ML-based transfer of knowledge in large-scale IT modernisation projects in the life insurance industry

Axel Helmert, Florian Karl, Zoom-Meeting, Vienna / Munich,
March, 17th, 2021, msg life ce gmbh, Fraunhofer IIS

Agenda: March, 17th, 2021, 16:30 - 17:30

1	Introduction, overview and objectives	16:30 – 16:35	
2	Applying ML in the life insurance industry	16:35 – 16:50	★
3	Uncertainty and supervisory authorities	16:50 – 17:00	★
4	AutoML and XAI	17:00 – 17:20	★
5	Resume, outlook and discussion	17:20 – 17:30	★

Agenda: March, 17th, 2021, 16:30 - 17:30

1	Introduction, overview and objectives	16:30 – 16:35
2	Applying ML in the life insurance industry	16:35 – 16:50
3	Uncertainty and supervisory authorities	16:50 – 17:00
4	AutoML and XAI	17:00 – 17:20
5	Resume, outlook and discussion	17:20 – 17:30

1 Introduction, overview and objectives

Abstract

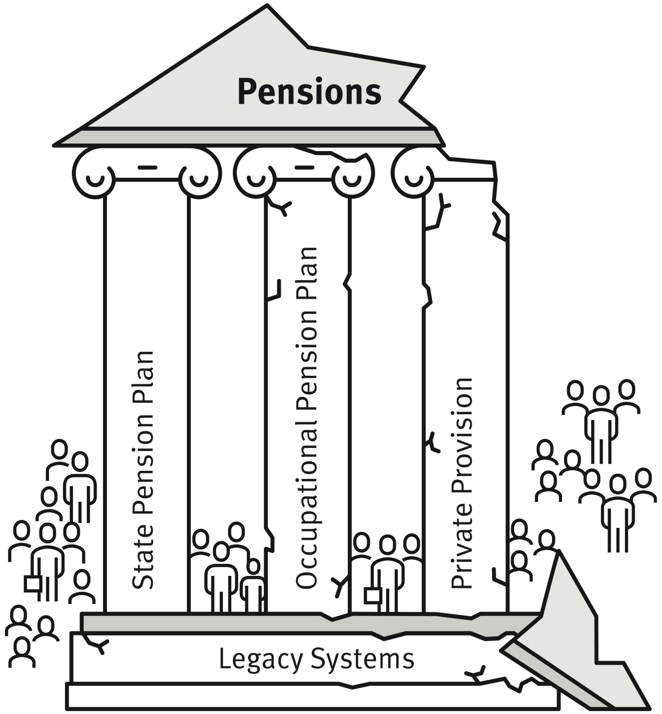
Low interest rates, the shift in demographics and Covid-19 have brought old-age provision to the brink of a systemic crisis, as current studies by Europe's leading policy institutions outline the overall situation.

Moreover, millions of insurance policies run on outdated IT systems, which make them excessively cumbersome to manage.

Hence, old contract data, together with the corresponding mathematical functions that define the behaviour of the contracts, need to be migrated, i.e. transferred from one or more source systems to a target system.

By using trustworthy deep neural networks **(**DNNs**)** or other methods, calculating values like a premium, a present value or a reserve can be learned by a machine.

The learned knowledge can be transferred and reactivated in a modern target system automatically.



Agenda: March, 17th, 2021, 16:30 - 17:30

1	Introduction, overview and objectives	16:30 - 16:35
2	Applying ML in the life insurance industry	16:35 – 16:50
3	Uncertainty and supervisory authorities	16:50 – 17:00
4	AutoML and XAI	17:00 – 17:20
5	Resume, outlook and discussion	17:20 – 17:30

2 Applying ML in the life insurance industry

Introduction and Goals: The economic background

1. Today migration is still done manually and comes at a high price. A mid-range migration costs a double-digit million amount and is thus often postponed or not carried out at all.
2. msg life is currently migrating several small and large portfolios at the same time (in Germany alone with a total of more than 10 million contracts).

➔ The approach is economically very promising and could help stabilise European pension systems

Introduction and Goals

In the application discussed here Machine learning is part of a four-step **automated** and **iterative** process:

1. **Learn** a function f of a source or reference system until the given quality is reached.
   The result is an approximation f* with this quality.
2. Perform a final quality assurance of the function (**Test**).
3. In the positive case, transfer the function to the target system (**Deploy**).
4. Use function f* (instead of f) in all operational business processes in the target system.

Consequences and Remarks

- We are dealing here (for the first time) with an industrial operational use of ML in life insurance.
- This results in tough requirements from a regulatory and actuarial point of view.
- Even if we look specifically at migrations in life insurance here, there is a greater field of use in all industries: Learn from old systems, transfer knowledge and reactivates it within modern IT-systems automatically!
- Despite the difficult conditions, we are already preparing a productive migration for next year with a customer, in which part of the functionality will be implemented with ML.
- Tomorrow I will give this lecture (in a slightly different form and supplemented by a prototypical demo) in our user group in front of more than 20 European life insurers.

