Presentation of OLPy

Online learning with Python.

Presented by: Vinny Adjibi

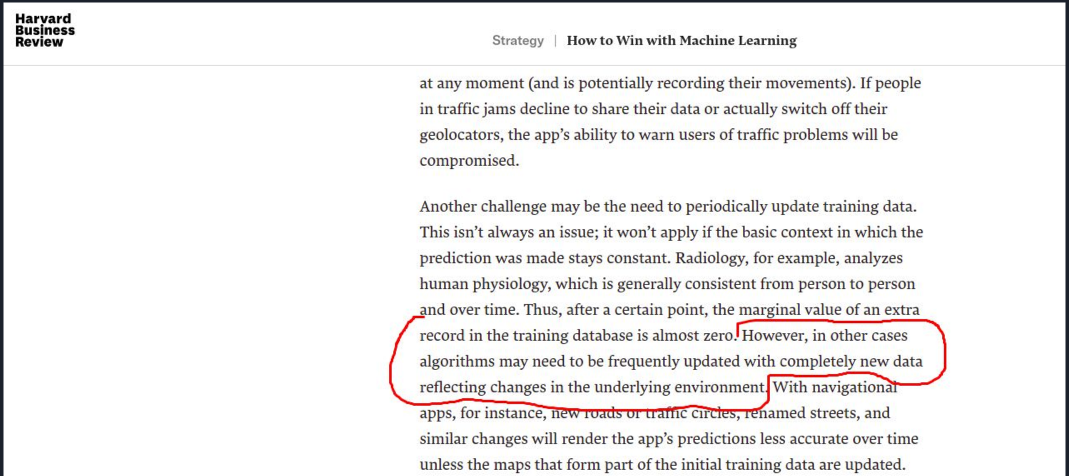
TOC

Introduction to Online Learning

State of the field

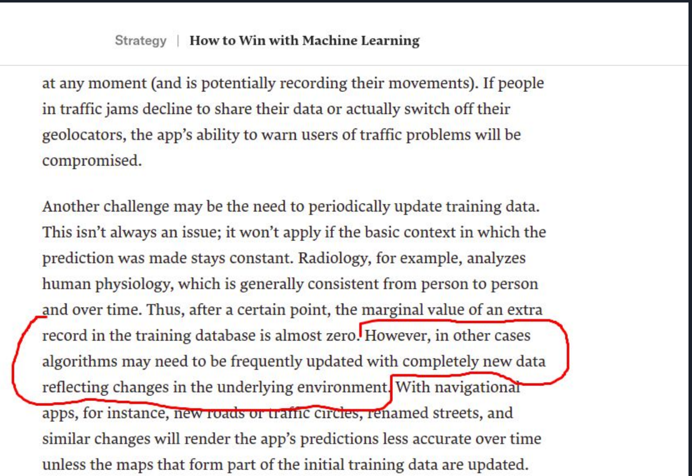
Presentation of OLPy

Conclusion



at any moment (and is potentially recording their movements). If people in traffic jams decline to share their data or actually switch off their geolocators, the app's ability to warn users of traffic problems will be compromised.

Another challenge may be the need to periodically update training data. This isn't always an issue; it won't apply if the basic context in which the prediction was made stays constant. Radiology, for example, analyzes human physiology, which is generally consistent from person to person and over time. Thus, after a certain point, the marginal value of an extra record in the training database is almost zero. However, in other cases algorithms may need to be frequently updated with completely new data reflecting changes in the underlying environment. With navigational apps, for instance, new roads or traffic circles, renamed streets, and similar changes will render the app's predictions less accurate over time unless the maps that form part of the initial training data are updated.

