An evaluation and comparison of AutoML solutions: *Azure AutoML* and *EvalML*

Tim Pachmann
Hochschule Darmstadt - University of Applied Sciences,
Haardtring 100, 64295 Darmstadt, Germany
tim.pachmann@stud.h-da.de

Abstract. Automated machine learning (AutoML) supports the development of machine learning systems by automating tasks that traditionally require manual, time-consuming labor and extensive expertise in the field. This paper presents two AutoML systems, one open-source (*EvalML*) and one commercial solution (*Azure AutoML*), and evaluates their range of functionality as well as practical usability. The results show that commercial tools provide better usability for beginners, while open-source alternatives tend to stand out with greater configurability.

Keywords: Machine Learning (ML) · Deep Learning · Automated Machine Learning (AutoML) · Hyperparameter Optimization (HPO).

1 Introduction

A data-driven model building and decision making can contribute to higher degrees of automation and more informed decisions throughout many industries. However, designing and tuning those models is a labor and time extensive task which requires extensive expertise, often limiting the speed of development and hindering its wide application.

The field of automated machine learning (AutoML) aims to solve this problem, by building methods to automate the entailed steps with as little human intervention as possible. This not only makes the topic of Machine Learning available to non-experts of the domain, but it also improves the general efficiency by freeing up time that is normally needed for performing the manual steps.

While the topic of AutoML comprises various problems, this paper focuses on systems that address the "Combined Algorithm Selection and Hyperparameter optimization" (CASH) problem by picking an algorithm from a list of options and then tune it to give the highest validation performance amongst all the combinations.

This paper is focusing on two AutoML solutions, mainly evaluating and comparing their practical usability by putting them to a defined set of tests and comparing the resulting benchmarks. Furthermore, additional meta information like the maturity of the systems, the licensing and their primary target user group should be considered.
The first system evaluated in this paper is Azure AutoML [4], a commercial web service by Microsoft. The cloud-based solution, which is based on Microsoft’s Azure Machine Learning library, may be used to perform various machine learning tasks, including classification, regression, forecasting as well as computer vision.

The second system is EvalML [1], an open-source AutoML software library written in Python, which builds, optimizes and evaluates machine learning pipelines using domain-specific objective functions. It may also be used for classification, regression and time series forecasting.

This paper is structured as follows: In section 2, related work is presented. After that, in section 3, the methodology and the framework for the comparison are introduced. After clarifying the procedure, section 4 focuses on the results of the evaluation, which are discussed in the following section 5. Finally, section 6 concludes the paper by summarizing the outcomes.

2 Related Work

This paper is used as one of the foundations for a broader survey [2], which presents a comprehensive survey of 20 state-of-the-art AutoML solutions. The procedure and evaluation criteria introduced in the article are largely inherited to this paper to allow for an unified comparison of all examined solutions.

3 Methodology

As defined in the survey [2], the following criteria are used for the comparison:

1. Solution Type
   Licensing model (e.g. commercial or open-source) and deployment model (e.g. software library, local application or cloud/web service) of the AutoML solution.

2. Maturity
   Information about the release status (e.g. pre-release or released) and the development status (e.g. actively developed) of each AutoML solution.

3. Target User Group
   The intended primary user group for the AutoML solution (e.g. domain expert or data / computer scientist).

4. ML Tasks
   List of machine learning tasks which the AutoML solution can process (e.g. classification, regression or time-series forecasting).

5. ML Library / Approaches
   Libraries and approaches that are used by the AutoML solution (e.g. scikit-learn, XGBoost, LightGBM, ...).
6. **Reporting**  
Scope / level of detail of the reporting functionality (e.g. basic or detailed).

7. **Result Type**  
Returned model type (e.g. single/multiple models or ensemble pipeline) and the export type (e.g. model instance / file or execution scripts) generated by the AutoML solutions.

In regards to the benchmarks, two tasks were performed with each AutoML solution:

1. **Binary Classification Task**  
The OpenML *PhishingWebsites* dataset was used for this task, with the goal of predicting whether a given website is malicious for an user.  
As the evaluation metric the $F1$ score was selected.