Math

- We want to approximate an actuarial function $\boldsymbol{f}(\vec{\boldsymbol{x}}_{\boldsymbol{c}}, \vec{\boldsymbol{x}}_{\boldsymbol{p}}) = \boldsymbol{y}$ of a product P of a source or reference system.
- In the conventional implementation, the input consists of (derived) contract values $\vec{\boldsymbol{x}}_{\boldsymbol{c}}$ (age, sex, term, …) and (modified) product parameters $\vec{\boldsymbol{x}}_{\boldsymbol{p}}$ (interest rate, mortality, charges and fees, …).
- If DNNs are used, a function $\boldsymbol{f}^*(\vec{\boldsymbol{x}}_{\boldsymbol{c}}, \boldsymbol{\theta}) = \hat{\boldsymbol{y}}$ of a product P* is learned through the training.
- In f*, the actuarial knowledge of the function f is represented by the internal network parameters $\boldsymbol{\theta}$ and the network architecture. (The DNN can contain actuarial knowledge in sub-models).

Model

In the first step we have reduced the scope:

- The aim is currently (only) to determine **all present values** of a product P in P* by ML.
- msg life has a complete calculation kernel inside the policy admin system:
  It is much easier to follow our approach step by step
- That means the solution in the target system is a cooperation of conventional mathematics and ML-components.
- The advantage of approximating present values is twofold:
  - The set of input data is smaller and manageable
  - It results in a consistent approach in the actuarial calculations

Model

- The results of the functions f and f * are then values $y, \hat{y} \in \mathbb{R}$.
- Additional inputs and outputs can be relevant for the training of the DNN. The input $\vec{\boldsymbol{x}}_{\boldsymbol{c}}$ and the output $\widehat{\boldsymbol{y}}$ refers to the test or the operational use of the function f *.
- In the last month we have tried to improve results by providing actuarial knowledge to the DNN.
- And we obtained good results:
  - **training is faster** and, a the same time,
  - **improved quality**
- **As an example:** Applying actuarial knowledge during the training seems to reduce the problem of local minima
- A blueprint of the idea is shown in the next diagram

2 Applying ML in the life insurance industry

Deep Neural Networks with actuarial knowledge, based on the equivalence principle

Input

age, sex, term

premium p, reserve V, benefits b

Submodel f_1^*

present value of premiums (pvp)

Submodel f_2^*

present value of benefits (pvb)

Actuarial Layer

$$p = \frac{b \cdot pvb - V}{pvp}$$

$$V = b \cdot pvb - p \cdot pvp$$

Output

pvp, pvb

Additional Output: p, V

Agenda: March, 17th, 2021, 16:30 - 17:30

1	Introduction, overview and objectives	16:30 – 16:35
2	Applying ML in the life insurance industry	16:35 – 16:50
3	Uncertainty and supervisory authorities	16:50 – 17:00
4	AutoML and XAI	17:00 – 17:20
5	Resume, outlook and discussion	17:20 – 17:30

3 Uncertainty and supervisory authorities

Quality, trustworthiness and explainability

1. The actuarial and regulatory requirements relate to many different aspects:
   a) **Quality**: Limits for the maximum deviation between values from the source and the target system.
   b) **Trustworthiness**: What quality is required for which calculations?
   c) **Explainability** and adaptability: How can the solution be explained and also adapted if necessary?
2. The size and structure of the relevant input data Φ and the training set X play an important role in assessing the quality, but also in selecting the method:
   a) The situation $X = \Phi$ for a finite ("small") set X is an important special case. In this case, other methods (than DNNs) can also be used successfully, for example a **decision tree**.
   b) An example for measuring the quality: For each result $\hat{y} = f^*(\vec{\boldsymbol{x}}_{\boldsymbol{v}}, \theta)$ with $\vec{\boldsymbol{x}}_{\boldsymbol{v}} \in \Phi$ relative and absolute limits for the maximum deviation are defined. If the quality is defined by a maximum absolute deviation $\varepsilon > 0$, then:
   $\forall \vec{\boldsymbol{x}}_{\boldsymbol{v}} \in \Phi$ applies: $|\hat{y} - y| < \varepsilon$

3 Uncertainty and supervisory authorities

Quality, trustworthiness and explainability

- The current state of research in mind, we assume that the fulfilment of the high requirements for quality and explainability in the context of calculating actuarial values can, in the next years, only be achieved if a reference system is available.
- Under this condition large and well-structured training sets (X, Y) can be generated automatically and with sufficient performance (and we are able to apply Meta-Learning).
- The good news is that such reference systems often exist and the use of DNNs no longer depends on the specific number of contracts in the portfolio.
- Reference systems can and are often developed and maintained directly in the actuarial office. If these form the basis for training the DNNs, it can be ensured that the values are calculated very precisely and with a high degree of certainty in productive operation as well.

