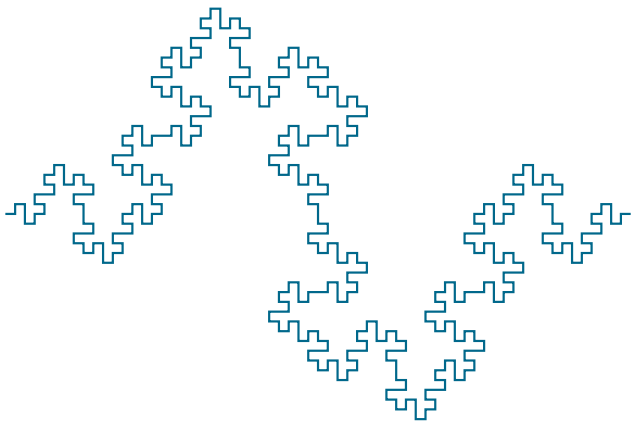
Python: out of core learning

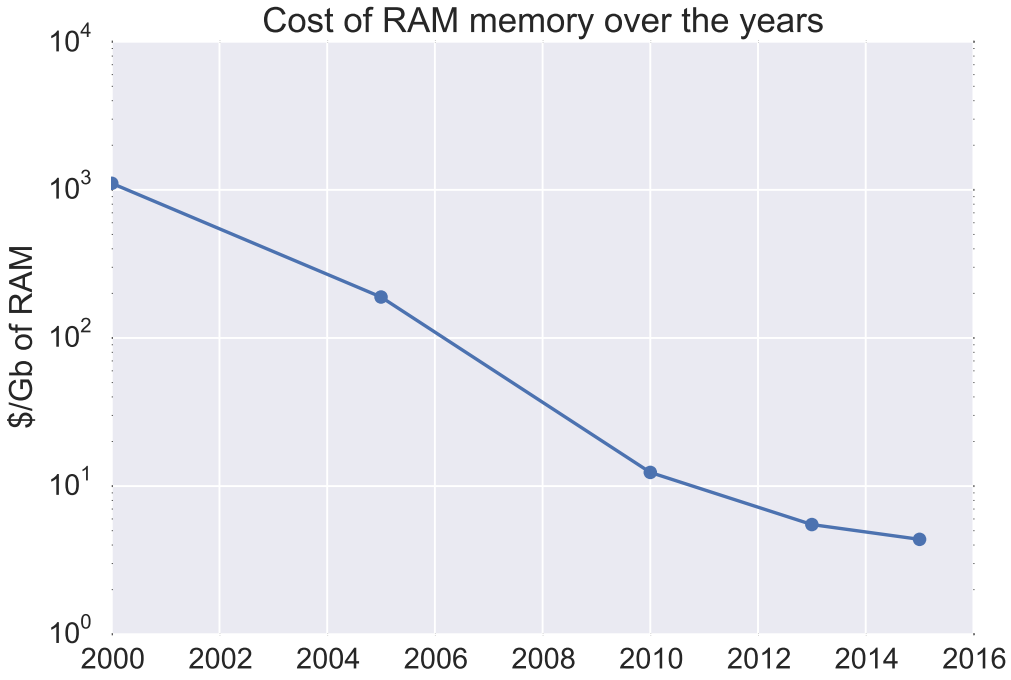
Georgios Balikas
University of Grenoble-Alpes



December 17, 2015

MOTIVATION

- Nobody ever got fired for using Hadoop on a cluster, but
- the data you apply ML on, are usually “small”, so
- it may be cheaper to add RAM!



