

$$\frac{dS}{dt} = T_{b,op} (N - N_{eff}) (1 - \epsilon S) S + \frac{I_{b,N}}{T_m} - \frac{S}{T_p}$$

$$\frac{S}{P} = \frac{T_p \lambda_0}{T_{act} \eta_{mc}} = \text{②}$$

$$\left[\begin{array}{c} S \\ \epsilon \end{array} \right]$$

$$N = N$$

$$P_f = (m$$

Beyond Python | Machine Learning in .NET

Why Python?

- No technical reasons for not using other languages
 - C# or Java (and friends) or C++ or Go or ...
- Interpreted, high-level, general-purpose
 - Multi-paradigm language (functional, OO, imperative)
- Created not to be a *programming language*
 - No compilers, no build steps, no CI/CD
 - Instant sharing of code



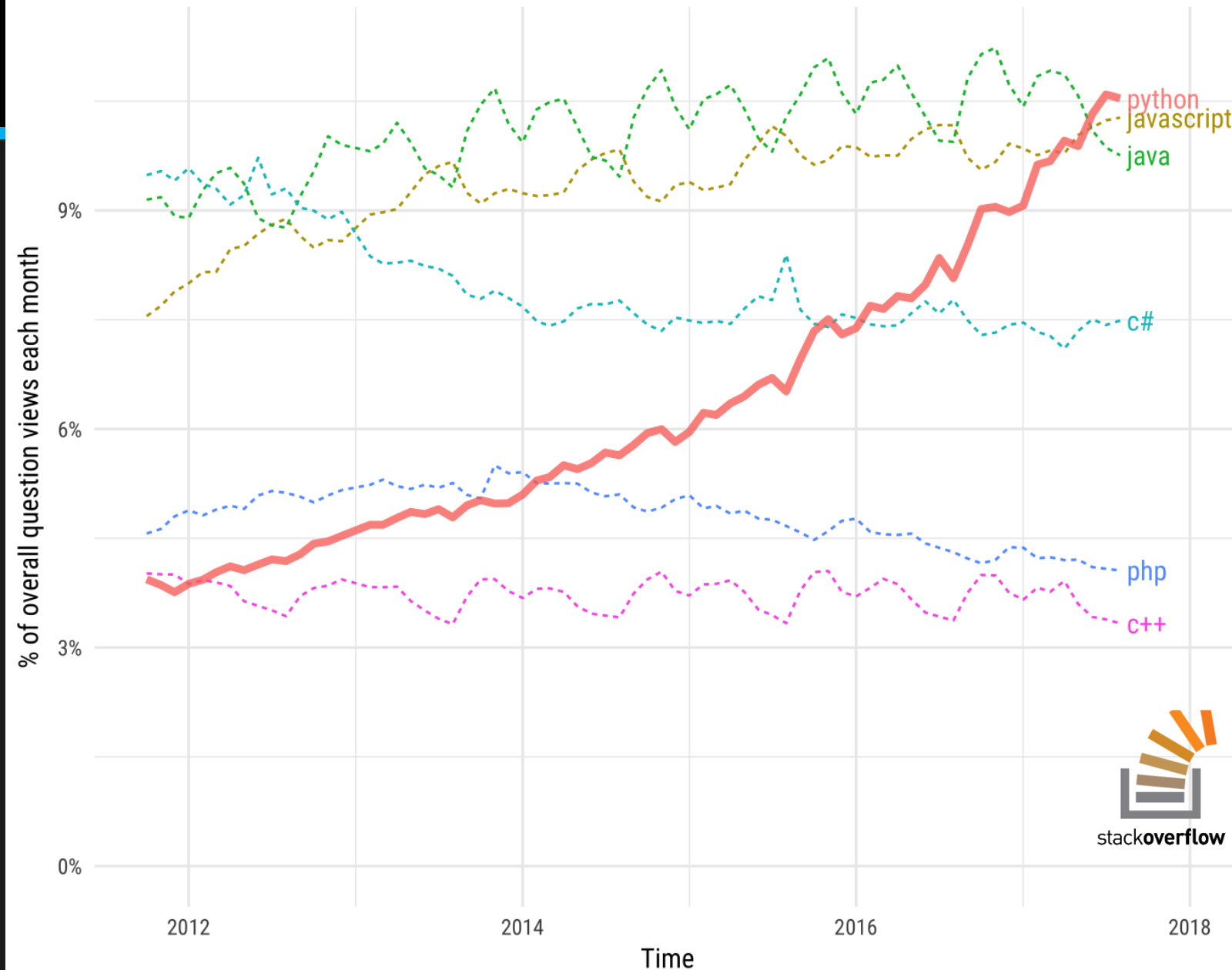
But then again ... **why Python?**

Python Facts

- Used for machine learning experiments
 - De facto standard
- Large tools ecosystem
 - Numeric computing
 - Data structures
 - Algorithms
 - Deep learning platforms

Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



But then again ... why Python?

It's a pure matter of **convenience!**

ML beyond Python

LANGUAGE	LIBRARY
Java, Scala, Clojure	DeepLearning4j
Java	Weka
C#	Accord.NET (stopped)
C++	MLPack
Go	GoML
C#, F#, VB.NET	ML.NET



More at <https://skymind.ai/wiki/java-ai>

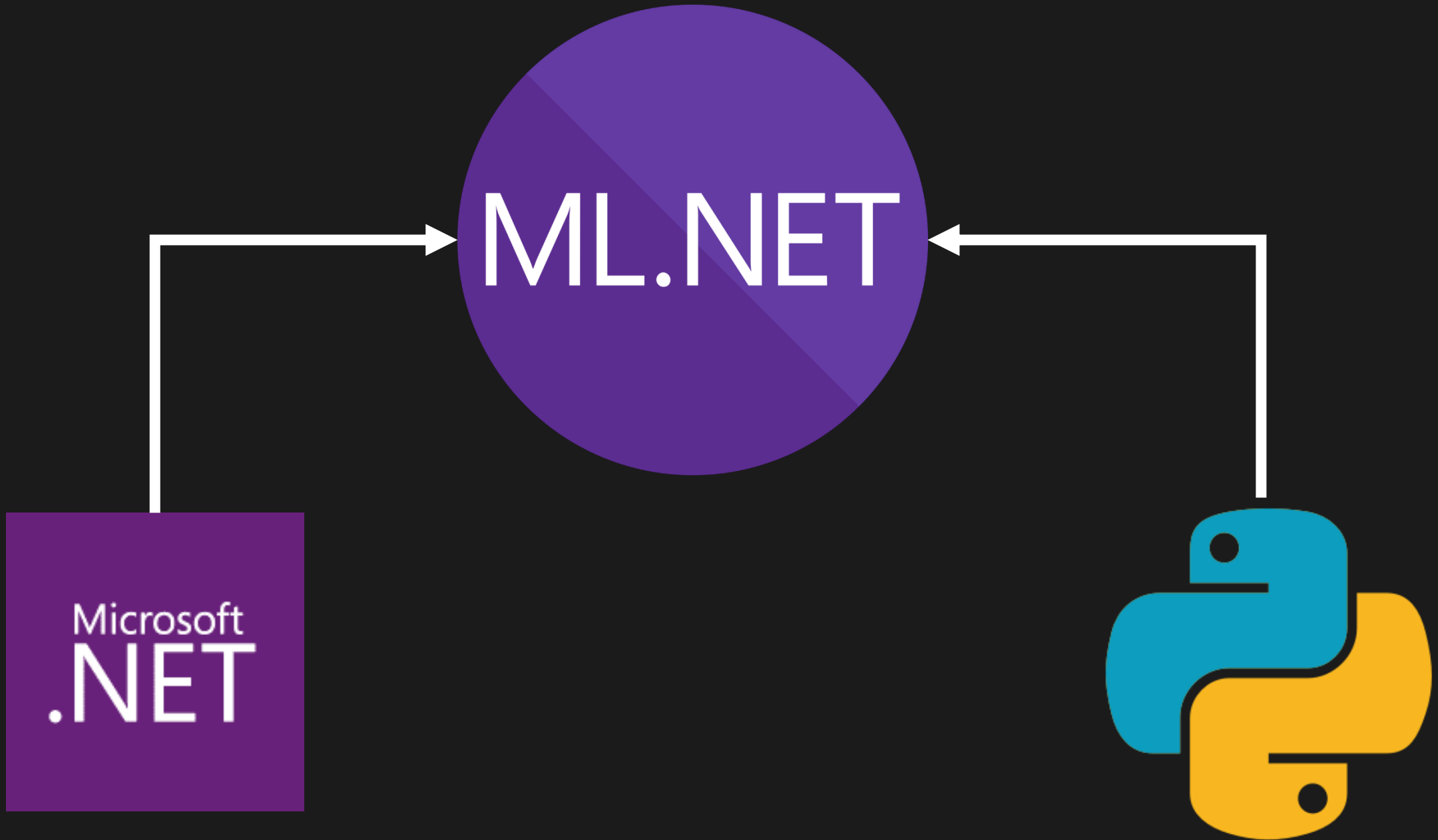
Why Should You Go Beyond Python?