Model

- One might wonder why ML is actually needed when the reference or source system has ready-made software.
- This question is important: The automated process is not only a logical, but also a technical transfer.
- The result of the process is a software component that meets the technical requirements of the target system and can collaborate with other components in the target system.
- An experienced software engineer knows that it is usually very difficult and expensive (or even impossible) to connect old software or software that was not written for operational use directly to a modern policy administration system.
- In other words, the normal approach to these large IT consolidation projects is a mixture of manual reimplementation of functionality of the old system based on the use of the new system.
- We will only bring ML into play if there is no adequate solution in the target system that we can use directly or with simple adjustments. With this in mind, ML is always a substitute for manual implementation, even if software is available for reference. This assumption is not unique to the life insurance industry.

3 Uncertainty and supervisory authorities

Quality, trustworthiness and explainability

- In addition to the economic goals: saving effort, time and human resources through automation, there are important logical and technical advantages:
- The additive use of DNNs, as a supplement to the conventionally programmed functions, is particularly useful if
  - there are no suitable templates in the target system or if
  - implementation is undesirable with the aim to reduce complexity in the target system or if
  - the implementation in the source system is very old and undocumented.

3 Uncertainty and supervisory authorities

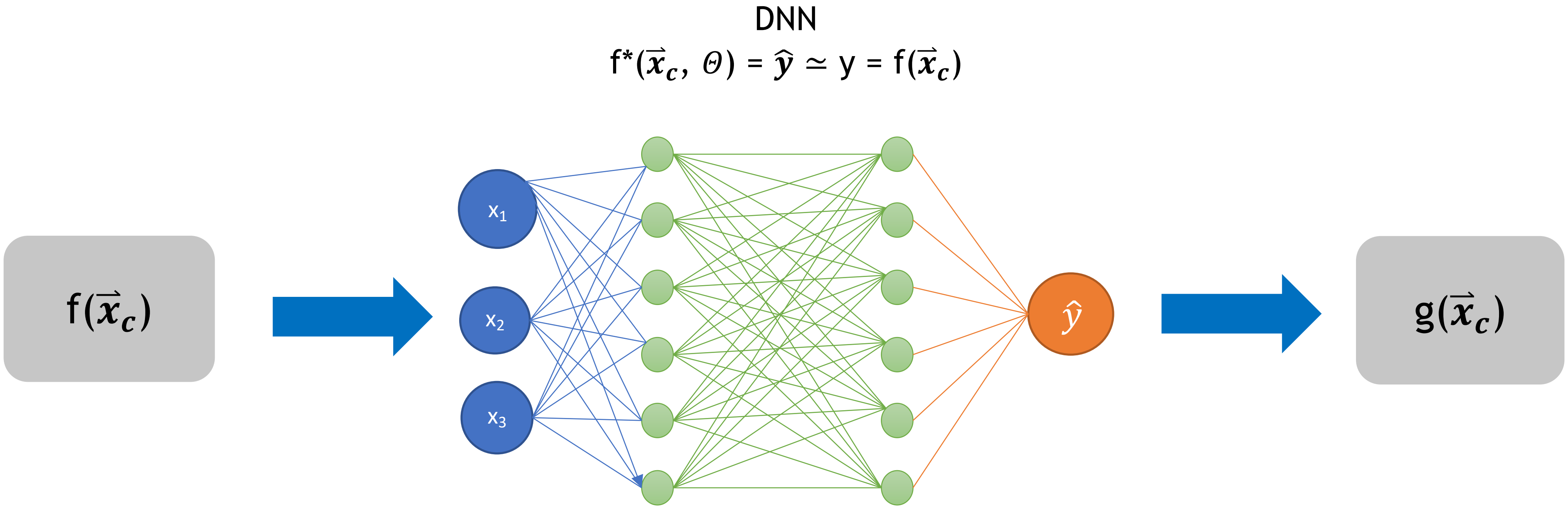
Quality, trustworthiness and explainability

1. The process applied currently provides a reference system even after migration.
2. This enables us to react in future error situations, complaints and extensions
(e.g. due to regulatory requirements).
3. We have learned to adapt the trained DNNs in the event of extensions.
The basis for this is again the (extended) reference system.
4. As long as we can start from this assumption, the problem of explainability is also addressed.

XAI

Research results from 2019/2020 enable the generation of formulas. It is a kind of reverse engineering

The metamodel $g(\vec{x}_c)$ is a closed-form expression, generated from f*.

