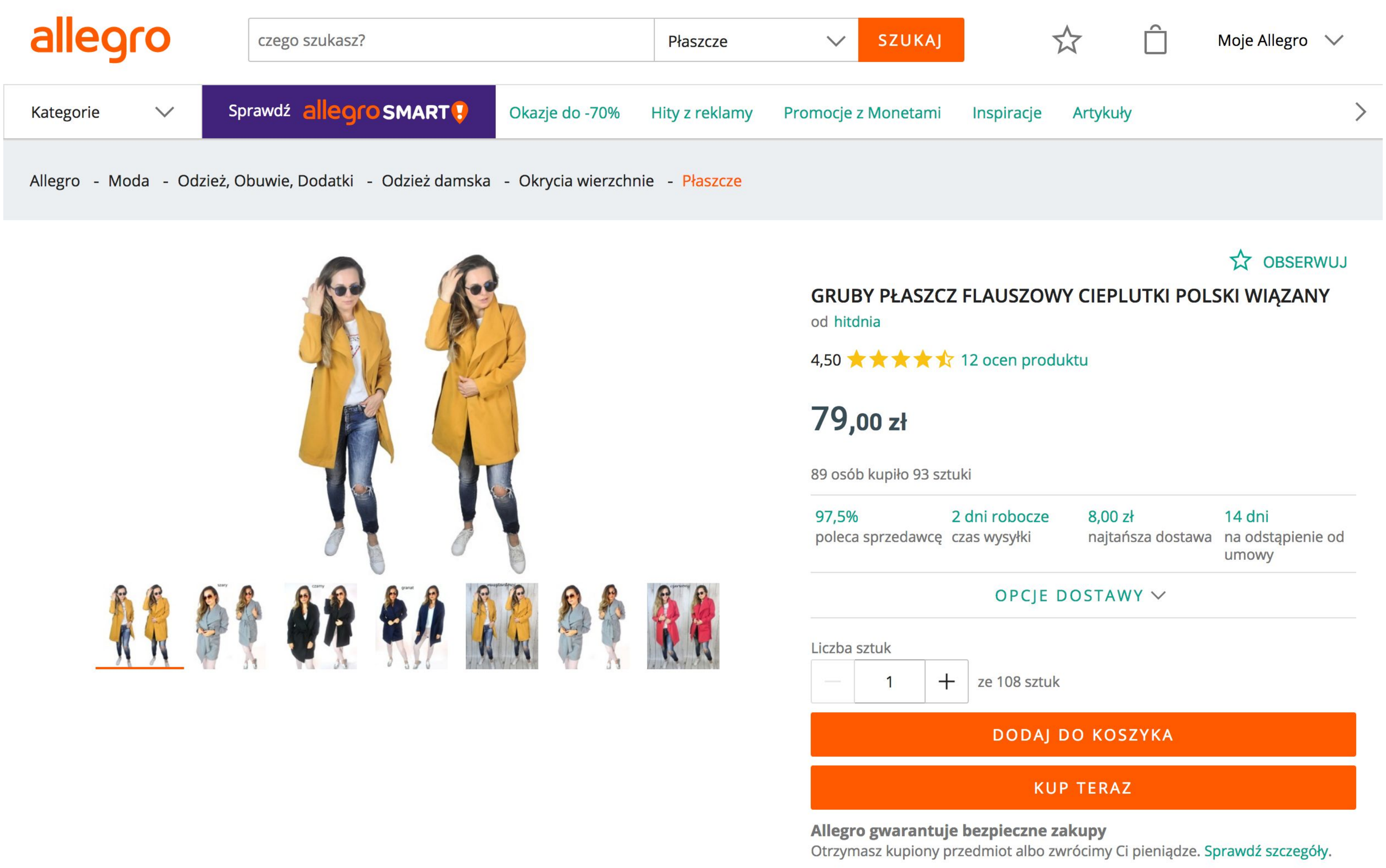
# Recognizing products from raw text descriptions using “shallow” and “deep” machine learning

Tomasz Płomiński
Allegro.pl

Tymoteusz Wołodźko
Allegro.pl
tymwol



allegro

czego szukasz?

Płaszcze

SZUKAJ

Moje Allegro

Kategorie

Sprawdź allegro SMART!

Okazje do -70% | Hity z reklamy | Promocje z Monetami | Inspiracje | Artykuły

Allegro - Moda - Odzież, Obuwie, Dodatki - Odzież damska - Okrycia wierzchnie - Płaszcze

☆ OBSERWUJ

GRUBY PŁASZCZ FLAUSZOWY CIEPLUTKI POLSKI WIĄZANY

od hitdnia

4,50 ★★★★☆ 12 ocen produktu

79,00 zł

89 osób kupiło 93 sztuki

| | | | |
|---|---|---|---|
| 97,5% | 2 dni robocze | 8,00 zł | 14 dni |
| poleca sprzedawcę | czas wysyłki | najtańsza dostawa | na odstąpienie od umowy |