2. **Regression Task**  
The OpenML *Colleges* dataset was used for this task, with the goal of predicting the likelihood of being accepted by a given university.  
As the evaluation metric the $RSME$ (Root Mean Square Error) score was selected.

As a technical framework for the benchmarks, each experiment should run on a machine with comparable resources (4 CPU cores, 16 GB RAM) and be configured to run within a time limit of 10 minutes.

For each benchmark task, three independent experiments were performed. To compare the results between the AutoML solution, an overall average score per task was calculated for each task.

4 Results

4.1 Azure AutoML

**Solution Type:** Commercial, cloud-based web service.

**Maturity:** As the product is neither listed as being in a public alpha/preview phase nor as discontinued, it is assumed to be fully released. Because the commercial solution is proprietary and therefore closed-source, a definitive development plan and contribution frequency is not publicly available. However, according to the public update announcements[^3] the tool is being actively maintained.

**Target User Group:** The tool is mainly targeted to be used by domain experts, but it also offers a SDK for data/computer scientists.

---

[^1]: OpenML "PhishingWebsites": [https://www.openml.org/d/4534](https://www.openml.org/d/4534)
[^2]: OpenML "Colleges": [https://www.openml.org/d/42727](https://www.openml.org/d/42727)
**ML Tasks**: The solution provides functionalities to perform classification, regression, time-series forecasting as well as computer vision tasks.

**ML Library / Approaches**: The solution is based on the Azure Machine Learning library, which internally uses the following estimators:

- LogisticRegression
- SGD
- MultinomialNaiveBayes
- BernoulliNaiveBayes
- SVM
- KNN
- DecisionTree
- RandomForest
- ExtremeRandomTrees
- LightGBM
- GradientBoosting
- XGBoostClassifier
- LinearSVM
- AveragedPerceptronClassifier

**Reporting**: The tool provides a user-friendly web frontend, which includes a comprehensive wizard for configuring and starting new automated runs (see figure 1) as well as detailed reporting of the current run status and its results (see figure 2):
Fig. 2. The run status summary of Azure AutoML.

**Result Type:** The best model determined by the tool may be exported, including the model file (either as a pickled python object or in a ONNX format) as well as a generated script file to serve an inference endpoint. Also, the user is presented with quick options to serve the model via Azure cloud resources.

**Benchmark Results:** Regarding the technical framework for the benchmarks, it has to be denoted that Azure only provides configuration options for runtime limitations with a minimum of 30 minutes.

All experiments were run on a virtual machine of the type *Standard_DS3_v2*.

<table>
<thead>
<tr>
<th>Run</th>
<th><strong>Classification</strong> (Score: $F_1$)</th>
<th><strong>Regression</strong> (Score: RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>0.97246 (in 27m 20.21s)</td>
<td>0.13371 (in 31m 37.93s)</td>
</tr>
<tr>
<td>#2</td>
<td>0.97140 (in 32m 40.43s)</td>
<td>0.13334 (in 32m 1.971s)</td>
</tr>
<tr>
<td>#3</td>
<td>0.97207 (in 26m 51.71s)</td>
<td>0.13132 (in 31m 22.61s)</td>
</tr>
<tr>
<td>$\approx$</td>
<td><strong>0.97198 (in ~27m 39.45s)</strong></td>
<td><strong>0.13279 (in ~31m 40.84s)</strong></td>
</tr>
</tbody>
</table>

*Azure Virtual Machine Type "Standard_DS3_v2":*
[https://pcr.cloud-mercato.com/providers/azure/flavors/standard_ds3_v2](https://pcr.cloud-mercato.com/providers/azure/flavors/standard_ds3_v2)
4.2 EvalML

Solution Type: Open-source software library.

Maturity: The library, which was created in 2019, is currently in a pre-release state (v0.14.0 as of Jan. 10, 2022) and is still actively and continuously being worked on - a new release is published about every two weeks. As the source code and project repository is open source, open issues and ongoing changes can be tracked publicly.

Target User Group: The project is clearly focused towards data/computer scientists and needs at least some programming experience as it is a software library written in Python. However, it still may be suitable for domain experts, since essentially only five lines of code are required to start the AutoML search pipeline.