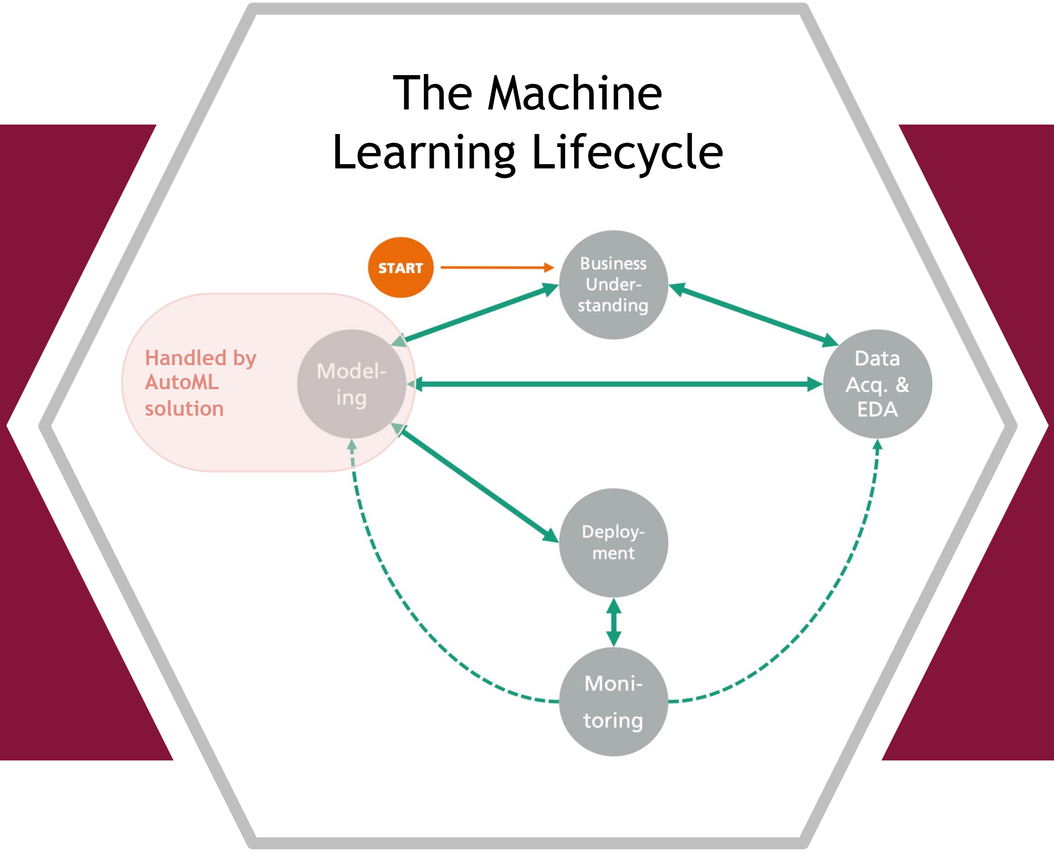


Agenda: March, 17th, 2021, 16:30 - 17:30

1	Introduction, overview and objectives	16:30 – 16:35
2	Applying ML in the life insurance industry	16:35 – 16:50
3	Uncertainty and supervisory authorities	16:50 – 17:00
4	AutoML and XAI	17:00 – 17:20
5	Resume, outlook and discussion	17:20 – 17:30

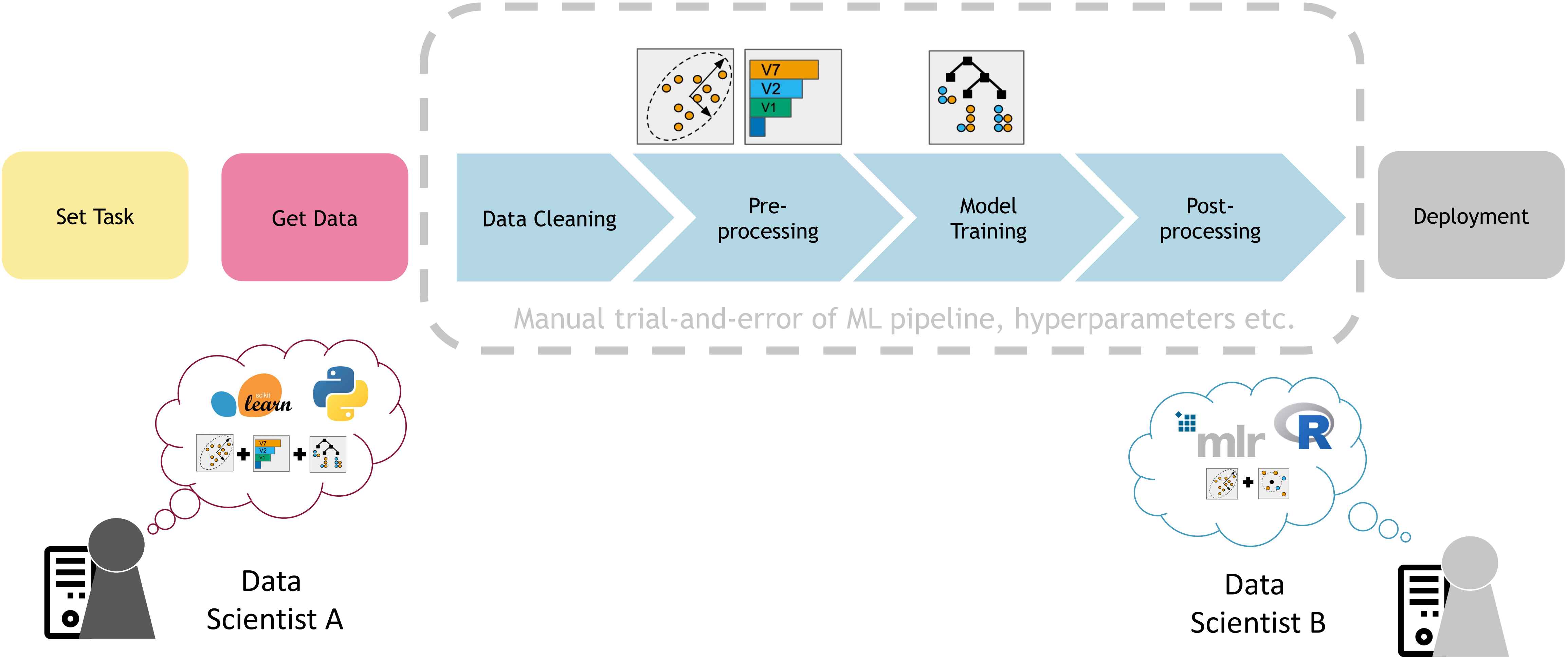
AutoML in the Context of the Machine Learning Lifecycle.

