|  |  |
| --- | --- |
| Panagiotis Papaemmanouil  **Course**: Machine Learning,  winter 2019-2020  **Professor**: I. Vlahavas |  |

An overview of “***Automated Machine Learning”***

Contents

[1 Introduction 2](#_Toc31484107)

[2 Automated Machine Learning 3](#_Toc31484108)

[2.1 What is Auto ML? 3](#_Toc31484109)

[2.2 Automated machine learning motivation. Why AutoML? 3](#_Toc31484110)

[3 Applications of Automated Machine Learning in every step of a Data Science / Machine Learning project 4](#_Toc31484111)

[3.1 Evolution of Automated Machine Learning 4](#_Toc31484112)

[3.2 Traditional ML pipeline vs Auto ML pipeline 4](#_Toc31484113)

[4 Automatically Network Architecture Search (NAS) 7](#_Toc31484114)

[5 Advantages and Risks of Automated Machine Learning 7](#_Toc31484115)

[6 Will Auto ml replace data scientists? 8](#_Toc31484116)

[7 Conclusions 9](#_Toc31484117)

[8 Auto ML tools 9](#_Toc31484118)

[8.1 Commercial Automated Machine Learning tools 9](#_Toc31484119)

[8.2 Open-source Automated Machine learning tools 9](#_Toc31484120)

[9 Material and Bibliography 10](#_Toc31484121)

# Introduction

The main purpose of this report is to explore the field of Automated Machine learning (or Auto ML).

Before I define the term “Auto ML” I need to first define “Machine Learning”.

A typical definition says that "Machine Learning is a field of study that gives computers the ability to learn without being explicitly programmed". In other words, Machine learning helps a computer to learn based on past data and use this knowledge to make predictions and create value from new unseen data. Even though computers can learn without being explicitly programmed, a lot of programming skills are required to develop an end-to-end machine learning system from scratch. Here come the Auto ML tools which promise to help data scientists in every step of their job, from automated data discovery, composition, and preparation to rapid automated model construction, deployment, and maintenance.

Auto ML tools aim to help data scientists and machine learning practitioners to ***accelerate*** and ***optimize*** the creation of machine learning and data science workflows.

Practitioners typically perform several tedious and time-consuming steps to derive insight from raw data. The process usually starts with data ingestion, cleansing, transformation (e.g. outlier removal, missing value imputation), then model building, and finally a presentation of predictions that align with the end-users objectives and preferences. It is a long and complex process requiring substantial time, skill and effort especially because of the **combinatorial explosion** in choices of algorithms and platforms, their parameters, and their compositions. Auto ML tools automate various steps in this process resulting in accelerated time-to-delivery of data products and machine learning insights, expand the reach of data science to non-experts, and offer a more systematic exploration of the available options.

Automated Machine Learning (AutoML) has been described as a "***quiet revolution in AI***" that is poised to dramatically change the data science landscape by using AI to automate many of the time-consuming aspects of applying Machine Learning to real-world problems. Academic researchers, startups, and tech giants alike have begun developing AutoML methods and tools ranging from simple open-source prototypes to industry-scale software products.

# Automated Machine Learning

## What is Auto ML?

“Automated Machine learning refers to the automated process of algorithm selection, hyperparameter tuning, iterative modeling, and model assessment. It is NOT automated data science, nor automated development of artificial intelligence. It is, however, transforming model building”, as ***Data Robot*** claims on its website.

A more general definition by Microsoft Azure is the following; "Automated Machine Learning automatically searches an enormous space of possible machine learning pipelines intelligently and efficiently. It’s essentially a recommender system for machine learning pipelines. Similar to how streaming services recommend movies for users, automated ML recommends machine learning pipelines for datasets."