OPCJE DOSTAWY

Liczba sztuk

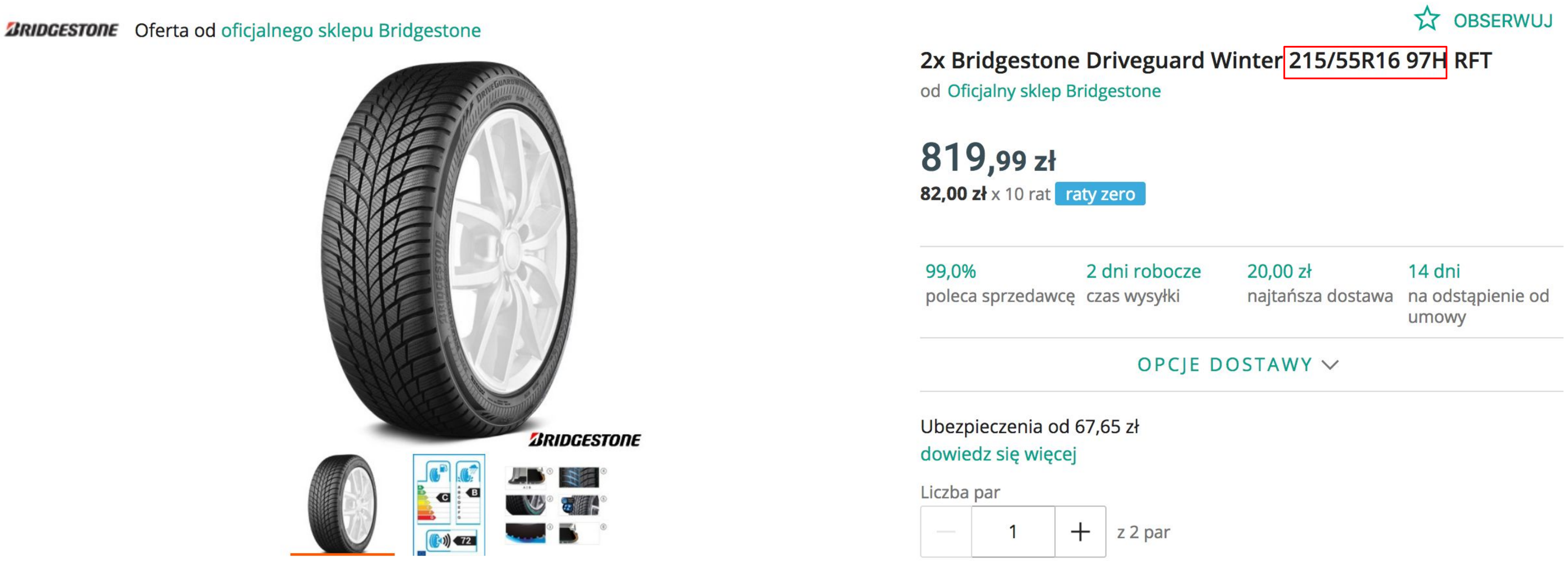
— 1 + ze 108 sztuk

DODAJ DO KOSZYKA

KUP TERAZ

**Allegro gwarantuje bezpieczne zakupy**

Otrzymasz kupiony przedmiot albo zwrócimy Ci pieniądze. Sprawdź szczegóły.



BRIDGESTONE Oferta od oficjalnego sklepu Bridgestone

☆ OBSERWUJ

2x Bridgestone Driveguard Winter 215/55R16 97H RFT

od Oficjalny sklep Bridgestone

819,99 zł

82,00 zł x 10 rat raty zero

| 99,0% | 2 dni robocze | 20,00 zł | 14 dni |
|---|---|---|---|
| poleca sprzedawcę | czas wysyłki | najtańsza dostawa | na odstąpienie od umowy |

OPCJE DOSTAWY ⌄

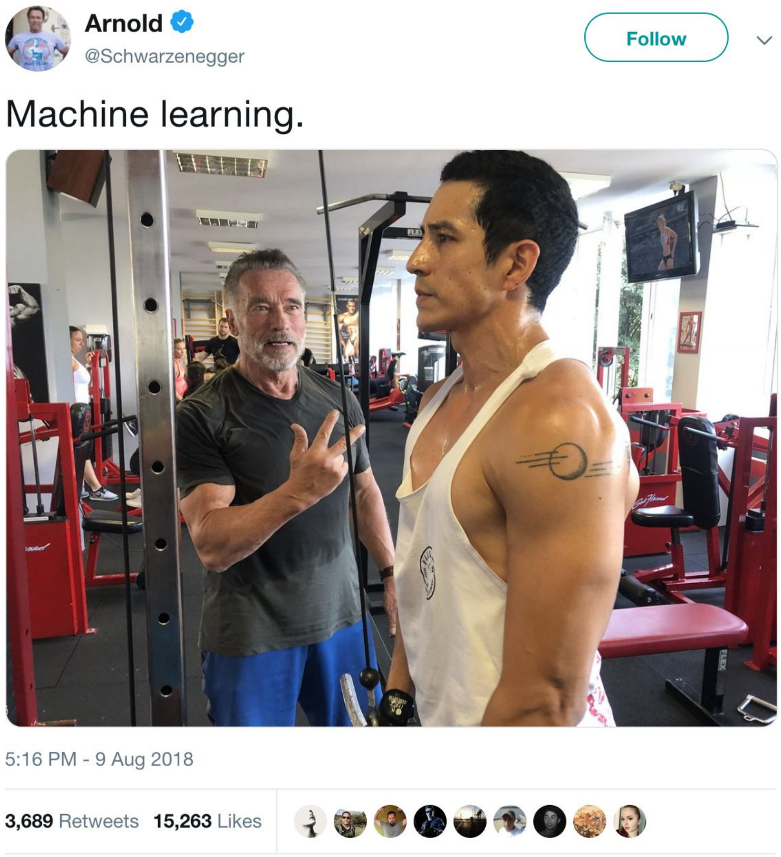
Ubezpieczenia od 67,65 zł

dowiedz się więcej

Liczba par

– 1 + z 2 par

```
^.*?\b([0-9]{2,3})\s?[\/]\s?([0-9]{2,3})\s?([zv])?t?[r\/]\s?([0-9]{2,3}\"?)(?:\s?([0-9]{1,3})\s?([a-z]{1,2}[1-9]?))?.*$
```