1. More and more tasks are approached via data driven methods.
2. Data Scientists often rely on trial-and-error.
3. Especially tedious for similar, recurring tasks.
4. Not the entire Machine Learning Lifecycle can be automated.



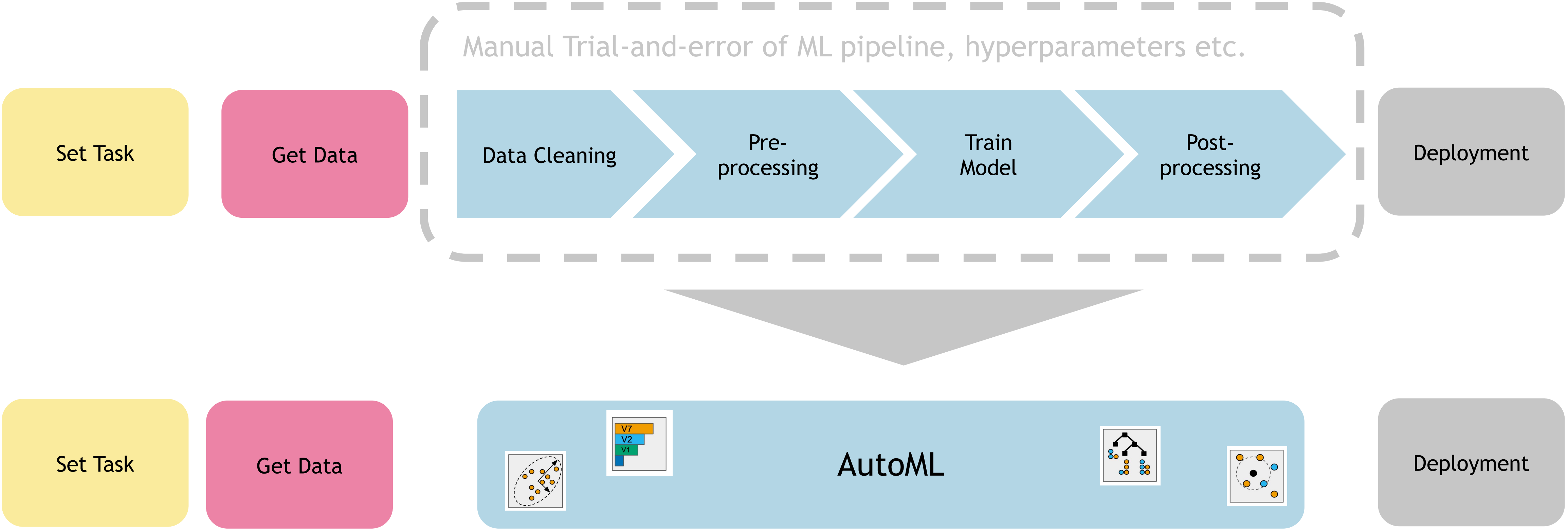
4 AutoML and XAI

Machine Learning is Messy Sometimes.



