Nordic probabilistic AI school

Introduction to probabilistic programming languages (PPLs)

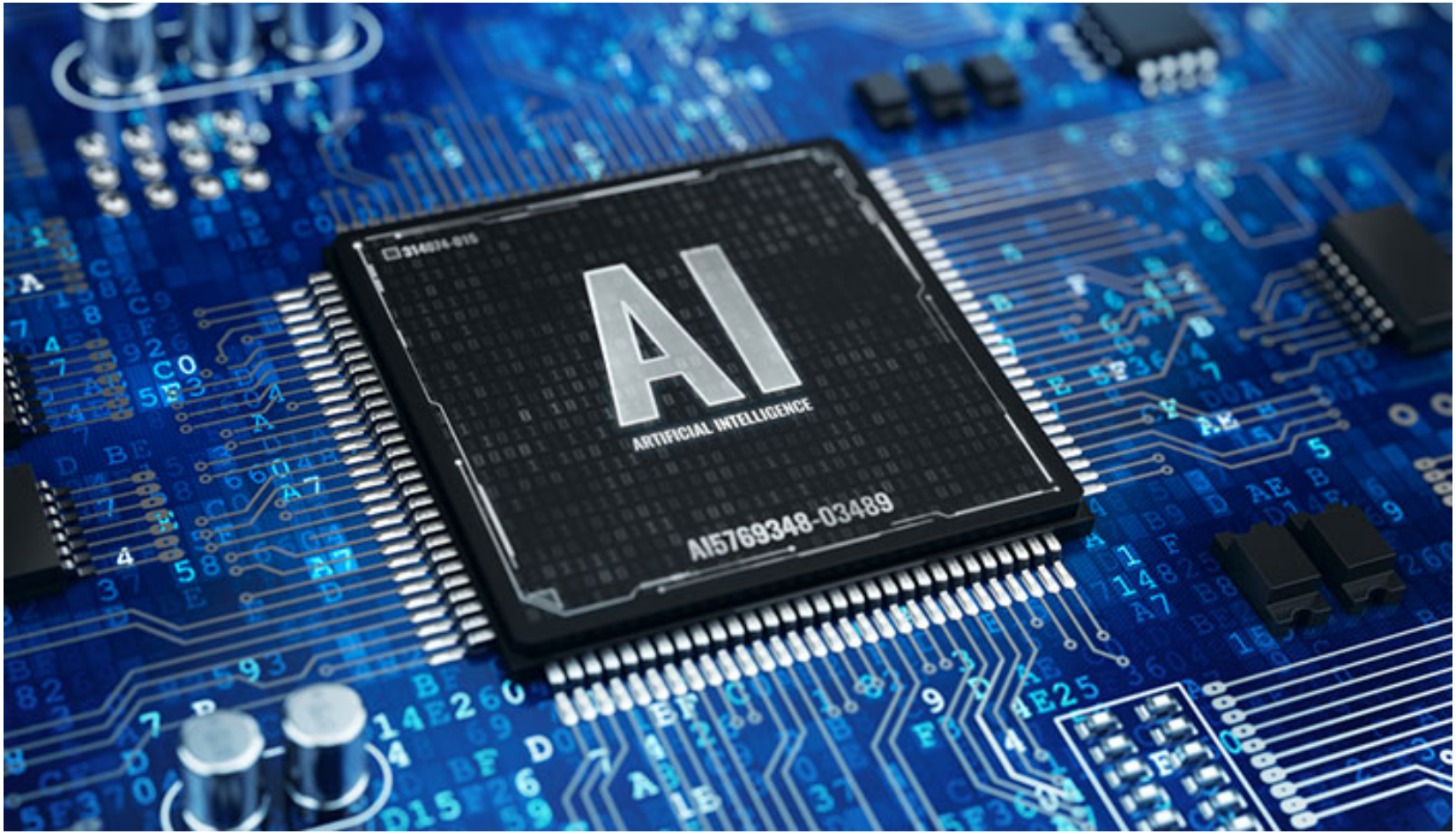
Andrés Masegosa, and Thomas Dyhre Nielsen

June 13, 2022

Plan

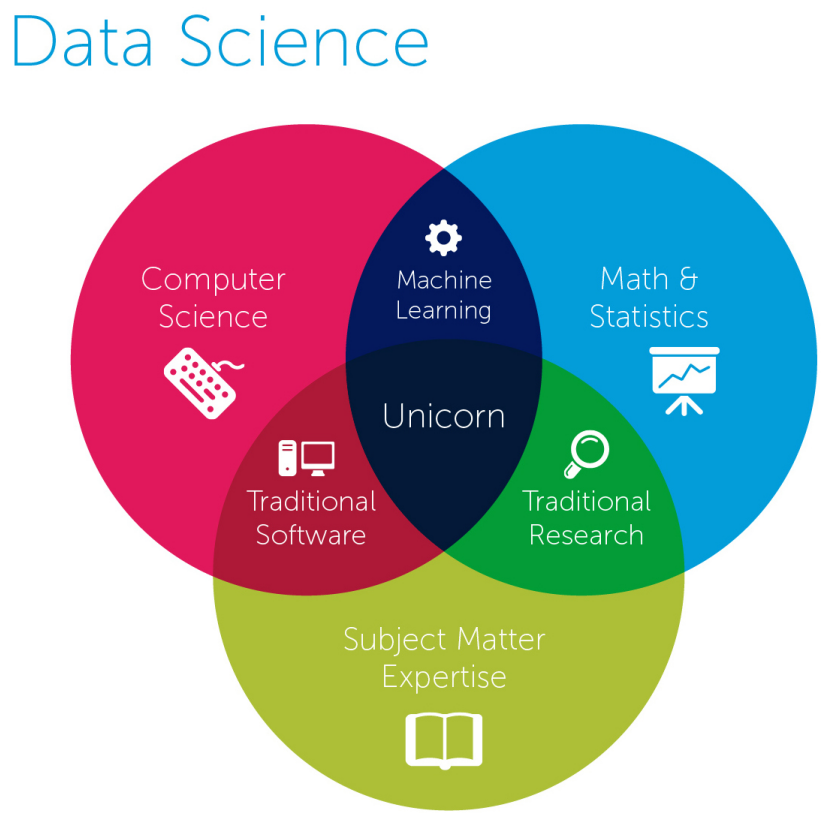
- **Day 1: Probabilistic programming**
  - Introduction to probabilistic programming
  - Probabilistic programming in Pyro
- **Day 2 (Before Lunch): Classical Variational Inference**
  - Introduction to Variational Inference
  - Mean-Field Approximation
  - Coordinate-ascent variational inference
- **Day 2 (After Lunch): Modern Variational Inference**
  - Black box variational inference
  - Variational inference in Pyro
- **Day 2 (Evening): Modern Variational Inference**
  - Variational AutoEncoders
  - Amortized Inference

Machine Learning Systems



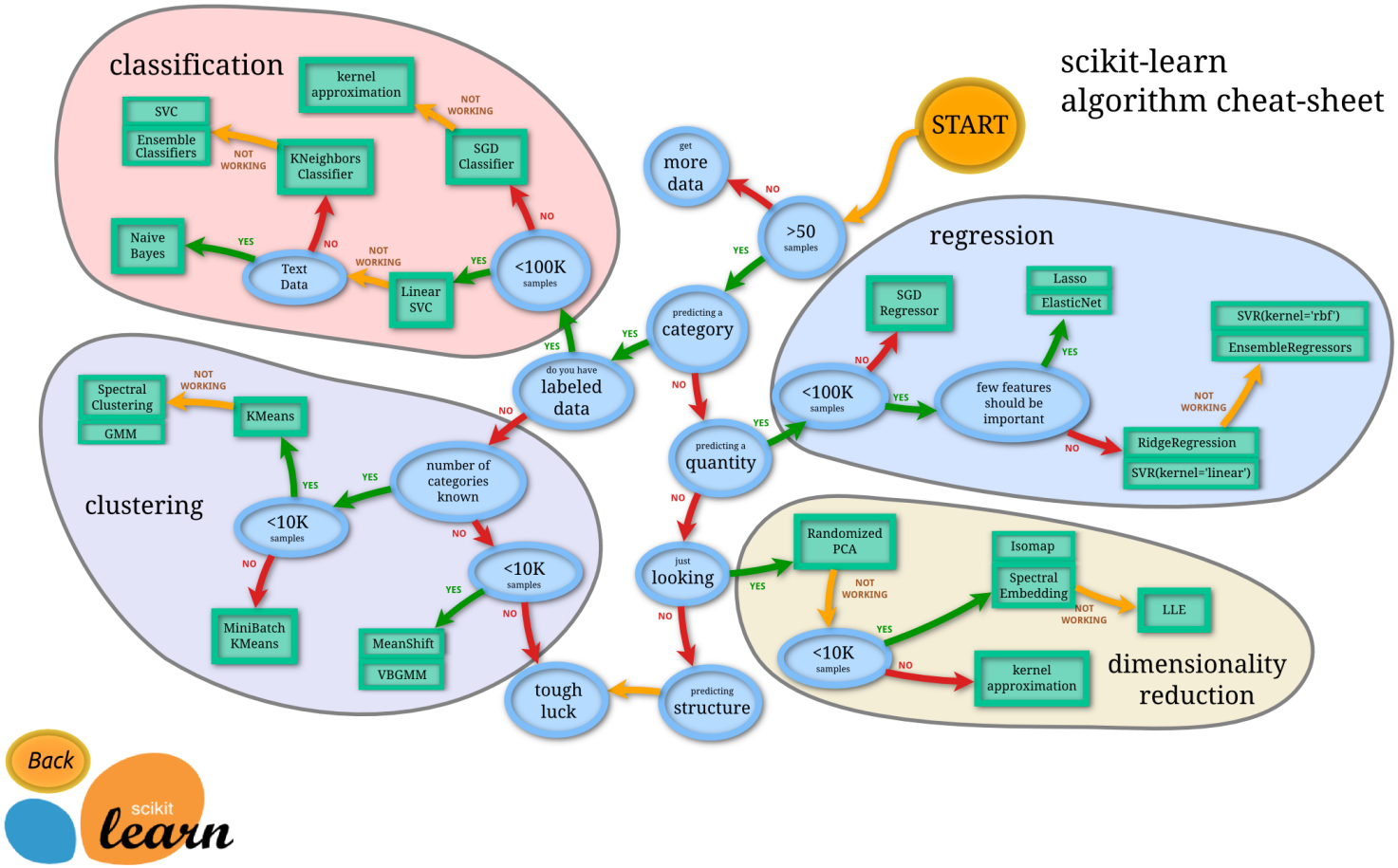
The development of **machine learning systems** requires enormous efforts.

