• Automatic Machine Learning (AutoML)
• Machine Learning Benchmarking
• Benchmarking in AutoML development
• Benchmark of OSS AutoML Systems

Slides ➡️ https://tinyurl.com/user19-amlbench
Automatic Machine Learning (AutoML)
Goals & Features of AutoML

- 🏆 Train the best model in the least amount of time.
- 📈 Reduce the human effort & expertise required in machine learning.
- 📈 Improve the performance of machine learning models.
- ✅ Increase reproducibility & establish a baseline for scientific research or applications.
Aspects of Automatic Machine Learning

Data Prep

Model Generation

Ensembles
Aspects of Automatic Machine Learning

Data Preprocessing
- Imputation, one-hot encoding, standardization
- Feature selection and/or feature extraction (e.g. PCA)
- Count/Label/Target encoding of categorical features

Model Generation
- Cartesian grid search or random grid search
- Bayesian Hyperparameter Optimization
- Individual models can be tuned using a validation set

Ensembles
- Ensembles often out-perform individual models
- Stacking / Super Learning (Wolpert, Breiman)
- Ensemble Selection (Caruana)
The different flavors of AutoML

By: Erin LeDell

In recent years, the demand for machine learning experts has outpaced the supply, despite the surge of people entering the field. To address this gap, there have been big strides in the development of user-friendly machine learning software (e.g. H2O, scikit-learn, kaggle). Although these tools have made it easy to train and evaluate machine learning models, there is still a good amount of data science knowledge that’s required in order to create the highest-quality model given your dataset. Writing the code to perform a hyperparameter search over many different types of algorithms can also be time consuming and repetitive work.

What is AutoML?

https://tinyurl.com/flavors-of-automl
Machine Learning Benchmarking
ML Benchmarking

• 📊 Compare model & runtime performance of machine learning tools
• 🔍 Provide accurate information for users to discriminate between tools
• 🔄 Best to run on fixed & publicly available hardware such as Amazon EC2
• 🟢 Best done by a third-party and not an author
ML Benchmarking Mistakes

• Not enough datasets, not enough diversity among the datasets and datasets are too small ❌

• Tools benchmarked incorrectly or unfairly:
  • Package authors are experts at using their own tool but make mistakes using others ❌
  • Inappropriate metrics used ❌
  • Tuning some algorithms more than others ❌
  • Insufficient memory or CPUs ❌
  • Over-generalization of results ❌
Benchmarking for AutoML development
Why is benchmarking so important for AutoML development?

- There is no “reference algorithm” in AutoML so we are creating new methods from scratch.
- It’s easy to overfit your tool to familiar datasets.
- Every time you make a change to the algorithm, you should justify the change via benchmarks.

Changes made to the H2O AutoML algorithm and the effect on performance:

- 3.20.0.10 – Baseline
- 3.22.0.1 – Add XGBoost
- 3.22.0.3 – Modify validation strategy
AutoML Benchmark
Collaboration between AutoML researchers and OpenML.org to develop a system for high quality benchmarks of the popular open source AutoML systems.

https://github.com/openml/automlbenchmark
openml.org

- Platform for reproducible ML experiments
- Unique IDs for datasets & ML tasks
- OpenML data is used in many ML benchmarks

https://www.openml.org/d/31
OpenML tasks are uniquely defined by dataset & response column, along with evaluation method (e.g. 10-fold CV).
• 🗄 Defined a diverse collection of datasets
• 🐳 Open source Dockerized framework for executing benchmarks locally or on Amazon EC2
• ✨ Extensible architecture (easy to add new tools)
• 📊 Results available on the web
• ⏪ Can re-run benchmarks on new tool versions & will expand to more tools, datasets & use cases
What qualifies as "AutoML" software?

- 👉 Point to a dataset & response column (no other required hyperparameters).
- 🏆 Returns the best model and optionally a list of all models trained.
- ⏰ Time or resource budget.
Example: H2O AutoML in R

```r
library(h2o)
h2o.init()

train <- h2o.importFile("train.csv")

aml <- h2o.automl(y = "response_colname",
                  training_frame = train,
                  max_runtime_secs = 600)

lb <- aml@leaderboard
```
**AutoML Software**

- **AutoWEKA**
- **auto-sklearn**
- **TPOT**
- **H2O AutoML**
- **Auto-Keras**
- **Hyperopt-sklearn**

---

<table>
<thead>
<tr>
<th>Tool</th>
<th>Back-end</th>
<th>Optimization</th>
<th>Meta-learning</th>
<th>Post-processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-WEKA</td>
<td>WEKA</td>
<td>Bayesian</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>auto-sklearn</td>
<td>scikit-learn</td>
<td>Bayesian</td>
<td>-</td>
<td>ensemble selection</td>
</tr>
<tr>
<td>TPOT</td>
<td>scikit-learn</td>
<td>Genetic Programming</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H2O AutoML</td>
<td>H2O</td>
<td>Random Search</td>
<td>-</td>
<td>stacked ensembles</td>
</tr>
</tbody>
</table>

Table 1: Simplified comparison of a selection of AutoML tools.
AutoML Benchmark Results

Scores on 4h binary classification problems

AUC

https://openml.github.io/automlbenchmark/results.html
In recent years, an active field of research has developed around automated machine learning (AutoML). Unfortunately, comparing different AutoML systems is hard and often done incorrectly. We introduce an open, ongoing, and extensible benchmark framework which follows best practices and avoids common mistakes. The framework is open-source, uses public datasets and has a website with up-to-date results. We use the framework to conduct a thorough comparison of 4 AutoML systems across 39 datasets and analyze the results.
Thank you!

@ledell on Github, Twitter
erin@h2o.ai