From the above definitions, we can conclude that the key purpose of Automate machine learning tools ***it's to automate some of the processes of the machine learning workflow.***

## Automated machine learning motivation. Why AutoML?

We are in the era of Artificial Intelligence. The evolution of Machine Learning algorithms leads to a growing demand for AΙ experts to build and deploy AI products for companies. This need doesn't apply only to tech companies, but essentially for all type of companies. A good rule of thumb to identify problems that maybe can be solved using AI is to follow the data; "Wherever there are data, there is a need for AI solutions".

From a **business perspective**, there are many challenges in the deployment of AI, in real-world problems. In an ideal world, the person who builds the AI for the business has domain knowledge, works for the company and has advanced computer programming and mathematical skills. But this is rarely the case because it is simply too difficult for a person to have all these interdisciplinary skills. Another obstacle that most companies face during this process is the lack of time to invest in in-house AI applications. Last but not least, a lot of concerns arise when the machine learning model played an important role in crucial decisions for the company. The trust in the AI models in terms of explainability/interpretability is a sensitive topic.

So, there are 3 main barriers for enterprises to become AI-driven; ***Expertise***, ***Speed***, ***Trust***.

From the **Data Science perspective**, there are also many obstacles in the process of developing machine learning solutions.

As it is known, there is no best model which fits for every problem. So, a large number of candidate ***models*** need to be tested, and each one of them has several hyperparameters that need to be a tuned. This adds a combinatorial complexity to the model selection problem. Also, there is a high number of ***features*** we have to choose from, without knowing for sure which one of them is relevant to the problem. On top of that, the model should generalize to unseen data, to have a good performance. Data scientists can best contribute their ingenuity to the model construction phase, yet anecdotally it seems that the most time-consuming pieces of machine learning are feature engineering and hyperparameter tuning. Thus, many models are not optimal as they move from experimental stages to production prematurely due to time constraints, or deployment in production is delayed.

To resolve all these issues addressed above, both from business and data science perspectives, there is need for a better way to shorten the time it takes to build machine learning models. Can some of the steps in the Machine Learning workflow be automated? Absolutely! Automated Machine Learning is one of the most important skills that successful data scientists need to have in their toolbox for improved productivity.

# Applications of Automated Machine Learning in every step of a Data Science / Machine Learning project

## Evolution of Automated Machine Learning

First applications of auto ML it was focused just on hyperparameter tuning of the machine learning algorithm, developing techniques such as grid search, random search and adaptive search. Nowadays will not call this Auto ML, but just hyperparameter tuning.

**The scope of Modern Auto ML** is to automate an entire machine learning pipeline, providing solutions to automate every step of the ml workflow.

## Traditional ML pipeline vs Auto ML pipeline

A **typical machine learning project** breaks down into discrete steps:

1. Translate Business problem to Machine Learning problem (ML task definition)
2. Collecting raw data
3. Merging data sources
4. Cleaning data
5. Feature engineering
6. Model construction
7. Hyperparameter tuning
8. Model validation
9. Deployment

In this section, I will provide a detailed explanation of how Auto ML tools can help in every step.

1. **Machine learning task detection based on target column**

|  |  |
| --- | --- |
| **Target/ Label type** | **Machine Learning task** |
| Boolean | Binary classification |
| Discrete Numerical / categorical | Multiclass classification (not to be confused with multilabel classification) |
| Continuous numerical | Regression |

**2. Machine Learning task detection based on input data**

In addition to these generic tasks, there are specific variations based on input data.

**Forecasting** is one such task type that is popular, given its relevance to many business problems like revenue forecasting, inventory management, predictive maintenance, and so on. If input data is time-series, determined by availability of a ***Date Time*** column, it is most likely a forecasting task. Different type of tasks can be ***clustering***, ***ranking*** etc, based that there is no target value and the machine learning problem falls in unsupervised learning category.

**3. Choosing evaluation metric**

Choosing a metric to evaluate your machine learning algorithm is fundamentally driven by the business outcome.