-



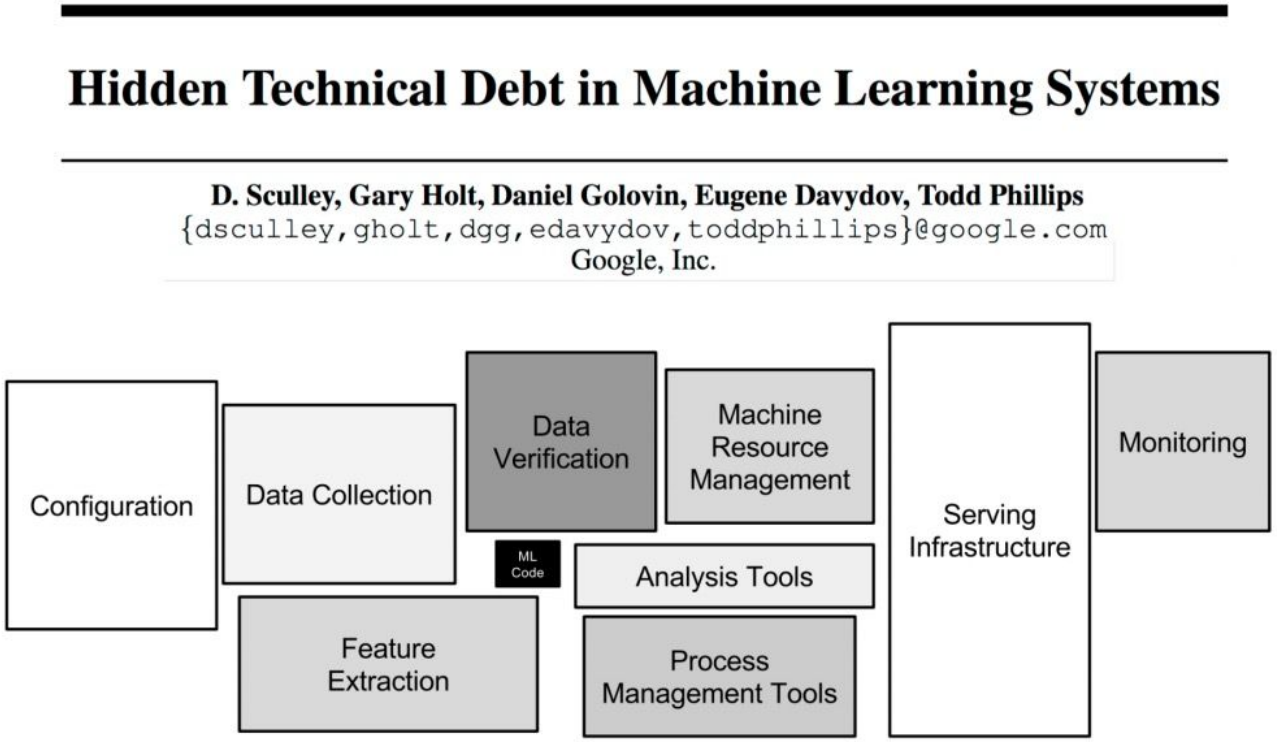
The development of **machine learning systems** requires enormous efforts.

- **It requires of highly qualified experts.**



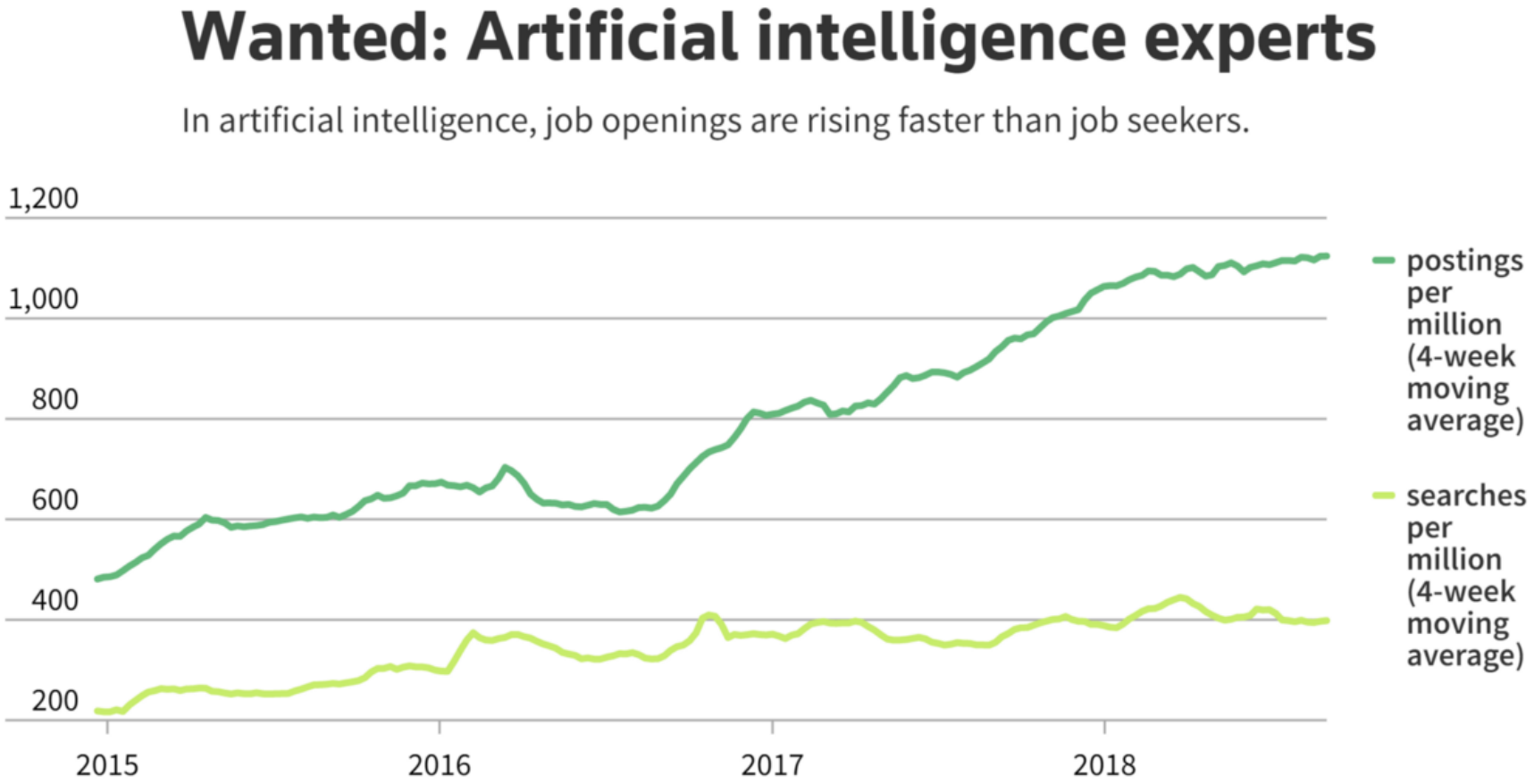
The development of machine learning systems requires enormous effort.

- It is necessary to have highly qualified experts.
- **It is difficult to find the ML model most suitable for an application.**