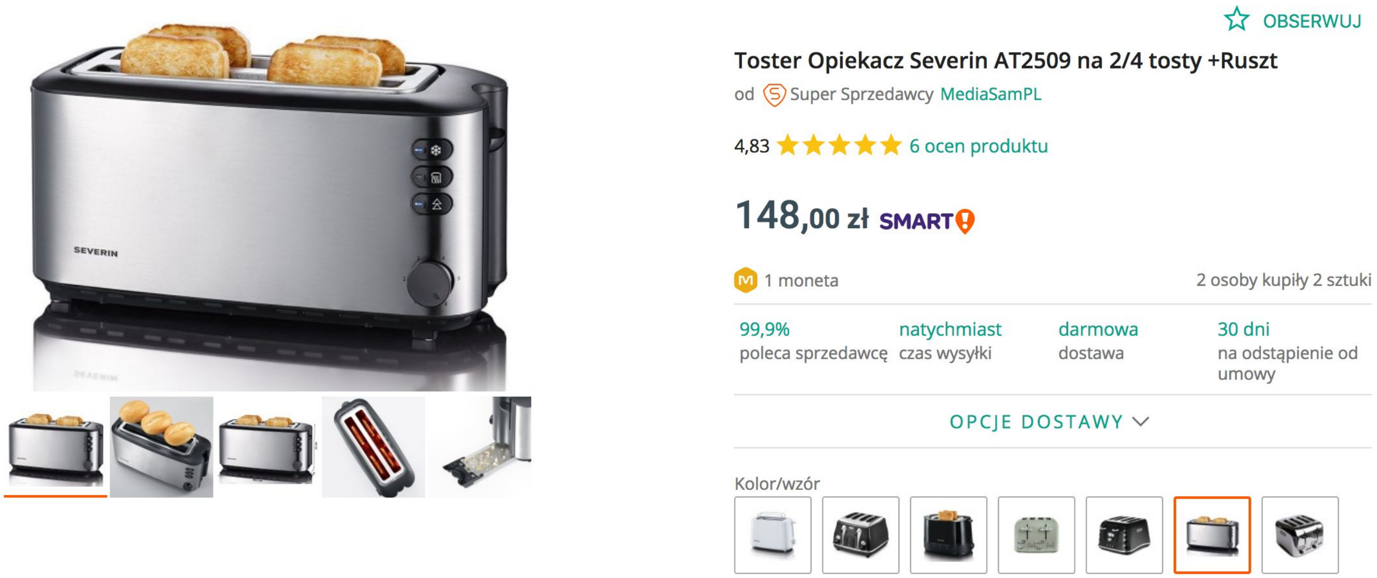


Arnold
@Schwarzenegger

Machine learning.

5:16 PM - 9 Aug 2018

3,689 Retweets 15,263 Likes



☆ OBSERWUJ

Toster Opiekacz Severin AT2509 na 2/4 tosty +Ruszt

od Super Sprzedawcy MediaSamPL

4,83 ★★★★★ 6 ocen produktu

148,00 zł SMART

1 moneta

2 osoby kupiły 2 sztuki

99,9% poleca sprzedawcę

natychmiast czas wysyłki

darmowa dostawa

30 dni na odstąpienie od umowy

OPCJE DOSTAWY

Kolor/wzór

| Toster | Opiekacz | **Severin** | **AT** | **2509** | na | 2 | 4 | tosty | Ruszt |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

# Features (at word level):

- Offer category

- Word length
- Is number
- Word position in name
- Is brand

... and same information about surrounding words:

- Length of word on position -2, -1, +1, +2
- Is number on position -2, -1, +1, +2
- Is brand ……

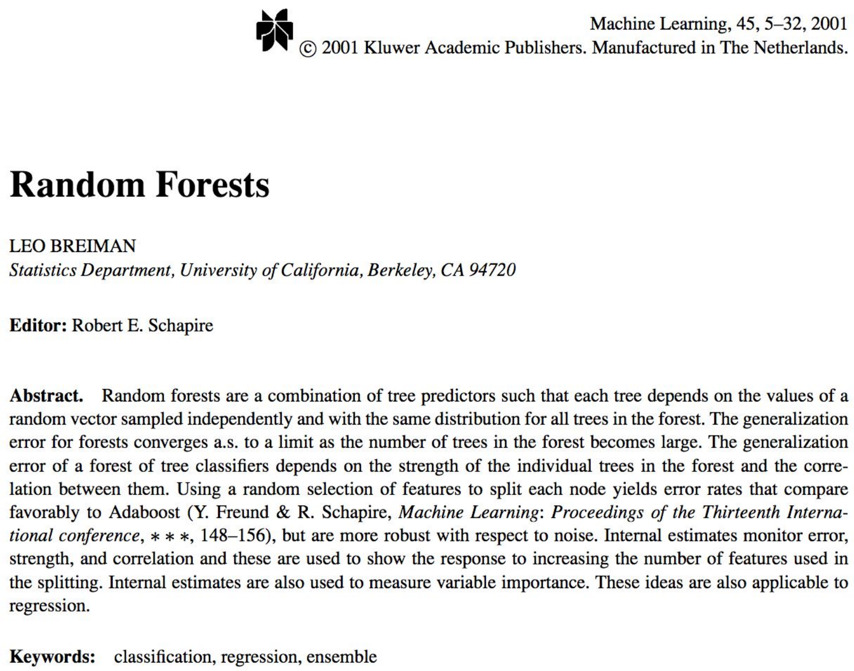
etc.

| | Toster | Opiekacz | Severin | AT | 2509 | na | 2 | 4 | tosty | Ruszt |
|---|---|---|---|---|---|---|---|---|---|---|
| **wordLength** | 6 | 8 | 7 | 2 | 4 | 2 | 1 | 1 | 5 | 5 |
| **isBrand** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **isBrand1p** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **isBrand1n** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **isNumber** | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |

# First attempt - cleaning product catalogue

- Well structured descriptions
- ≈ 50% words marked as positives
- ≈ 37K observations in training set

- Very short fitting time
- Similar performance of XGBoost and Random Forest models
- Accuracy: ≈ 99% in train, >97% in test
- Precision, recall: ≈ 97%
- Position of word in name most important for prediction



Machine Learning, 45, 5–32, 2001
© 2001 Kluwer Academic Publishers. Manufactured in The Netherlands.

