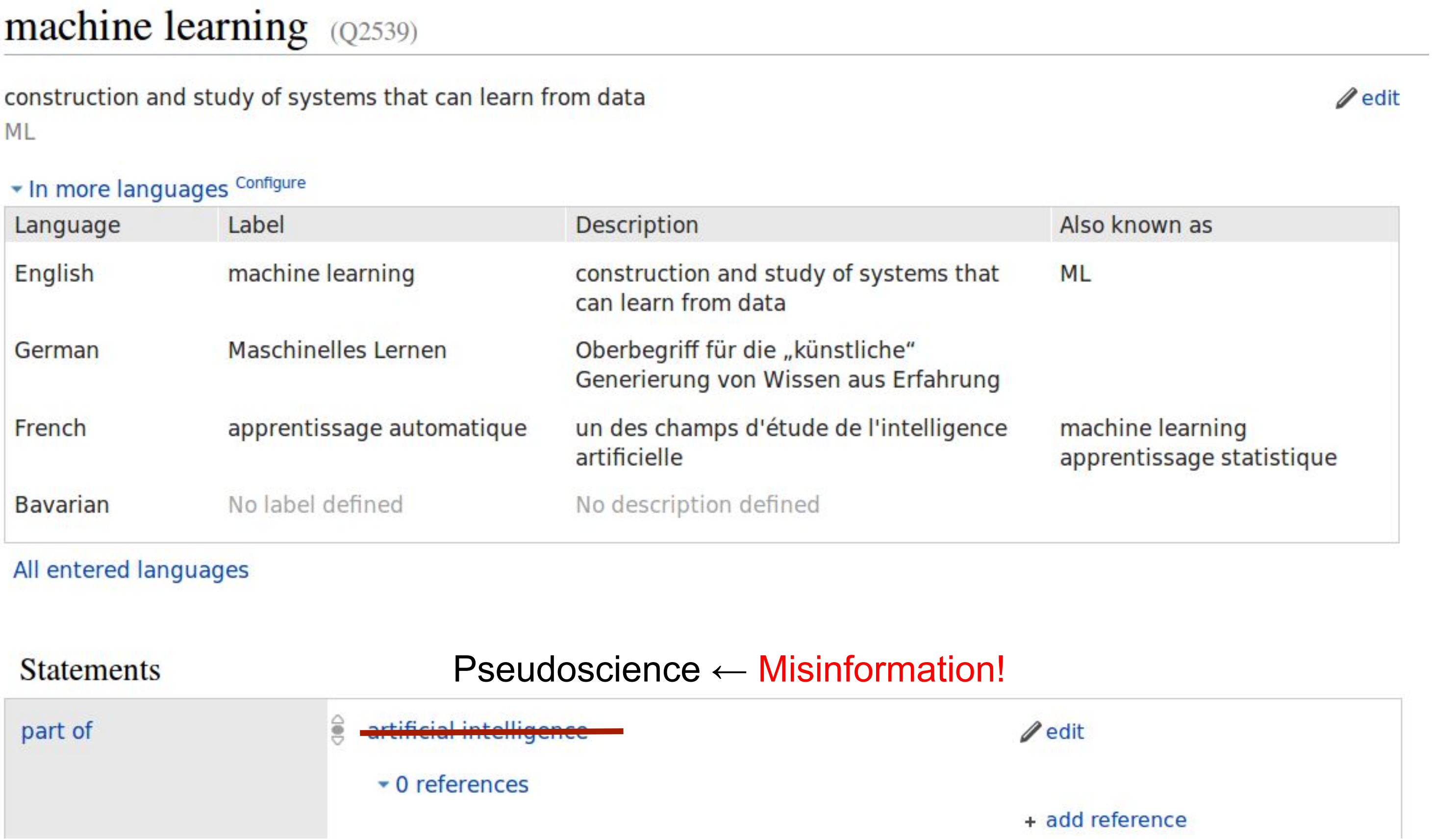
WSDM Cup 2017:
Vandalism Detection

Alexey Grigorev
21.01.2017



machine learning (Q2539)

construction and study of systems that can learn from data

ML

Language	Label	Description	Also known as
English	machine learning	construction and study of systems that can learn from data	ML
German	Maschinelles Lernen	Oberbegriff für die „künstliche“ Generierung von Wissen aus Erfahrung	
French	apprentissage automatique	un des champs d'étude de l'intelligence artificielle	machine learning apprentissage statistique
Bavarian	No label defined	No description defined	