Automated Machine Learning can automate the process of selecting the right evaluation metric for a given input dataset and machine learning task. For instance, scenarios **like fraud detection** (which is a classification task) inherently have imbalanced data in that a very small percentage of data would be fraudulent. In this case, area under curve (AUC) is a much better evaluation metric than accuracy. Automatically detecting class imbalance in the data can help automatically choose AUC as an evaluation metric for this classification task.

**4. Automated Data preprocessing and cleaning**

Modern Auto ML tools have the ability to automatically find the type of each feature (Boolean, discrete, continuous numerical, text) and detect issues with input data and automatically flag them. Also, they can drop columns that are not useful as features, and apply various techniques to deal with missing values.

**5. Automated Feature engineering**

The feature engineering step needs the most effort and takes the highest portion of the time, during the development of a Machine Learning project. Tasks such Feature Extraction and Creation, Feature selection, Scaling, Balancing treatment, dealing with categorical variables (binning, encoding) etc. considers more as “Art” than Science. Auto ML tools can handle this kind of issues following predefined recipes given the type or raw data and the type of ML task.

**6. Model / Algorithm selection.**

As it is proven by the research experience, ***there is no single best algorithm, which fits for every problem.***

Auto ML tools accept as input training data along with labels and try a whole bunch of different machine learning pipelines. At first at random, but eventually, after some repetition, the process is guided. During model selection, automated ML runs several iterations. Each iteration uses different data preprocessing methods and algorithms.

Furthermore, a machine learning model is described by a combination of an algorithm and associated Hyperparameter values. So for every model tested, hyperparameter tuning must be done.

**7. Hyperparameter tuning**

Model hyperparameters are used during the model training process to establish the correct values of model parameters. They are external to the model, and their values cannot be estimated from data. The choice of the hyperparameters will affect the duration of the training and the accuracy of the predictions. As part of the model training process, data scientists usually specify hyperparameters based on heuristics or knowledge and often tune the hyperparameters manually.

Here are some examples of model hyperparameters for various machine learning algorithms:

* The k in k-nearest neighbors
* The desired depth and number of leaves in a decision tree
* The C and sigma in support vector machines (SVMs)
* The learning rate for a neural network training
* Simple linear regression doesn’t have any hyperparameters, but variants of linear regression, like Ridge regression and Lasso, do

The first applications of Automated Machine learning had focused on Hyperparameter tuning because it *relies more on experimental results than theory*, and thus the best method to determine the optimal settings is to try many combinations and evaluate the performance for each model.

There are two general approaches for hyperparameter tuning, implemented in almost every auto ml tool and machine learning framework.

***Brute force approaches***; picking machine learning algorithms at random and applying grid-search /random search / adaptive search for the hyperparameter values.

***Smarter approaches***.

For real-world problems, the search space is very large, and brute-force approaches will not be effective, because the size of the search space grows exponentially, as there are more hyperparameters. This has led to the emergence of smarter selection and optimization approaches, mostly powered by advanced statistics and machine learning techniques. Some of these approaches include *Bayesian optimization*, genetic programming, *multiarmed bandit*, and *meta-learning.*

**8. Model evaluation automation**

Programmatically automate various sampling techniques such as Hold-out, Cross-validation, etc to evaluate model performance.

**9. Model interpretation - Prediction analysis**

As has already been said the final adoption of Machine learning solutions from companies, heavily relies on the trust that the company has to the model. So almost every Auto Ml tool incorporates various methods to provide **Transparency** and **Explainability** about model predictions, features used and expected behavior.

**10. Monitoring and retraining**

Model performance during training can be very different from its performance after deployment on real data. Thus, it is important to continuously measure model performance even after deployment. Poor model performance is typically caused by a change in characteristics of input data over time, which is known as **concept drift.** Techniques exist to automatically monitor this drift and model performance over time.

As soon as the poor model performance is detected, corrective actions can be taken to minimize the damage.