AMA TEAM

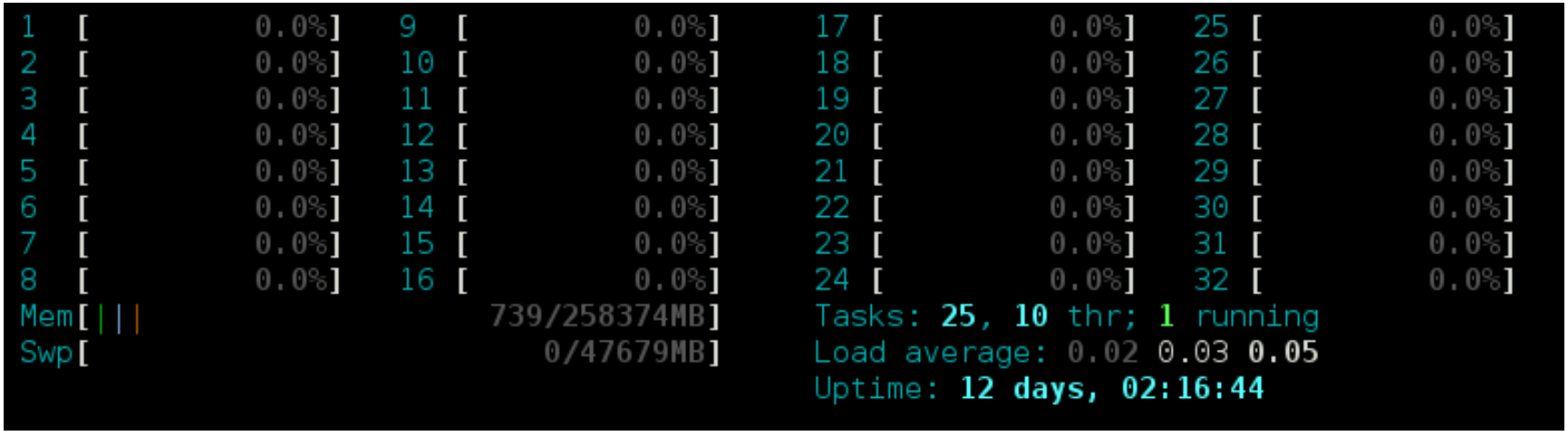


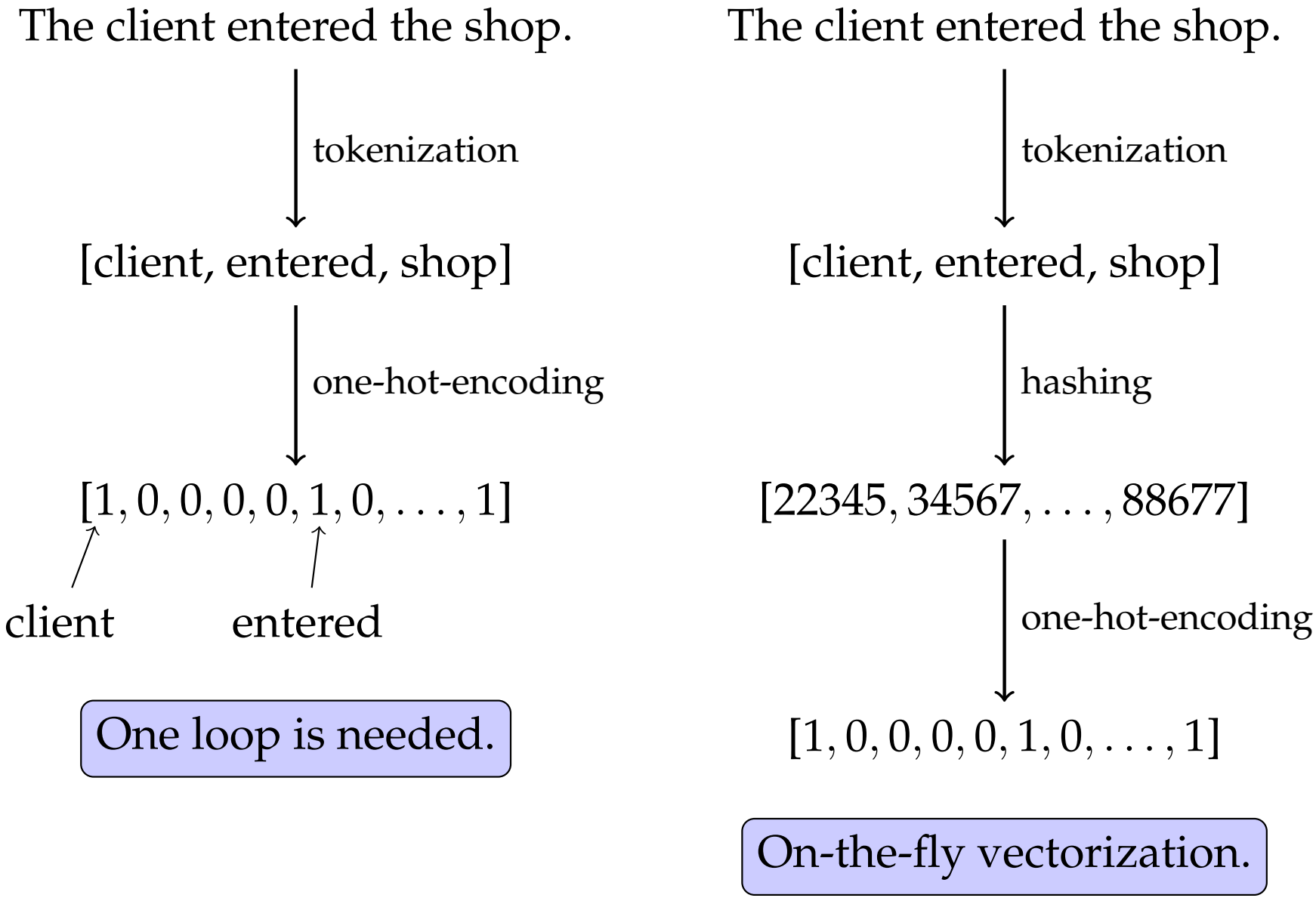
Figure : Our tiger.

