

Better than Deep Learning: Gradient Boosting Machines (GBM)

Szilard Pafka, PhD

Chief Scientist, Epoch (USA)

e-Rum conference, Milano, Italy / Online

June 2020



Szilard [Deeper than Deep Learning]

@DataScienceLA

physics PhD, chief (data) scientist, meetup organizer, [datascience.la](#), (visiting) professor, machine learning benchmarks

⌚ Santa Monica, California ⚡ [linkedin.com/in/szilard](#)

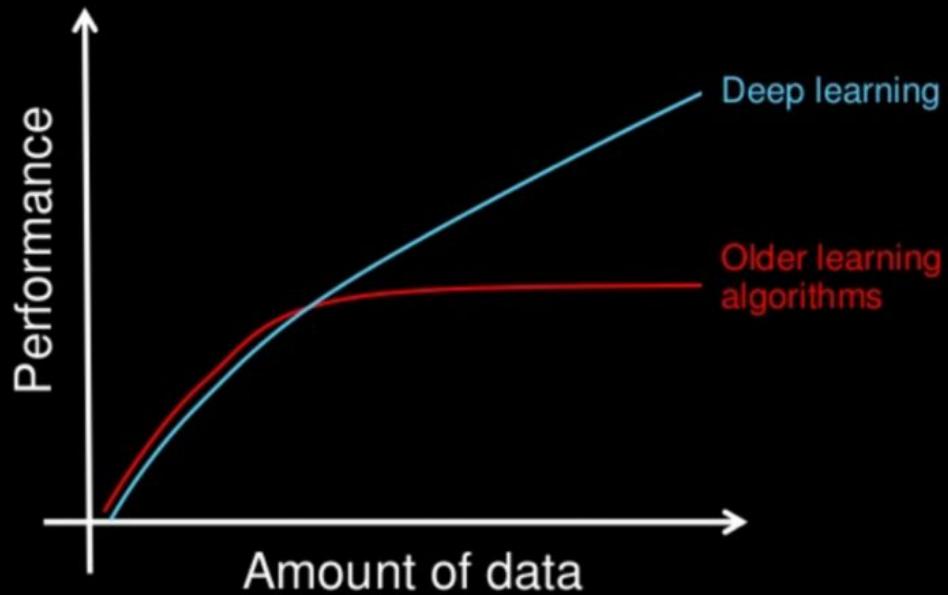
228 Following 3,311 Followers

Disclaimer:

I am not representing my employer (Epoch) in this talk

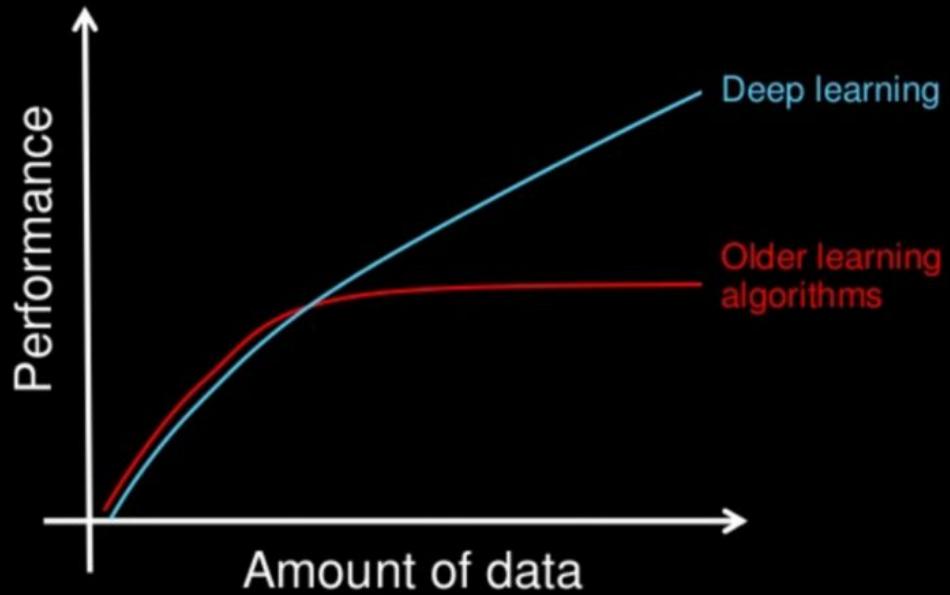
I cannot confirm nor deny if Epoch is using any of the methods, tools, results etc. mentioned in this talk