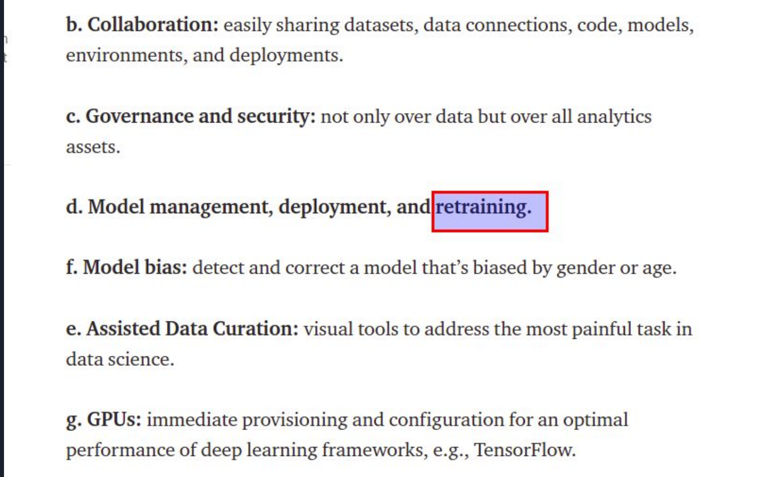


Harvard Business Review

Strategy | How to Win with Machine Learning

at any moment (and is potentially recording their movements). If people in traffic jams decline to share their data or actually switch off their geolocators, the app's ability to warn users of traffic problems will be compromised.

Another challenge may be the need to periodically update training data. This isn't always an issue; it won't apply if the basic context in which the prediction was made stays constant. Radiology, for example, analyzes human physiology, which is generally consistent from person to person and over time. Thus, after a certain point, the marginal value of an extra record in the training database is almost zero. However, in other cases algorithms may need to be frequently updated with completely new data reflecting changes in the underlying environment. With navigational apps, for instance, new roads or traffic circles, renamed streets, and similar changes will render the app's predictions less accurate over time unless the maps that form part of the initial training data are updated.



b. Collaboration: easily sharing datasets, data connections, code, models, environments, and deployments.

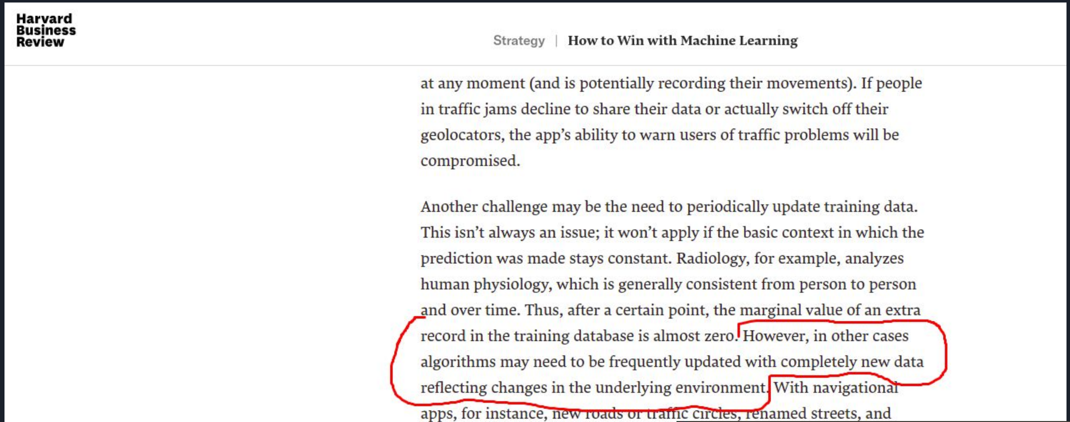
c. Governance and security: not only over data but over all analytics assets.

d. Model management, deployment, and retraining.

f. Model bias: detect and correct a model that's biased by gender or age.

e. Assisted Data Curation: visual tools to address the most painful task in data science.

g. GPUs: immediate provisioning and configuration for an optimal performance of deep learning frameworks, e.g., TensorFlow.