Statements

Pseudoscience ← Misinformation!

part of ~~artificial intelligence~~

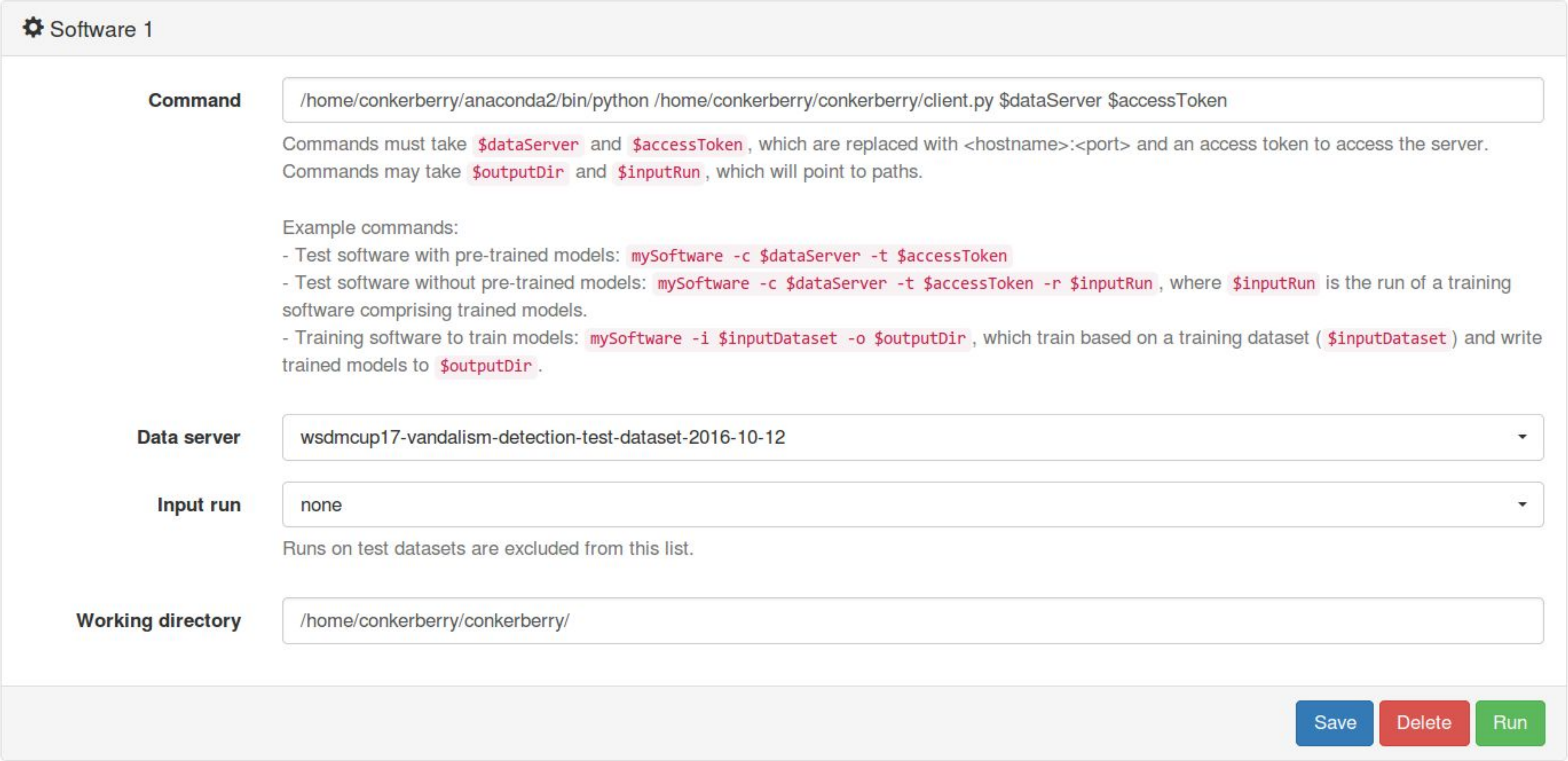
https://www.wikidata.org/wiki/Q2539

WSDM Cup 2017: Vandalism Detection

- http://www.wsdm-cup-2017.org/vandalism-detection.html
- Vandalism in KBs is very bad
- Goal of the competition:
- **Predict if a Wikidata revision should be rolled back or not**
- Solutions ranked by AUC