Why deep learning



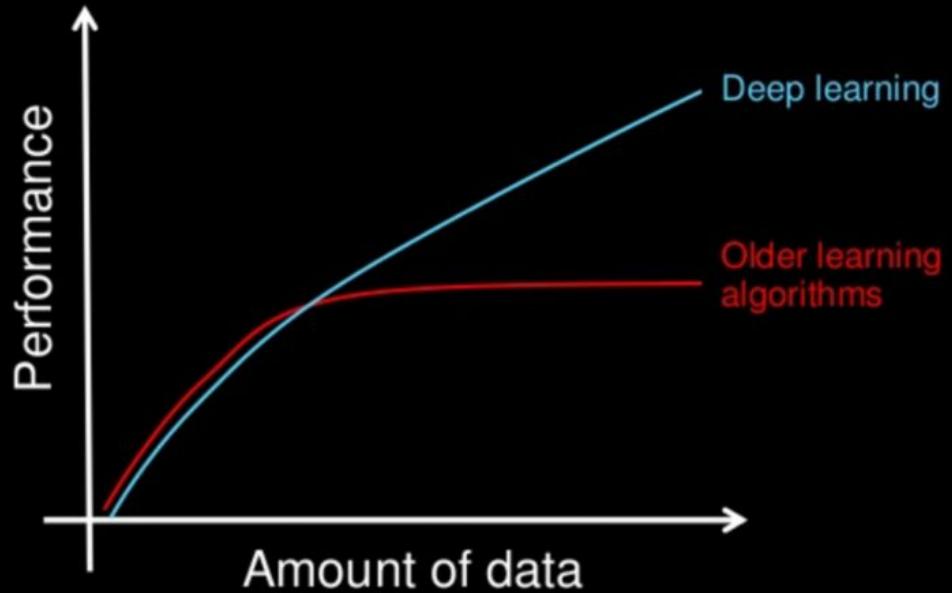
Source: Andrew Ng

Why deep learning

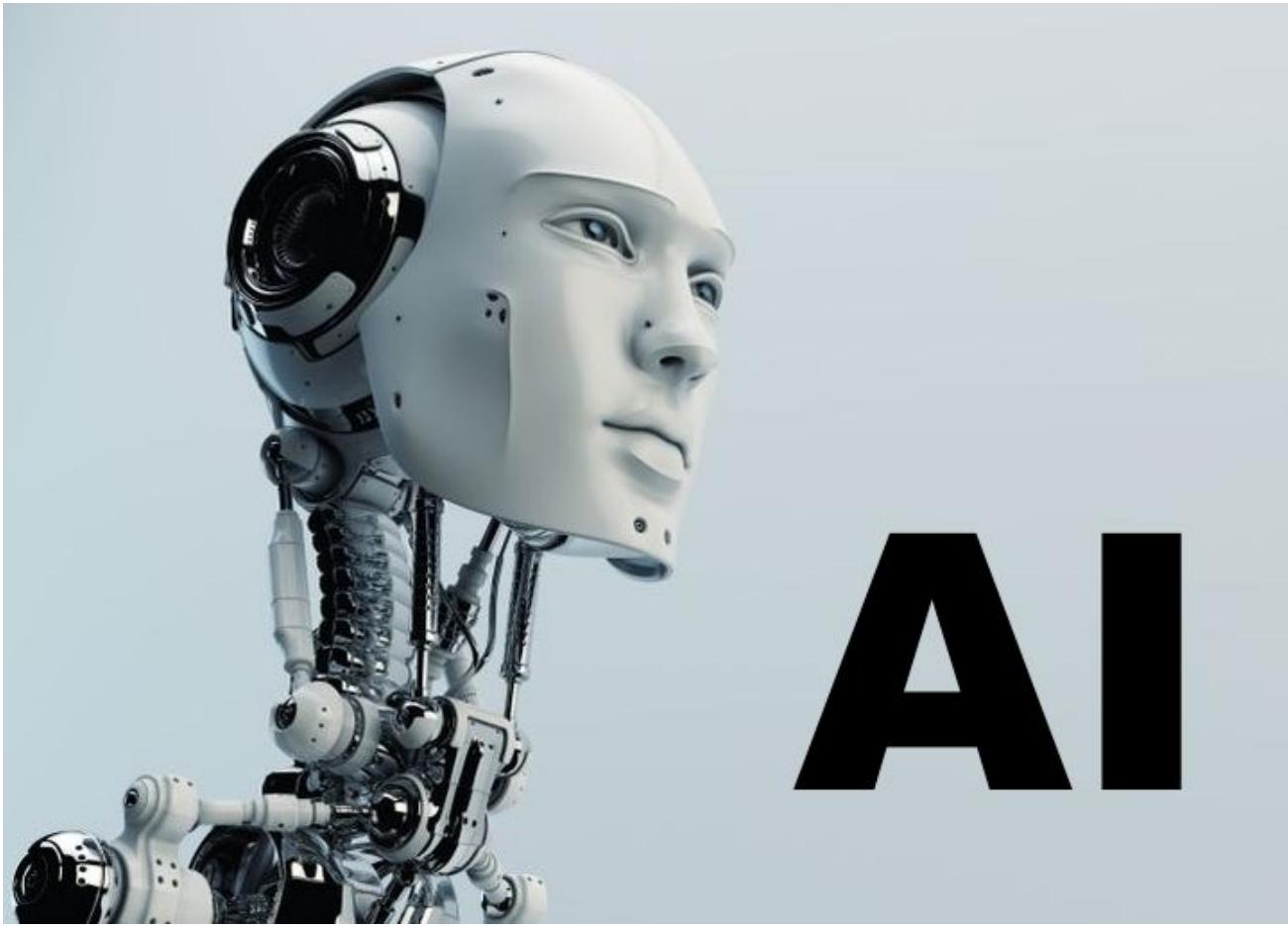


Source: Andrew Ng

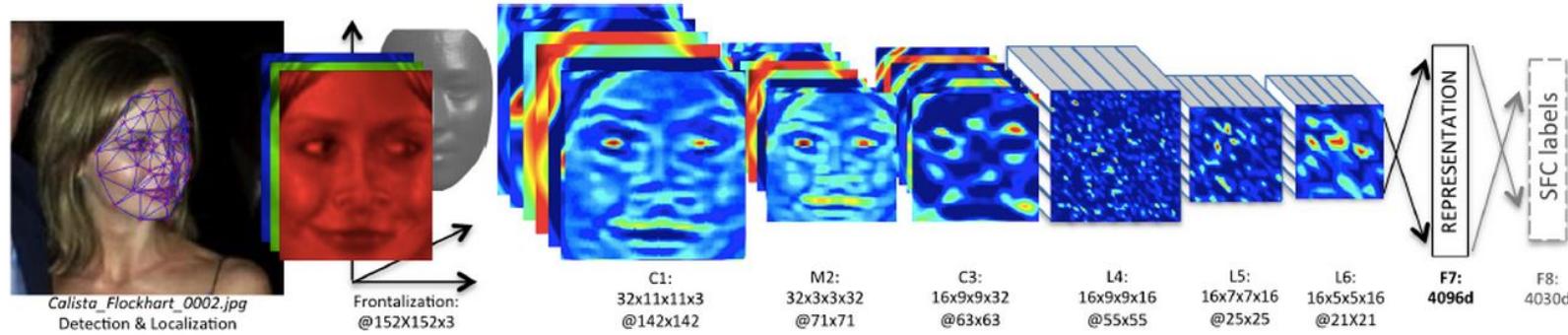
Why deep learning

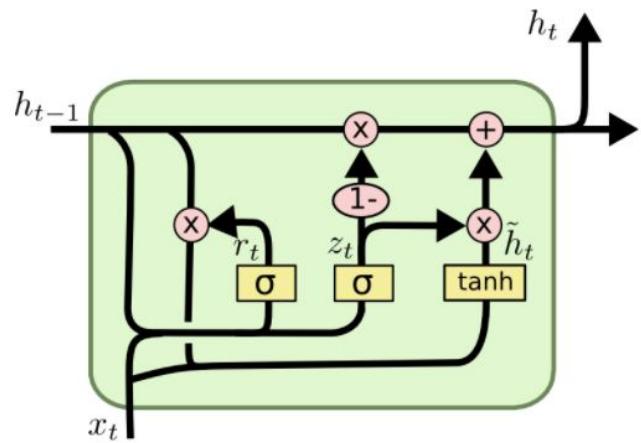
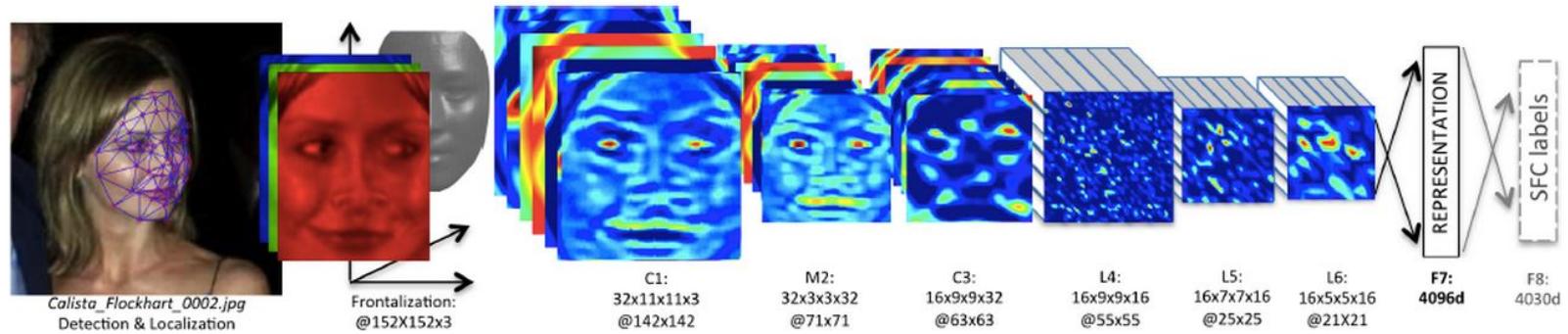


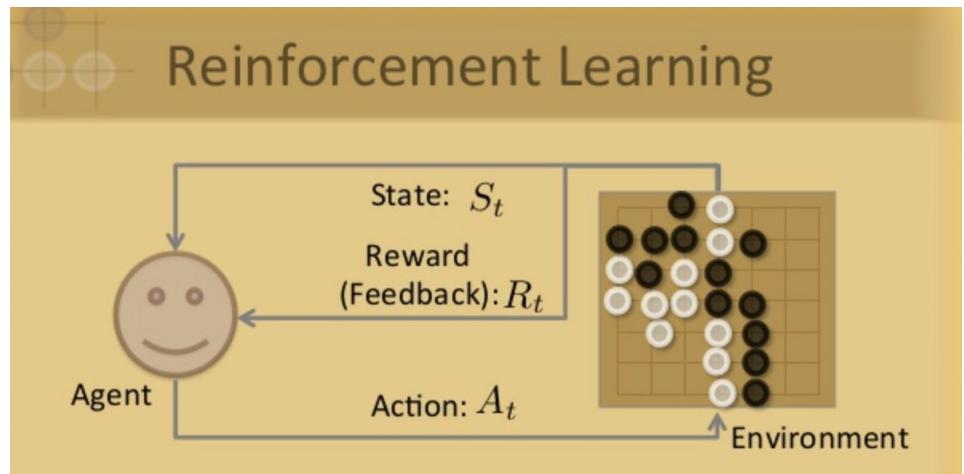
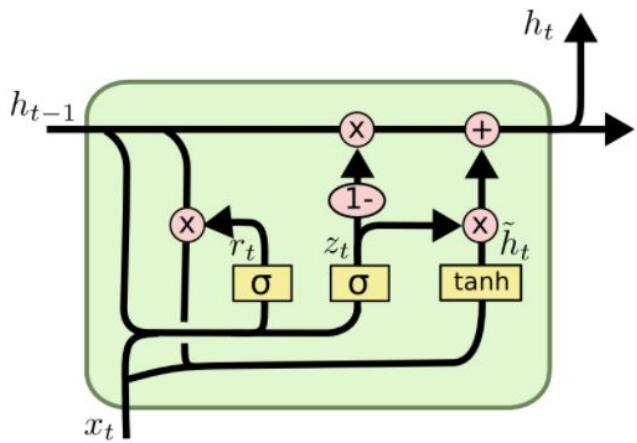
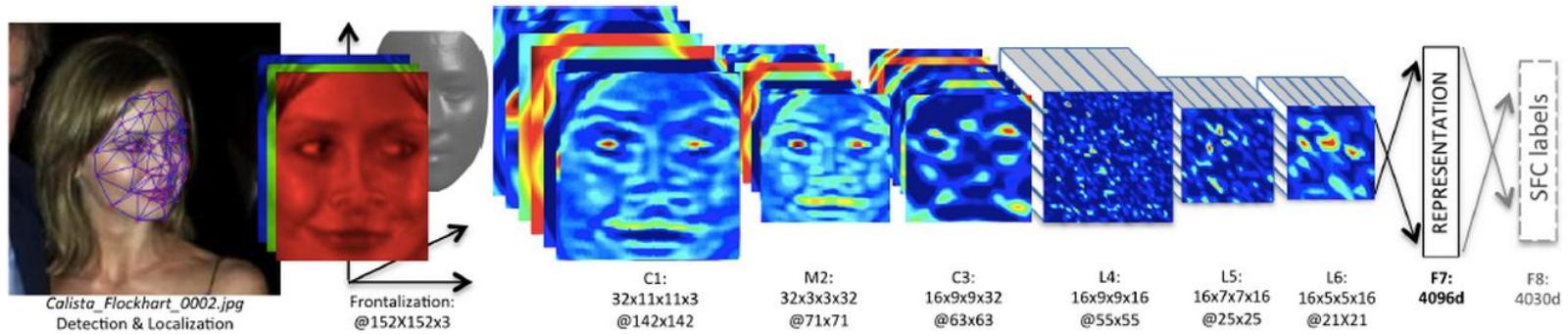
Source: Andrew Ng



AI









THAT'S NO AI

ITS JUST MORE IF STATEMENTS

0101010101011010100000001001
11000101010101010101010101010101
111000101010100010100010100010100
001010101010101010101010100001000
0101010101010101010101010100000000
0100010101010101010101010101010101
00101010101010101010101010101010101
010101010101010101010101010101010101
0010101010101010101010101010101010101
010101010101010101010101010101010101
11011010101010101010101010101010101
11111000000010101
111110101010101010101000010101010101
010101010101010101010101010101010101
010101010101010101010101010101010101
0110111110101010111110101010101010101
1101010101010000001010101010101010101

FRAUD





Params	AUC	Time (s)	Epochs
default: activation = "Rectifier", hidden = c(200,200)	73.1	270	1.8
hidden = c(50,50,50,50), input_dropout_ratio = 0.2	73.2	140	2.7
hidden = c(50,50,50,50)	73.2	110	1.9
hidden = c(20,20)			
hidden = c(20)			
RectifierWithDropout, c(200,200,200)			
ADADELTA rho = 0.95, epsilon = 1e-08	73.3	270	1.9
adaptive = FALSE default: rate = 0.005, decay = 1, momentum = 0	73.0	340	1.1
rate = 0.001, momentum = 0.5 / 1e5 / 0.99	73.2	410	0.7
rate = 0.01, momentum = 0.5 / 1e5 / 0.99	73.3	280	0.9
rate = 0.01, rate_annealing = 1e-05, momentum = 0.5 / 1e5 / 0.99	73.5	360	1
rate = 0.01, rate_annealing = 1e-04, momentum = 0.5 / 1e5 / 0.99	72.7	3700	8.7
rate = 0.01, rate_annealing = 1e-05, momentum = 0.5 / 1e5 / 0.99	73.4	350	0.9



szilard commented Nov 27, 2015

Trying to see if DL can match RF/GBM in accuracy on the airline dataset (which covers years 2005-2006, while validation and test sets sampled disjunctly from 2007). The categorical variables are kept categorical artificially and are intentionally not encoded as ordinal variables (as is common for business datasets).

kaggle

Machine Learning Challenge Winning Solutions

- The most frequently used tool by data science competition winners
 - 17 out of 29 winning solutions in kaggle last year used XGBoost
 - Solve wide range of problems, such as digit recognition, image classification, text classification; customer behavior prediction; motion detection; ad click through rate prediction; malware classification; product categorization; hazard risk prediction; massive online course dropout rate prediction
- Present and Future of KDDCup. Ron Bekkerman (KDDCup 2015 chair): "Something dramatic happened in Machine Learning over the past couple of years. It is called XGBoost - a package implementing Gradient Boosted Decision Trees that works wonders in data classification. Apparently, every winning team used XGBoost, mostly in ensembles with other classifiers. Most surprisingly, the winning teams report very minor improvements that ensembles bring over a single well-configured XGBoost."
- A lot contributions from the kaggle community

56:01 / 1:16:29

CC HD

XGBoost A Scalable Tree Boosting System June 02, 2016

26,599 views

1212

1

SHARE

SAVE

...



