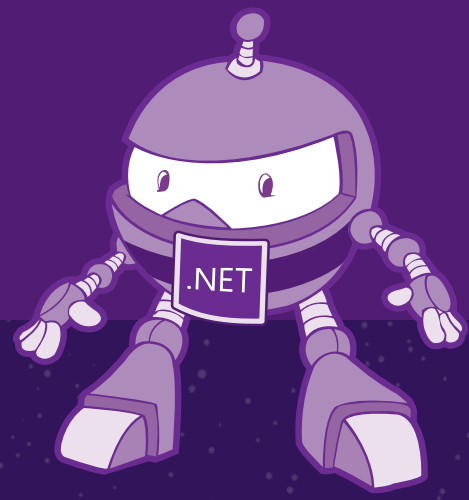

.NET Conf 2019
Discover the world of .NET
.NET
dotnet new
www.dotnetconf.net

What's new in ML.NET?

(Machine Learning for .NET)

Alejandro Giménez
/in/alejandrogiga

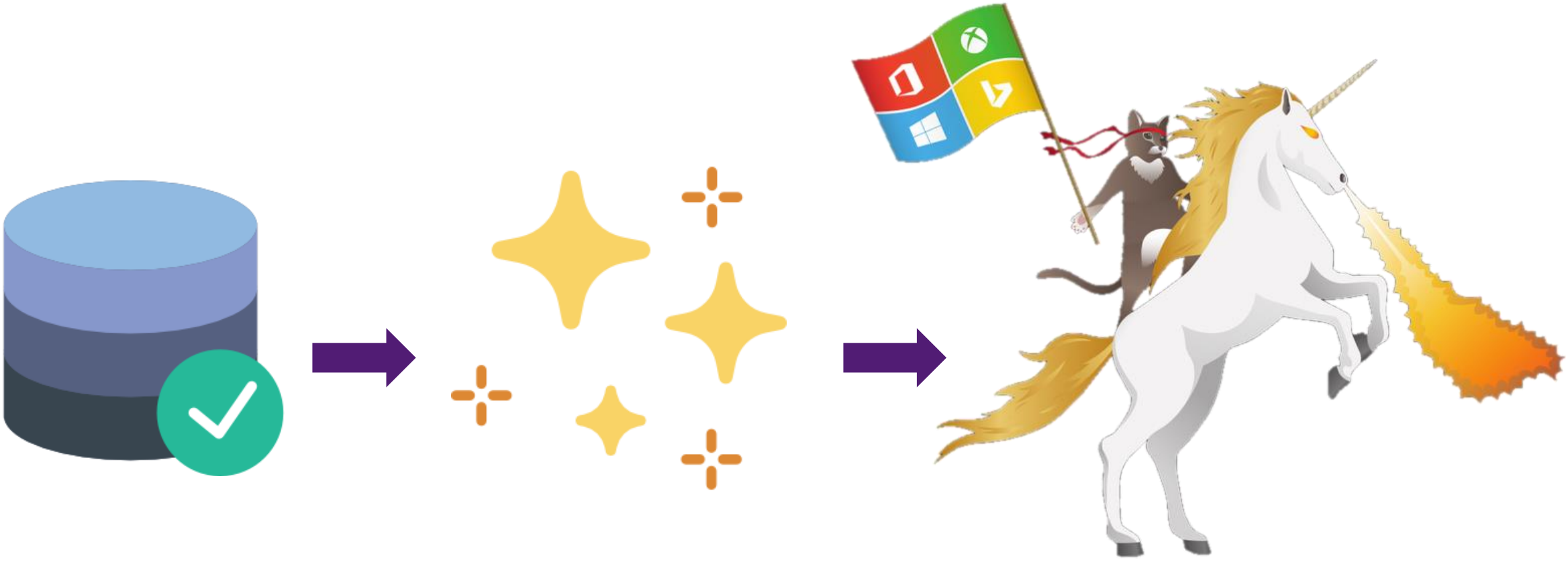
Gabriel García
/in/gabrielgar



Special thanks to:
Cesar de la Torre
@cesardelatorre

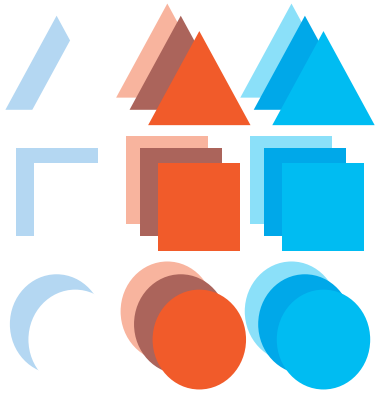
Quick intro on **ML.NET**

ML.NET...