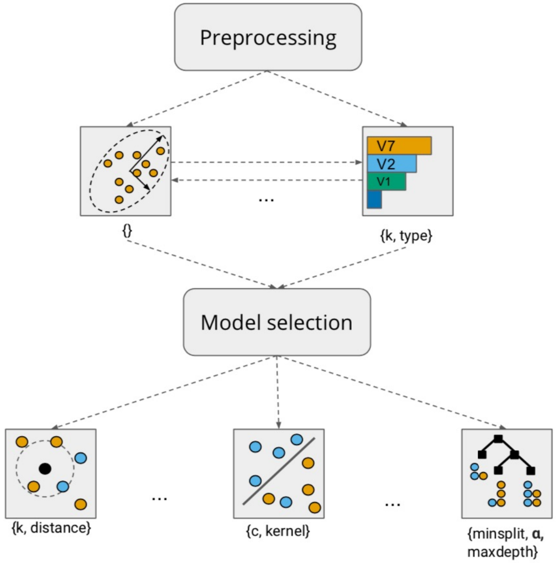
AutoML Tries to Replace Manual Tuning Effort with an Automated Solution*.

*and a lot of computational effort

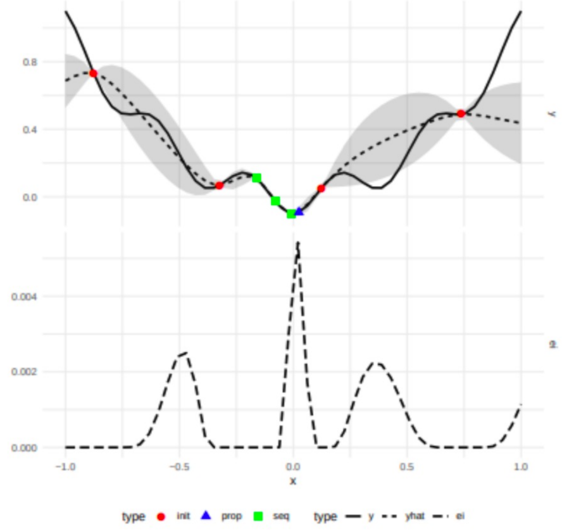


AutoML - Improving Machine Learning with Machine Learning (and Optimization).

AutoML is the **optimization** over a **search space of machine learning pipelines** to minimize a metric of choice.



Representation



Optimization

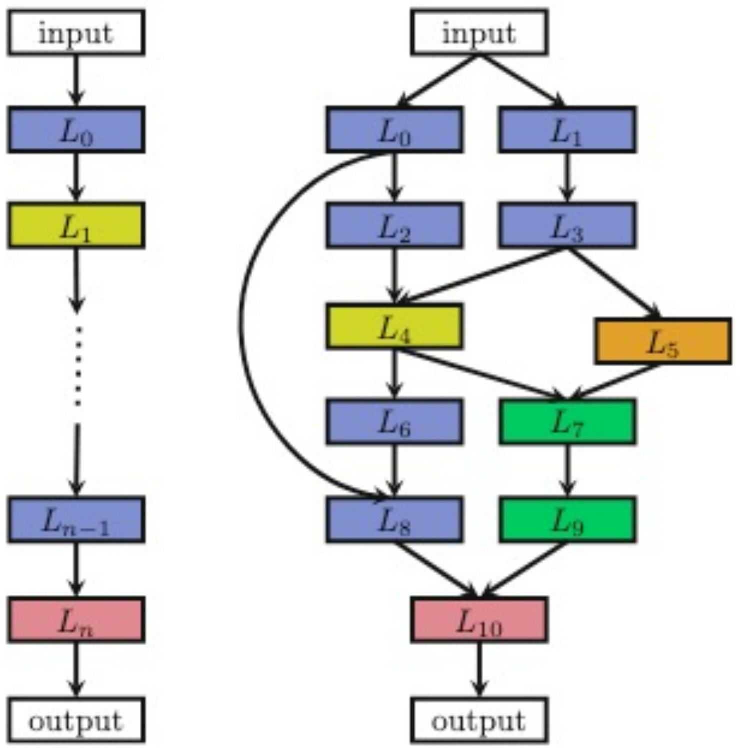
Optimization Methods for AutoML / Neural Architecture Search

- Random/Grid-Search
- Bayesian Optimization
- Evolutionary Algorithms
- Reinforcement Learning — ENAS: Efficient Neural Architecture Search via Parameter Sharing
- Gradient-Based Methods — DARTS: Differentiable Architecture Search

No Free Lunch Theorem
Choose your optimization wisely!

Neural Architecture Search (NAS): The Quest for Better (and Less Affordable) Deep Learning.

- Neural Architecture Search is a **subfield of AutoML.**
- Mostly cell-based approaches to search space to keep the space simple.
- Search space often quite limited and geared towards specific applications (e.g. InceptionTime).



[Hutter, Frank, Lars Kotthoff, and Joaquin Vanschoren. *Automated Machine Learning*. Springer: New York, NY, USA, 2019.]

Why Even Go Towards Deep Learning?

Deep Learning is actually good on tabular data.
- ➔ From experience in the insurance domain.
- ➔ Deep Learning approaches geared specifically at tabular data (e.g. TabNet) have shown success.
- ➔ Recent NAS approaches for Tabular Data: Auto-Pytorch Tabular, AutoGluon Tabular.