Corrective actions could include the following:

* Immediately take the model offline (and disable the corresponding user experience)
* Retrain the model with the latest data and deploy the retrained model

**Wrapping up**

Automated Machine Learning empowers users (with or without machine learning expertise) to identify an end-to-end machine learning pipeline for any problem, achieving higher accuracy while spending far less of their time. And it enables a significantly larger number of experiments to be run, resulting in faster iteration toward production-ready intelligent experiences. Given input data, it can automate the entire Machine Learning pipeline.

# Automatically Network Architecture Search (NAS)

This is a popular area of research, trying to solve the problem of the automatic design of neural network architectures. With neural networks gaining popularity, meta-learning approaches have been applied to this problem. The most known tool was developed by Google AI researchers from the Brain team. Also, NAS techniques have been used to design networks that are on par or outperform hand-designed architectures.

Methods for NAS can be categorized according to the search space, search strategy, and performance estimation strategy used.

* **Search space** (degrees of freedom in the experimental space) – define the set of neural network architectures
* **Search strategy** – find a good performing model quickly, an exhaustive search doesn’t be a good idea because of the size of search space (Try Bayesian optimization, reinforcement learning, gradient-based approaches, evolutionary methods)
* **Performance estimation strategy** (ideally we will train a candidate model through an entire dataset and test its performance in the test set, but this is very time consuming, so to save time various shortcuts are used to gen an estimate of how a model will perform)
  + *lower fidelity estimates* (fewer training epochs, subset the data, downsample the data)

Test error will be of course lower than a model trained on the full train data, but the idea is that you can get a relative ranking between candidate models

* + *learning curve extrapolation*
  + *one-shot models with weight sharing*

# Advantages and Risks of Automated Machine Learning

**Advantages**

1. **Democratization of machine learning**

Automated ML is democratizing AI and empowering people across the enterprise to do machine learning with familiar tools. Also, AutoML is a really interesting area, because you can apply machine learning algorithms in real production without any knowledge about it. It opens Data Science for many developers, and allow them to create many interesting projects.

1. **Speed up the development and experimentation process**

Auto ML it is a huge time saver. Auto ML is actually a great productivity tool. It is going to help the data scientist or machine learning engineer be more productive when it comes to applying machine learning pipelines to datasets.

1. **No need for programming skills**

The standard structure of input-output data, adopted by almost any Auto ML tool, meaning that there is no need for programming skills

**Risks. Auto ML is overhyped?**

1. **Not fully Automated ML**

As one can understand, before any automated machine learning tool can be used, a lot of prior work required from a Data Scientist, to transform the business problem into a machine learning problem. Auto ML doesn't have (at least until now), any way to address this problem.

1. **Not customize solutions**

Moreover, Auto ML can't be yet competitive with humans in terms of accuracy and can't provide high complexity customized models. Because of its nature, the final models can't be designed to meet specific needs that a business may have.

1. **Poor feature creation**

Additionally, even though many Auto ML tools claim that can extract automatically features from a database, the truth is far from this statement. Data scientist can be create more complex features inspired by domain knowledge, while auto ml featurization is almost limited to naive statistics of raw data.

# Will Auto ml replace data scientists?

**Are the Data Scientist role in danger?**

A key ingredient for successful machine learning implementations is being able to map the business problem and the desired outcome to the appropriate machine learning problem. This is something that Auto ML is not able to find on its own, so the role of Data Scientist is not threatened. Auto ML will not replace the data scientist, in the same way, that the scikit learn or any other popular machine learning framework hasn’t replaced the data scientist. It is just a productivity tool. It saved data scientists from having to rewrite from scratch all the ml algorithms that they wanted to apply.

We should look at the Auto ML tools like an assistant. You can set an auto ml tool working on (for a day or two) and you can keep on with your analysis. The purpose of AutoML is to free data scientists from the burden of repetitive and time-consuming tasks.

**So, what is the core job of the modern Data Scientist?**

The core job of a modern Data Scientist has to be the transformation of the business problem to a machine learning problem.