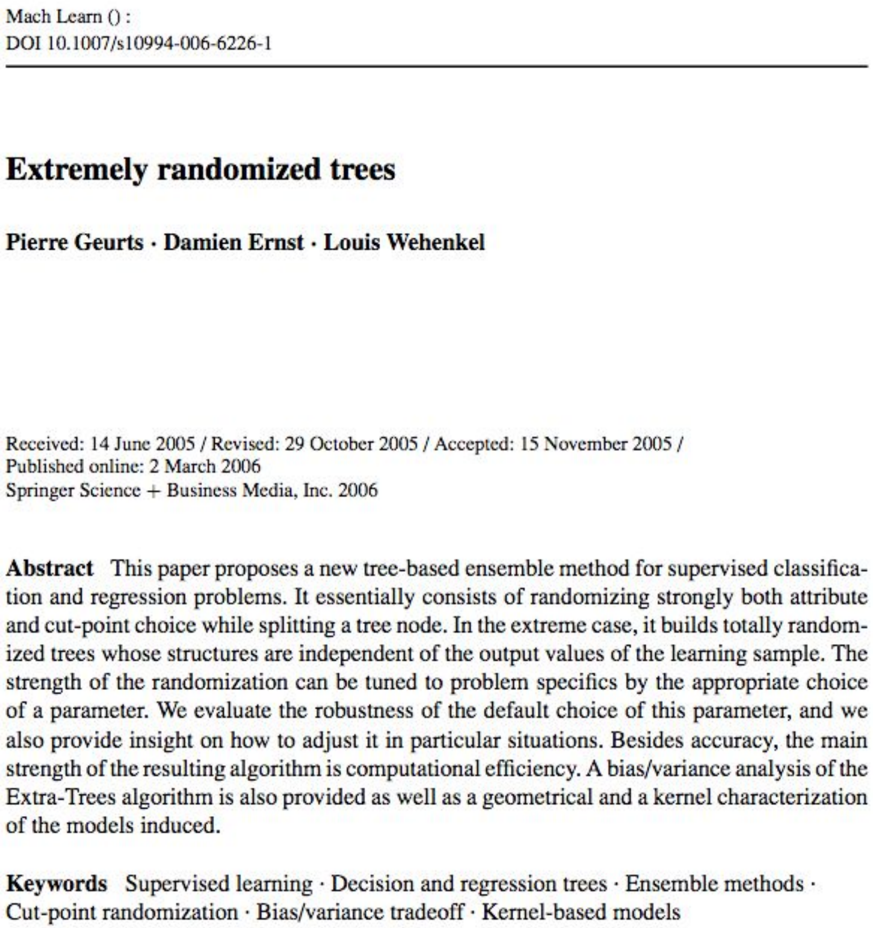
# Random Forests

LEO BREIMAN
*Statistics Department, University of California, Berkeley, CA 94720*

**Editor:** Robert E. Schapire

**Abstract.** Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges a.s. to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. Using a random selection of features to split each node yields error rates that compare favorably to Adaboost (Y. Freund & R. Schapire, *Machine Learning*: *Proceedings of the Thirteenth International conference*, ∗ ∗ ∗, 148–156), but are more robust with respect to noise. Internal estimates monitor error, strength, and correlation and these are used to show the response to increasing the number of features used in the splitting. Internal estimates are also used to measure variable importance. These ideas are also applicable to regression.

**Keywords:** classification, regression, ensemble



Mach Learn () :
DOI 10.1007/s10994-006-6226-1

# Extremely randomized trees

**Pierre Geurts · Damien Ernst · Louis Wehenkel**

Received: 14 June 2005 / Revised: 29 October 2005 / Accepted: 15 November 2005 /
Published online: 2 March 2006
Springer Science + Business Media, Inc. 2006

**Abstract** This paper proposes a new tree-based ensemble method for supervised classification and regression problems. It essentially consists of randomizing strongly both attribute and cut-point choice while splitting a tree node. In the extreme case, it builds totally randomized trees whose structures are independent of the output values of the learning sample. The strength of the randomization can be tuned to problem specifics by the appropriate choice of a parameter. We evaluate the robustness of the default choice of this parameter, and we also provide insight on how to adjust it in particular situations. Besides accuracy, the main strength of the resulting algorithm is computational efficiency. A bias/variance analysis of the Extra-Trees algorithm is also provided as well as a geometrical and a kernel characterization of the models induced.

**Keywords** Supervised learning · Decision and regression trees · Ensemble methods · Cut-point randomization · Bias/variance tradeoff · Kernel-based models