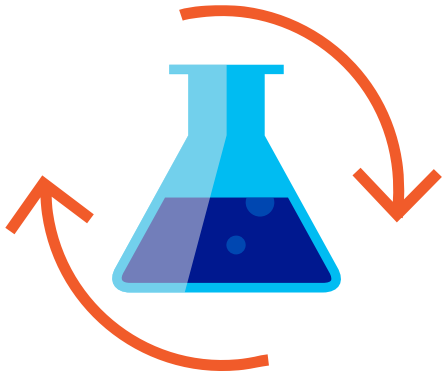


DATA **AI** **MAGIC**

Machine Learning Workflow



Datasets



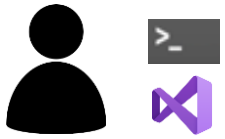
ML model

ML model

Prepare Your Data

Build & Train

Run



App/tools for training the ML model

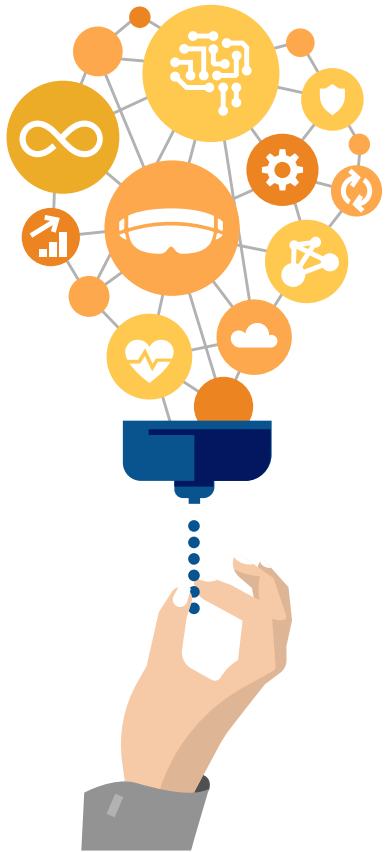
Model creation

End-user app using the ML model

Model consumption

ML.NET

An **open source** and **cross-platform** machine learning framework



C#
F#

Built for .NET developers

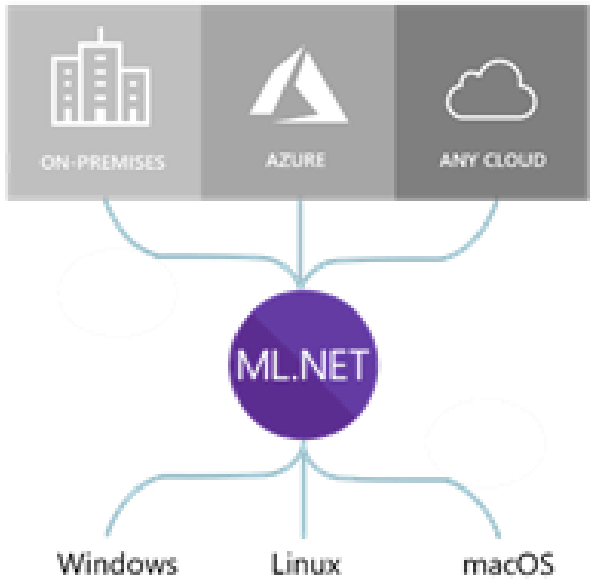
stom ML made easy with tools

Extended with TensorFlow & more

Trusted & proven at scale

http://dot.net/ml

ML.NET runs anywhere



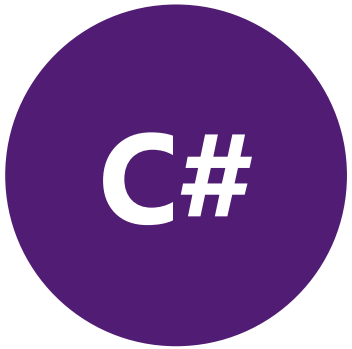
Supported Frameworks:

- .NET Core *(Natively)*
- .NET Framework *(Natively)*
- Python with *NimbusML* *(Python bindings)*

Supported processor arch.