DataScience.LA

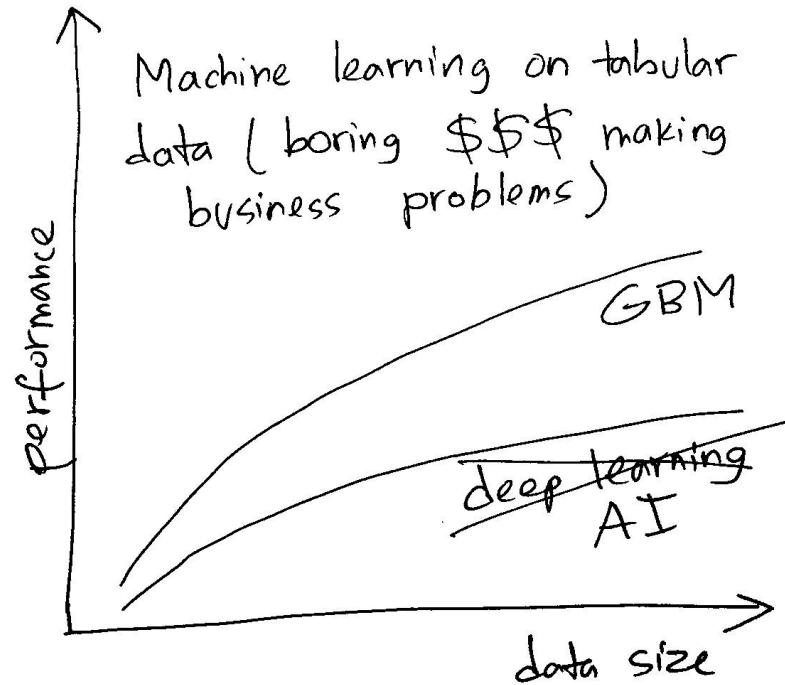
Published on Jun 3, 2016

SUBSCRIBE 3.4K

Szilard [Deeper than Deep Learning] @DataScienceLA · 2 Nov 2016

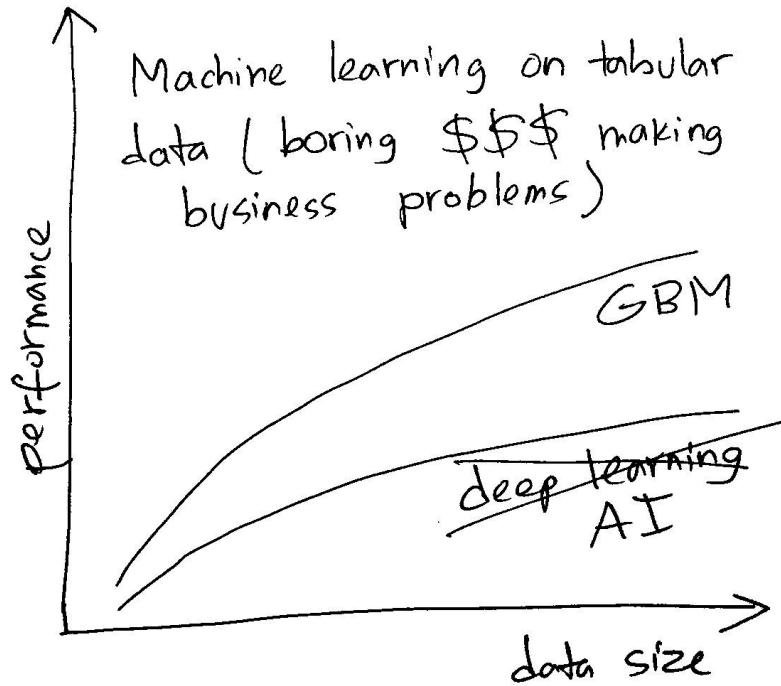


Can anyone beat GBMs with deep learning (ahem, AI) on the airline dataset (or generally tabular/business data)? [github.com/szilard/benchm...](https://github.com/szilard/benchmarks)



Szilard [Deeper than Deep Learning] @DataScienceLA · 2 Nov 2016

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



3. Parameter tuning and ensembling

```
# train xgboost
xgb <- xgboost(data = data.matrix(tr
label = train$destina
eta = 0.001,
max_depth = 15,
nround=25,
subsample = 0.5,
colsample_bytree = 0.
seed = 1,
eval_metric = "merror
objective = "multi:sc
num_class = 12,
nthread = 4|
```

▶ ▷ 🔍 2:58 / 4:06

What Kaggle has learned from almost a million data scientists - Anthony Goldbloom
18,153 views



O'Reilly

Published on May 25, 2017

MODEL	1ST	2ND	AVG	1ST	2ND
BST-DT	0.580	0.228	RF	0.727	0.207
RF	0.390	0.525	ANN	0.053	0.172
BAG-DT	0.030	0.232	BSTD T	0.059	0.228
SVM	0.000	0.008	SVM	0.043	0.195
ANN	0.000	0.007	LR	0.089	0.132
KNN	0.000	0.000	BAGDT	0.002	0.012
BST-STMP	0.000	0.000	KNN	0.023	0.045
DT	0.000	0.000	BSTST	0.004	0.009
LOGREG	0.000	0.000	PRC	0	0
NB	0.000	0.000	NB	0	0

An Empirical Comparison of Supervised Learning Algorithms

<http://www.cs.cornell.edu/~alexn/papers/empirical.icml06.pdf>

An Empirical Evaluation of Supervised Learning in High Dimensions

<http://lowrank.net/nikos/pubs/empirical.pdf>

MODEL	1ST	2ND
BST-DT	0.580	0.228
RF	0.390	0.525
BAG-DT	0.030	0.232
SVM	0.000	0.008
ANN	0.000	0.007
KNN	0.000	0.000
BST-STMP	0.000	0.000
DT	0.000	0.000
LOGREG	0.000	0.000
NB	0.000	0.000

AVG	1ST	2ND
RF	0.727	0.207
ANN	0.053	0.172
BSTD	0.059	0.228
SVM	0.043	0.195
LR	0.089	0.132
BAGDT	0.002	0.012
KNN	0.023	0.045
BSTST	0.004	0.009
PRC	0	0
NB	0	0

An Empirical Comparison of Supervised Learning Algorithms

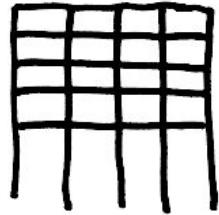
<http://www.cs.cornell.edu/~alexn/papers/empirical.icml06.pdf>

An Empirical Evaluation of Supervised Learning in High Dimensions

<http://lowrank.net/nikos/pubs/empirical.pdf>

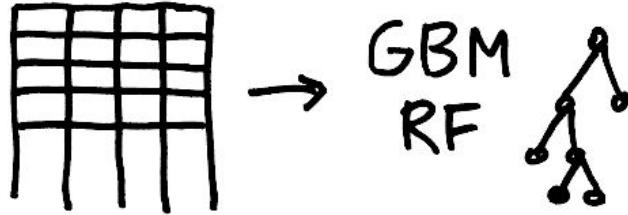
-  [gbm 1.5-3.tar.gz](#) 2005-10-07 22:49 249K
-  [gbm 1.5-5.tar.gz](#) 2006-01-21 12:58 249K
-  [gbm 1.5-7.tar.gz](#) 2006-04-18 11:58 254K
-  [gbm 1.5.tar.gz](#) 2005-05-09 22:56 250K
-  [gbm 1.6-1.tar.gz](#) 2007-06-14 08:29 257K

-  [randomForest 4.5-12.tar.gz](#) 2005-06-21 09:36 80K
-  [randomForest 4.5-15.tar.gz](#) 2005-09-22 19:35 81K
-  [randomForest 4.5-16.tar.gz](#) 2006-01-24 10:21 81K
-  [randomForest 4.5-18.tar.gz](#) 2006-12-10 16:07 67K
-  [randomForest 4.5-19.tar.gz](#) 2007-10-16 20:38 67K