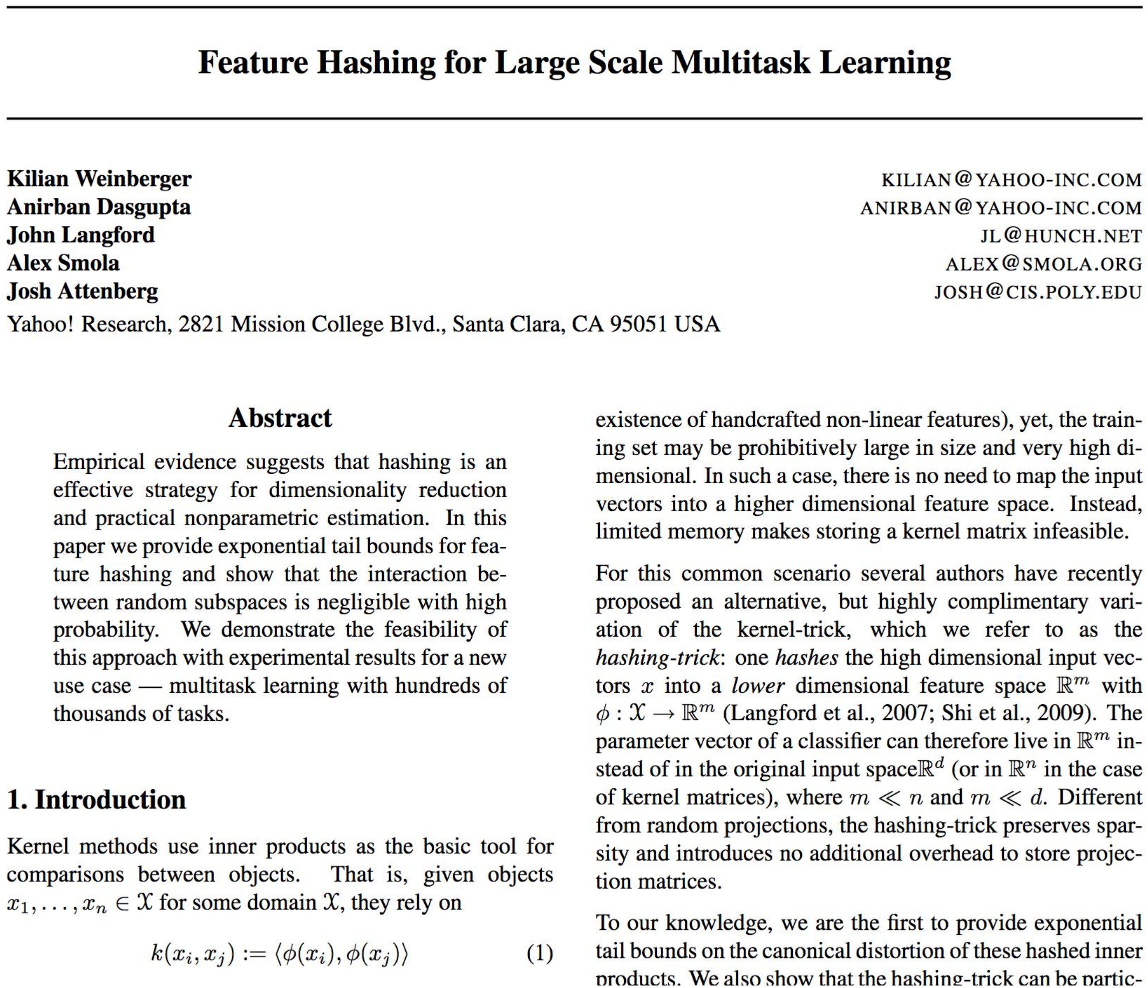
# Test set predictions

- 76% accuracy
- 82% precision, 79% recall
- 24% perfect match

***But...***

- 73% had ≤ 2 words misclassified (out of 7 words on average)
- 80% had >2 words in product name (vs 80% in train data)

Image source: Fredrik Sørlie



# Feature Hashing for Large Scale Multitask Learning

**Kilian Weinberger** KILIAN@YAHOO-INC.COM
**Anirban Dasgupta** ANIRBAN@YAHOO-INC.COM
**John Langford** JL@HUNCH.NET
**Alex Smola** ALEX@SMOLA.ORG
**Josh Attenberg** JOSH@CIS.POLY.EDU
Yahoo! Research, 2821 Mission College Blvd., Santa Clara, CA 95051 USA

## Abstract

Empirical evidence suggests that hashing is an effective strategy for dimensionality reduction and practical nonparametric estimation. In this paper we provide exponential tail bounds for feature hashing and show that the interaction between random subspaces is negligible with high probability. We demonstrate the feasibility of this approach with experimental results for a new use case — multitask learning with hundreds of thousands of tasks.

## 1. Introduction

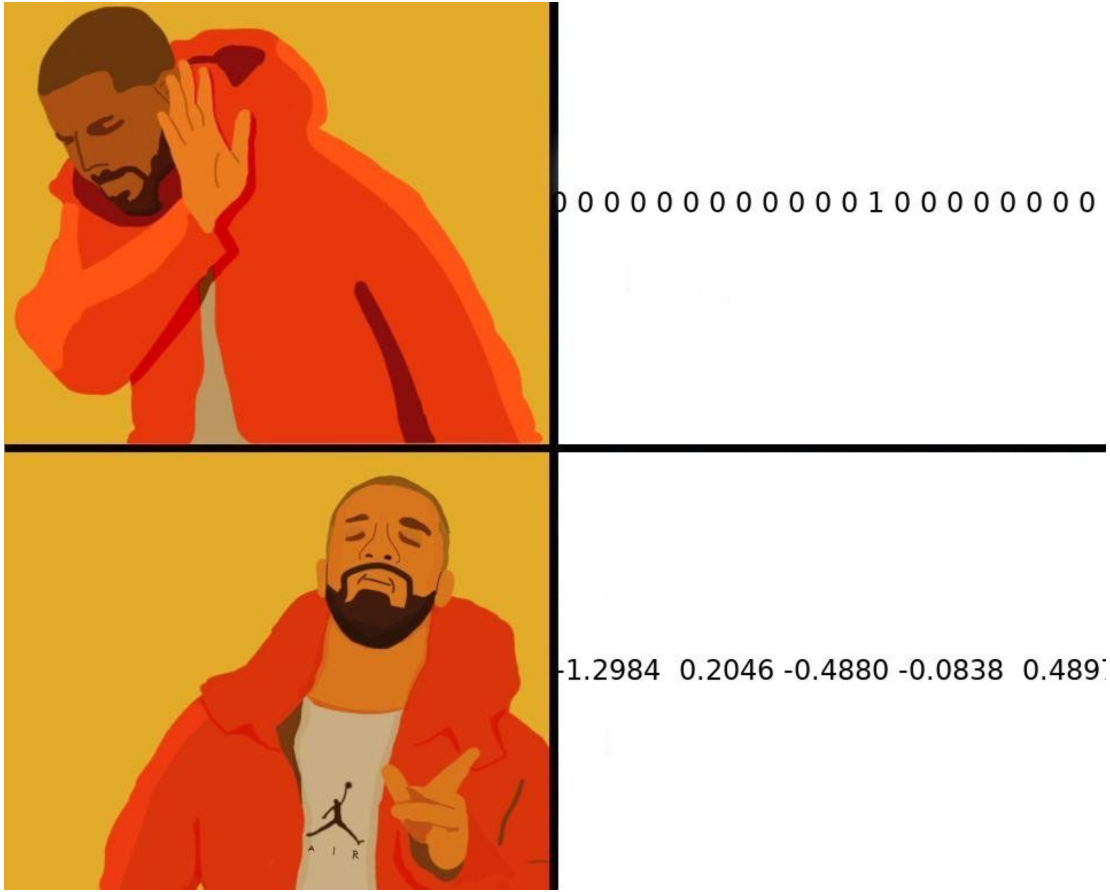
Kernel methods use inner products as the basic tool for comparisons between objects. That is, given objects $x_1, \ldots, x_n \in \mathcal{X}$ for some domain $\mathcal{X}$, they rely on

$$k(x_i, x_j) := \langle \phi(x_i), \phi(x_j) \rangle \qquad (1)$$

existence of handcrafted non-linear features), yet, the training set may be prohibitively large in size and very high dimensional. In such a case, there is no need to map the input vectors into a higher dimensional feature space. Instead, limited memory makes storing a kernel matrix infeasible.