Given access to significant computational resources, one needs to find ways to scale and harness the power.

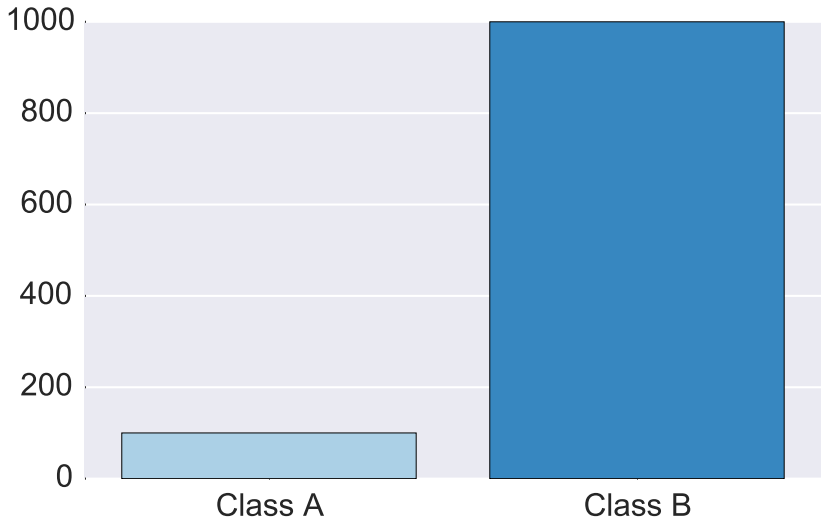
TODAY

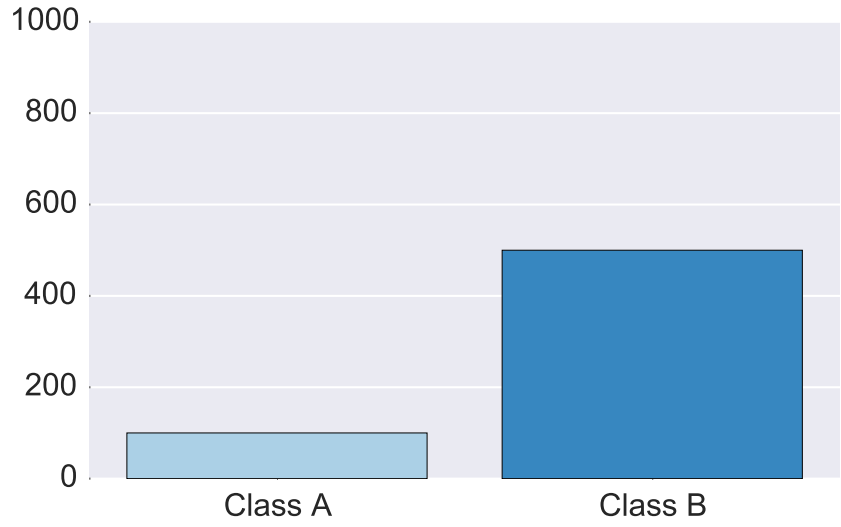
- Text pre-processing
- Sub-sampling
- Linear Classification
- Non-linear classification via kernel approximations
- Conclusion: Apart from Python, what?

BAG OF WORDS


The client entered the shop.
tokenization
[client, entered, shop]
one-hot-encoding
[1, 0, 0, 0, 0, 1, 0, . . . , 1]
client
entered
One loop is needed.
The client entered the shop.
tokenization
[client, entered, shop]
hashing
[22345, 34567, . . . , 88677]
one-hot-encoding
[1, 0, 0, 0, 0, 1, 0, . . . , 1]
On-the-fly vectorization.

