Advertisement targeting with machine learning

Story about keywords and Adtech

Agenda

1. Data analysis
2. Data Parsing
3. Sentence as vector
4. TFIDF
5. LSA
6. Clustering + k-means
7. Conclusion + future

Boring slide with me

1. I’m a Python Programmer
2. I’m a Hackerspace Silesia’s board member
3. I’m working in the adtech area
4. My job is counting site visits, purchases and reporting

Problem

- Client supports advertisement’s campaigns in many shops
- He dreams about machine learning
- What we only have is NGINX logs from trackers
- Data must be transformed to keywords per line

Analise

- What type is this data?
- How is look this data?
- Do we need all words?

Parsing

1. We have only logs from NGINX/Apache
2. We only interested with URL (without host)
3. We discard URLs like a admin / login / logout
4. We discard numbers (useless IDs)
5. We discard common and useless words and special characters
6. Keywords group by IP or cookieID
7. Document = keywords per cookieID

Parsing

```
YY.YY.YY.YY - - [DATE] "GET /xxx HTTP/1.1" 200 72 "https://www.url.pl/womens-road-shoes"
YY.YY.YY.YY - - [DATE] "GET /xxx HTTP/1.1" 200 72 "https://www.url.pl/mens-road-shoes"
ZZ.ZZ.ZZ.ZZ - - [DATE] "GET /xxx HTTP/1.1" 200 72 "http://www.url.pl/ShoeFinder"
```

```
www url pl womens road shoes www url pl mens road shoes
www url pl ShoeFinder
```

Machine learning

- We need transform data to “countable”
- If we make it to matrix of numbers, then we can make model to machine learning

Sentence as vector

- Document = 1 line of grouped keywords (per IP/cookieID)
- Document has “Terms”
- Term can be a char, word, pair words, source link etc.
- Document can be as a vector with frequency of terms
- Many document = many vectors = many columns = matrix ^_^

1. John has cats
2. Cat has claws
3. Mark has cats

Row	John	has	cats	cat	claws	Mark
1	**1**	**1**	**1**	0	0	0
2	0	**1**	0	**1**	**1**	0
3	0	**1**	**1**	0	0	**1**

1. John has cats
2. Mark has cats

Wiersz	John	John has	Has cats	cats	Mark	Mark has
1	**1**	**1**	**1**	1	0	0
2	0	0	**1**	**1**	**1**	**1**

TFIDF

- TF = **T**erm **F**requency (in document)
- IDF = **I**nverse **D**ocument **F**requency
  (rarer terms in document has a higher ranking)

$$\mathrm{TFIDF}(\mathrm{term}, \mathrm{doc}) = \mathrm{TF}(\mathrm{term}, \mathrm{doc}) \times \mathrm{IDF}(\mathrm{term})$$

$$\mathrm{TF} = \mathrm{count}(\mathrm{term}) \,/\, \mathrm{count}(\mathrm{terms\ in\ doc})$$

$$\mathrm{IDF} = \log_{10}\left(\mathrm{count}(\mathrm{docs}) \,/\, \mathrm{count}(\mathrm{docs\ with\ term}) \right)$$

term	TF	IFD	TFIFD
John	0.33	log(3/1) ≈ 1.1	0.363
has	0.33	log(3/3) = 0	0
cats	0.33	log(3/2) ≈ 0.4	0.132

1. John has cats
2. Cat has claws
3. Mark has cats

term	TF	IFD	TFIFD
Cat	0.33	log(3/1) ≈ 1.1	0.363
has	0.33	log(3/3) = 0	0
claws	0.33	log(3/1) ≈ 1.1	0.363