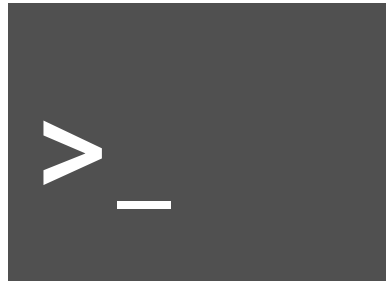
- x64
- x86

Three ways to use ML.NET…



ML.NET
API
(Code)

ML.NET
Model Builder
(Visual Studio UI)

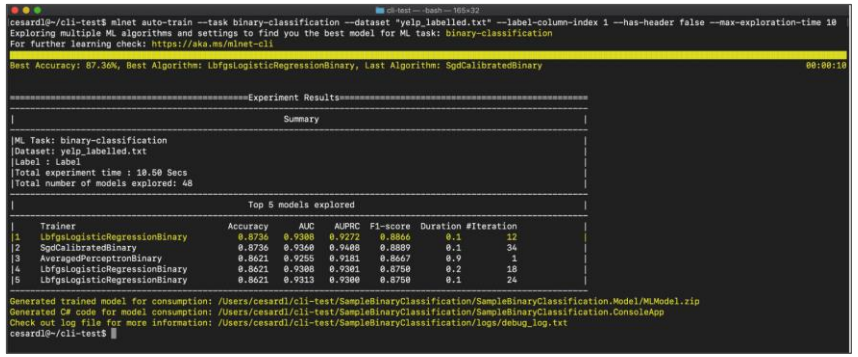


ML.NET
CLI
(Command-Line Interface)

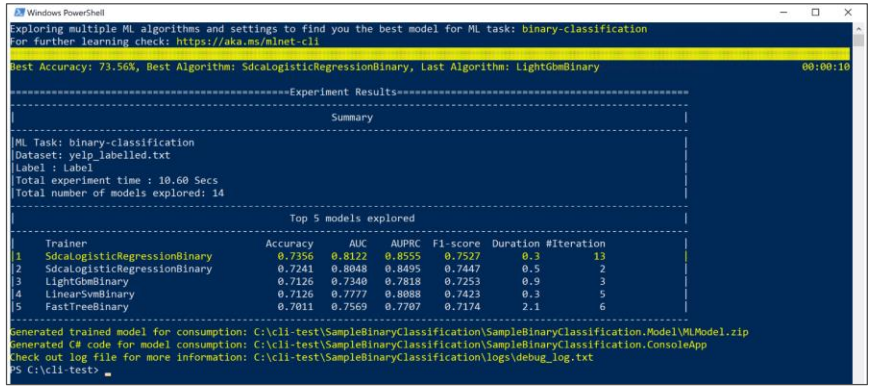
ML.NET CLI

- Use the CLI to easily build custom ML models with Automated ML
- Cross platform (Windows, Linux, MacOS)
- Generate code for training & consumption

macOS / Linux (Bash)



Windows (PowerShell and CMD)