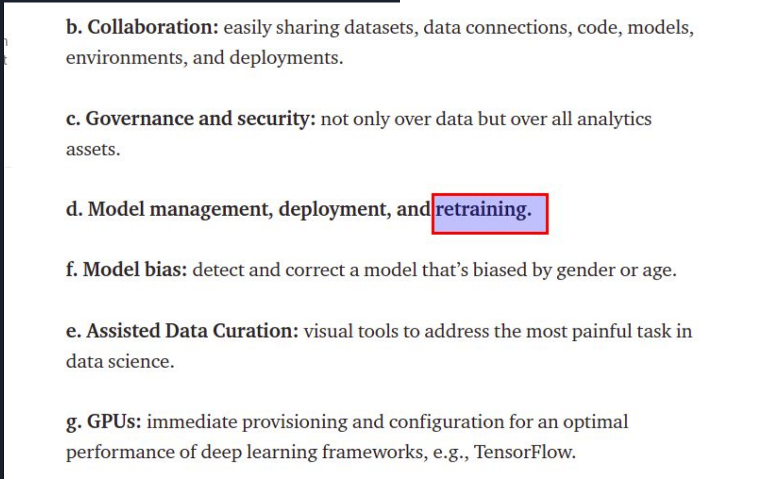
Harvard Business Review

Strategy | How to Win with Machine Learning

at any moment (and is potentially recording their movements). If people in traffic jams decline to share their data or actually switch off their geolocators, the app's ability to warn users of traffic problems will be compromised.

Another challenge may be the need to periodically update training data. This isn't always an issue; it won't apply if the basic context in which the prediction was made stays constant. Radiology, for example, analyzes human physiology, which is generally consistent from person to person and over time. Thus, after a certain point, the marginal value of an extra record in the training database is almost zero. However, in other cases algorithms may need to be frequently updated with completely new data reflecting changes in the underlying environment. With navigational apps, for instance, new roads or traffic circles, renamed streets, and similar changes will render the app's

unless the maps that form part of the



b. Collaboration: easily sharing datasets, data connections, code, models, environments, and deployments.

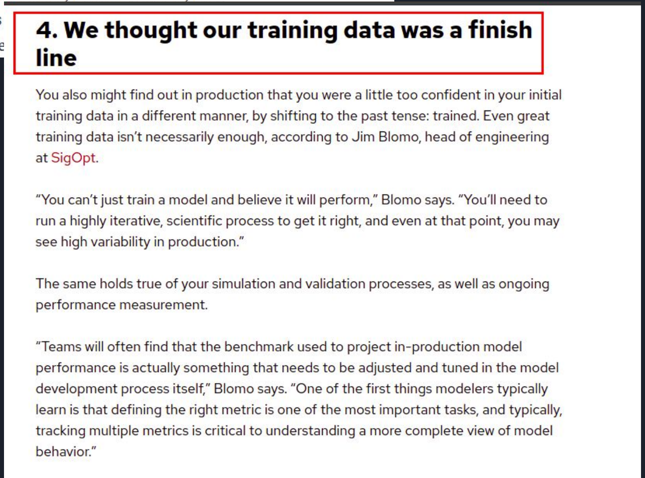
c. Governance and security: not only over data but over all analytics assets.

d. Model management, deployment, and retraining.

f. Model bias: detect and correct a model that's biased by gender or age.

e. Assisted Data Curation: visual tools to address the most painful task in data science.

g. GPUs: immediate provisioning and configuration for an optimal performance of deep learning frameworks, e.g., TensorFlow.



4. We thought our training data was a finish line

You also might find out in production that you were a little too confident in your initial training data in a different manner, by shifting to the past tense: trained. Even great training data isn't necessarily enough, according to Jim Blomo, head of engineering at SigOpt.

"You can't just train a model and believe it will perform," Blomo says. "You'll need to run a highly iterative, scientific process to get it right, and even at that point, you may see high variability in production."

The same holds true of your simulation and validation processes, as well as ongoing performance measurement.