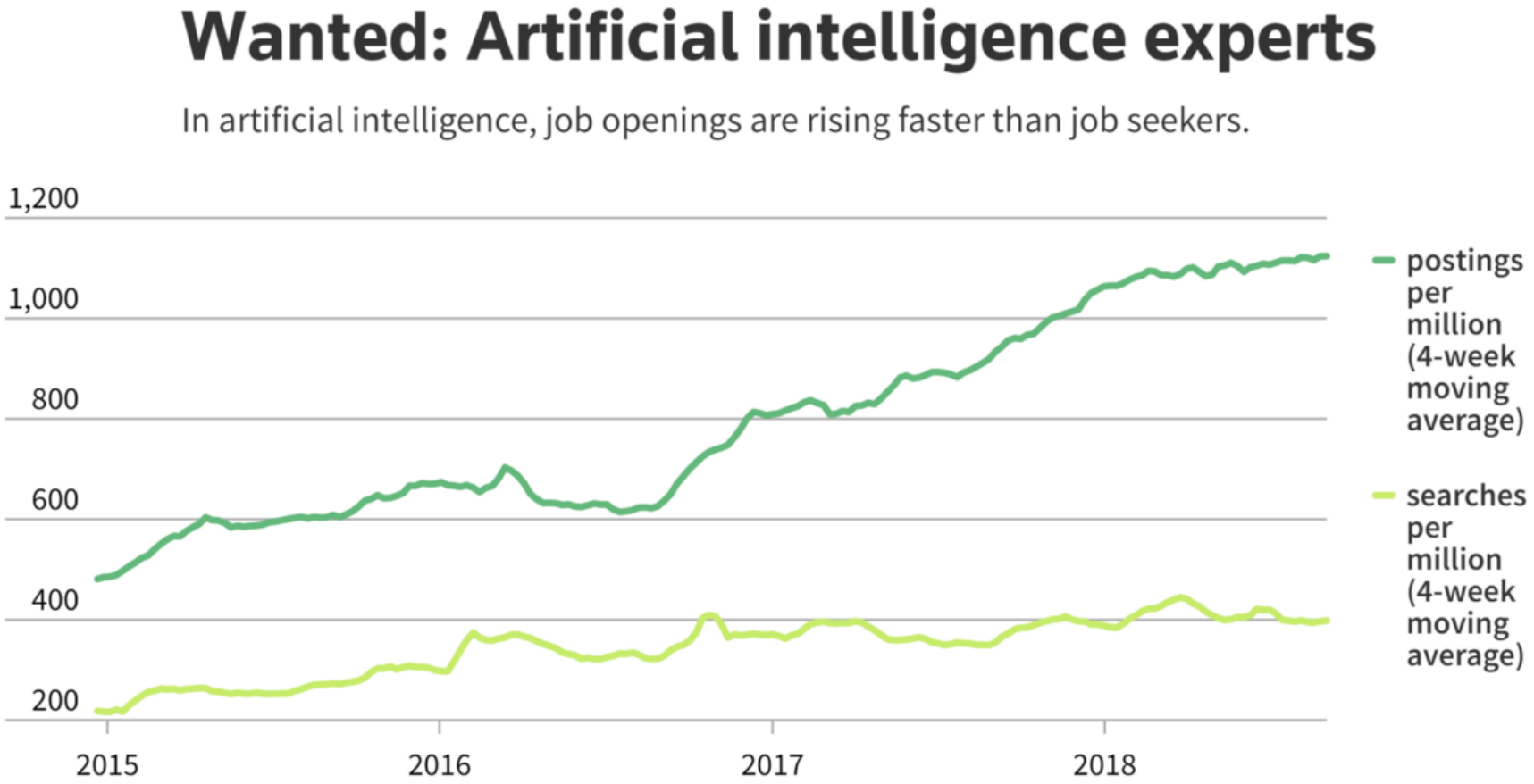
The development of machine learning applications requires enormous effort.

- It is necessary to have highly qualified experts.
- It is difficult to find the ML model most suitable for an application.
- **Programming a ML model is a complex task where many problems are intermingled**.



Consequences:

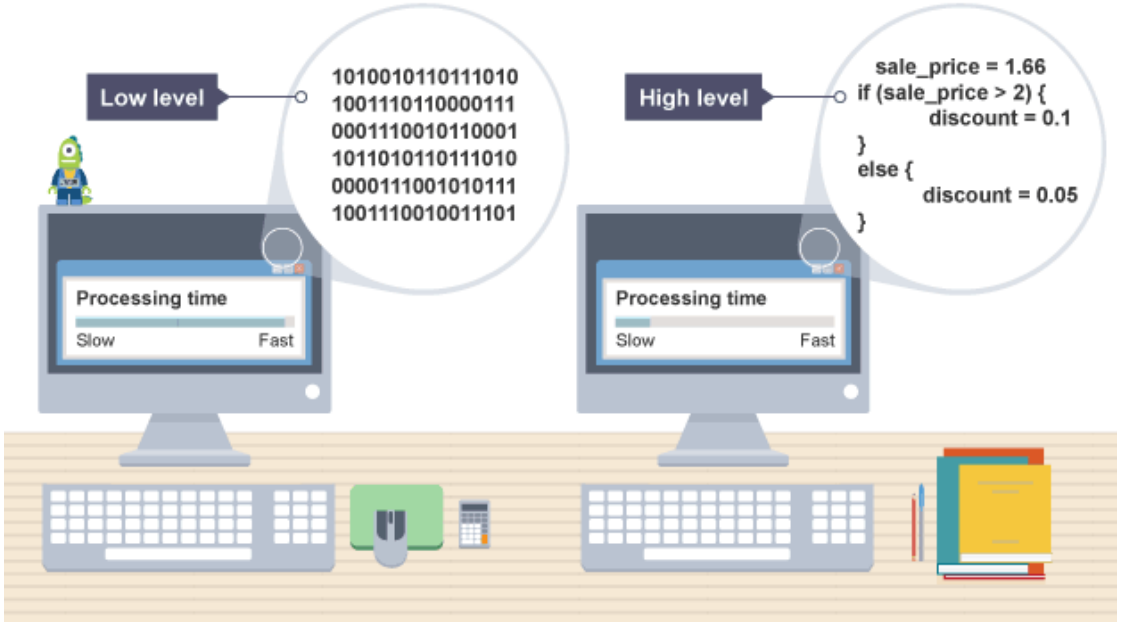
- Shortage of AI experts (and high salaries).



Consequences:

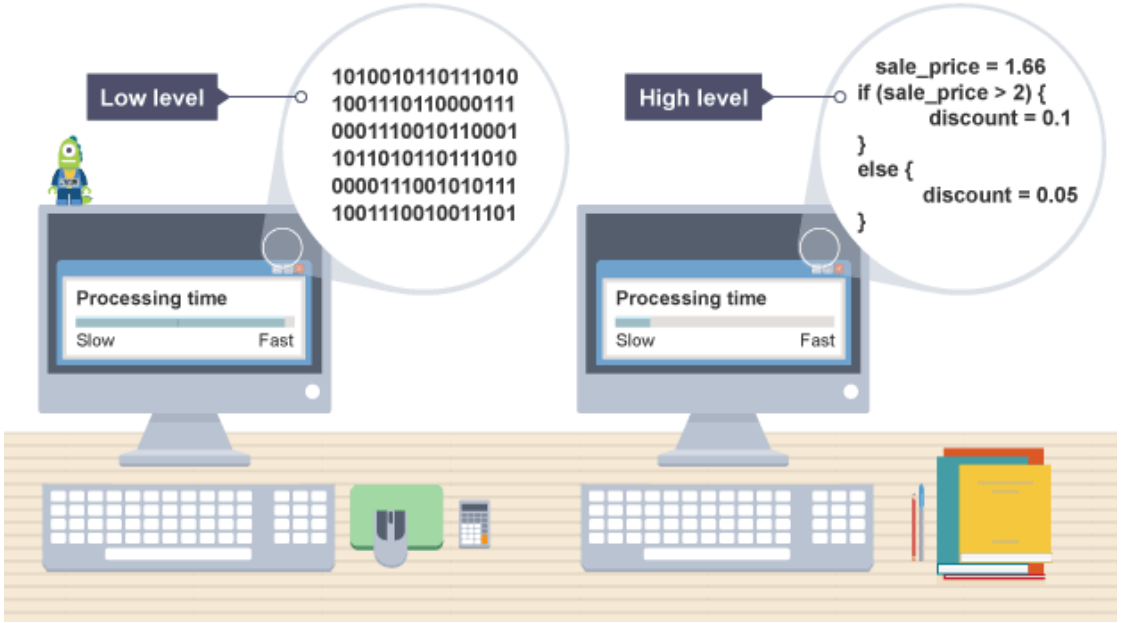
- Shortage of AI experts (and high salaries).
- Only big corporations have the resources for developing ML systems.

Programming Languages



Similar situation than 50 years ago:

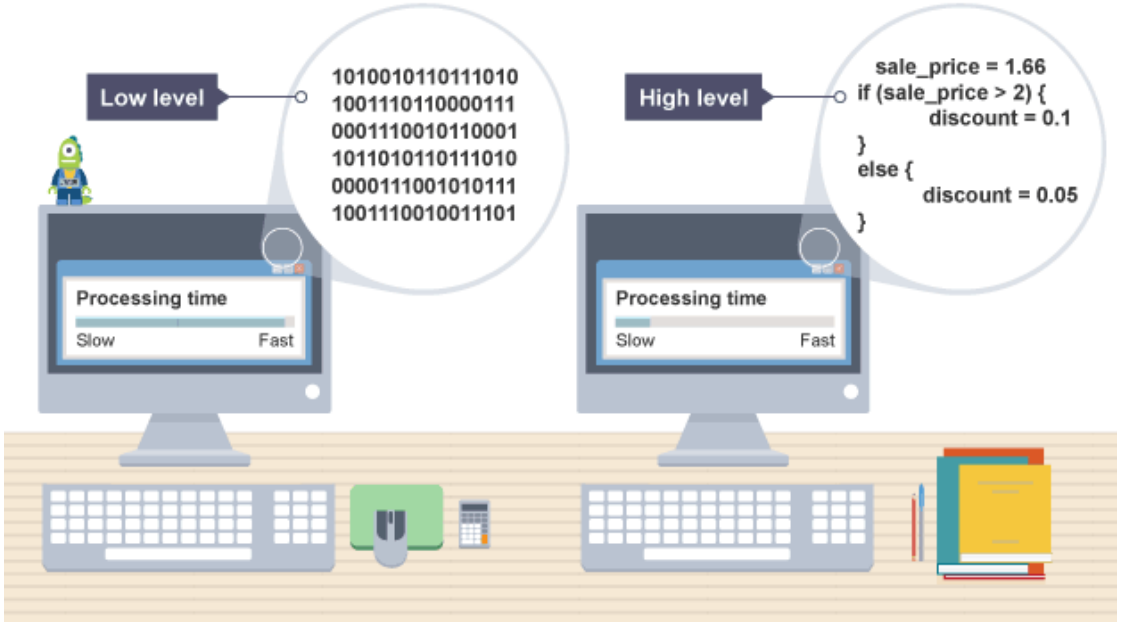
Programming Languages



Similar situation than 50 years ago:

- People used to program in low-level programming languages.