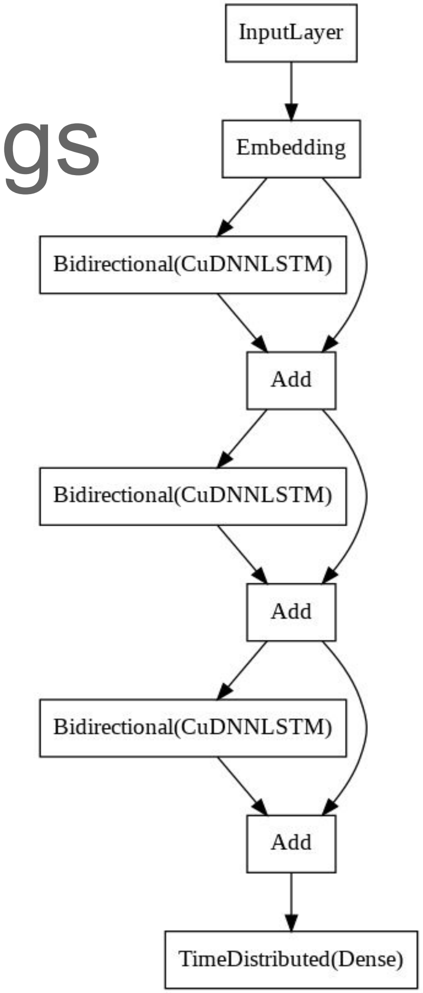
For this common scenario several authors have recently proposed an alternative, but highly complimentary variation of the kernel-trick, which we refer to as the *hashing-trick*: one *hashes* the high dimensional input vectors $x$ into a *lower* dimensional feature space $\mathbb{R}^m$ with $\phi : \mathcal{X} \to \mathbb{R}^m$ (Langford et al., 2007; Shi et al., 2009). The parameter vector of a classifier can therefore live in $\mathbb{R}^m$ instead of in the original input space$\mathbb{R}^d$ (or in $\mathbb{R}^n$ in the case of kernel matrices), where $m \ll n$ and $m \ll d$. Different from random projections, the hashing-trick preserves sparsity and introduces no additional overhead to store projection matrices.

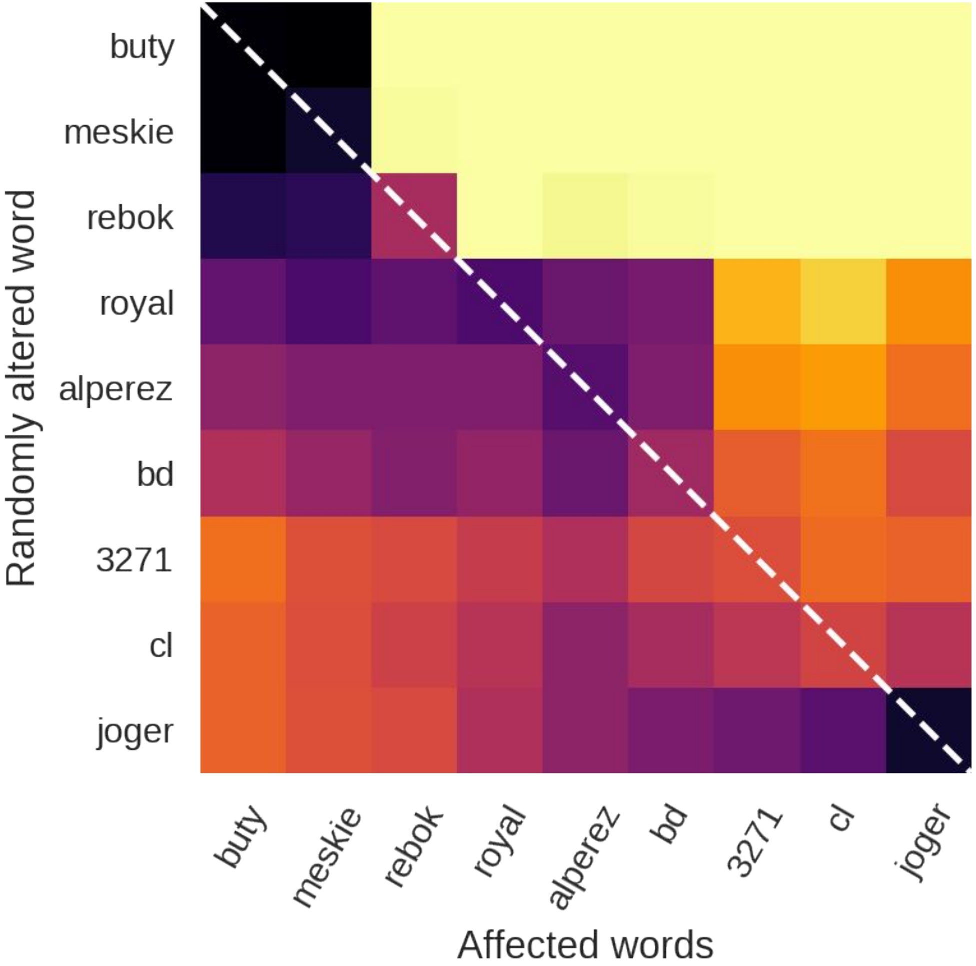
To our knowledge, we are the first to provide exponential tail bounds on the canonical distortion of these hashed inner products. We also show that the hashing-trick can be partic-



Using image from: https://www.reddit.com/r/dankmemes/comments/8qvfd4/think_i_found_a_loophole_here/

- Embeddings
- Bi-LSTM
- Skip
- Bi-LSTM
- Skip
- Bi-LSTM
- Skip
- Dense




Randomly altered word
buty
meskie
rebok
royal
alperez
bd
3271
cl
joger
Affected words
buty
meskie
rebok
royal
alperez
bd
3271
cl
joger