Not a usual competition

- No forum
- Code to put on VM
  - VM constraints: 1 core, 4 Gb RAM
- Evaluation
  - Read data from the socket
  - Transform the data, make a prediction
  - Write the prediction back to the socket
- Test data not available
- Results of test runs aren’t shown, only validation runs
- No leaderboard
- Only one attempt at the end - by default, last successful run on test



Software 1

Command /home/conkerberry/anaconda2/bin/python /home/conkerberry/conkerberry/client.py $dataServer $accessToken

Commands must take `$dataServer` and `$accessToken`, which are replaced with <hostname>:<port> and an access token to access the server. Commands may take `$outputDir` and `$inputRun`, which will point to paths.

Example commands:
- Test software with pre-trained models: `mySoftware -c $dataServer -t $accessToken`
- Test software without pre-trained models: `mySoftware -c $dataServer -t $accessToken -r $inputRun`, where `$inputRun` is the run of a training software comprising trained models.
- Training software to train models: `mySoftware -i $inputDataset -o $outputDir`, which train based on a training dataset (`$inputDataset`) and write trained models to `$outputDir` .

Data server wsdmcup17-vandalism-detection-test-dataset-2016-10-12

Input run none

Runs on test datasets are excluded from this list.

Working directory /home/conkerberry/conkerberry/

Data

- Wikidata dump with edits
  - From 2012 to 2016
  - 72 mln revisions
  - 24.8 Gb compressed
  - 400 Gb uncompressed
- Very skewed:
  - 127k rollbacks (out of 72m)
  - i.e. 0.0025 - fraction of positives
- For each revision know:
  - Title of page, timestamp
  - Comment and json data
  - Username or IP + geo information
  - Was reverted or not

```
<page>
  <title>Q5704066</title>
  <ns>0</ns>
  <id>5468191</id>
  <revision>
    <id>185142906</id>
    <parentid>185051367</parentid>
    <timestamp>2015-01-01</timestamp>
    <contributor>
      <username>username</username>
      <id>52267</id>
    </contributor>
    <comment>/* wbsetdescription-add:1|es */
futbolista irlandes</comment>
    <model>wikibase-item</model>
    <format>application/json</format>
    <text xml:space="preserve">{{PAGE_JSON}}</text>
  </revision>
</page>
```

IP if anonymous & geo info in meta file

Cross-Validation

Train	2012-10-29 to 2016-02-29	65 mln
Validation	2016-03-01 to 2016-04-30	7.2 mln
Test	from 2016-05-01	10.4 mln

← Available from start

← Available towards end

← Not available

Train	2015-01-01 to 2015-31-31	27.9 mln
Validation	2016-01-01 to 2016-02-29	9.3 mln
Test	2016-03-01 to 2016-04-30	7.2 mln

Features

3 groups:

- Page feature: page title
- User features
- Comment features

Not considered:

- Timestamps
- Counters
- JSON Content

Problematic due to 4Gb constraint

User Features

- Username, if logged in, if not “anonymous=True”
- IP: 90.219.230.105 → 90, 90_219, 90_219_230, 90_219_230_105
- Geo information from the meta file:
    - country_code=GB
    - continent_code=EU
    - time_zone=GMT
    - regio_code=EN
    - city_name=LEEDS
    - county_name=WEST_YORKSHIRE
- OHE: put this together as one string
    - `anonymous 90 90_219 90_219_230 90_219_230_105 country_code=GB ...`
    - Use CountVectorizer