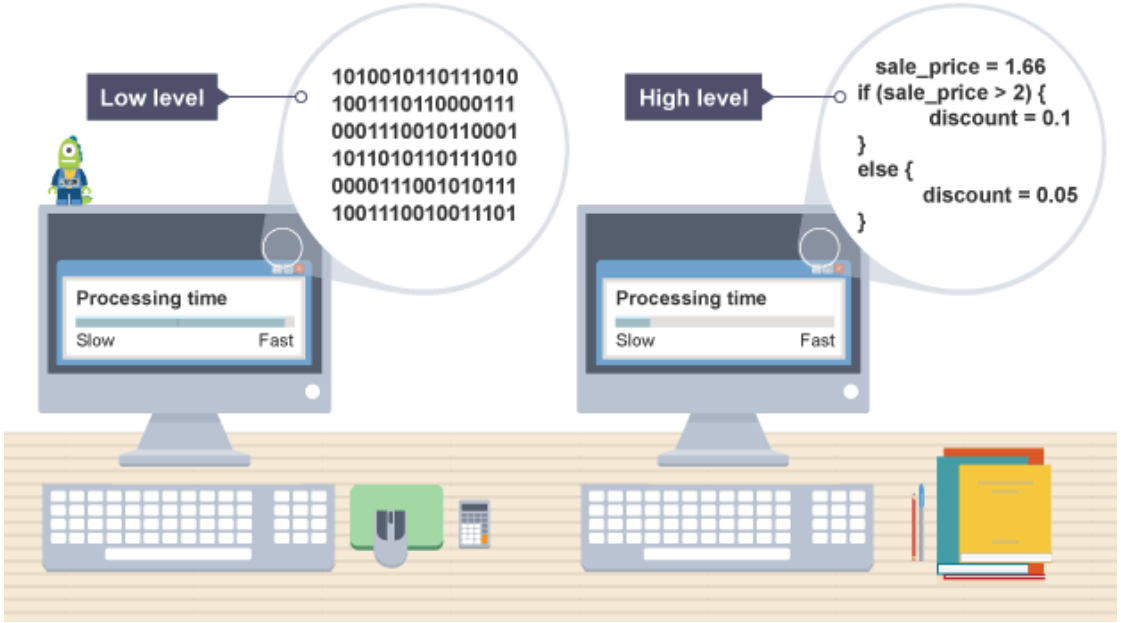
Programming Languages



Similar situation than 50 years ago:

- People used to program in low-level programming languages.
- Programming was complex and demand high-expertise.

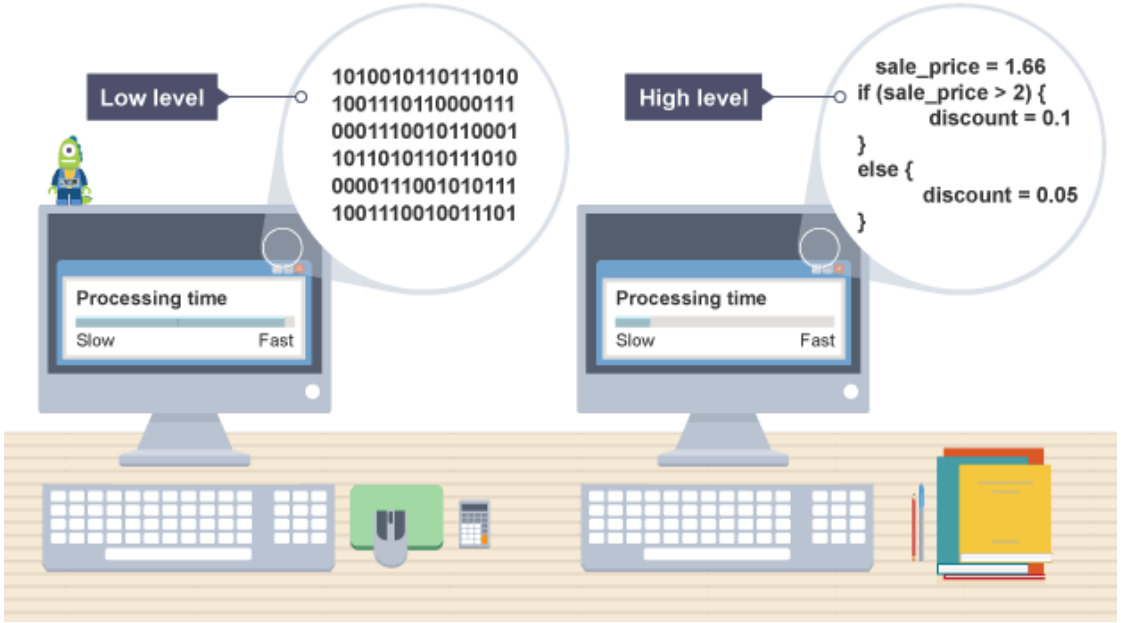
Programming Languages



Similar situation than 50 years ago:

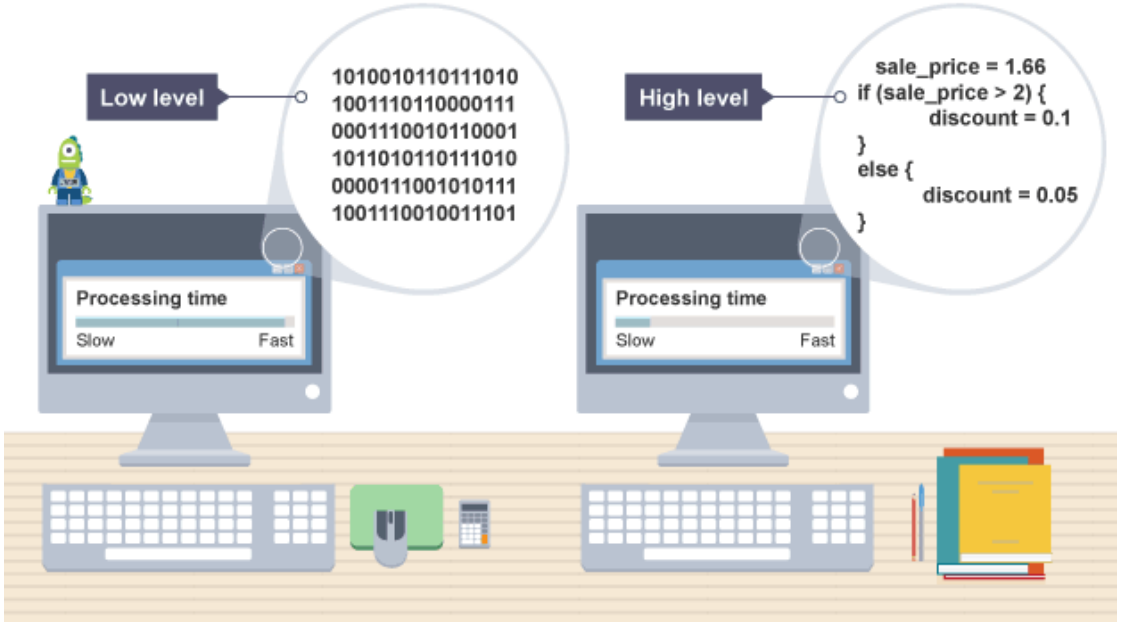
- People used to program in low-level programming languages.
- Programming was complex and demand high-expertise.
- Focus on application and low-level hardware details.

Programming Languages



High-level programming languages brought many advantages:

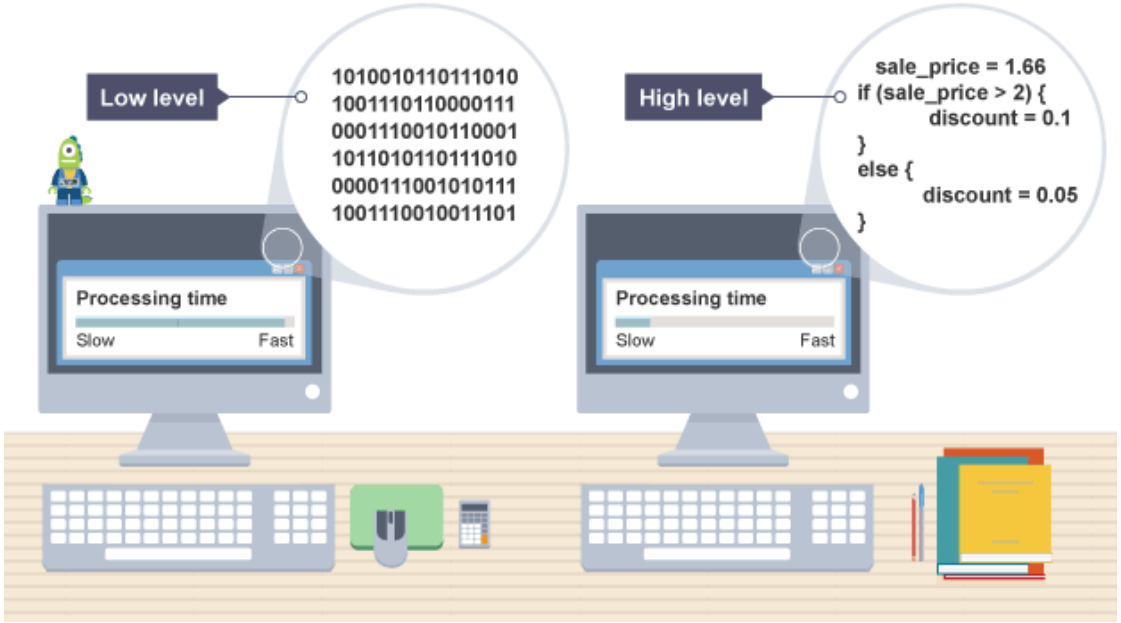
Programming Languages



High-level programming languages brought many advantages:

- Programmers focused on the applications.