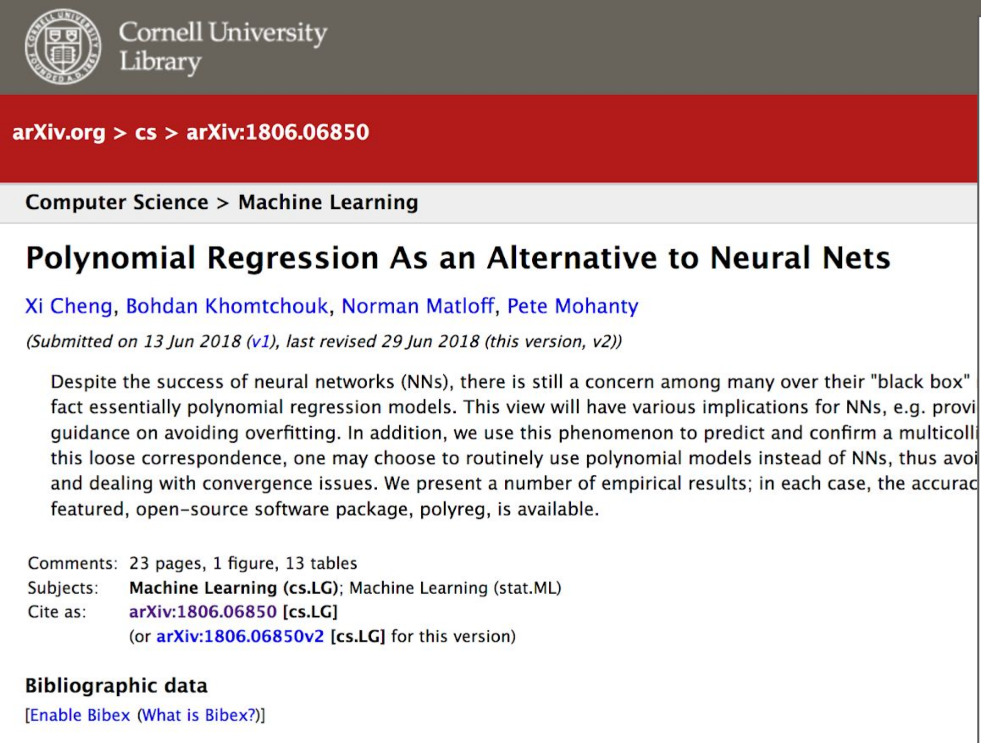
# Test set predictions

- 88% accuracy (vs 91% in train set)
- 80% precision & recall (vs 82% in train set)
- 28% perfect match

***But...***

- 73% had ≤ 2 words misclassified (out of 7 words on average)
- 4 words per product name (vs 4 words in train data)
- 80% had >2 words in product name (vs 80% in train data)
- 2% did not predict any product name (vs 1.8% blanks in train data)
- 3 distinct names per product (per 50 offers/product)
- for each product 64% (60-84%) of offers were assigned same name

# “Shallow” vs “deep” learning



Cornell University Library

We gratefully ack

arXiv.org > cs > arXiv:1806.06850

Search or Article ID

Help | Advanced search)

Computer Science > Machine Learning

## Polynomial Regression As an Alternative to Neural Nets

Xi Cheng, Bohdan Khomtchouk, Norman Matloff, Pete Mohanty

*(Submitted on 13 Jun 2018 (v1), last revised 29 Jun 2018 (this version, v2))*

Despite the success of neural networks (NNs), there is still a concern among many over their "black box" … are in fact essentially polynomial regression models. This view will have various implications for NNs, e.g. provi… …gh guidance on avoiding overfitting. In addition, we use this phenomenon to predict and confirm a multicolli… …ly, given this loose correspondence, one may choose to routinely use polynomial models instead of NNs, thus avoi… …meters and dealing with convergence issues. We present a number of empirical results; in each case, the accurac… …y-featured, open-source software package, polyreg, is available.

|  |  |
|---|---|
| Comments: | 23 pages, 1 figure, 13 tables |
| Subjects: | **Machine Learning (cs.LG)**; Machine Learning (stat.ML) |
| Cite as: | arXiv:1806.06850 [cs.LG] |
|  | (or arXiv:1806.06850v2 [cs.LG] for this version) |

**Bibliographic data**

[Enable Bibex (What is Bibex?)]

**Submissi**

From: Norm

[v1] Wed, 1

[v2] Fri, 29

*Which auth*

Link back to

If you have a

Down
- PDF
- Other

(license)

Current
cs.LG
< prev
new | rece

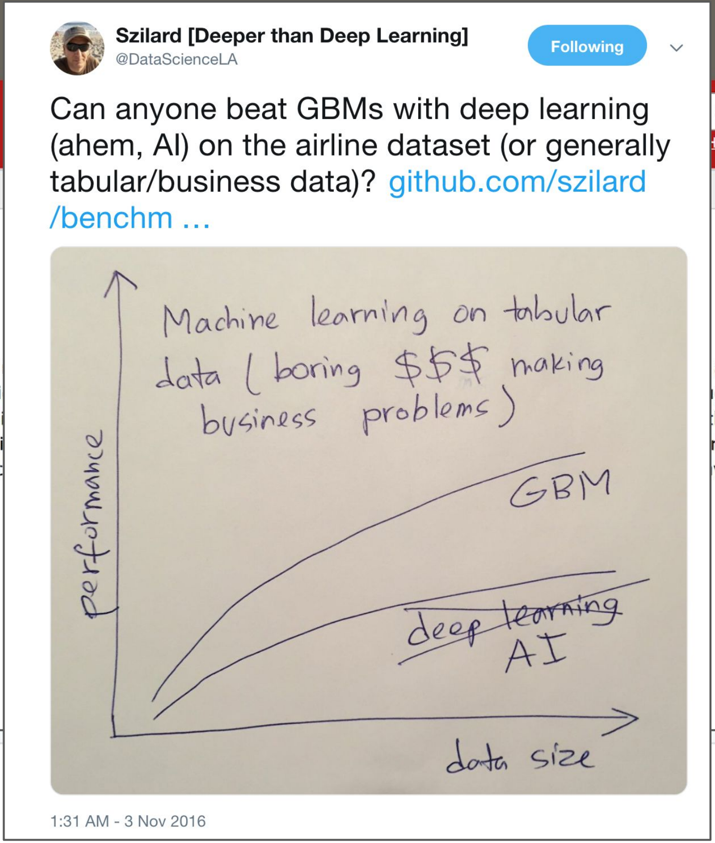
Change
cs
stat
stat.ML

Referen
- NASA

1 blog l

Google S

Bookma



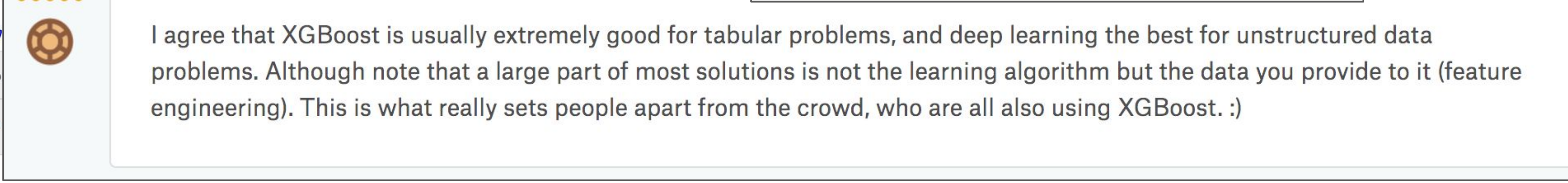
**Szilard [Deeper than Deep Learning]** @DataScienceLA