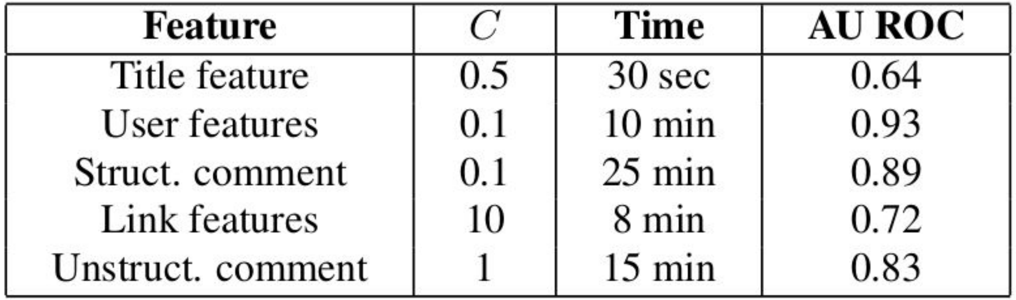
Comment Features

```
/* wbsetdescription-add:1|es */ futbolista irlandes
/* wbcreateclaim-create:1| */ [[Property:P31]]: [[Q5]], #autolist2
/* wbsetsitelink-add:1|idwiki */ Megaloharpya
/* clientsitelink-remove:1||frwiki */ Origyn Web Browser
```

- “Structured” part: inside /* */ - split on “:” and “|”:
  - wbsetdescription-add 1 es
  - wbsetsitelink-add 1 idwiki
  - clientsitelink-remove 1 frwiki
- Wiki-Links: [[Property:P31]], [[Q5]]
- Free text: outside of /* */:
  - futbolista irlandes
  - autolist2
  - origyn web browser

Models

- LinearSVM in primal with L1 regularization



Feature	C	Time	AU ROC
Title feature	0.5	30 sec	0.64
User features	0.1	10 min	0.93
Struct. comment	0.1	25 min	0.89
Link features	10	8 min	0.72
Unstruct. comment	1	15 min	0.83

- Stacking: XGBoost and L2-LogReg
  - Both good on validation, but overfit testing
- Online Learning: worse than SVM on one year
- Upsampling and Undersampling: always worse than as-is