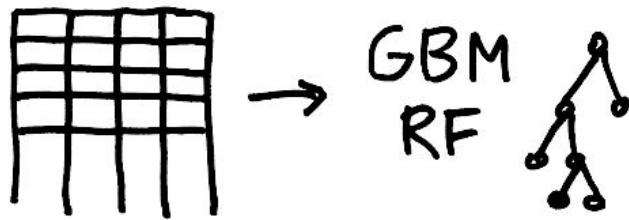


GBM
RF



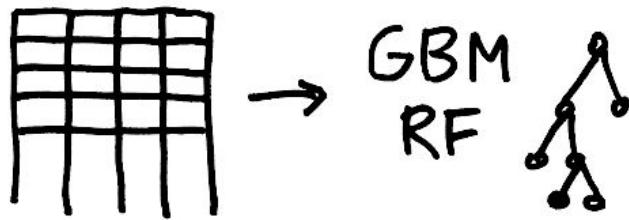


用 → LR ~~決策~~

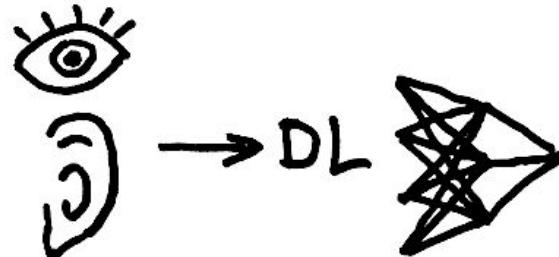


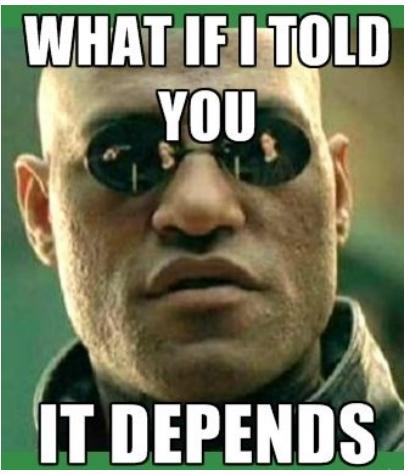
用 → LR





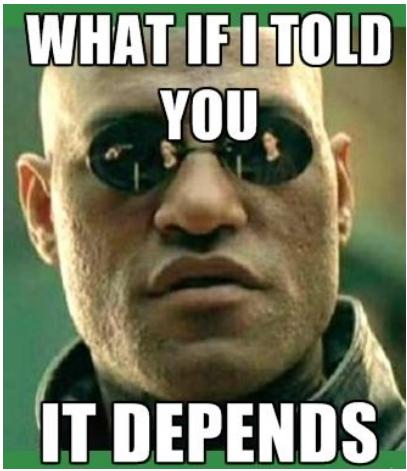
用 → LR



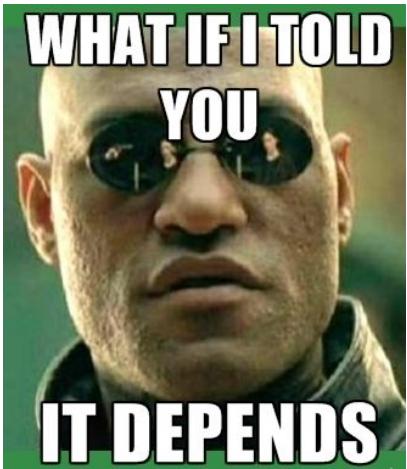


**WHAT IF I TOLD
YOU**

IT DEPENDS



**WHAT IF I TOLD
YOU**

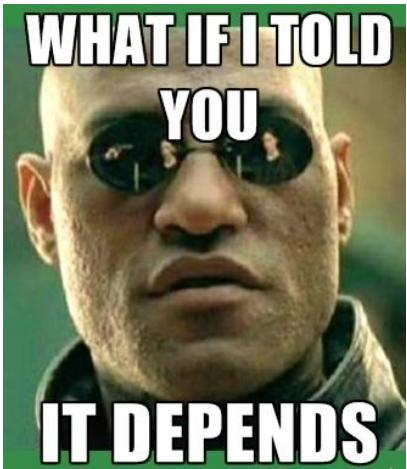


IT DEPENDS



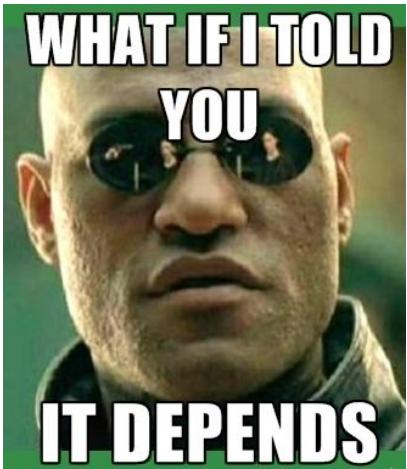
Hyperparameter
tuning





Hyperparameter
tuning



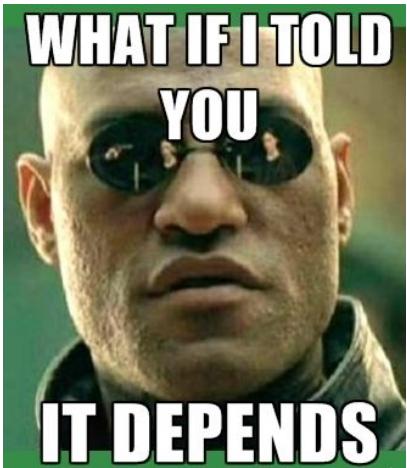


Hyperparameter tuning

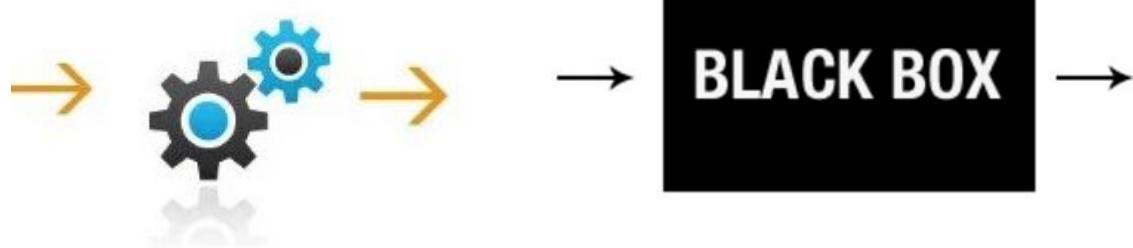


Features



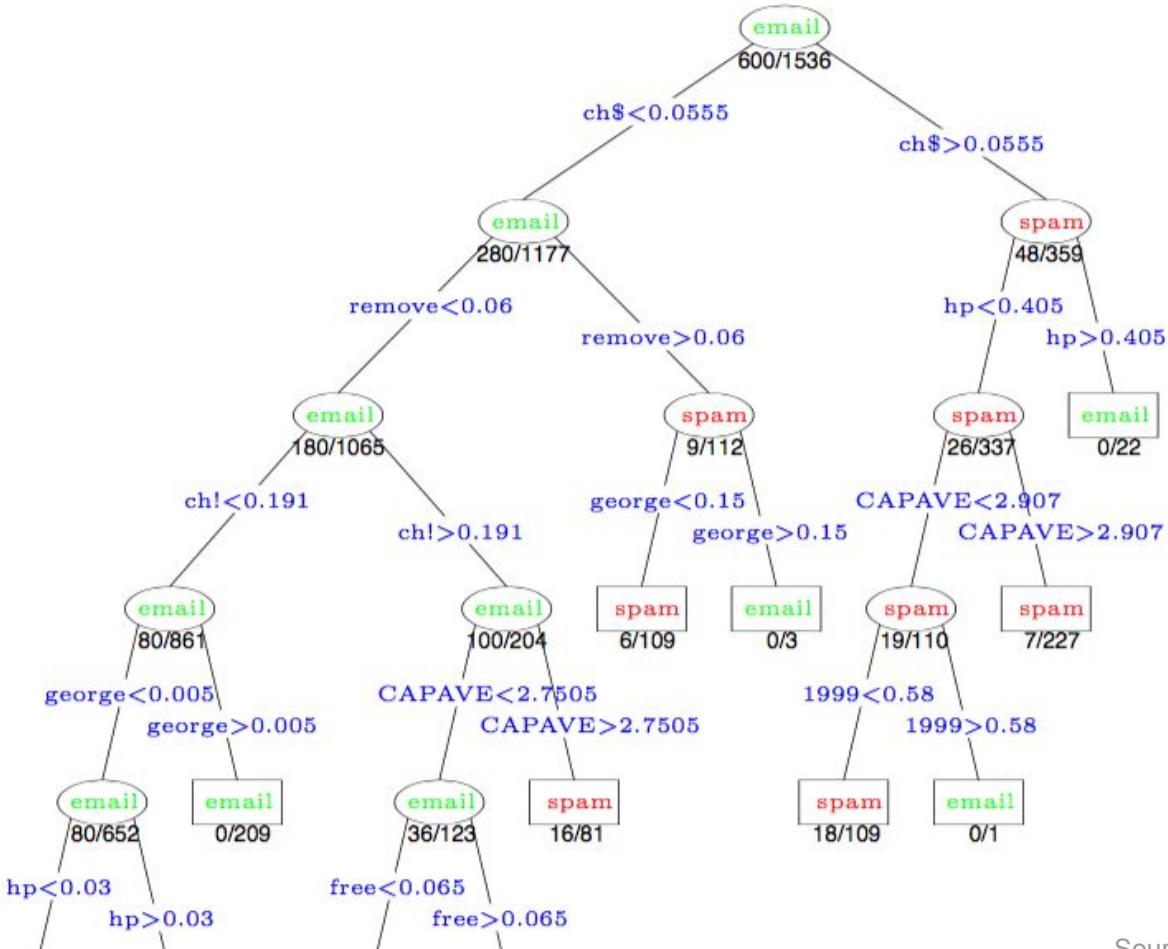


Features



GBM?

TELL ME MORE



Source: Hastie et al, ESL 2ed

Algorithm 10.1 AdaBoost.M1.

1. Initialize the observation weights $w_i = 1/N, i = 1, 2, \dots, N$.
2. For $m = 1$ to M :
 - (a) Fit a classifier $G_m(x)$ to the training data using weights w_i .
 - (b) Compute
$$\text{err}_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}.$$
 - (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
 - (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))], i = 1, 2, \dots, N$.
3. Output $G(x) = \text{sign} \left[\sum_{m=1}^M \alpha_m G_m(x) \right]$.



open source

- R packages
- Python scikit-learn
- Vowpal Wabbit
- H2O
- xgboost
- Spark MLlib
- a few others



open source

- R packages
- Python scikit-learn
- Vowpal Wabbit
- H2O
- xgboost
- Spark MLlib
- a few others