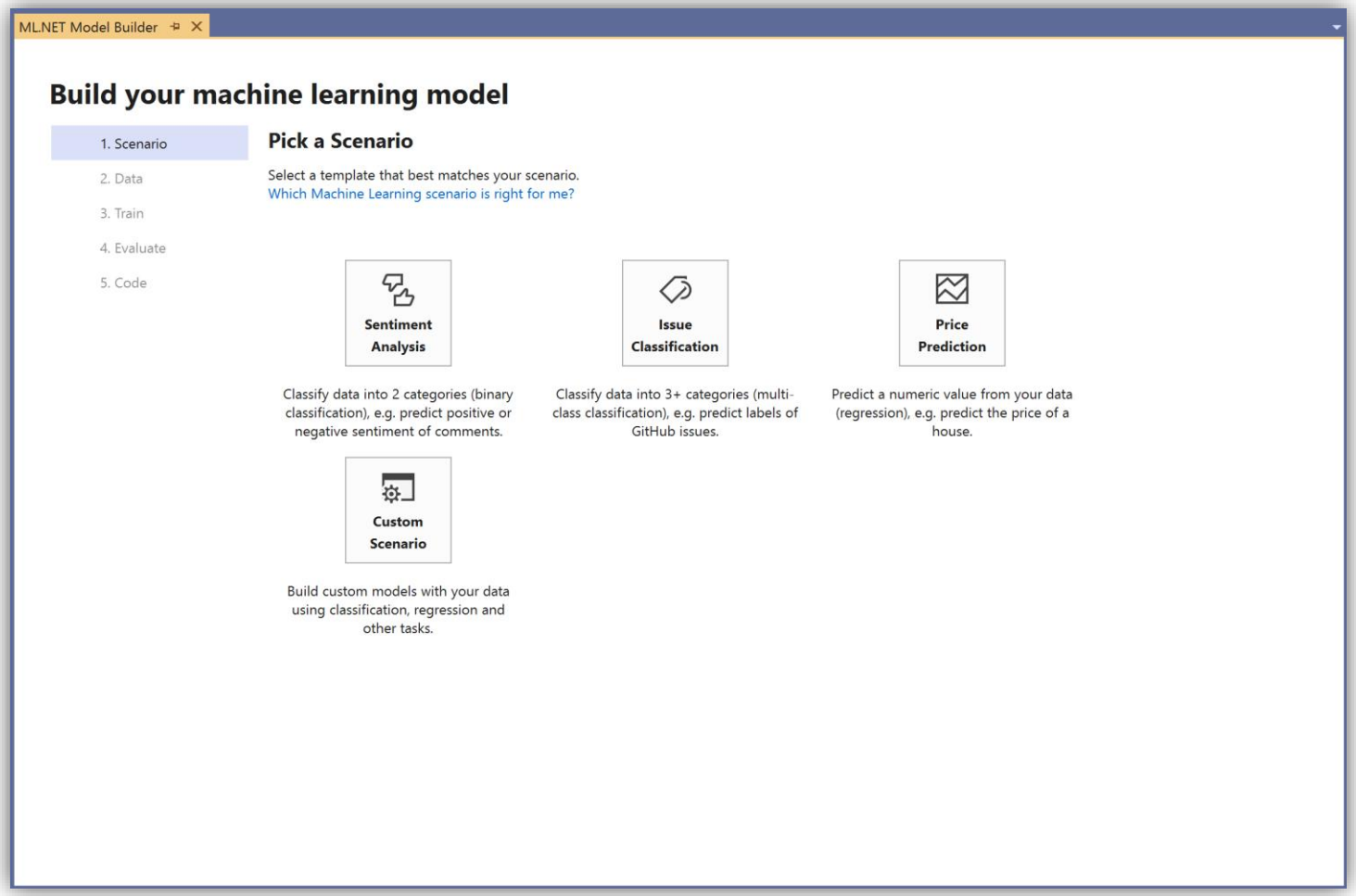



```
> mlnet auto-train --ml-task binary-classification --dataset "customer-reviews.tsv" --label-column-name Sentiment
```

ML.NET Model Builder
Approachable machine learning in Visual Studio

- A simple UI to easily build custom ML models with Automated ML
- Load from files and databases
- Generate code for training and consumption
- Run everything local

Download VS vsix:
http://aka.ms/mlnetmodelbuilder



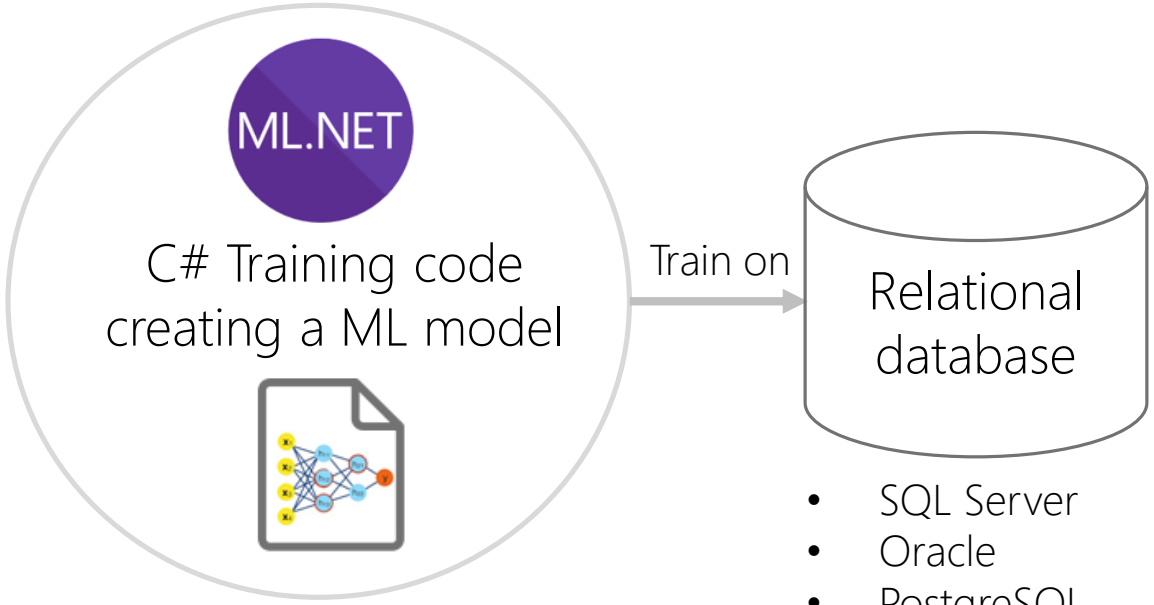
What's new for data reading: **Database Loader**