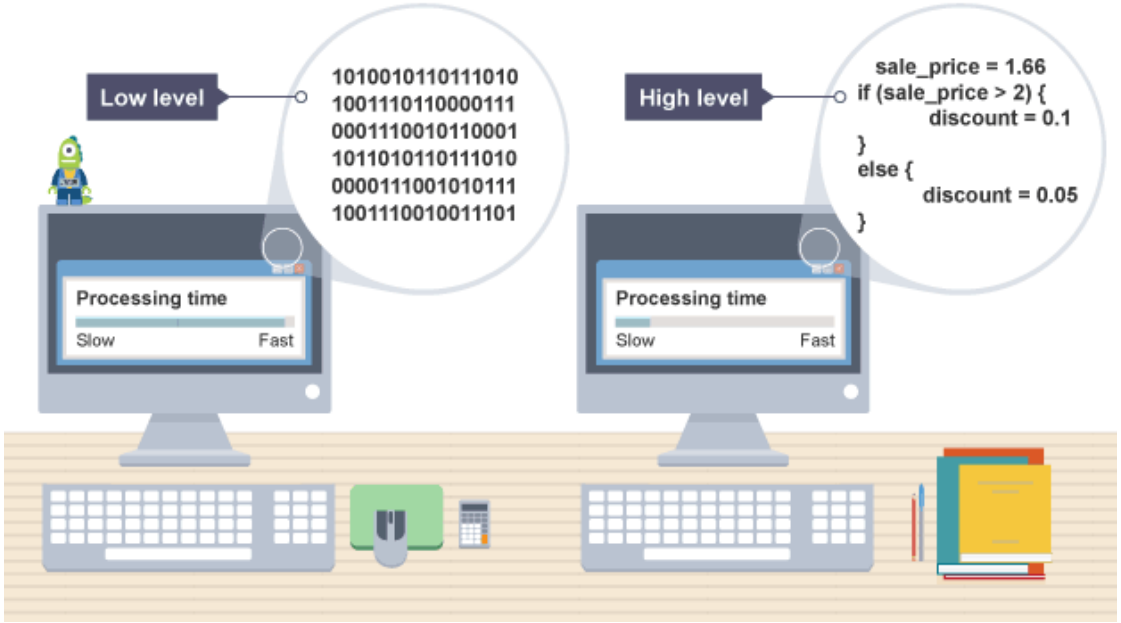
Programming Languages



High-level programming languages brought many advantages:

- Programmers focused on the applications.
- Hardware Experts focused on compilers.

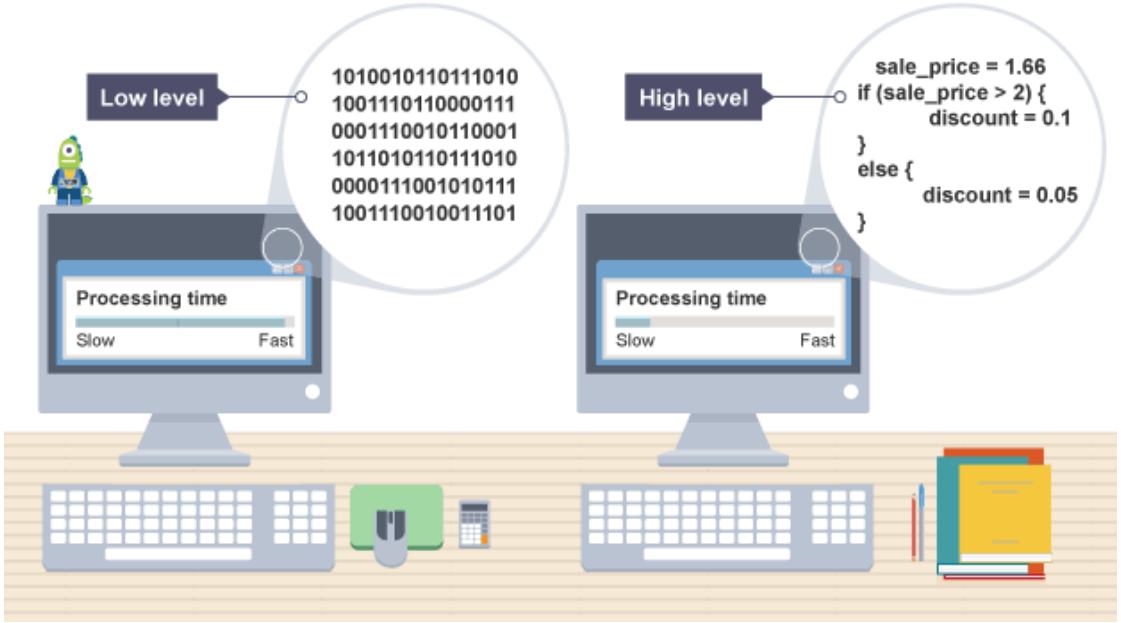
Programming Languages



High-level programming languages brought many advantages:

- Programmers focused on the applications.
- Hardware Experts focused on compilers.
- High gains in productivity.

Programming Languages



High-level programming languages brought many advantages:

- Programmers focused on the applications.
- Hardware Experts focused on compilers.
- High gains in productivity.
- “Democratization” of the software development.

Big Data and Machine Learning Ecosystem

Claire D. Costa. Best Python Libraries for Machine Learning and Deep Learning.
https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c

Big Data and Machine Learning Libraries:

- **High-quality**, well-maintained and open-source libraries

Big Data and Machine Learning Ecosystem

Claire D. Costa. Best Python Libraries for Machine Learning and Deep Learning.
https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c

Big Data and Machine Learning Libraries:

- **High-quality**, well-maintained and open-source libraries
- They try to provide **high-level abstractions**.

Big Data and Machine Learning Ecosystem

Claire D. Costa. Best Python Libraries for Machine Learning and Deep Learning.
https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c

Big Data and Machine Learning Libraries:

- **High-quality**, well-maintained and open-source libraries
- They try to provide **high-level abstractions**.
- Hiding under the hood **low level details**.

Big Data and Machine Learning Ecosystem

Claire D. Costa. Best Python Libraries for Machine Learning and Deep Learning.
https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c

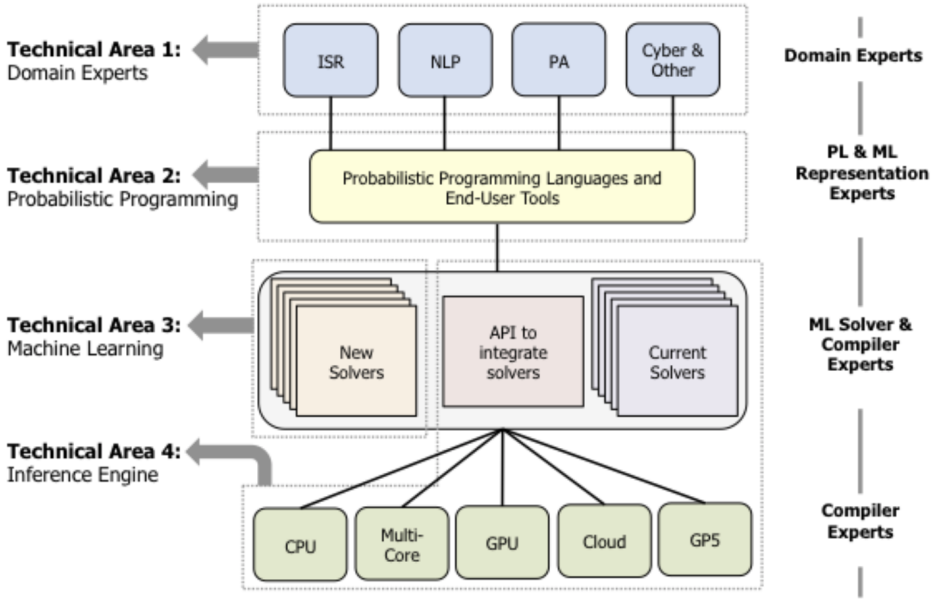
Big Data and Machine Learning Libraries:

- **High-quality**, well-maintained and open-source libraries
- They try to provide **high-level abstractions**.
- Hiding under the hood **low level details**.
- Increase the **adoption** of these technologies.

What about in Probabilistic AI?

Which are the "high-level libraries" in Probabilistic AI?