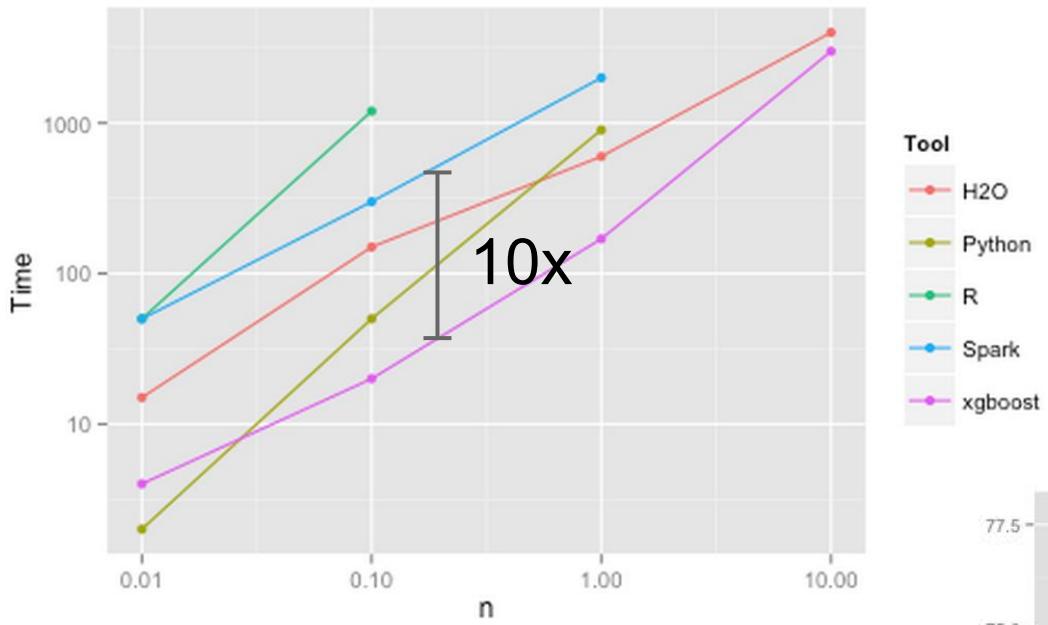


[szilard / benchm-ml](#)

Star

1,203

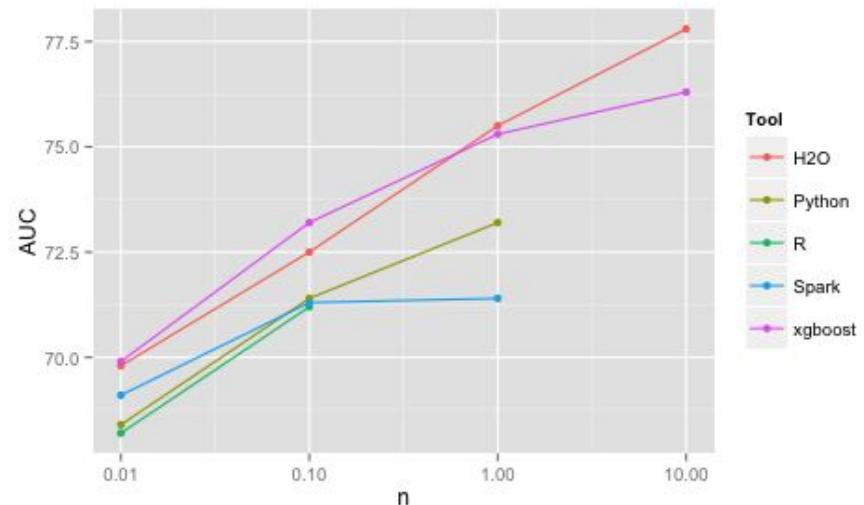
Simple/limited/incomplete benchmark



szilard / **benchm-ml**

Tool

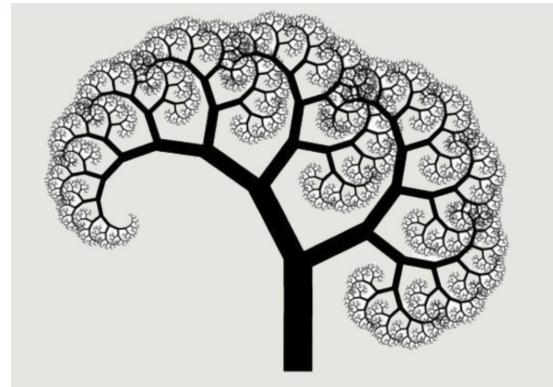
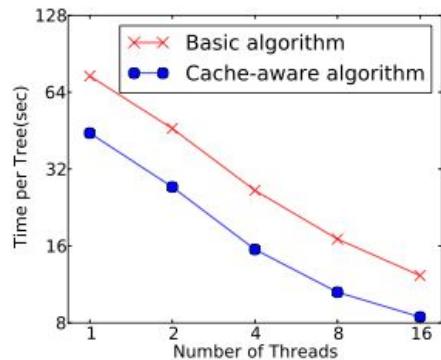
- H2O
- Python
- R
- Spark
- xgboost



Tool

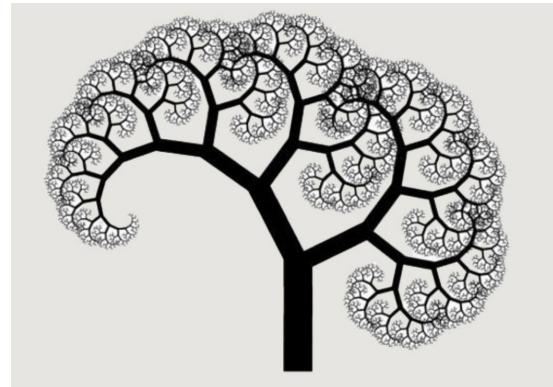
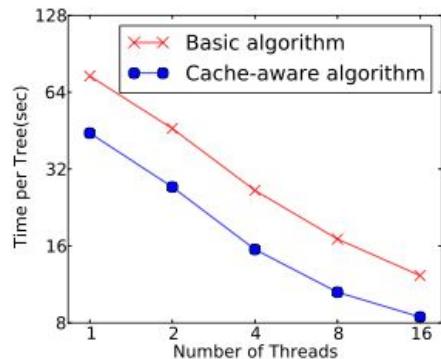
- H2O
- Python
- R
- Spark
- xgboost

XGBoost: A Scalable Tree Boosting System



Microsoft / LightGBM

XGBoost: A Scalable Tree Boosting System



Microsoft / LightGBM



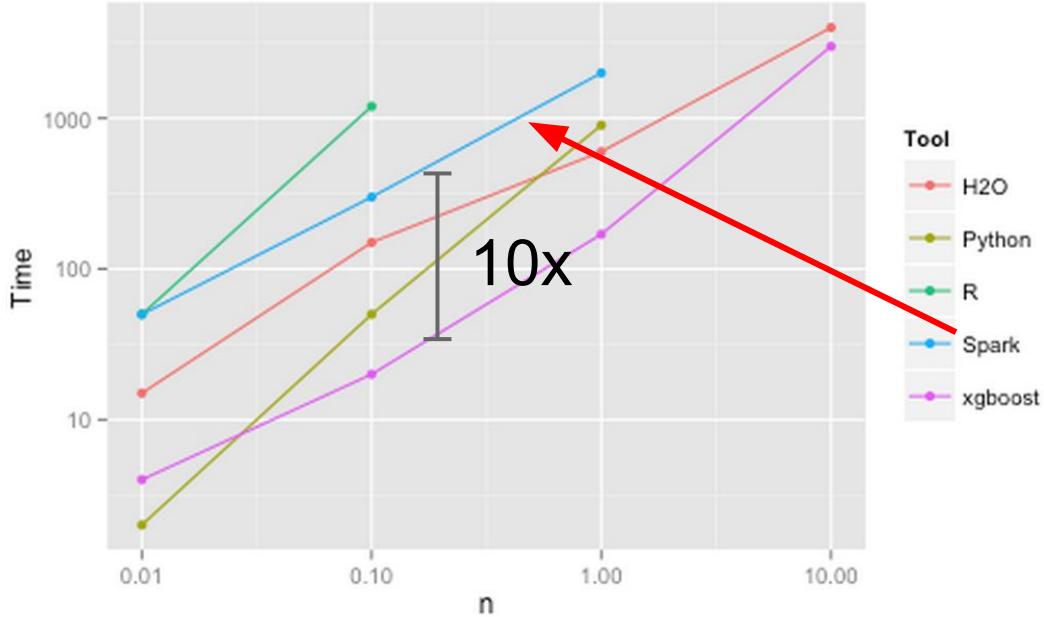
← → C <https://cran.r-project.org/web/pa>

xgboost: Extreme Gradient Boosting

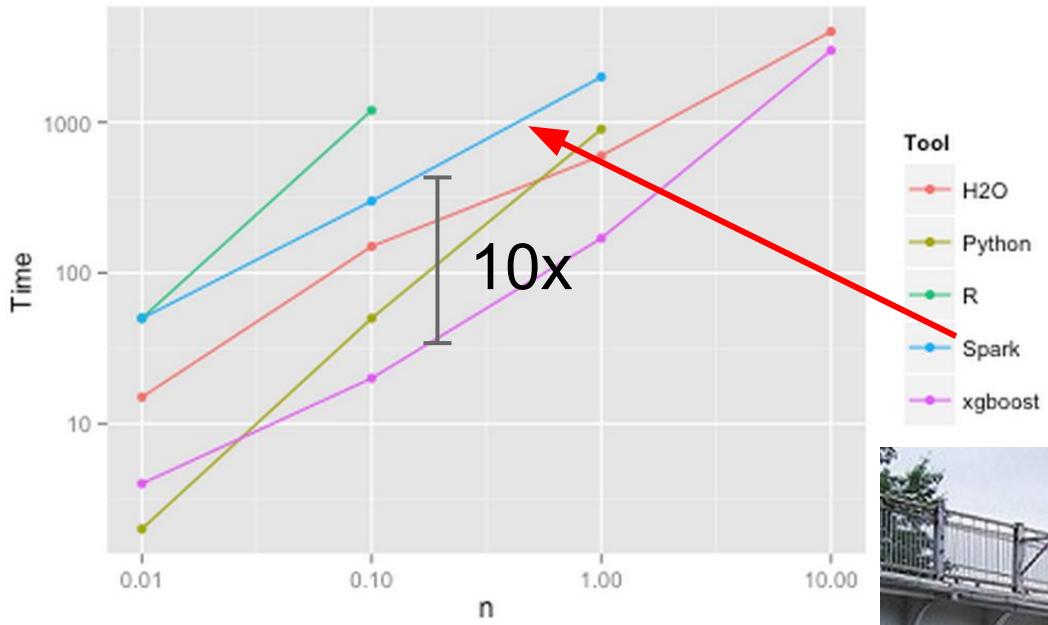
← → C <https://cran.r-project.org/>

h2o: R Interface for H2O





szilard / **benchm-ml**

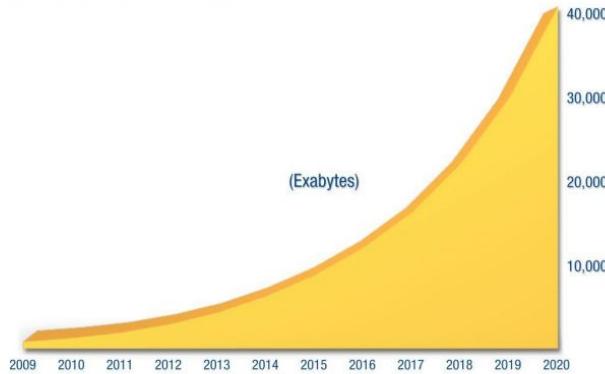


szilard / benchm-ml



Figure 1

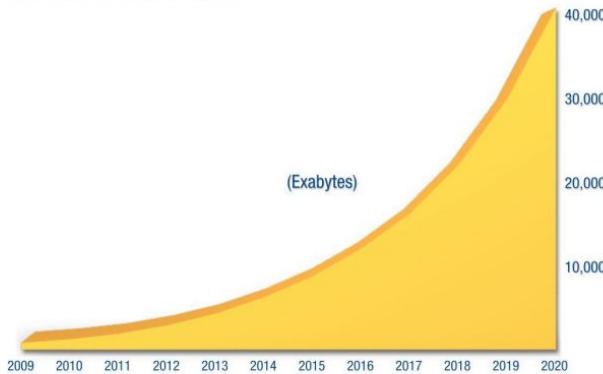
The Digital Universe: 50-fold Growth from the Beginning of 2010 to the End of 2020