"Teams will often find that the benchmark used to project in-production model performance is actually something that needs to be adjusted and tuned in the model development process itself," Blomo says. "One of the first things modelers typically learn is that defining the right metric is one of the most important tasks, and typically, tracking multiple metrics is critical to understanding a more complete view of model behavior."

Understanding the problems

01 Data is not (mutually) exclusive.

02 Time to have labelled data is very uncertain.

03 Long Short Term Memory??

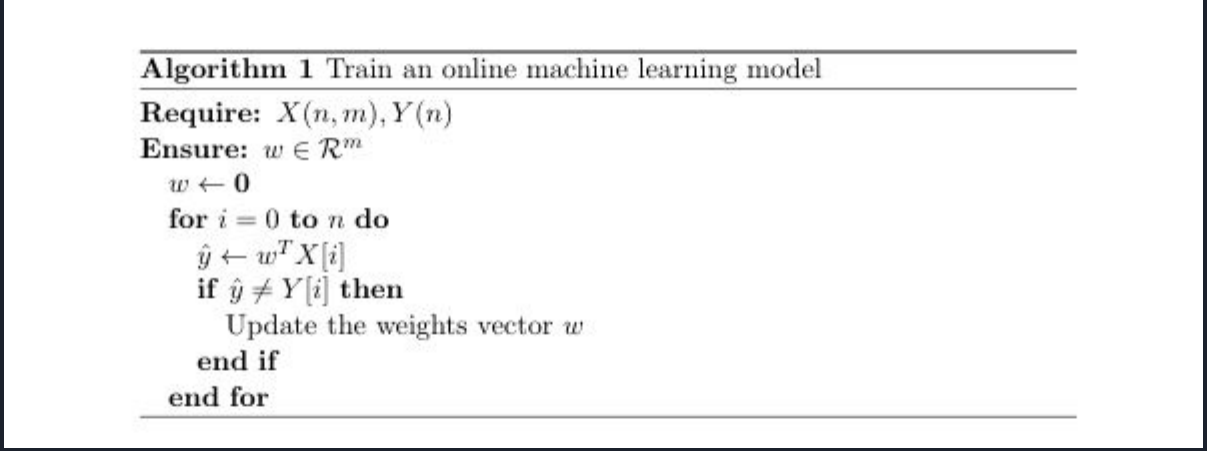
Desired capabilities

1. Low computational overhead while retraining the model.

2. The amount of data that we have should not limit how much we learn - ideally

3. The model should adapt to new trends effectively

Online learning



```
Algorithm 1 Train an online machine learning model
Require: $X(n, m), Y(n)$
Ensure: $w \in \mathcal{R}^m$
  $w \leftarrow \mathbf{0}$
  for $i = 0$ to $n$ do
    $\hat{y} \leftarrow w^T X[i]$
    if $\hat{y} \neq Y[i]$ then
      Update the weights vector $w$
    end if
  end for
```

Desired capabilities

1. **Low computational overhead while retraining the model.**
    - Online learners only deal with one data point at a time.

Desired capabilities

1. **Low computational overhead while retraining the model.**
    - Online learners only deal with one data point at a time.

2. **The amount of data that we have should not limit how much we learn - ideally**
    - We can update an online learning model with as little as one new data point.

Desired capabilities

1. **Low computational overhead while retraining the model.**
   - Online learners only deal with one data point at a time.

2. **The amount of data that we have should not limit how much we learn - ideally**
   - We can update an online learning model with as little as one new data point.