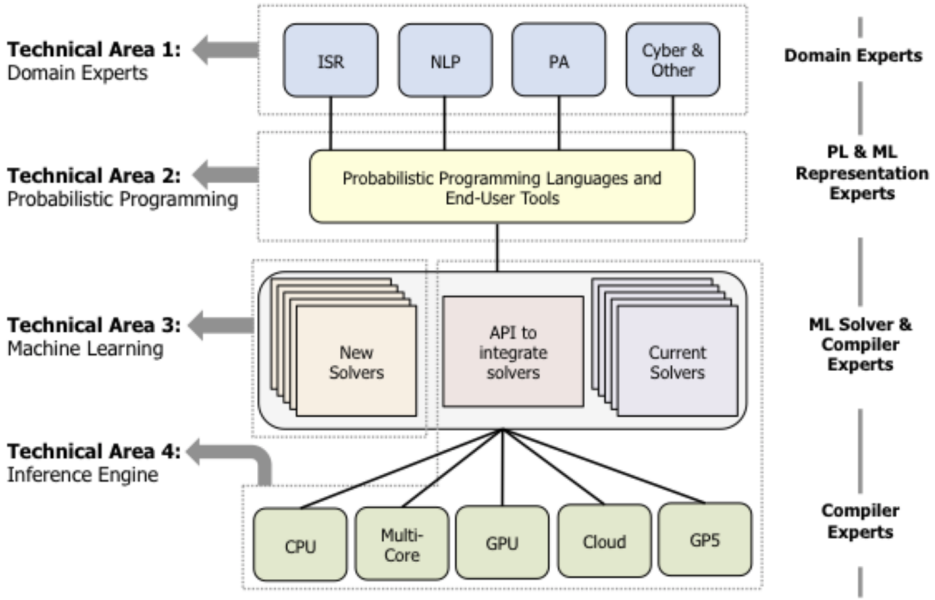
Why PPLs?



PPLs as high-level programming languages for **probabilistic machine learning systems**:

- Stacked architecture

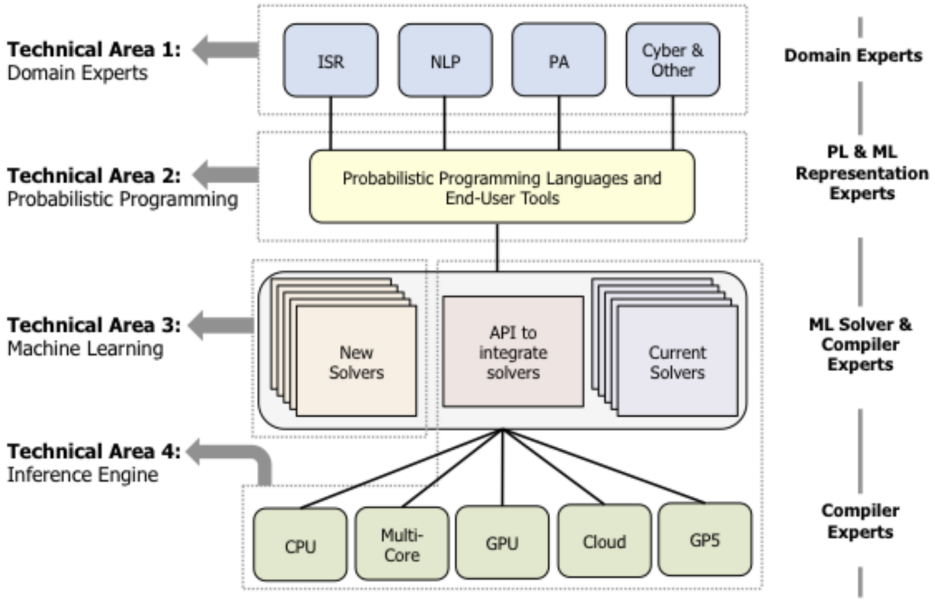
Why PPLs?



PPLs as high-level programming languages for **probabilistic machine learning systems**:

- Stacked architecture
- Different Domain Experts will code their models using the same language.

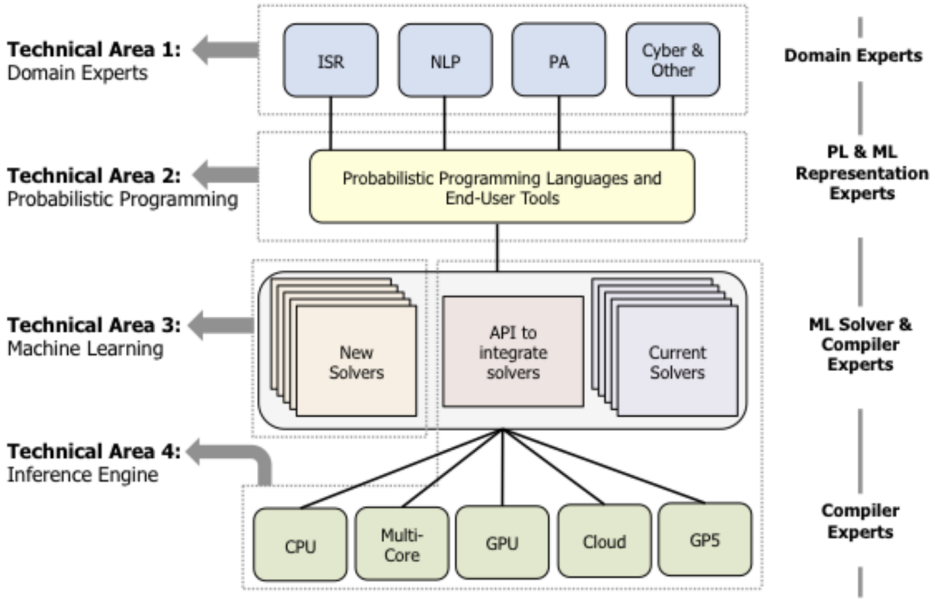
Why PPLs?



PPLs as high-level programming languages for **probabilistic machine learning systems**:

- Stacked architecture
- Different Domain Experts will code their models using the same language.
- ML experts will focus on the development of new ML solvers.

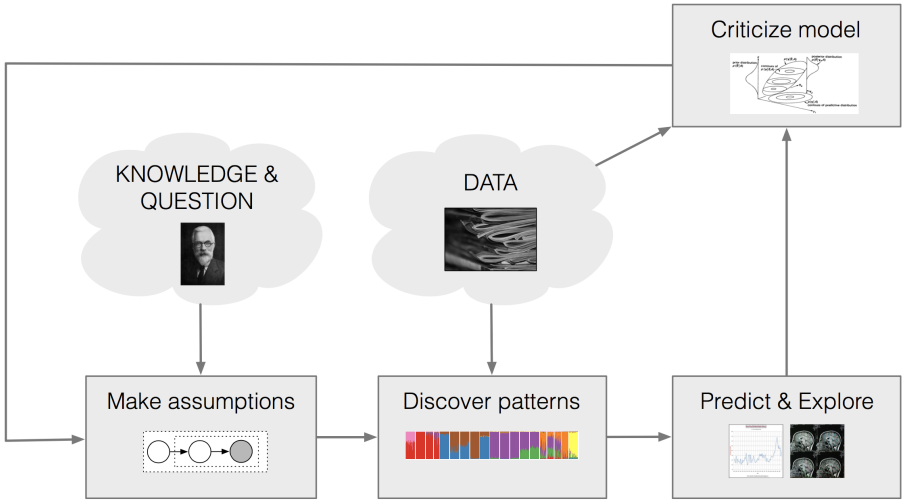
Why PPLs?



PPLs as high-level programming languages for **probabilistic machine learning systems**:

- Stacked architecture
- Different Domain Experts will code their models using the same language.
- ML experts will focus on the development of new ML solvers.
- Compile experts will focus on running these ML solvers on specialized hardware.

Box's Loop

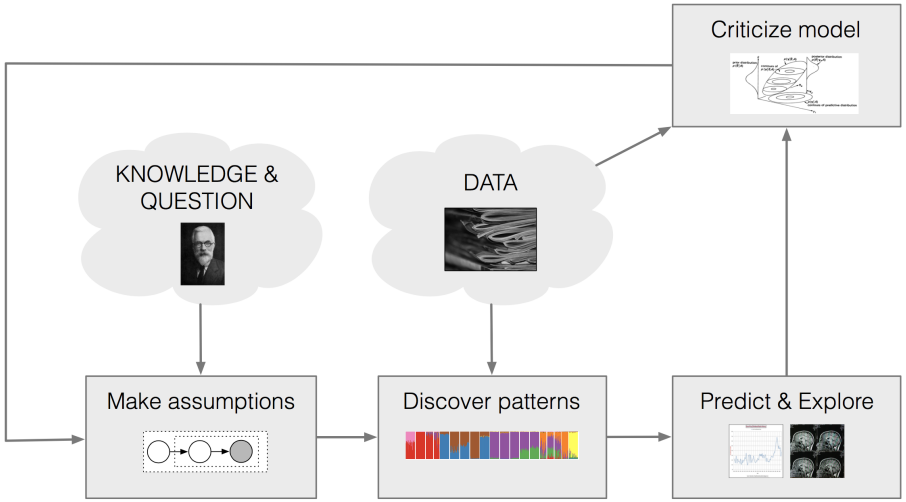


[Box, 1980; Rubin, 1984; Gelman+ 1996; Blei, 2014]

Benefits of PPLs for developing probabilistic machine learning systems:

- Simplify probabilistic machine learning model code.

Box's Loop

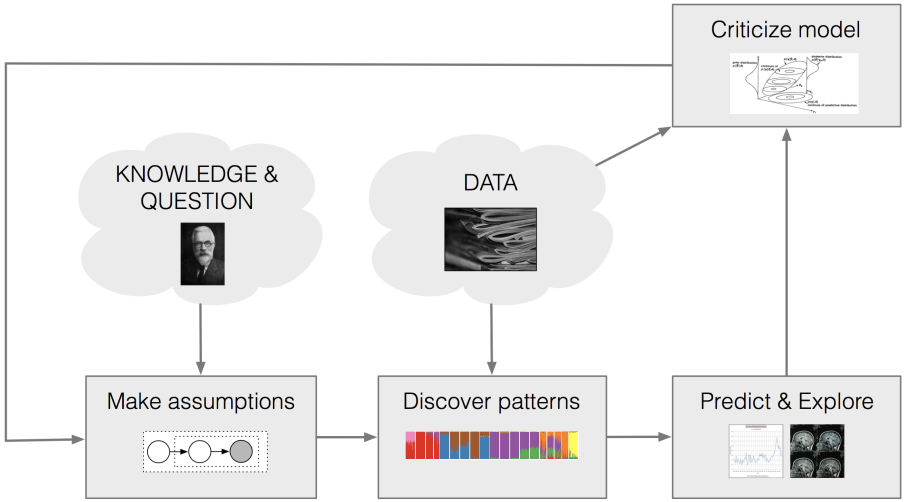


[Box, 1980; Rubin, 1984; Gelman+ 1996; Blei, 2014]

Benefits of PPLs for developing probabilistic machine learning systems:

- Simplify probabilistic machine learning model code.
- Reduce development time and cost to encourage experimentation.