Source: IDC's Digital Universe Study, sponsored by EMC, December 2012

Figure 1

The Digital Universe: 50-fold Growth from the Beginning of 2010 to the End of 2020



Source: IDC's Digital Universe Study, sponsored by EMC, December 2012



Hadley Wickham
@hadleywickham

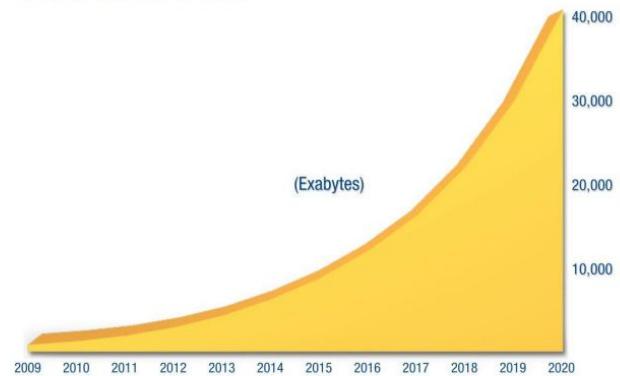


Following

"It takes a big man to admit his data is small" —
@jcheng

Figure 1

The Digital Universe: 50-fold Growth from the Beginning of 2010 to the End of 2020



Hadley Wickham
@hadleywickham



Following

"It takes a big man to admit his data is small" —
@jcheng

TYPICAL SIZE OF DATASETS

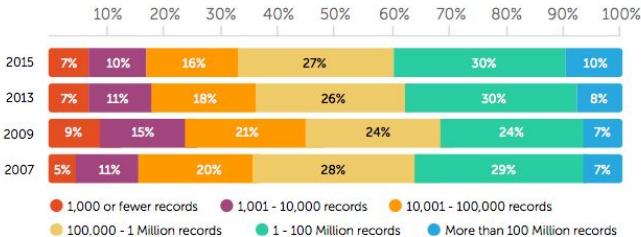
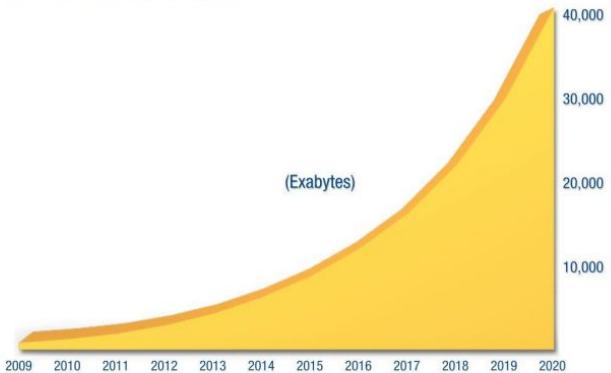


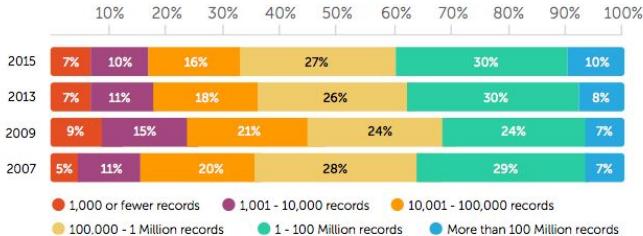
Figure 1

The Digital Universe: 50-fold Growth from the Beginning of 2010 to the End of 2020



Source: IDC's Digital Universe Study, sponsored by EMC, December 2012

TYPICAL SIZE OF DATASETS

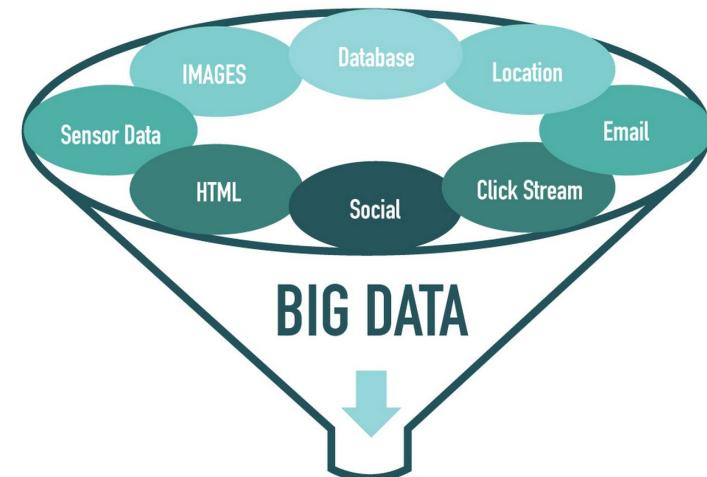


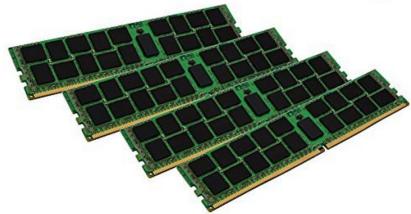
Hadley Wickham
@hadleywickham



Following

"It takes a big man to admit his data is small" —
@jcheng





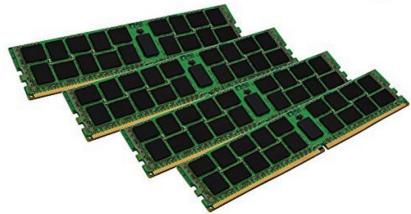
Kingston Technology Value RAM 128GB Kit (4x32GB) 2133MHz DDR4 ECC Reg CL15 (KVR21R15D4K4/128)

by [Kingston Technology](#)

[Be the first to review this item](#)

Was: \$743.99

Price: **\$743.96 & FREE Shipping.** [Details](#)



Kingston Technology Value RAM 128GB Kit (4x32GB) 2133MHz DDR4 ECC Reg CL15 (KVR21R15D4K4/128)

by [Kingston Technology](#)

[Be the first to review this item](#)

Was: \$743.99

Price: **\$743.96 & FREE Shipping.** [Details](#)



Model	vCPU	Mem (GiB)
r3.8xlarge	32	244
x1e.32xlarge	128	3,904
u-12tb1.metal	448	12 (TiB)



Szilard @DataScienceLA · Aug 3

I wish my #machinelearning worked... ("both" is not a choice 😊) #bigdata
#datascience #rstats #pydata cc @h2o @databricks @cloudera @kaggle

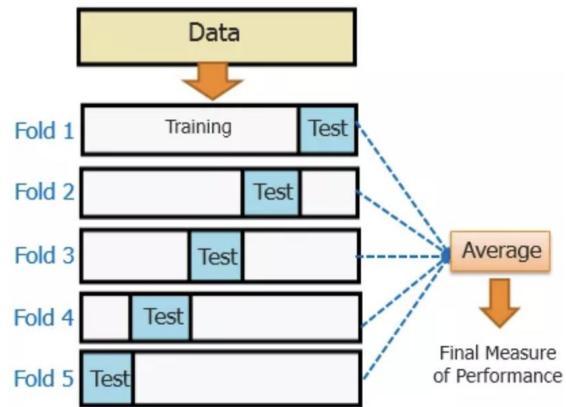
10% on 10x bigger data

70% 10x faster

20% I don't care about either

104 votes • Final results





Hyperparameter
tuning



szilard / GBM-perf

r4.8xlarge (32 cores, but run on physical cores only/no hyperthreading)

```
git clone https://github.com/szilard/GBM-perf
cd GBM-perf/cpu
sudo docker build -t gbmperf_cpu
sudo docker run --rm gbmperf_cpu
```

Tool	Time[s] 100K	Time[s] 1M	Time[s] 10M	AUC 1M	AUC 10M
h2o	16	20	100	0.762	0.776
xgboost	3.8	12	78	0.749	0.755
lightgbm	2.4	5.2	42	0.764	0.774
catboost	5.4	50	490	0.740	0.744

szilard / GBM-perf

r4.8xlarge (32 cores, but run on physical cores only/no hyperthreading)

```
git clone https://github.com/szilard/GBM-perf
cd GBM-perf/cpu
sudo docker build -t gbmperf_cpu
sudo docker run --rm gbmperf_cpu
```

Tool	Time[s] 100K	Time[s] 1M	Time[s] 10M	AUC 1M	AUC 10M
h2o	16	20	100	0.762	0.776
xgboost	3.8	12	78	0.749	0.755
lightgbm	2.4	5.2	42	0.764	0.774
catboost	5.4	50	490	0.740	0.744

p3.2xlarge (1 GPU, Tesla V100)



Tool	Time[s] 100K	Time[s] 1M	Time[s] 10M	AUC 1M	AUC 10M
h2o xgboost	9	14	60	0.749	0.756
xgboost	2.4	4.8	13	0.750	0.756
lightgbm	10	16	67	0.766	0.774
catboost	3.9	10	135	0.742	0.750

CPU (m5.12xlarge):