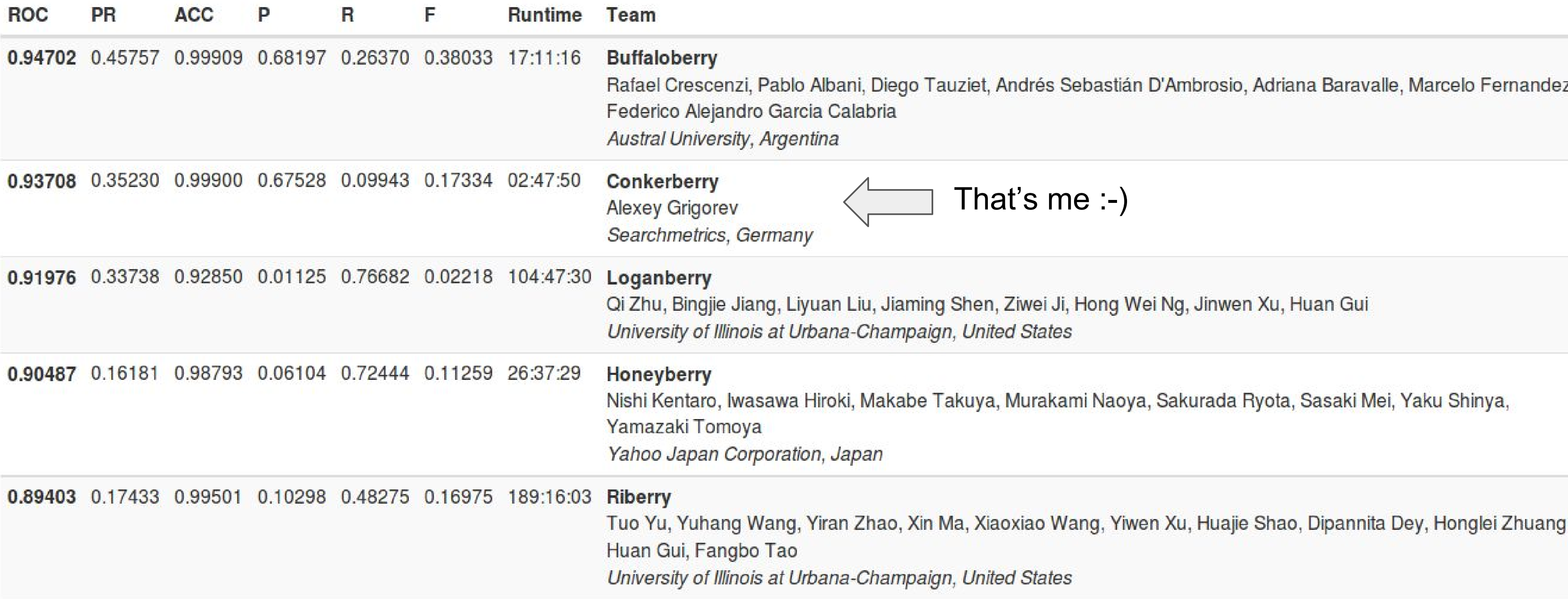
Final Model

- Put all features together as one string
- `title=Q123 username=user wbsetdescription-add 1 es P31 Q5 …`
- OHE?
  - CountVectorizer - dictionary too large
  - HashingVectorizer with 10m columns

Final model:

- OHE with HashingVectorizer (no memory)
- SVM on OHE matrix (300 mb)
- ~ 0.96 AUC on my test

Final Results



ROC	PR	ACC	P	R	F	Runtime	Team
0.94702	0.45757	0.99909	0.68197	0.26370	0.38033	17:11:16	**Buffaloberry** Rafael Crescenzi, Pablo Albani, Diego Tauziet, Andrés Sebastián D'Ambrosio, Adriana Baravalle, Marcelo Fernandez Federico Alejandro Garcia Calabria *Austral University, Argentina*
0.93708	0.35230	0.99900	0.67528	0.09943	0.17334	02:47:50	**Conkerberry** Alexey Grigorev *Searchmetrics, Germany*
0.91976	0.33738	0.92850	0.01125	0.76682	0.02218	104:47:30	**Loganberry** Qi Zhu, Bingjie Jiang, Liyuan Liu, Jiaming Shen, Ziwei Ji, Hong Wei Ng, Jinwen Xu, Huan Gui *University of Illinois at Urbana-Champaign, United States*
0.90487	0.16181	0.98793	0.06104	0.72444	0.11259	26:37:29	**Honeyberry** Nishi Kentaro, Iwasawa Hiroki, Makabe Takuya, Murakami Naoya, Sakurada Ryota, Sasaki Mei, Yaku Shinya, Yamazaki Tomoya *Yahoo Japan Corporation, Japan*
0.89403	0.17433	0.99501	0.10298	0.48275	0.16975	189:16:03	**Riberry** Tuo Yu, Yuhang Wang, Yiran Zhao, Xin Ma, Xiaoxiao Wang, Yiwen Xu, Huajie Shao, Dipannita Dey, Honglei Zhuang Huan Gui, Fangbo Tao *University of Illinois at Urbana-Champaign, United States*

That's me :-)