Java (and C#) is the most widely used programming language in the enterprise. Large organizations have enormous Java (and C#) codebases, and rely heavily on the JVM (and CLR) as a computing environment.

- ML is perceived as a task for data scientists. Developers invoke API.
- What about **full-stack** ML experts?
- Data scientists will never reach the heights of programming ...
- ...but developers have a longer way to go to get acquainted in ML

Full-stack Machine Learning Development



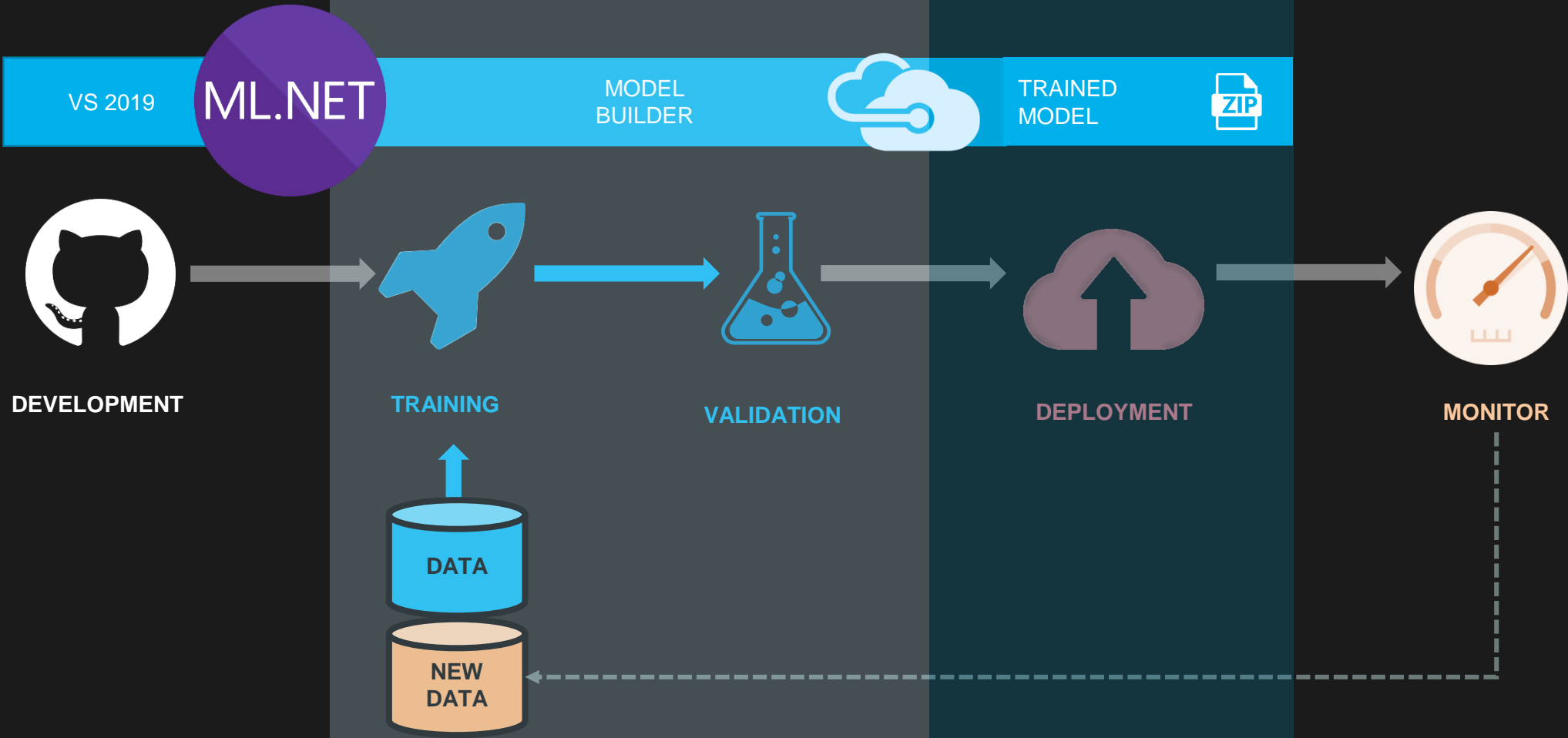


Microsoft
.NET

ML.NET



INTEGRATED WITH AZURE DEVOPS



IN ACTION

- MODEL BUILDER
- MANUAL TUNING
- AUTO-ML

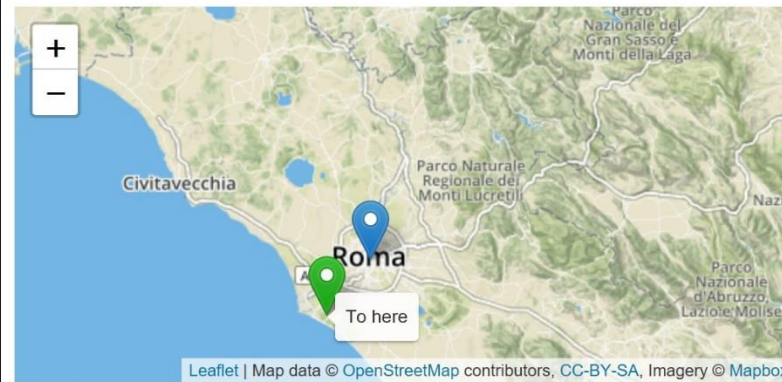


Worldwide Fare Estimator powered by ML.NET

Show that you paid more than we suggested and get the extra back. Immediately. No strings attached. ×

Where do you want to go from here? RESET

Via delle Terme Deciane 13, 00153, Roma



Destination

Pineta di Castel Fusano

21.5 KM

Car

Normal

Payment

Cash

Passengers

1

ESTIMATE

ESTIMATED FARE

\$ Calculating ...

ESTIMATED TIME

🕒

ASP.NET Core



Top5 Laws of Software at the Time of ML

- We won't write new algorithms. **Period.**