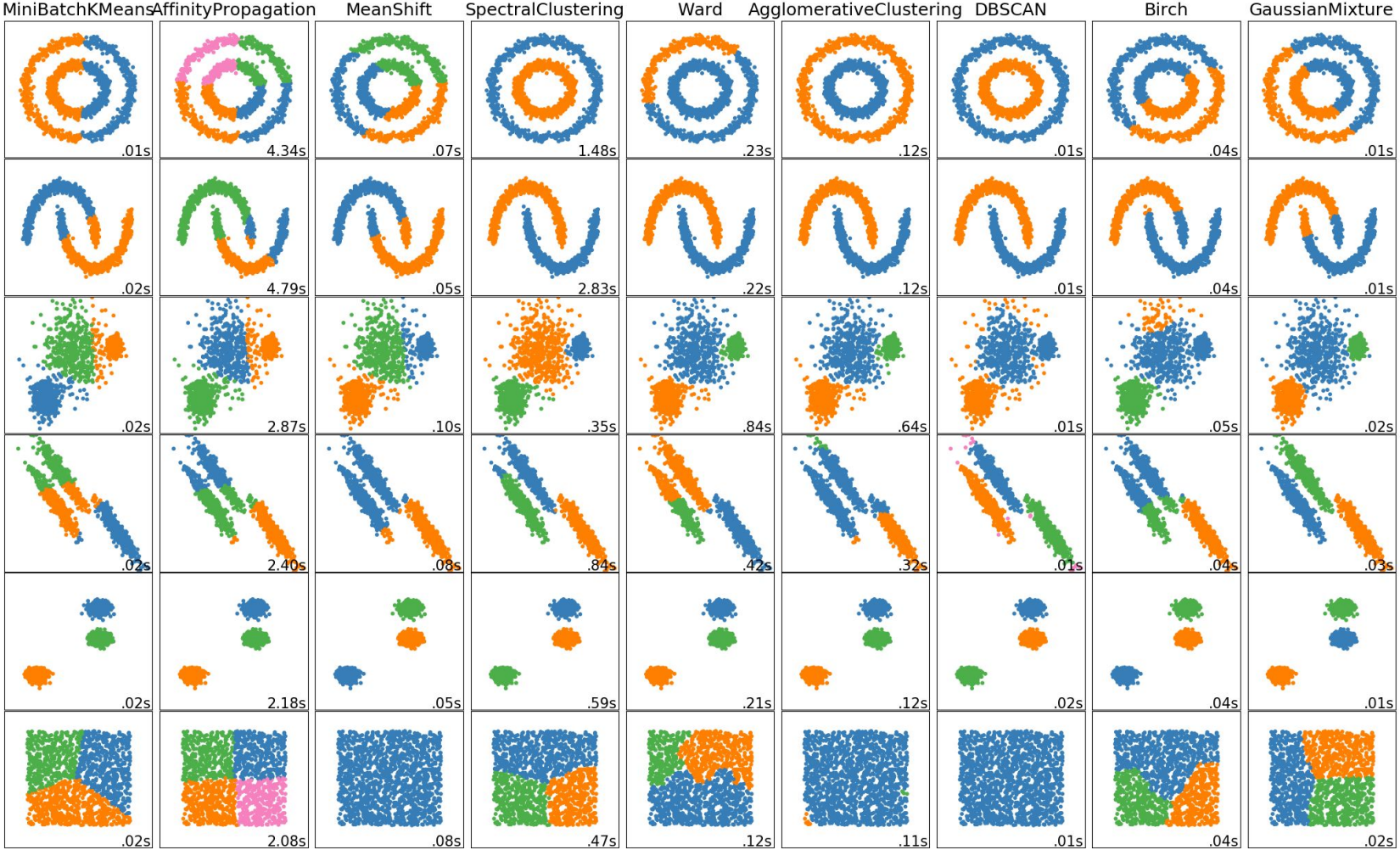
term	TF	IFD	TFIFD
Mark	0.33	log(3/1) ≈ 1.1	0.363
has	0.33	log(3/3) = 0	0
cats	0.33	log(3/2) ≈ 0.4	0.132

LSA

- LSA - Latent semantic analysis
- Groups vectors according to rank terms (ex. TFIDF)
- Groups terms according to similarities ranks
- I can’t explain without animation: https://commons.wikimedia.org/wiki/File:Topic_model_scheme.webm

Clustering

- We don’t know nothing about what we can get from data
- We need to use unsupervised learning
- Clustering - accordion to values of vectors we can classificate group of terms, see next slide


MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
Birch
GaussianMixture
.01s
4.34s
.07s
1.48s
.23s
.12s
.01s
.04s
.01s
.02s
4.79s
.05s
2.83s
.22s
.12s
.01s
.04s
.01s
.02s
2.87s
.10s
.35s
.84s
.64s
.01s
.05s
.02s
.02s
2.40s
.08s
.84s
.42s
.32s
.01s
.04s
.03s
.02s
2.18s
.05s
.59s
.21s
.12s
.02s
.04s
.01s
.02s
2.08s
.08s
.47s
.12s
.11s
.01s
.04s
.02s

Clustering

- Every one row from matrix we can treat as vector in N-dimension
- K-means algorithm - find the N nearest vectors to create group

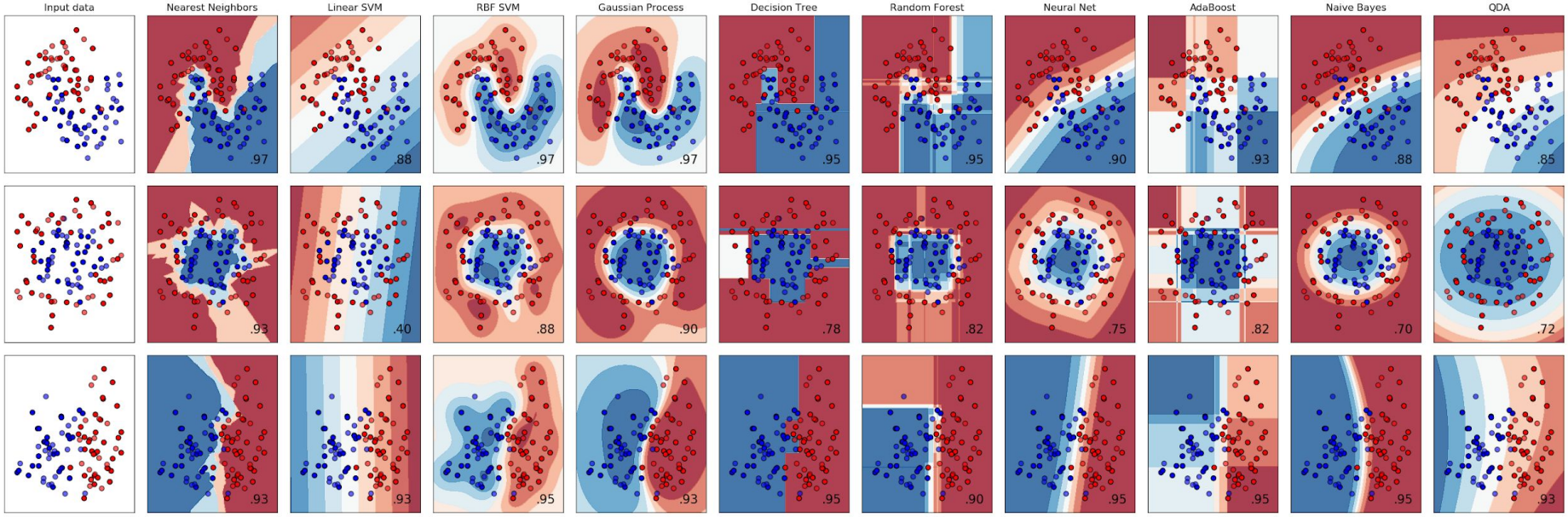
Clustering

```
www url pl womens road shoes www url pl mens road shoes
www url pl ShoeFinder
```