SUBSAMPLING





Depending on the problem, sub-sampling:

- Reduces complexity,
- In several cases, does not result in loss of information
- Instead of hurting performance, it can actually benefit it.

LINEAR CLASSIFICATION

```
from sklearn.linear_model import SGDClassifier

sgd = SGDClassifier()

csv_iter = pd.read_csv('a_big_file.csv',\
    chunksize=1000)
for chunk in csv_iter:
    X = csv_iter[features]
    y = csv_iter['label']
    sgd.partial_fit(x,y)
```

KERNEL APPROXIMATION

```
from sklearn.kernel_approximation import RBFSampler

#Load data (X,y) here..

rbf_feature = RBFSampler(gamma=1, random_state=1)
X_features = rbf_feature.fit_transform(X)
clf = SGDClassifier()
clf.fit(X_features, y)
```

EXPERIMENTAL SETTING

- Wikipedia documents
- 500 classes
- 108,468 vocabulary size
- min docs/class = 1

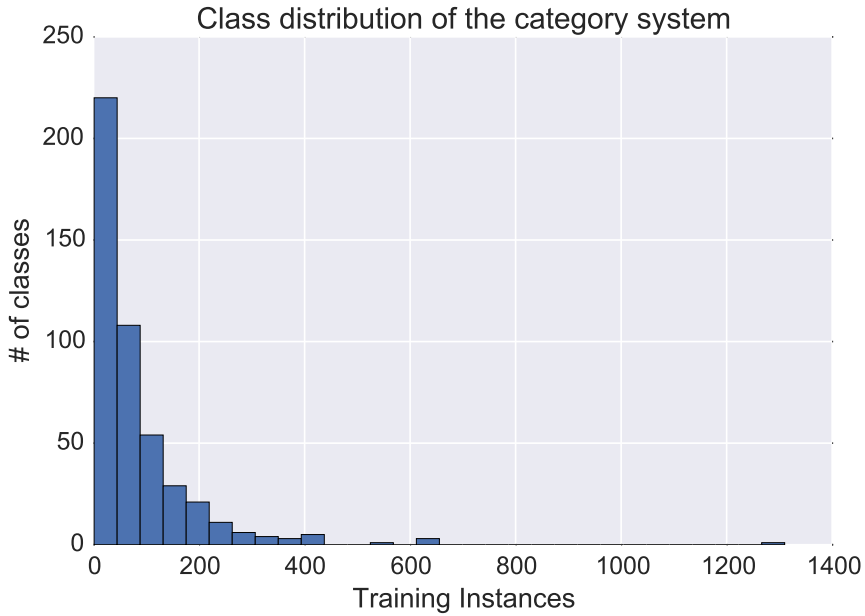


Figure : Distribution of training instances per class.