- Smarter people will do it for us (and much better). **Period.**
- *We're all moving from the Age of Programming to the Age of Training*
- Will be training other people's algorithm to serve our goals. **Period.**
 - Mostly (but not necessarily) **about development**
- Will be using quality data and properly shaped. **Period.**
 - Mostly (but not necessarily) **about data science**

SHALLOW
Learning

MULTI-LINEAR REGRESSION
MULTIPLE CLASSIFICATION
GROUP CLUSTERING

DEEP
Learning

NEURAL NETWORKS



**ONNX compatible
Frontend/Export
Frameworks**

TensorFlow
PyTorch
CNTK
Scikit-Learn
ML.NET
CoreML
...

Export
file



ONNX



Pre-trained model file

Used
by

ML.NET NuGet



.NET
Core

or

.NET
Framework

application

Predict



Conclusions

- Python is OK. **Period.**
- Python is not necessarily the way to go. **Period.**
- ML in .NET is coming and will grow. **Period.**
- *Not talking about Azure services.*
- ML solutions need a problem and a lot of domain knowledge. **Period.**
- ML solutions are a *trial-and-error* thing. **Period.**
- ML solutions are not as easy as running a wizard. **Period.**
- Not all ML solutions are a *blood-sweat-tears* thing. **Period.**

THANK
YOU!