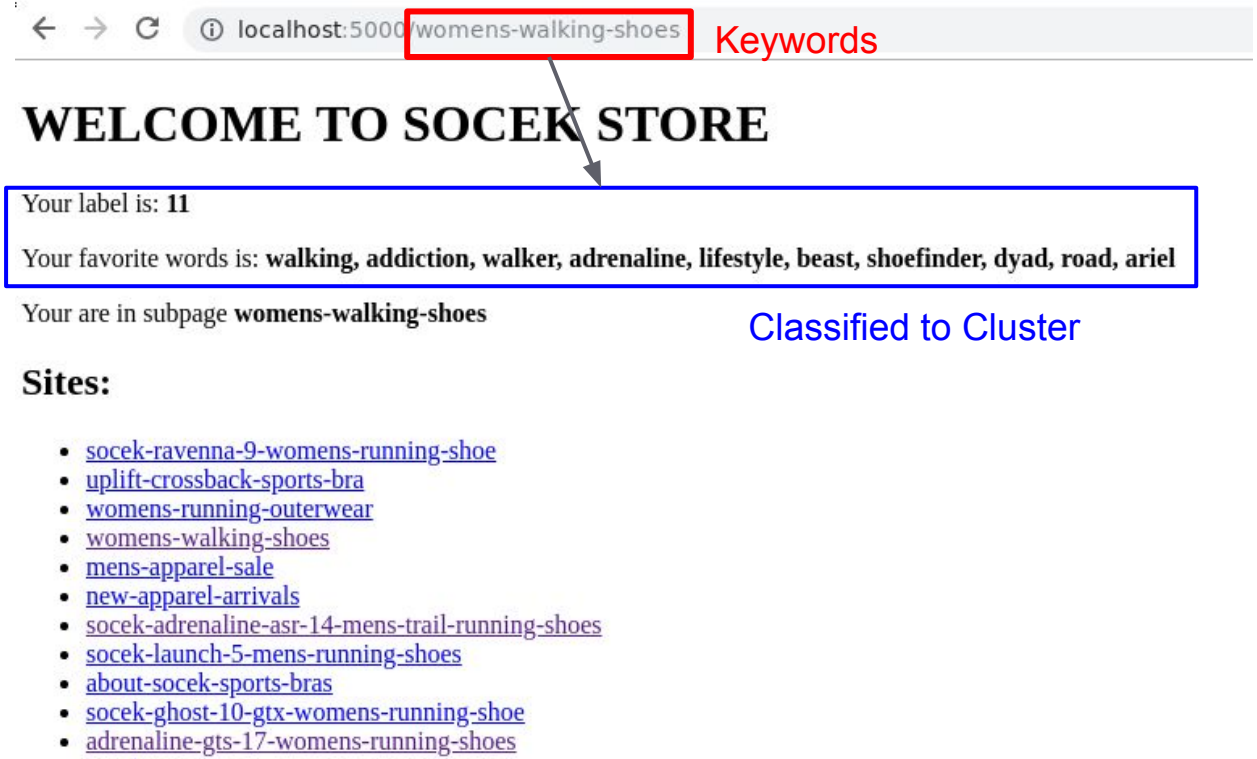

```
Cluster 01: sports, track, spikes, bras, bra, purecadence, impact, walking, juno, apparel
Cluster 02: show, coshipping, cosummary, submit, start, account, adrenaline, apparel, home, road
Cluster 03: start, coshipping, cosummary, submit, adrenaline, apparel, ghost, locator, returns, return
```

Classification

- According to created groups (cluster), we can make a classificator.
- Created classificator will be select ‘group’ based on input (ex. Keywords from visited sites)


Input data
Nearest Neighbors
Linear SVM
RBF SVM
Gaussian Process
Decision Tree
Random Forest
Neural Net
AdaBoost
Naive Bayes
QDA
.97
.88
.97
.97
.95
.95
.90
.93
.88
.85
.93
.40
.88
.90
.78
.82
.75
.82
.70
.72
.93
.93
.95
.93
.95
.90
.95
.95
.95
.93

Classification



Conclusion + future

- With clustering keywords, we can make special ad campaigns or change product categories on site.
- Based on cookieID and keywords used by visitor, we can show special targeted ads or propose another products
- **MORE DATA** - from many days + another metrics (like age or city)

Source

1. https://en.wikipedia.org/wiki/Latent_semantic_analysis
2. http://scikit-learn.org/stable/auto_examples/text/document_clustering.html
3. http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html
4. http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html
5. https://pandas.pydata.org/
6. https://github.com/firemark/word-analyser