PRE-PROCESSING: EXPERIMENTS

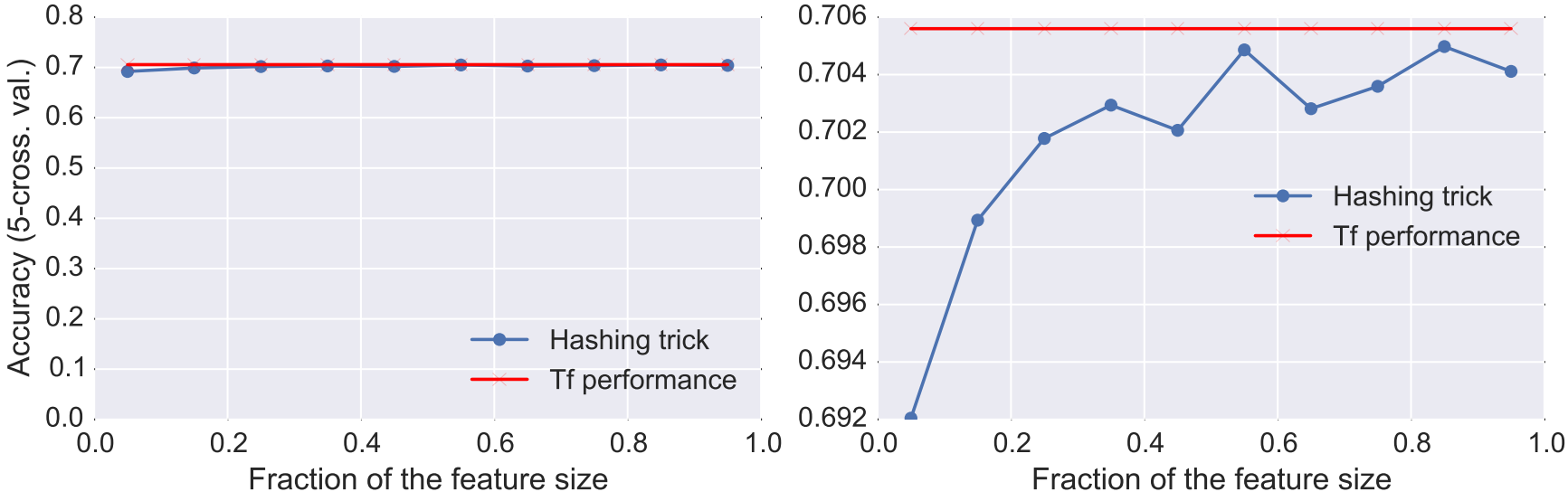
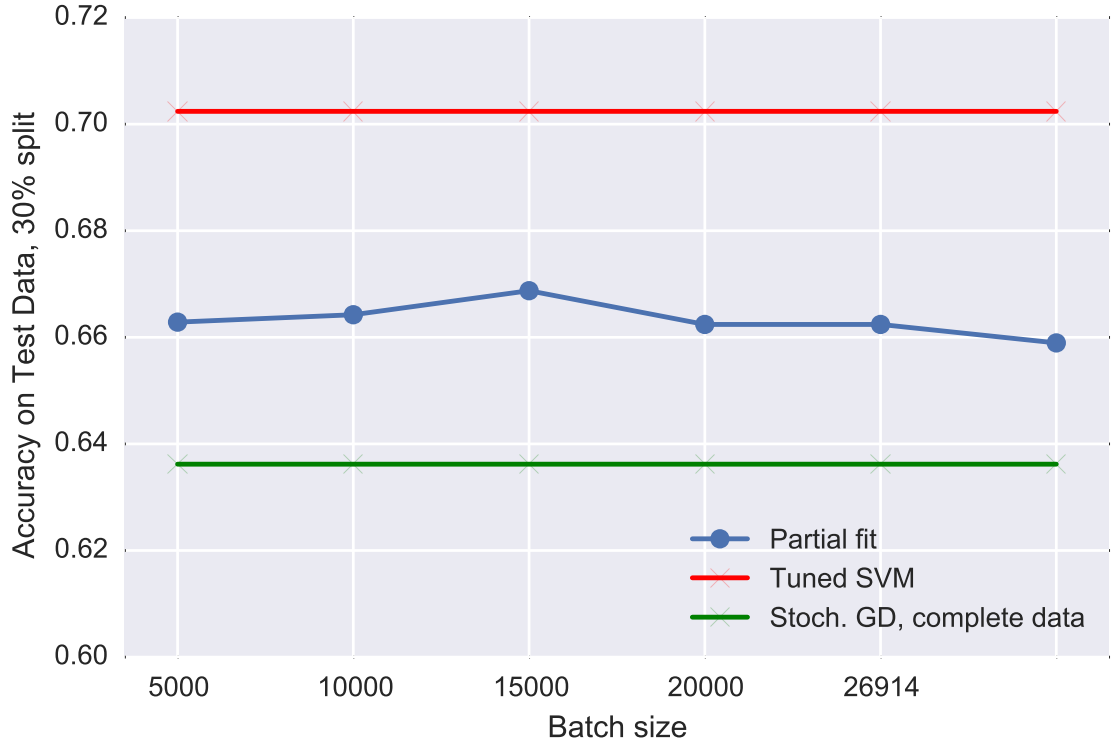


Figure : The effect of feature reduction in performance.

LEARNING: BATCH TRAINING



- Taking of-the-shelf is not advised!
- Tuning helps performance.
- Memory efficient does not imply lacking in performance.

Conclusion

I tried to provide evidence that:

- Large scale machine learning can be done out-of-the core.
- It requires a good ~~deep~~ understanding of the learning pipeline.
- It can result in significant gains in performance.

What else exists out there:

- Vowpal Wabbit
- Xgboost
- ...

QUESTIONS??

Code and Data:
`https://github.com/balikasg/OutOfCoreLearning/`