Tool	time [s]	AUC	RAM train [GB]
h2o	520	0.775	8
xgboost	510	0.751	15
lightgbm	310	0.774	5
catboost	3360	0.723 ?!	140

100M records and RAM usage

CPU (m5.12xlarge):

Tool	time [s]	AUC	RAM train [GB]
h2o	520	0.775	8
xgboost	510	0.751	15
lightgbm	310	0.774	5
catboost	3360	0.723 ?!	140

100M records and RAM usage

GPU (Tesla V100):

Tool	time [s]	AUC	GPU mem [GB]	extra RAM [GB]
h2o xgboost	270	0.755	4	30
xgboost	80	0.756	6	0
lightgbm	400	0.774	3	6
catboost	crash (OOM)		>16	14

```
## exporting model for scoring
```

```
h2o.download_mojo(md_rf, path = "./h2o")
```

```
## building prediction service
```

```
# (need jetty-runner.jar ROOT.war from Steam)
```

```
java -jar jetty-runner.jar ROOT.war
```

```
curl -X POST --form mojo=@h2o_RF.zip --form jar=@h2o-genmodel.jar \  
localhost:8080/makewar > h2o_RF_MOJO.war
```

GitHub Gist

Search...



szilard / [h2o_scoring.R](#)

H₂O.ai

```
## exporting model for scoring
```

```
h2o.download_mojo(md_rf, path = "./h2o")
```

```
## building prediction service
```

```
# (need jetty-runner.jar ROOT.war from Steam)
```

```
java -jar jetty-runner.jar ROOT.war
```

```
curl -X POST --form mojo=@h2o_RF.zip --form jar=@h2o-genmodel.jar \  
localhost:8080/makewar > h2o_RF_MOJO.war
```

```
## run prediction service
```

```
java -jar jetty-runner.jar --port 20000 h2o_RF_MOJO.war
```

```
## score via REST API
```

```
time curl "http://localhost:20000/predict?Month=c-8&DayofMonth=c-21&Da  
# (fast scoring needs JVM to warm up with a few requests)
```



```
## read CSV (e.g. data.table::fread) or get data from database (SQL connector)
## do one-hot-encoding of categorical variables (e.g. Matrix::sparse.model.matrix)

## special optimized data structure
dxgb_train <- xgb.DMatrix(data = X_train, label = y_train)

## TRAIN
md <- xgb.train(data = dxgb_train,
                  objective = "binary:logistic",
                  nround = 100, max_depth = 10, eta = 0.1,
                  tree_method = "hist")

## SCORE
yhat <- predict(md, newdata = X_test)

## evaluation (score distribution, ROC curve, AUC etc.)
```



```
h2o.gbm(x, y, training_frame, model_id, checkpoint, ignore_const_cols = TRUE,
  distribution = c("AUTO", "gaussian", "bernoulli", "multinomial", "poisson",
  "gamma", "tweedie", "laplace", "quantile", "huber"), quantile_alpha = 0.5,
  tweedie_power = 1.5, huber_alpha = 0.9, ntrees = 50, max_depth = 5,
  min_rows = 10, learn_rate = 0.1, learn_rate_annealing = 1,
  sample_rate = 1, sample_rate_per_class, col_sample_rate = 1,
  col_sample_rate_change_per_level = 1, col_sample_rate_per_tree = 1,
  nbins = 20, nbins_top_level = 1024, nbins_cats = 1024,
  validation_frame = NULL, balance_classes = FALSE, class_sampling_factors,
  max_after_balance_size = 5, seed, build_tree_one_node = FALSE,
  nfolds = 0, fold_column = NULL, fold_assignment = c("AUTO", "Random",
  "Modulo", "Stratified"), keep_cross_validation_predictions = FALSE,
  keep_cross_validation_fold_assignment = FALSE,
  score_each_iteration = FALSE, score_tree_interval = 0,
  stopping_rounds = 0, stopping_metric = c("AUTO", "deviance", "logloss",
  "MSE", "AUC", "misclassification", "mean_per_class_error"),
  stopping_tolerance = 0.001, max_runtime_secs = 0, offset_column = NULL,
  weights_column = NULL, min_split_improvement = 1e-05,
  histogram_type = c("AUTO", "UniformAdaptive", "Random", "QuantilesGlobal",
  "RoundRobin"), max_abs_leafnode_pred, pred_noise_bandwidth = 0,
  categorical_encoding = c("AUTO", "Enum", "OneHotInternal", "OneHotExplicit",
  "Binary", "Eigen"))
```

Arno Candel in GBM, R, Technical, Tutorials | June 16, 2016

H2O GBM Tuning Tutorial for R

In this tutorial, we show how to build a well-tuned H2O GBM model for a supervised classification task. and use a small dataset to allow you to reproduce these results in a few minutes on a laptop. This script ca dreds of GBs large and H2O clusters with dozens of compute nodes.

① machinelearningmastery.com/configure-gradient-boosting-algorithm/



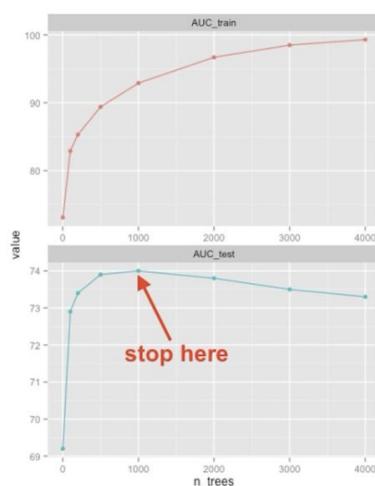
[Start Here](#)

Search...

How to Configure the Gradient Boosting Algorithm

by Jason Brownlee on September 12, 2016 in XGBoost

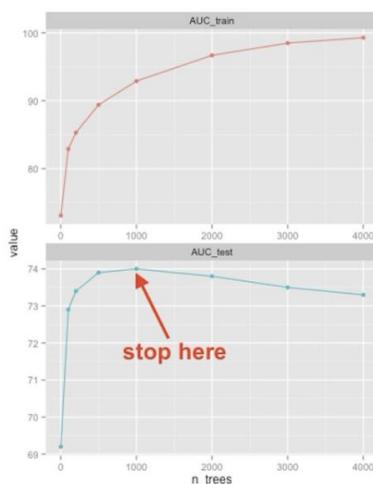
GBM/GBDT tip of the day: If it's not using early stopping, it's crap. 😬😬😬 (early stopping: checking accuracy metric on a validation set and stopping when it starts deteriorating by X) If not, you are either overfitting or wasting CPU/training time (or both). Don't do either!



Szilard [Deeper than Deep Learning]

@DataScienceLA

GBM/GBDT tip of the day: If it's not using early stopping, it's crap. 😬😬😬 (early stopping: checking accuracy metric on a validation set and stopping when it starts deteriorating by X) If not, you are either overfitting or wasting CPU/training time (or both). Don't do either!



Szilard [Deeper than Deep Learning]

@DataScienceLA

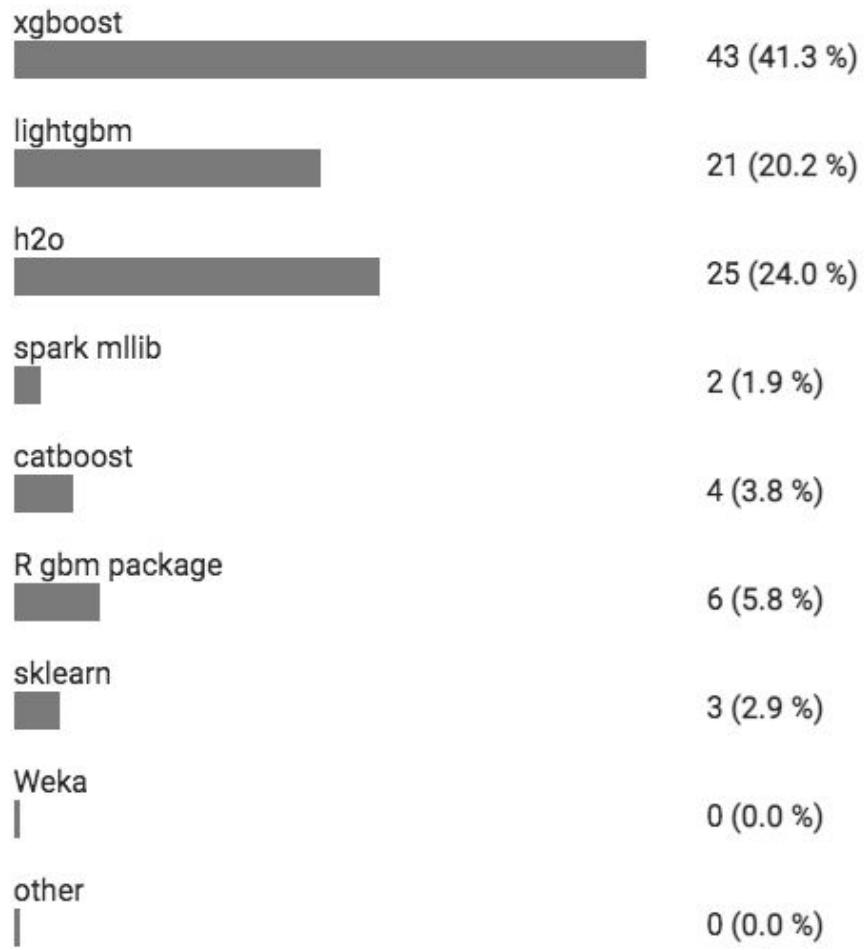
in all 3 top GBM implementations ([#xgboost](#), [#lightgbm](#), [#h2oai](#)) all you have to do (for early stopping) is to set a few params in the training function

```
md <- xgb.train(data = dxgb_train,
+                 objective = "binary:logistic",
+                 max_depth = 10, eta = 0.1,
+                 nround = 10000, early_stopping_rounds = 10, watchlist = list(valid=dxgb_valid),
+                 tree_method = "hist")
```