Data Situation is impeccable.
- ➔ Data can be generated through reference engines.
- ➔ Practically no noise present in the data.

A specific search space can be formulated.
- ➔ This is critical for successful NAS (e.g. InceptionTime, Auto-Pytorch Tabular).
- ➔ Domain and Machine Learning Knowledge will be needed.

A multi-objective approach may be needed.
- ➔ Model efficiency /model size are important.
- ➔ Alternative: constrained optimization.

… and all other advantages that Deep Learning will bring to the table.

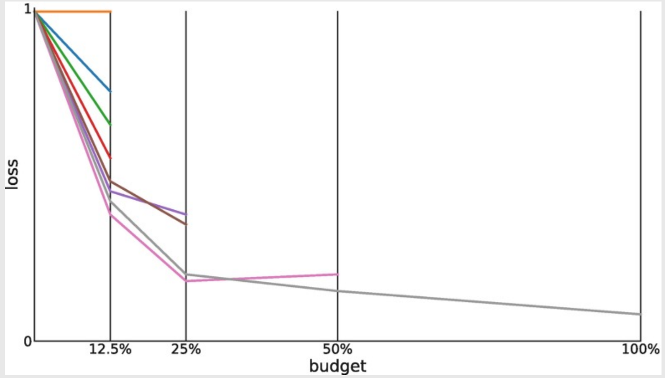
The Crux with Neural Architecture Search.

I. It is expensive.

Speedups of the NAS process are extremely important:
- Weight sharing.
- Meta-Learning.
- Multi-Fidelity approaches (less data, less epochs, resolution).

AutoGluon

Benchmarking also very expensive (surrogate benchmarks help).



[Hutter, Frank, Lars Kotthoff, and Joaquin Vanschoren. *Automated Machine Learning*. Springer: New York, NY, USA, 2019.]

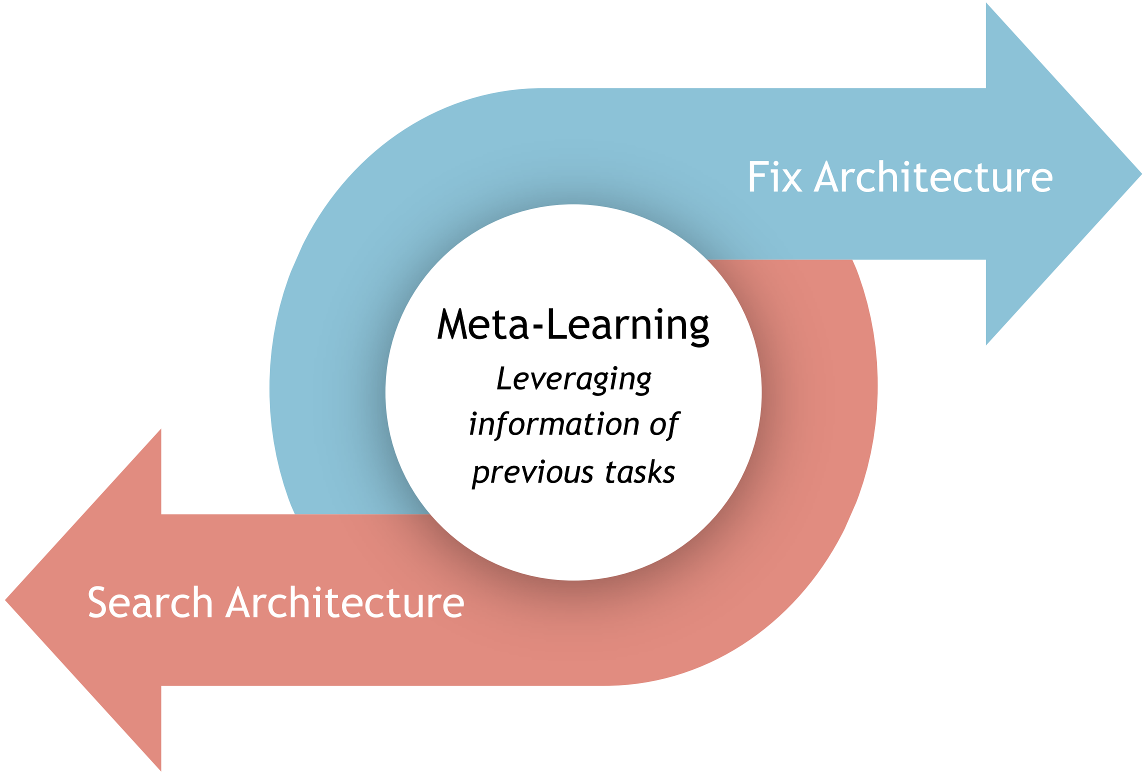
II. Proper benchmarking is hard.
"NAS evaluation is frustratingly hard." *by Yang et al*. tells the tale.