ML Tasks: The library provides functionalities to perform classification (binary / multiclass), regression as well as time-series forecasting.

ML Library / Approaches: The library includes the following estimators:

- CatBoost (Classifier / Regressor)
- Decision Tree (Classifier / Regressor)
- Elastic Net (Classifier / Regressor)
- Extra Trees (Classifier / Regressor)
- K-Nearest Neighbors Classifier
- LightGBM (Classifier / Regressor)
- Logistic Regression Classifier
- Random Forest (Classifier / Regressor)
- Support Vector Machine (Classifier / Regressor)
- Vowpal Wabbit Binary (Classifier / Regressor)
- Vowpal Wabbit Multiclass (Classifier / Regressor)
- XGBoost (Classifier / Regressor)
- Linear Regressor
- Prophet Regressor
- ARIMA Model Regressor

Reporting: The library features a very verbose status reporting (may be configured), which includes information about the current pipeline status (see figure) as well as a detailed ranking result after completion:

---

5 EvalML Documentation, Usage Example: [https://github.com/alteryx/evalml#run-automl](https://github.com/alteryx/evalml#run-automl)
Result Type: The library returns a pre-trained instance of the best pipeline, which may also be exported as a pickled model file. Aside from the model, the library generates a detailed result report and ranking table, which is accessible via pipeline attributes.

Benchmark Results: The full documentation for the conducted experiments can be found in the project repository. The run time was limited to 10 minutes and the experiment was run on a virtual machine of the type m5dn.xlarge.

Table 2. Benchmark Results for EvalML

<table>
<thead>
<tr>
<th>Run</th>
<th>Classification (Score: F1)</th>
<th>Regression (Score: RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>0.90625 (in 10m 1s)</td>
<td>0.21902 (in 10m 1s)</td>
</tr>
<tr>
<td>#2</td>
<td>0.90625 (in 10m 0s)</td>
<td>0.21902 (in 10m 0s)</td>
</tr>
<tr>
<td>#3</td>
<td>0.90625 (in 10m 0s)</td>
<td>0.21902 (in 10m 1s)</td>
</tr>
<tr>
<td>≈</td>
<td>0.90625 (in ~10m 0s)</td>
<td>0.21902 (in ~10m 1s)</td>
</tr>
</tbody>
</table>

8. GitHub repository for the AutoML experiments: [https://github.com/hochschule-darmstadt/AutoML_Hauptseminar](https://github.com/hochschule-darmstadt/AutoML_Hauptseminar)

9. AWS EC2 Instance Type "m5dn.xlarge": [https://aws.amazon.com/de/ec2/instance-types/m5/](https://aws.amazon.com/de/ec2/instance-types/m5/)
5 Discussion

Both evaluated AutoML solutions succeeded at the task of building, optimizing and evaluating various machine learning pipelines. While Azure AutoML yielded considerably better results, it has to be noted that the runtime limit was set to about three times more than the one for EvalML due to provider constraints (minimum of 30 minutes).

In many regards, EvalML was much more flexible to adjust to custom needs, providing vast configuration options and possibilities for further customization (e.g. by defining custom objective functions). Azure AutoML did not have as many options, which on the one hand constraints the ways of usage, but on the other hand this limitation allows for a very comprehensible and user-friendly UI.

This underlines the fact that Azure AutoML, as a commercial cloud solution, is more suitable for novices at machine learning and provides an easier "off-the-shelf" solution for domain experts. The open-source alternative EvalML provides extensive functionalities at the cost of usability. However, the basic usage of the library is fairly simple and well documented, which also allows the usage by rather unexperienced data/computer scientists.

6 Conclusion

AutoML is still an active field of study, the solutions being developed and offered are equally in a state of progression. More and more commercial and open-source tools are emerging and existing projects are continuously being enhanced.

While this work covers only a small fraction of the solutions available, the presented results indicate that commercial solutions tend to target domain experts and novices in the field and provide better usability, while open-source alternatives rather address data/computer scientists and enable a higher degree of customization and tunability.

References

1. Alteryx Development: EvalML, [https://github.com/alteryx/evalml](https://github.com/alteryx/evalml), GitHub repository (last accessed January 11th, 2022)