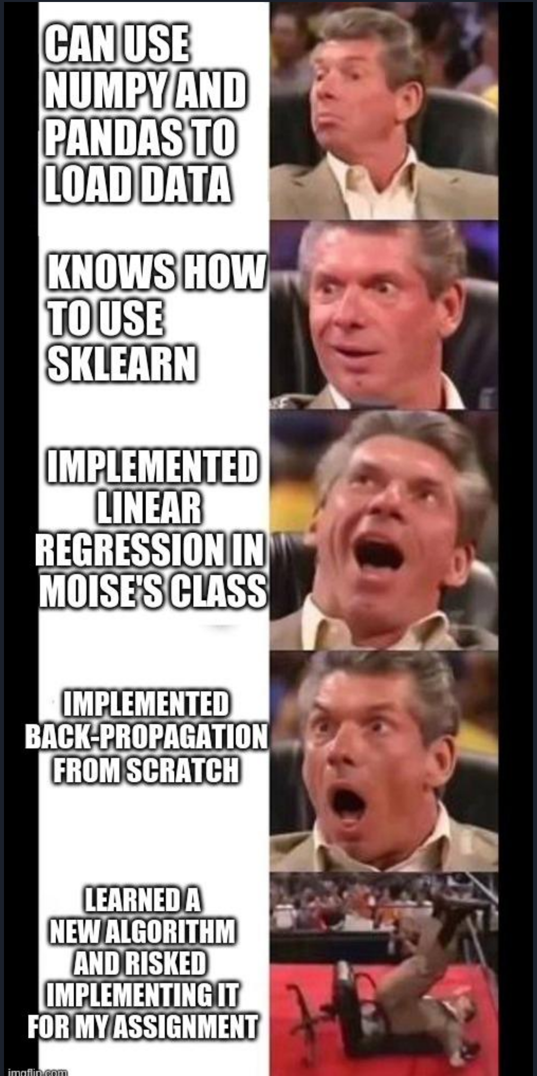
3. **The model should adapt to new trends effectively**
   - For every received data point, the model is updated every time the loss function reaches a certain threshold. Could be as simple as a 0/1 loss function or more advanced ones.

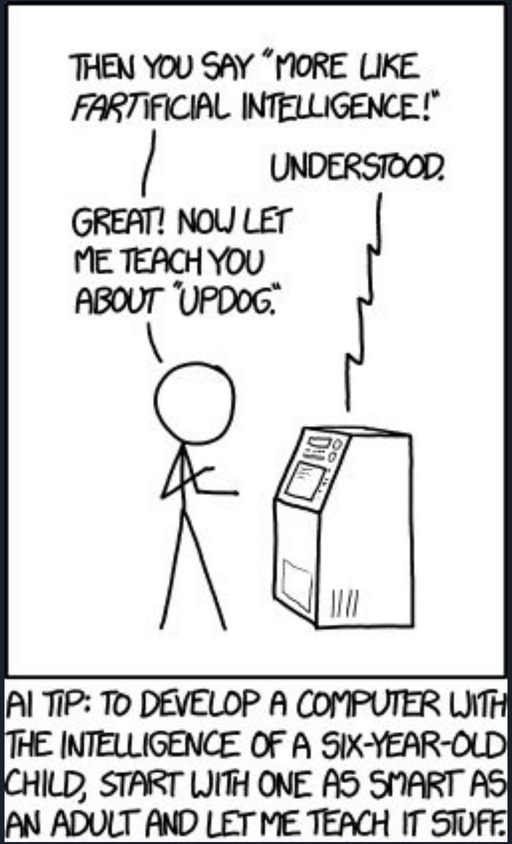
Why aren’t they used?

(solely my opinion here)