- Training pipeline often matters more than architecture.
  ➔ Results are not all that comparable (hyperparameters, code, evaluation etc.)
- Are architecture and hyperparameters optimized jointly?

Information is King: Using Information from Other ML Tasks with Meta-Learning.

Recommender

- Use previous information to make model search efficient.
- Pipeline/architecture **recommendation.**
- **Warm-starting** AutoML and Neural Architecture Search.
- **Meta-Features** can be helpful.



Learning to Learn

- Use previous information to create/update model.
- Utilize pre-trained models with **Transfer Learning.**
- **Few-Shot Learning** for new tasks with only few samples.

Using Meta-Learning to Create an “Insurance Mathematics Expert”.

Recommender

- Learning to test cheap and promising models first.
- Awkward search space with Deep Learning architectures and traditional Machine Learning model.
- Proven in several existing AutoML and NAS systems[1].

Learning to Learn

- Transfer learning can not only be used for computer vision tasks.
- Few Shot scenario may be of interest – very similar recurring tasks.

1 e.g. autosklearn, Auto-PyTorch Tabular

When Your Models are Too Complex: Explainable Machine Learning / XAI.

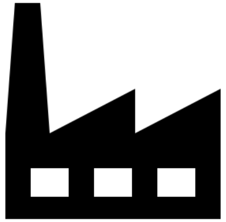
Machine Learning Pipelines and Deep Learning approaches are usually black-box.

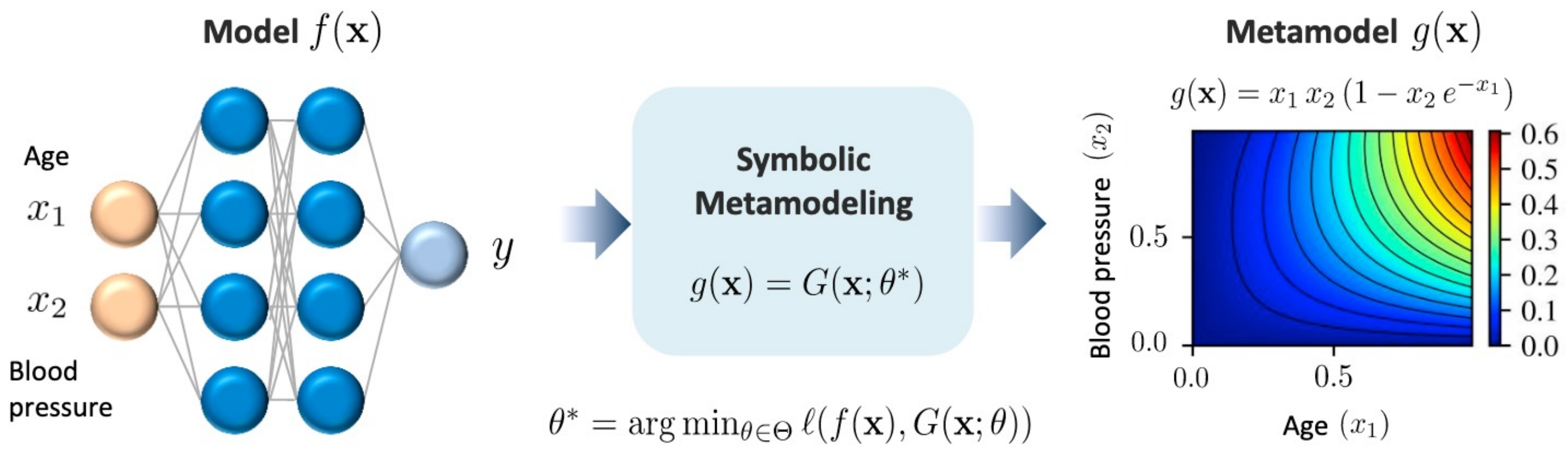
The case for Interpretability.

Justify:	Investigate if and why biased or unexpected predictions were made.
Control:	Debug models, identify and correct vulnerabilities.
Improve:	Understanding allows model improvement.
Discover:	Learn new facts, gather information and gain insights.

Metamodeling: From black-box to white-box.

In some domains, like insurance, decisions have to be completely trustworthy and interpretable.





[Alaa, Ahmed, and Mihaela van der Schaar. "Demystifying black-box models with symbolic metamodels." (2019).]

An (interpretable) metamodel approximates the black-box.

Evaluation of suitability is tricky.

Agenda: March, 17th, 2021, 16:30 - 17:30

1	Introduction, overview and objectives	16:30 – 16:35
2	Applying ML in the life insurance industry	16:35 – 16:50
3	Uncertainty and supervisory authorities	16:50 – 17:00
4	AutoML and XAI	17:00 – 17:20
5	Resume, outlook and discussion	17:20 – 17:30

Thank you for your attention.

CONTACT

Axel Helmert, Florian Karl

Axel.helmert@msg-life.com