Data scientists will able to provide custom Machine Learning solutions, which are capable to achieve better performance that Auto ML and meet specific requirements. They can mainly focus on problem formulation and modeling, instead of time-consuming tasks such as model selection and hyperparameter optimization.

Furthermore, the data scientist will be able to create more complex features inspired by domain knowledge, on top of the naive features created by Auto ML tools. Another crucial task for modern Data Scientist will be the debugging of the ML algorithms. What happened if a model did not behave as it is expected? Moreover, the implementation of the remaining pipeline is a job that has to be done by a human (data scientist or software developer).

# Conclusions

Automated Machine learning frameworks provide valuable solutions to common data science problems and they can dramatically improve the productivity of data science teams who then spend less time implementing algorithms and more time thinking about theory.

Auto ML should be another, yet very powerful tool, in the toolkit of the modern Data Scientist.

# Auto ML tools

**How Auto ML works?**

Auto ML services provide machine learning at the click of a button, or, at the very least, promise to keep algorithm implementation, data pipelines, and code, in general, hidden from view.

There are 2 categories of Automated Machine Learning tools.

Some tools offer only **partial solutions** such as Feature Engineering, model selection, Hyperparameter optimization, etc, while other tools provide an **end-to-end solution**, building the entire Machine Learning pipeline.

**How one can access Auto ML tools?**

Automated Machine learning tools can be used through an intuitive web interface or a simple API using a scripting language (R/Python).

## Commercial Automated Machine Learning tools

1. Microsoft Azure Automated Machine Learning (<https://docs.microsoft.com/en-us/azure/machine-learning/concept-automated-ml>)
2. Google AutoML – Google Cloud Platform (GCP) AutoML (<https://cloud.google.com/automl/>)
3. Amazon Web Services (AWS) (<https://aws.amazon.com/machine-learning/>)
4. Oracle Machine Learning Database (<https://www.oracle.com/database/technologies/datawarehouse-bigdata/oml-notebooks.html>)
5. Data Robot (<https://www.datarobot.com/>)
6. H2O.ai (<https://www.h2o.ai/products/h2o-driverless-ai/>)
7. SAS factory miner (<https://www.sas.com/el_gr/software/factory-miner.html>)
8. IBM Watson studio (<https://www.ibm.com/cloud/watson-studio/autoai>)
9. ML-jar (<https://mljar.com/>)
10. RapidMiner (<https://rapidminer.com/>)

## Open-source Automated Machine learning tools

1. Auto-sklern (<https://automl.github.io/auto-sklearn/master/>)
2. Auto-pytorch (<https://github.com/automl/Auto-PyTorch>)
3. Auto-weka (<https://www.cs.ubc.ca/labs/beta/Projects/autoweka/>)
4. TPOT
5. H2O.ai (<http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html>)
6. Devol
7. Auto-keras (<https://autokeras.com/>)
8. ML-box (<https://github.com/AxeldeRomblay/MLBox>)
9. featuretools (<https://www.featuretools.com/>)
10. SMAC (Sequential model-based algorithm configuration) (<https://www.automl.org/automated-algorithm-design/algorithm-configuration/smac/>)
11. BOHB (Robust and Efficient Bayesian Optimization framework) (<https://www.automl.org/automl/bohb/>)

# Material and Bibliography

**Books**:

1. Frunk Hutter, Lars Kotthoff, Joaquin Vanschoren (editors), *Automated Machine Learning, Methods, Systems, Challenges.* The Springer series on Challenges in Machine Learning.
2. Deepak Mukunthu, Parashaar Shah, Wee Hyong Tok, *Practical Automated Machine learning on Azure,* O’Reilly

**Github repositories:**

* Awesome Auto ML papers: <https://github.com/hibayesian/awesome-automl-papers>
* Awesome Auto ML: <https://github.com/dragen1860/awesome-AutoML>

**web-pages:**

* <https://www.automl.org/>
* <https://automl.info/>