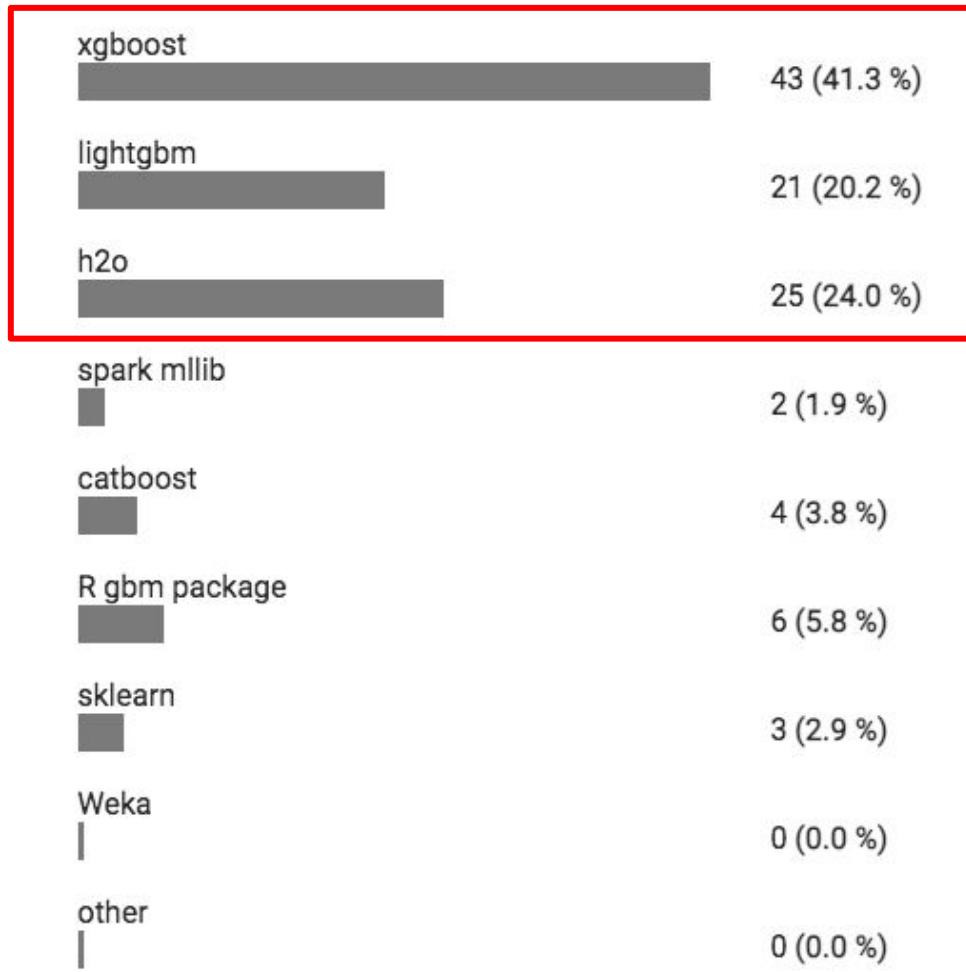
[1] valid-error:0.207010
Will train until valid_error hasn't improved in 10 rounds.

[2] valid-error:0.203230
[3] valid-error:0.202950
[4] valid-error:0.202810
[5] valid-error:0.202550
[6] valid-error:0.202380
[7] valid-error:0.202410
[8] valid-error:0.202380
....
[113] valid-error:0.197940
[114] valid-error:0.197960
[115] valid-error:0.197940
[116] valid-error:0.197950
[117] valid-error:0.197820
Stopping. Best iteration:
[107] valid-error:0.197750

	gbm (R pkg)	xgboost	lightgbm	h2o
easy R install	cran	cran	linux OK	java+cran
maintained	retired	yes	yes	yes
preprocessing	not needed	1-hot	1-hot/categ int	not needed
new cats scoring	yes	no	no	yes
early stopping	no	yes	yes	yes
speed (CPU)	1 core	ok	fastest	slow (small data)
GPU supported	no	yes	yes	via xgboost
speed GPU	NA	fastest	ok/slow	indirectly/slower
REST scoring	no	no	no	yes
other algos	no	RF	RF	RF/GLM/NN
best for	teaching/historic	Kaggle	Kaggle	prod/real-time



show the configuration of this Ferendum



show the configuration of this Ferendum

Winning Solution of KaggleDays 2019 Competition in San Francisco

Winning Solution of KaggleDays 2019 Competition in San Francisco

During the first two to three hours of the competition, we focused on exploratory data analysis (EDA). We first analyzed the distributions of raw features between train and test

Winning Solution of KaggleDays 2019 Competition in San Francisco

During the first two to three hours of the competition, we focused on exploratory data analysis (EDA). We first analyzed the distributions of raw features between train and test

For model training, LightGBM is chosen as the baseline model for doing quick experiments on raw and engineered features. It turned out that our engineered features improved AUC in

Winning Solution of KaggleDays 2019 Competition in San Francisco

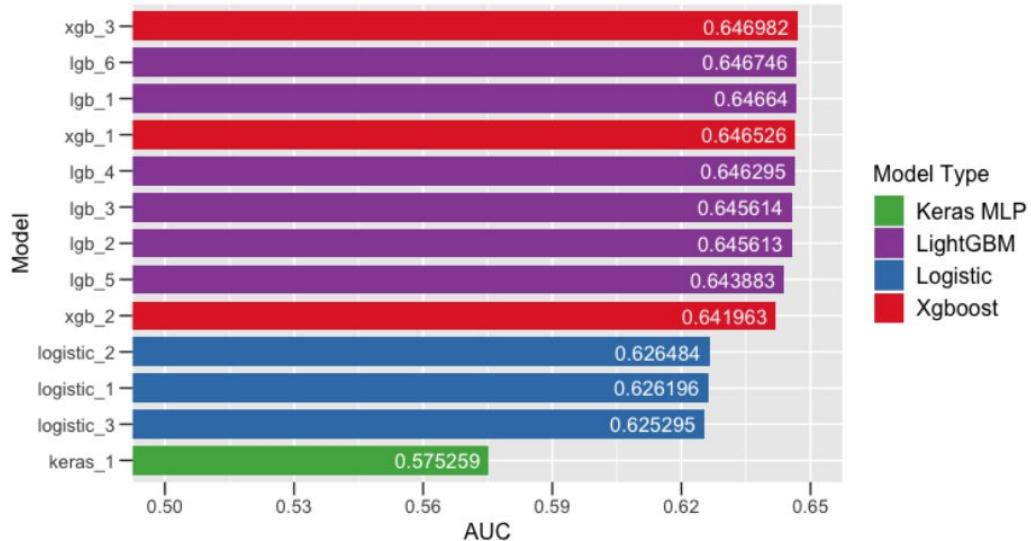
During the first two to three hours of the competition, we focused on exploratory data analysis (EDA). We first analyzed the distributions of raw features between train and test

For model training, LightGBM is chosen as the baseline model for doing quick experiments on raw and engineered features. It turned out that our engineered features improved AUC in

Based on the LightGBM baseline notebook, I created other notebooks for Xgboost, Logistic Regression, Random Forests and Extra Trees models. To save some time, I ran Random

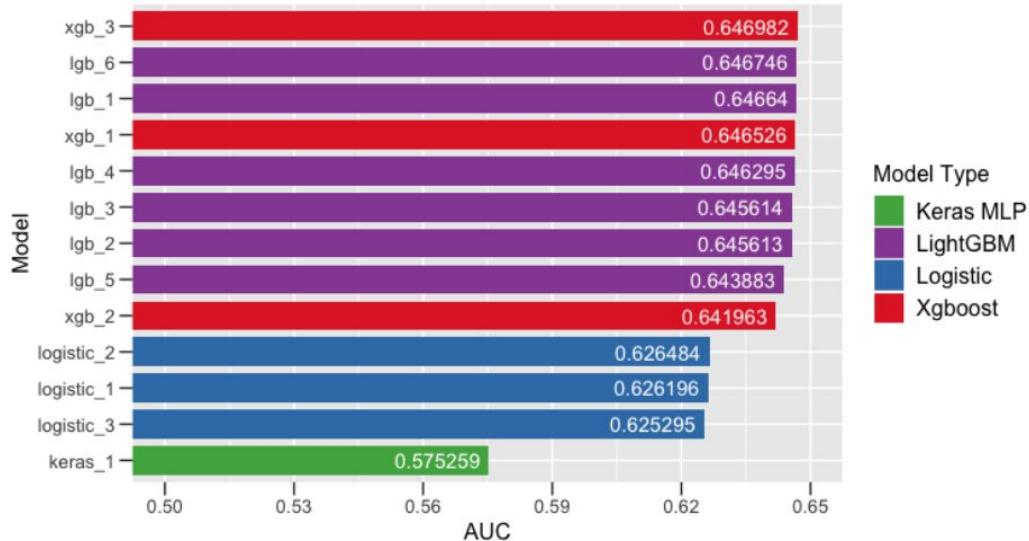
Winning Solution of KaggleDays 2019 Competition in San Francisco

During the first two to three hours of the competition, we focused on exploratory data analysis (EDA). We first analyzed the distributions of raw features between train and test



Winning Solution of KaggleDays 2019 Competition in San Francisco

During the first two to three hours of the competition, we focused on exploratory data analysis (EDA). We first analyzed the distributions of raw features between train and test



For blending and stacking, we looked for diverse models that perform relatively well, but are not highly correlated with each other. Xgboost and LightGBM had very similar

TABULAR DATA YOU HAVE



GBM USE YOU MUST

More:



Berlin Buzzwords 2019: Szilard Pafka—Better than Deep Learning: Gradient Boosting Machines (GBM)

More:



Berlin Buzzwords 2019: Szilard Pafka—Better than Deep Learning: Gradient Boosting Machines (GBM)



[szilard / talks-main](#)

Most recent/important talks I gave at conferences/meetups

In the last 5 years I gave about 50 talks at various data science and machine learning conferences and meetups. :

More:



[szilard / benchm-ml](#)



Star

1,203



[szilard / teach-data-science-UCLA-master-appl-stats](#)



[szilard / teach-ML-CEU-master-bizanalytics](#)

[szilard / GBM-intro](#)

[szilard / GBM-workshop](#)



[szilard / ML-scoring](#)

[szilard / GBM-perf](#)

[szilard / GBM-tune](#)

[szilard / GBM-multicore](#)

GitHubGist

Search...



[szilard / h2o_scoring.R](#)



[szilard / ML_with_H2O.R](#)



✉️ spafka@gmail.com

🐦 [@DataScienceLA](https://twitter.com/DataScienceLA)

linkedin linkedin.com/in/szilard

github github.com/szilard