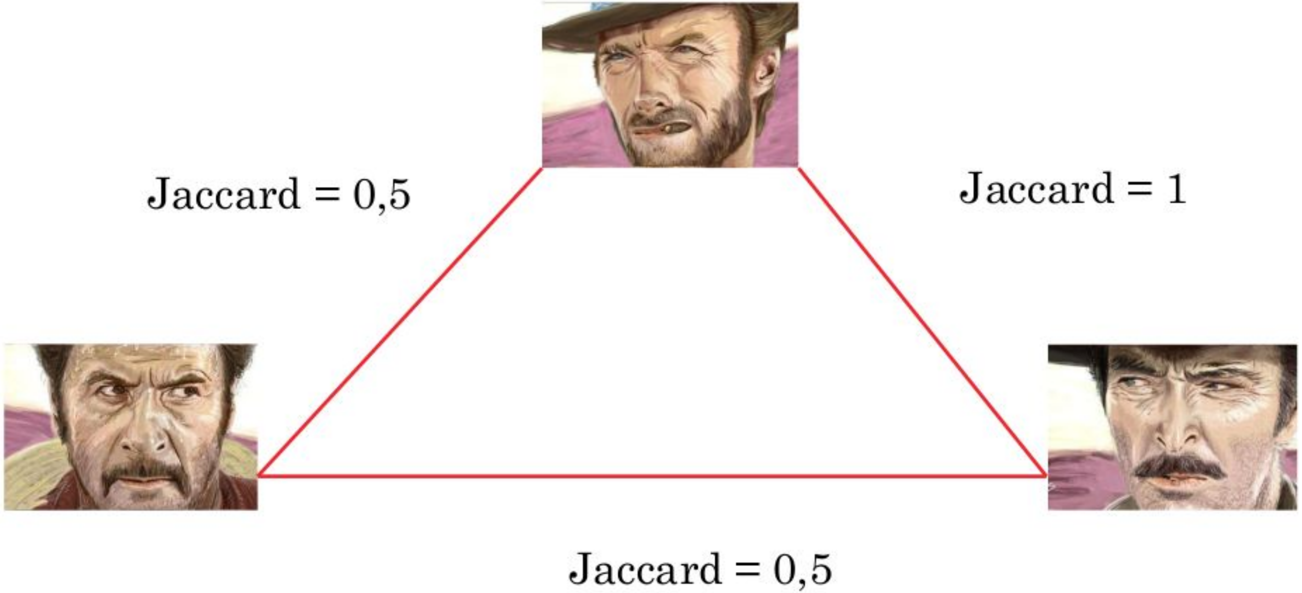


	index
kaggle.com	**1**
machinelearning.ru	**3**



index	kaggle.com	machinelearning.ru	Minhash
(index + 1) mod 3	**2**	**1**	**1**
(2*index + 1) mod 3	**0**	**1**	**0**