Box's Loop

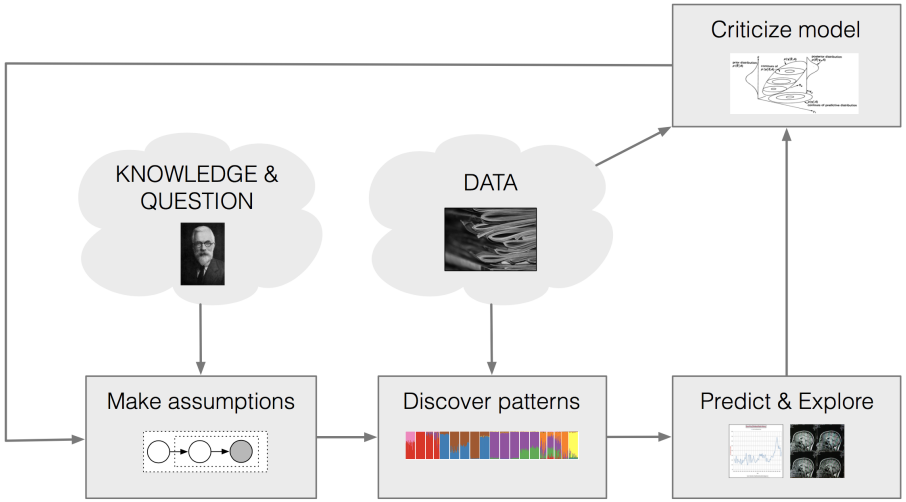


[Box, 1980; Rubin, 1984; Gelman+ 1996; Blei, 2014]

Benefits of PPLs for developing probabilistic machine learning systems:

- Simplify probabilistic machine learning model code.
- Reduce development time and cost to encourage experimentation.
- Reduce the necessary level of expertise.

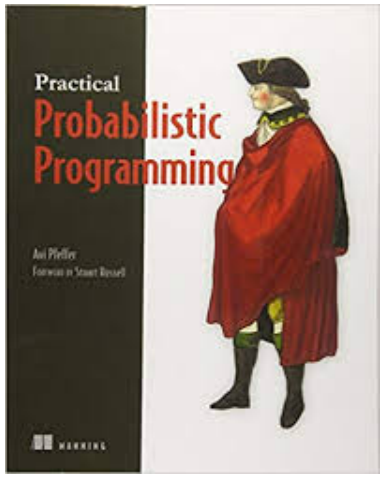
Box's Loop



[Box, 1980; Rubin, 1984; Gelman+ 1996; Blei, 2014]

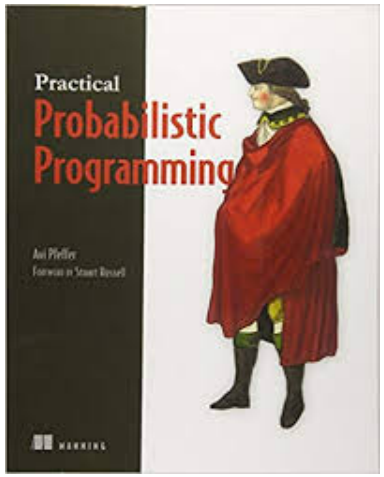
Benefits of PPLs for developing probabilistic machine learning systems:

- Simplify probabilistic machine learning model code.
- Reduce development time and cost to encourage experimentation.
- Reduce the necessary level of expertise.
- "Democratization" of the development of probabilistic ML systems.



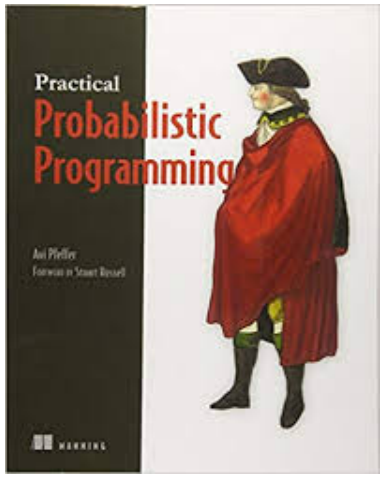
1st Generation of PPLs :

- Bugs, WinBugs, Jags, Figaro, etc.



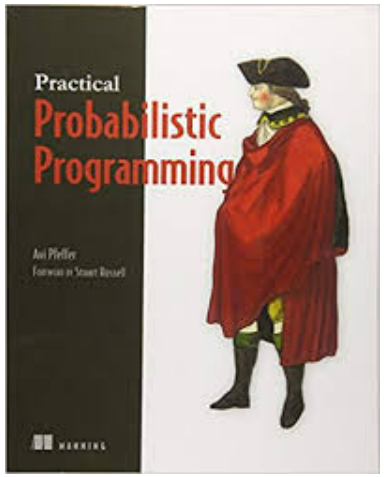
1st Generation of PPLs :

- Bugs, WinBugs, Jags, Figaro, etc.
- Turing-complete probabilistic programming languages. (i.e. they can represent any computable probability distribution).



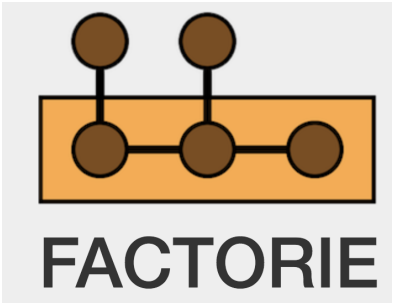
1st Generation of PPLs :

- Bugs, WinBugs, Jags, Figaro, etc.
- Turing-complete probabilistic programming languages. (i.e. they can represent any computable probability distribution).
- Inference engine based on Monte Carlo methods.



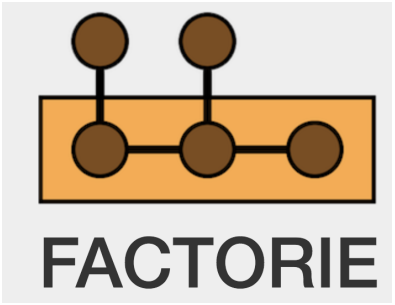
1st Generation of PPLs :

- Bugs, WinBugs, Jags, Figaro, etc.
- Turing-complete probabilistic programming languages. (i.e. they can represent any computable probability distribution).
- Inference engine based on Monte Carlo methods.
- They did not scale to large data samples/high-dimensional models.



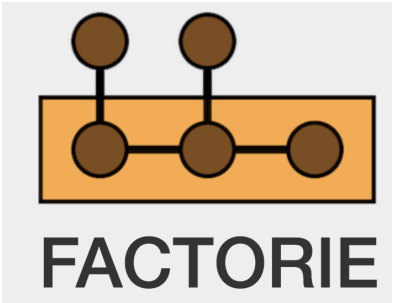
2nd Generation of PPLs :

- Infer.net, Factorie, Amidst, etc.



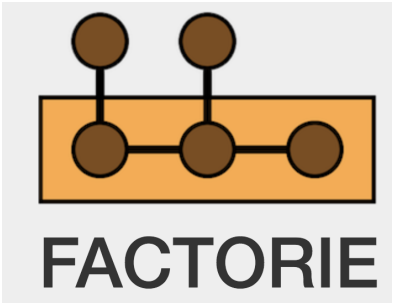
2nd Generation of PPLs :

- Infer.net, Factorie, Amidst, etc.
- Inference engine based on message passage algorithms and/or variational inference methods.



2nd Generation of PPLs :

- Infer.net, Factorie, Amidst, etc.
- Inference engine based on message passage algorithms and/or variational inference methods.
- They did scale to large data samples/high-dimensional models.



2nd Generation of PPLs :

- Infer.net, Factorie, Amidst, etc.
- Inference engine based on message passage algorithms and/or variational inference methods.
- They did scale to large data samples/high-dimensional models.
- Restricted probabilistic model family (i.e. factor graphs, conjuage exponential family, etc.)

3rd Generation of PPLs :

- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.

3rd Generation of PPLs :

- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.

3rd Generation of PPLs :

- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.
- They did scale to large data samples/high-dimensional models.

3rd Generation of PPLs :

- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.
- They did scale to large data samples/high-dimensional models.
- Turing-complete probabilistic programming languages.