Following

Can anyone beat GBMs with deep learning (ahem, AI) on the airline dataset (or generally tabular/business data)? github.com/szilard/benchm …

Machine learning on tabular data (boring $$$ making business problems)
performance
GBM
~~deep learning~~
AI
data size

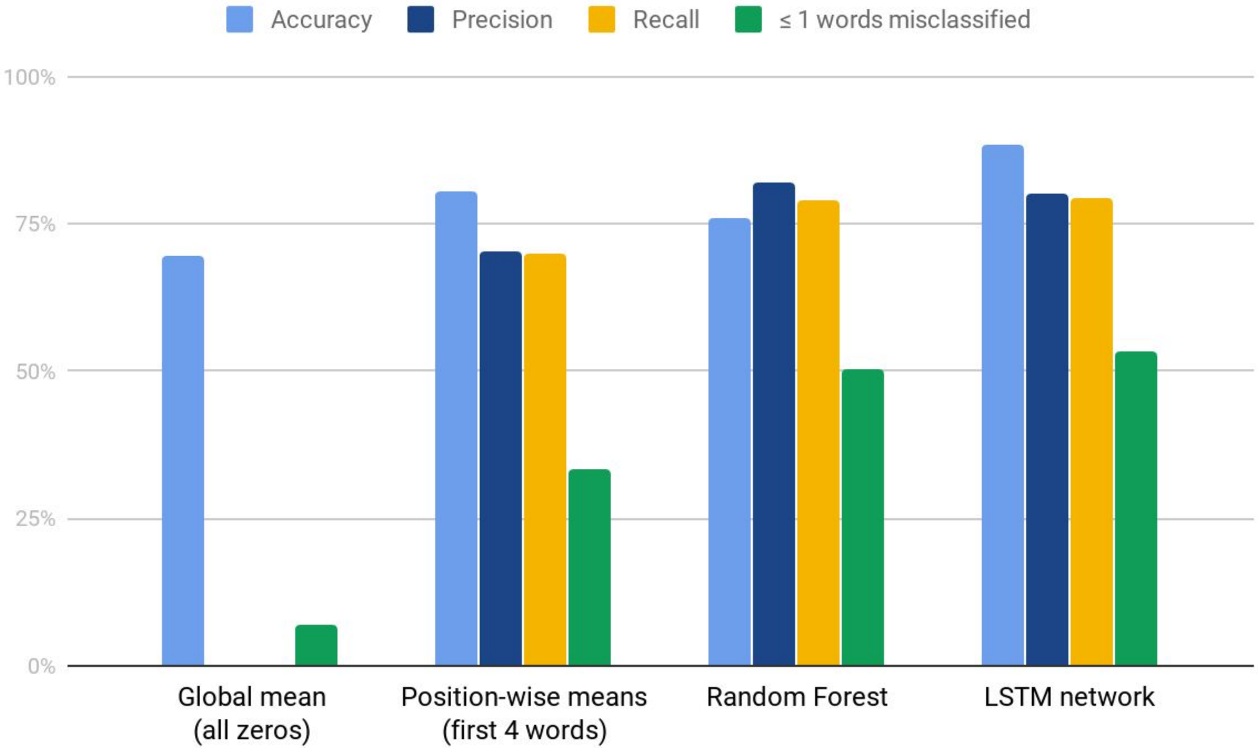
1:31 AM - 3 Nov 2016



anokas • 2 years ago • Options • Reply

3

I agree that XGBoost is usually extremely good for tabular problems, and deep learning the best for unstructured data problems. Although note that a large part of most solutions is not the learning algorithm but the data you provide to it (feature engineering). This is what really sets people apart from the crowd, who are all also using XGBoost. :)


Accuracy
Precision
Recall
≤ 1 words misclassified
100%
75%
50%
25%
0%
Global mean
(all zeros)
Position-wise means
(first 4 words)
Random Forest
LSTM network

## “Shallow” machine learning

- Relatively fast to train
- Good results with tailor-made features
- Less computational resources needed
- Available out-of-the-box
- Less hyperparameters to tune
- Impact of model choice & hyperparameters on results better understood
- Relatively easy to interpret
- Good for structured/tabular data

## “Deep” learning

- Takes *almost* raw data as input
- Learns the features itself
- Less feature engineering needed
- ...but architecture engineering instead
- Very flexible family of models
- Performance depends on many “tricks”
- Black box-ish
- Data hungry
- Slower to learn, need GPU’s
- ...but designed to work with large datasets
- Greater risk of overfitting
- Good for pattern recognition

# If something wasn’t clear enough, check:

https://medium.com/value-stream-design/introducing-one-of-the-best-hacks-in-machine-learning-the-hashing-trick-bf6a9c8af18f

https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

https://scikit-learn.org/stable/modules/ensemble.html

https://brohrer.github.io/how_decision_trees_work.html

http://ruder.io/word-embeddings-1/

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

https://danijar.com/tips-for-training-recurrent-neural-networks/

https://stats.stackexchange.com/a/352037/35989

https://www.jeremyjordan.me/nn-learning-rate/

DEEP LEARNING with Python
François Chollet
O'REILLY
Feature Engineering for Machine Learning
PRINCIPLES AND TECHNIQUES FOR DATA SCIENTISTS
Alice Zheng & Amanda Casari
Ian Goodfellow • Yoshua Bengio • Aaron Courville
DEEP LEARNING
Systemy uczące się