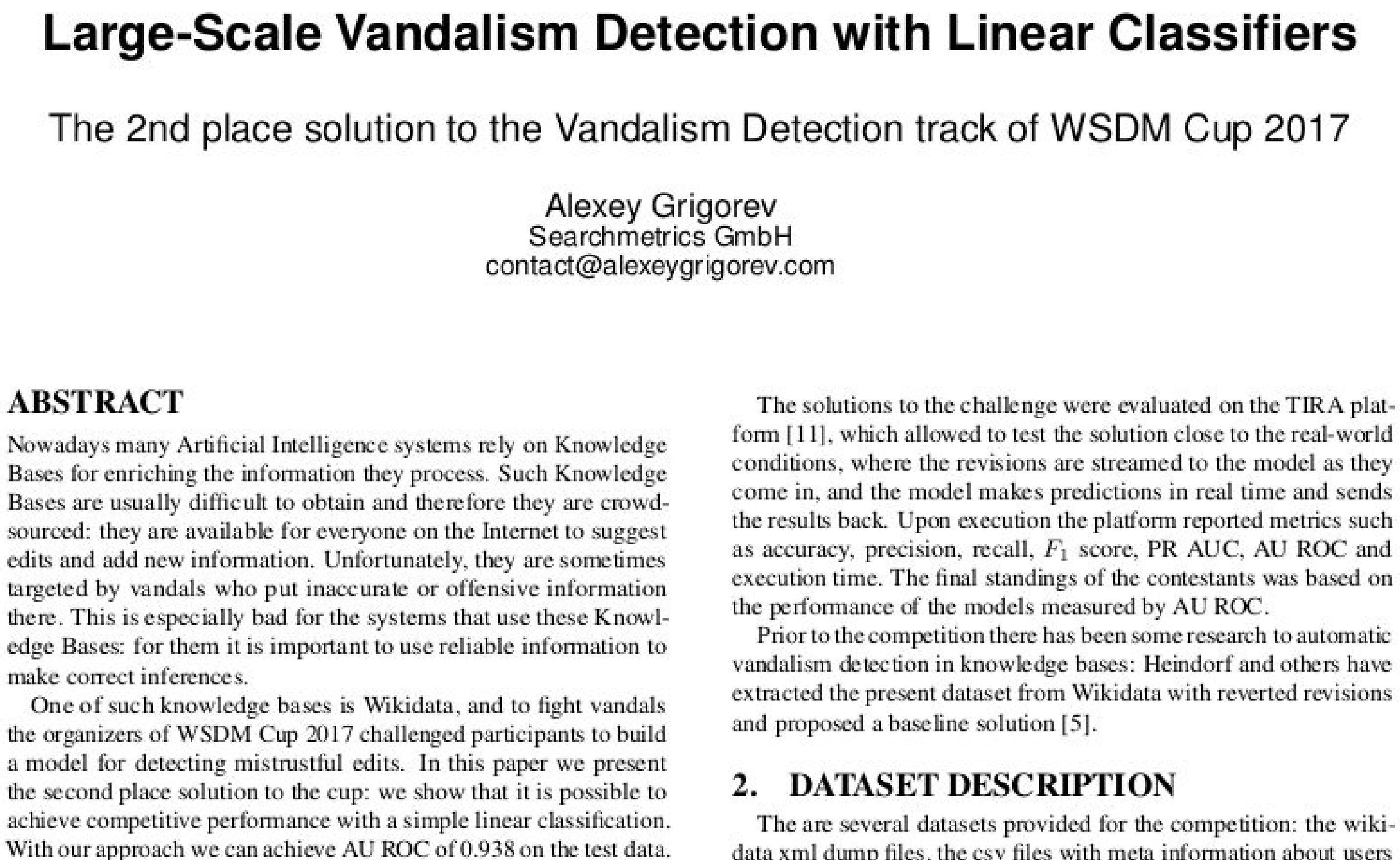
Conclusions

New things I learned/tried:

- Feather - very fast!
- Twisted

Lessons Learned:

- Trust your CV
- Sometimes ensembling does not work
- Prefer simpler models to avoid overfitting



Large-Scale Vandalism Detection with Linear Classifiers

The 2nd place solution to the Vandalism Detection track of WSDM Cup 2017

Alexey Grigorev
Searchmetrics GmbH
contact@alexeygrigorev.com

ABSTRACT

Nowadays many Artificial Intelligence systems rely on Knowledge Bases for enriching the information they process. Such Knowledge Bases are usually difficult to obtain and therefore they are crowdsourced: they are available for everyone on the Internet to suggest edits and add new information. Unfortunately, they are sometimes targeted by vandals who put inaccurate or offensive information there. This is especially bad for the systems that use these Knowledge Bases: for them it is important to use reliable information to make correct inferences.

One of such knowledge bases is Wikidata, and to fight vandals the organizers of WSDM Cup 2017 challenged participants to build a model for detecting mistrustful edits. In this paper we present the second place solution to the cup: we show that it is possible to achieve competitive performance with a simple linear classification. With our approach we can achieve AU ROC of 0.938 on the test data.

The solutions to the challenge were evaluated on the TIRA platform [11], which allowed to test the solution close to the real-world conditions, where the revisions are streamed to the model as they come in, and the model makes predictions in real time and sends the results back. Upon execution the platform reported metrics such as accuracy, precision, recall, F_1 score, PR AUC, AU ROC and execution time. The final standings of the contestants was based on the performance of the models measured by AU ROC.

Prior to the competition there has been some research to automatic vandalism detection in knowledge bases: Heindorf and others have extracted the present dataset from Wikidata with reverted revisions and proposed a baseline solution [5].

2. DATASET DESCRIPTION

The are several datasets provided for the competition: the wikidata xml dump files, the csv files with meta information about users

Links and Further Info

- Competition website: http://www.wsdm-cup-2017.org/
- Competition platform: http://tira.io/
- My solution:
    - https://github.com/alexeygrigorev/wsdmcup17-vandalism-detection

Thank you
Questions?