3rd Generation of PPLs :

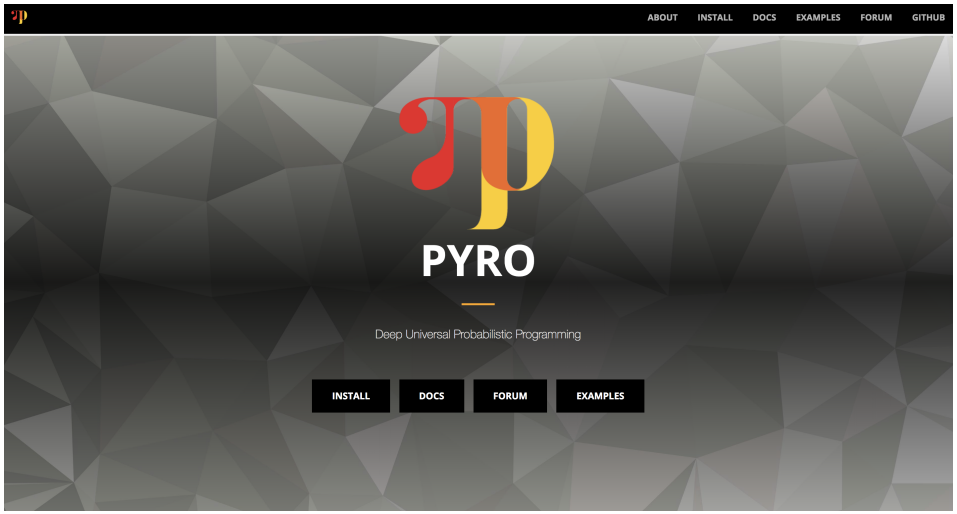
- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.
- They did scale to large data samples/high-dimensional models.
- Turing-complete probabilistic programming languages.
- Rely on deep learning frameworks (TensorFlow, Pytorch, Theano, etc).

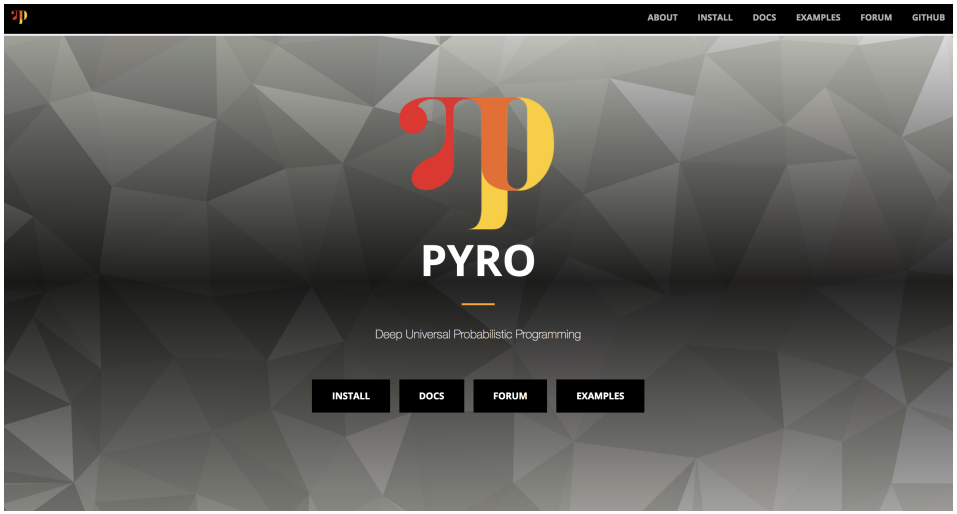
3rd Generation of PPLs :

- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.
- They did scale to large data samples/high-dimensional models.
- Turing-complete probabilistic programming languages.
- Rely on deep learning frameworks (TensorFlow, Pytorch, Theano, etc).
  - Specialized hardware like GPUs, TPUs, etc.

3rd Generation of PPLs :

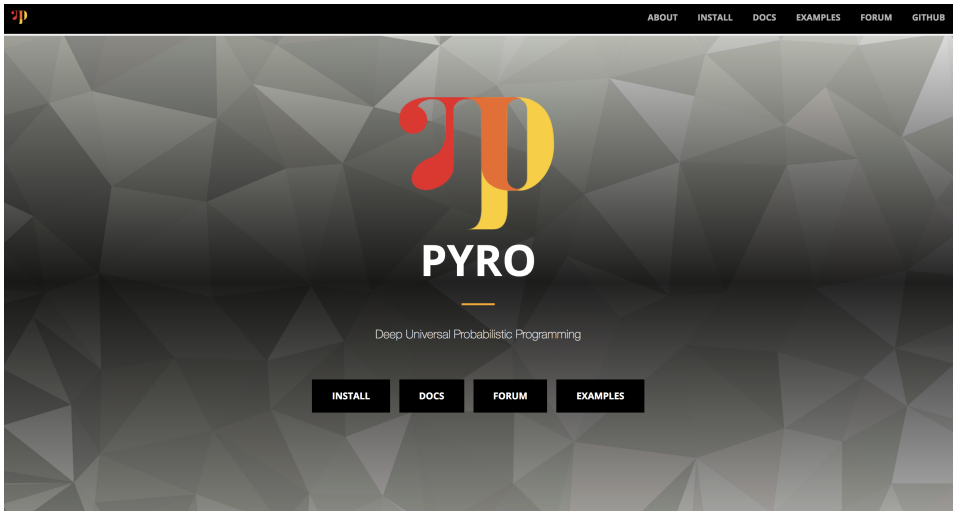
- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.
- They did scale to large data samples/high-dimensional models.
- Turing-complete probabilistic programming languages.
- Rely on deep learning frameworks (TensorFlow, Pytorch, Theano, etc).
  - Specialized hardware like GPUs, TPUs, etc.
  - Automatic differentiation methods.


ABOUT
INSTALL
DOCS
EXAMPLES
FORUM
GITHUB
PYRO
Deep Universal Probabilistic Programming
INSTALL
DOCS
FORUM
EXAMPLES



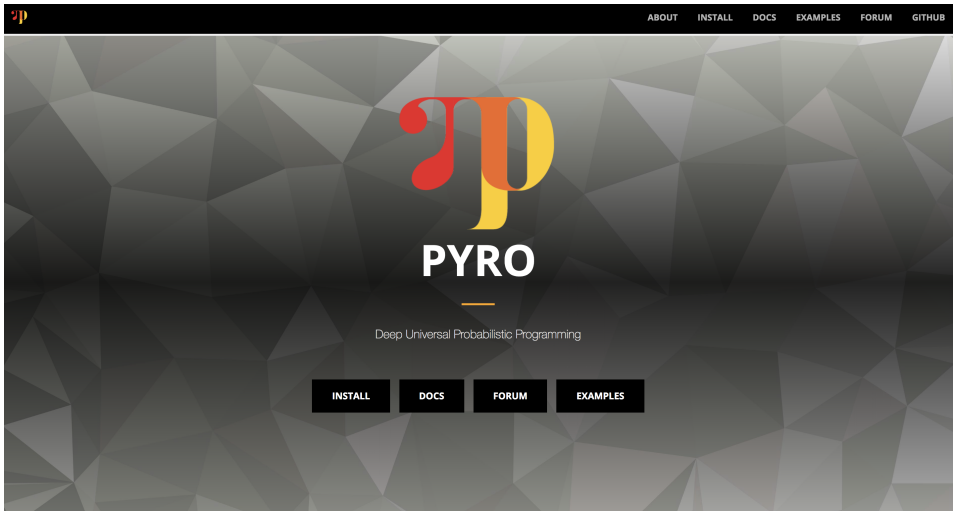
Pyro's main features (`www.pyro.ai`) :

- Initially developed by UBER (the car riding company).
- Actively maintained by community contributors (there is dedicated team at Broad Institute)



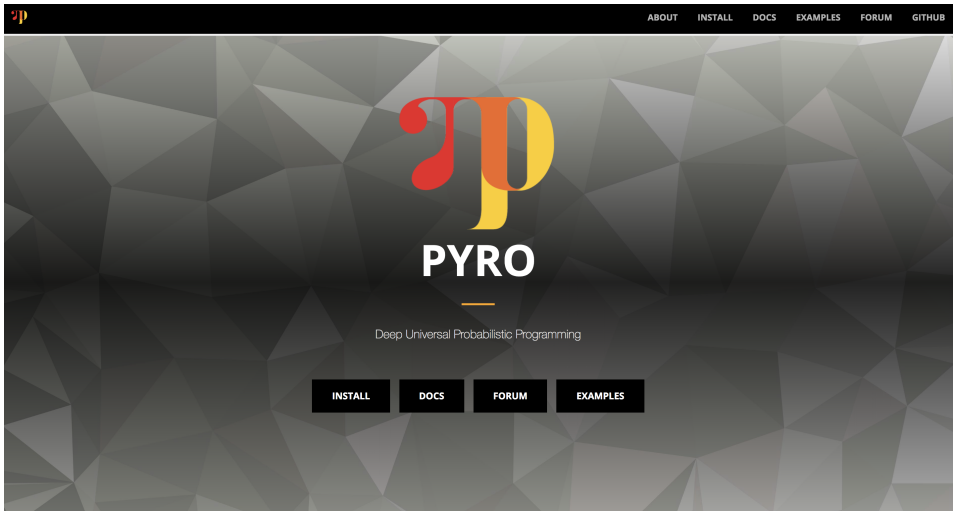
Pyro's main features (`www.pyro.ai`) :

- Initially developed by UBER (the car riding company).
- Actively maintained by community contributors (there is dedicated team at Broad Institute)
- Rely on Pytorch (Deep Learning Framework).



Pyro's main features (`www.pyro.ai`) :

- Initially developed by UBER (the car riding company).
- Actively maintained by community contributors (there is dedicated team at Broad Institute)
- Rely on Pytorch (Deep Learning Framework).
- Enable GPU accelaration and distributed learning.



Pyro's main features (`www.pyro.ai`) :

- Initially developed by UBER (the car riding company).
- Actively maintained by community contributors (there is dedicated team at Broad Institute)
- Rely on Pytorch (Deep Learning Framework).
- Enable GPU accelaration and distributed learning.

`https://github.com/PGM-Lab/2022-ProbAI`