Two existing libraries

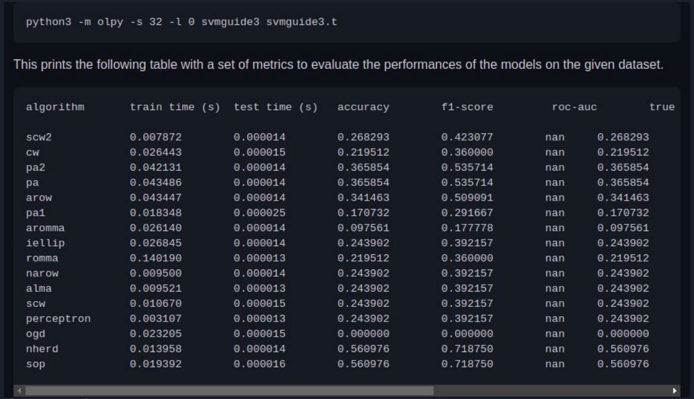
1. LIBOL - Matlab
2. SOL - CPython


CAN USE NUMPY AND PANDAS TO LOAD DATA
KNOWS HOW TO USE SKLEARN
IMPLEMENTED LINEAR REGRESSION IN MOISE'S CLASS
IMPLEMENTED BACK-PROPAGATION FROM SCRATCH
LEARNED A NEW ALGORITHM AND RISKED IMPLEMENTING IT FOR MY ASSIGNMENT
imgflip.com


THEN YOU SAY "MORE LIKE FARTIFICIAL INTELLIGENCE!"
UNDERSTOOD.
GREAT! NOW LET ME TEACH YOU ABOUT "UPDOG."
AI TIP: TO DEVELOP A COMPUTER WITH THE INTELLIGENCE OF A SIX-YEAR-OLD CHILD, START WITH ONE AS SMART AS AN ADULT AND LET ME TEACH IT STUFF.

https://xkcd.com/1696/

About OLPy



```
python3 -m olpy -s 32 -l 0 svmguide3 svmguide3.t
```

This prints the following table with a set of metrics to evaluate the performances of the models on the given dataset.

```
algorithm        train time (s)  test time (s)   accuracy        f1-score         roc-auc        true

scw2             0.007872        0.000014        0.268293        0.423077        nan     0.268293
cw               0.026443        0.000015        0.219512        0.360000        nan     0.219512
pa2              0.042131        0.000014        0.365854        0.535714        nan     0.365854
pa               0.043486        0.000014        0.365854        0.535714        nan     0.365854
arow             0.043447        0.000014        0.341463        0.509091        nan     0.341463
pa1              0.018348        0.000025        0.170732        0.291667        nan     0.170732
aromma           0.026140        0.000014        0.097561        0.177778        nan     0.097561
iellip           0.026845        0.000014        0.243902        0.392157        nan     0.243902
romma            0.140190        0.000013        0.219512        0.360000        nan     0.219512
narow            0.009500        0.000014        0.243902        0.392157        nan     0.243902
alma             0.009521        0.000013        0.243902        0.392157        nan     0.243902
scw              0.010670        0.000015        0.243902        0.392157        nan     0.243902
perceptron       0.003107        0.000013        0.243902        0.392157        nan     0.243902
ogd              0.023205        0.000015        0.000000        0.000000        nan     0.000000
nherd            0.013958        0.000014        0.560976        0.718750        nan     0.560976
sop              0.019392        0.000016        0.560976        0.718750        nan     0.560976
```

- Currently usable for binary classification.
- Implements 16 different online learning models including the Perceptron.
- Uses an API similar to that of scikit-learn
- Provides a module that you can run your tests with
- Comes bundled with three datasets (svmguide1, svmguide3, a1a)

Links

https://github.com/boladjivinny/olpy

https://olpy.readthedocs.io/

https://pypi.org/project/olpy/

Importance of the scikit-learn interface

- Habit
- Possibility to “ensemble” with existing models
- GridSearchCV
- … and much more

Recap

- Online learning models can address some real world problems.
- Less known because not many implementations exists
- OLPy addresses the shortage of implementation of the models
- Built in Python and easy to use for anyone familiar with scikit-learn

Plus

- Training on a subset of data from the current point
- Classes weights for imbalanced data.

Demo

What’s next?

- Use the models for your projects, hackathons, etc.
- Maybe turn into an evangelist
- Contribute to it in any possible way: https://github.com/boladjivinny/olpy

Thank you.
Questions/Remarks?