15

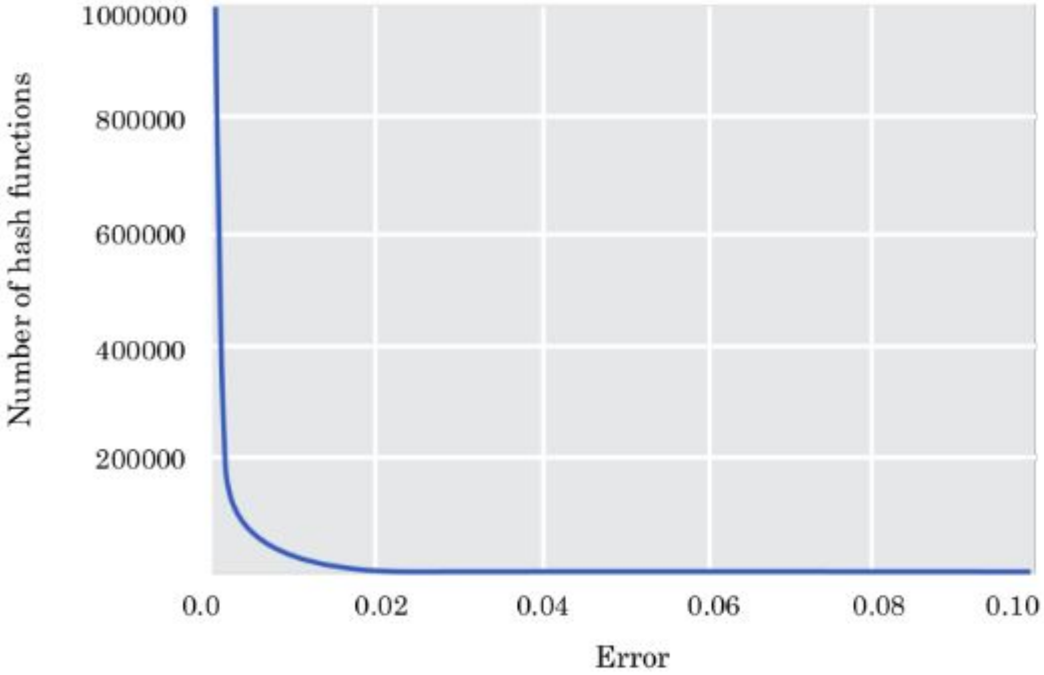


Jaccard = 0,5

Jaccard = 1

Jaccard = 0,5

How to choose number of hash functions?



$$k = [1/\epsilon^2]$$

How to choose parameters for hash functions?

$$h(x) = (ax + b) \bmod c$$

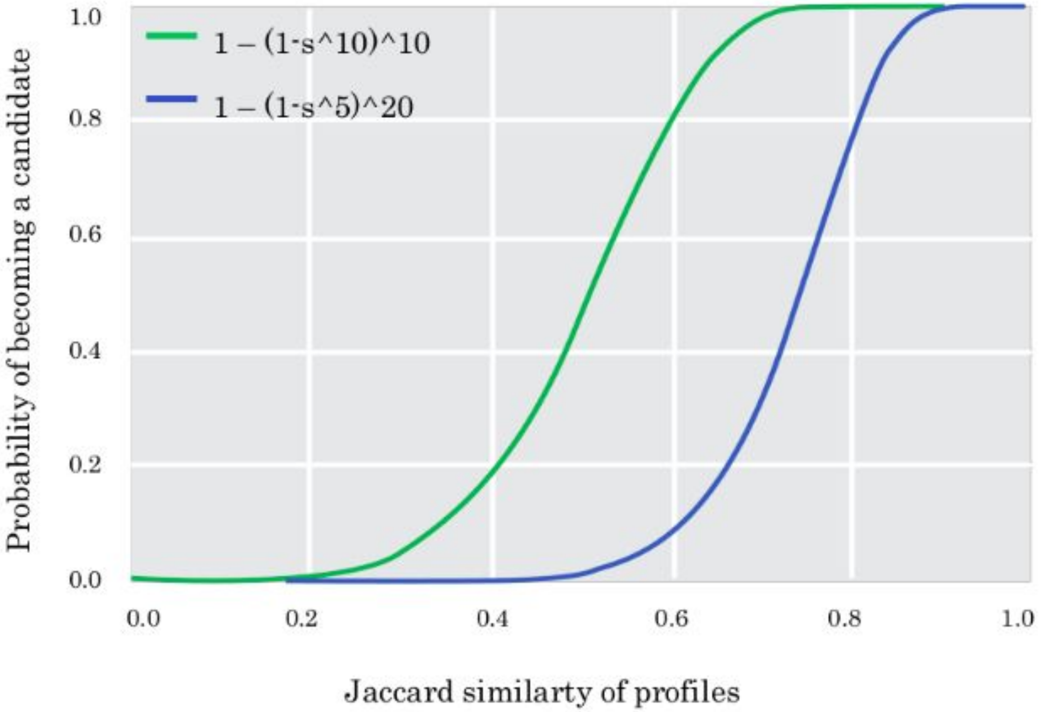
a, b - random integers < max(x)
c - common prime integer > max(x)

Locality sensitive hashing

Banding

band1	hash1	1	1
	hash2	3	3
band2	hash3	1	1
	hash4	2	4

How to choose quantity bands?



$$1-(1-x^r)^b$$

Thanks!