.NET

Database Loader

Enabled scenarios

- Training directly against relational databases.
- Simple and out-of-the-box code
- Supports any RDBMS supported by System.Data
- Currently in preview (1.4-preview release)

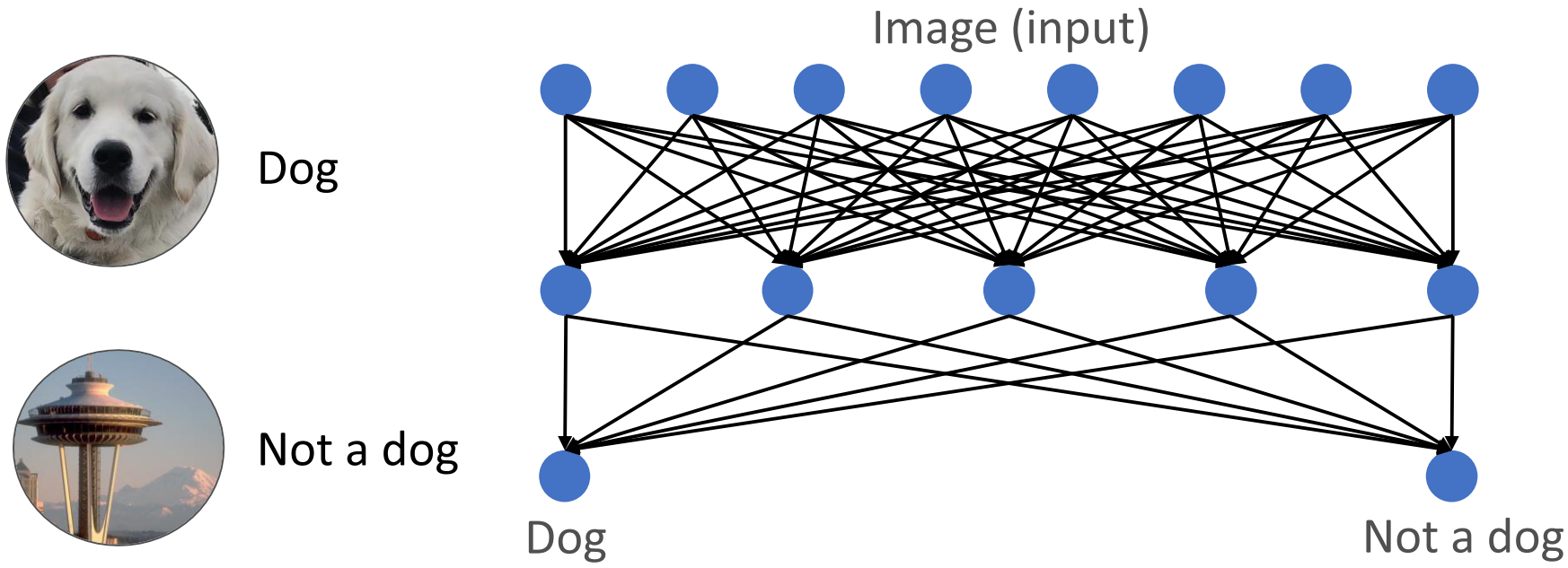


What's new for **Deep Learning in ML.NET**

.NET

Use Deep Learning models with ML.NET

- Add intelligence based on DNN-based models to your .NET apps
- Enables **computer vision** and many more Deep Learning domains



Leading deep learning 'pre-trained models'
(DNN architectures)

Computer Vision

Image classification

- Google InceptionV3
- Microsoft ResNet
- NASNet
- Oxford VGG Model
- MobileNetV2
- etc.

Object detection

- Yolo (You Only Look Once)
- R-CNN
- SPP-net
- Fast R-CNN
- Faster R-CNN
- etc.

Audio and speech

- Wavenet
- espnet
- waveblow
- deepspeech2
- loop
- tacotron
- etc.

NLP (Natural language processing)

Generative Models

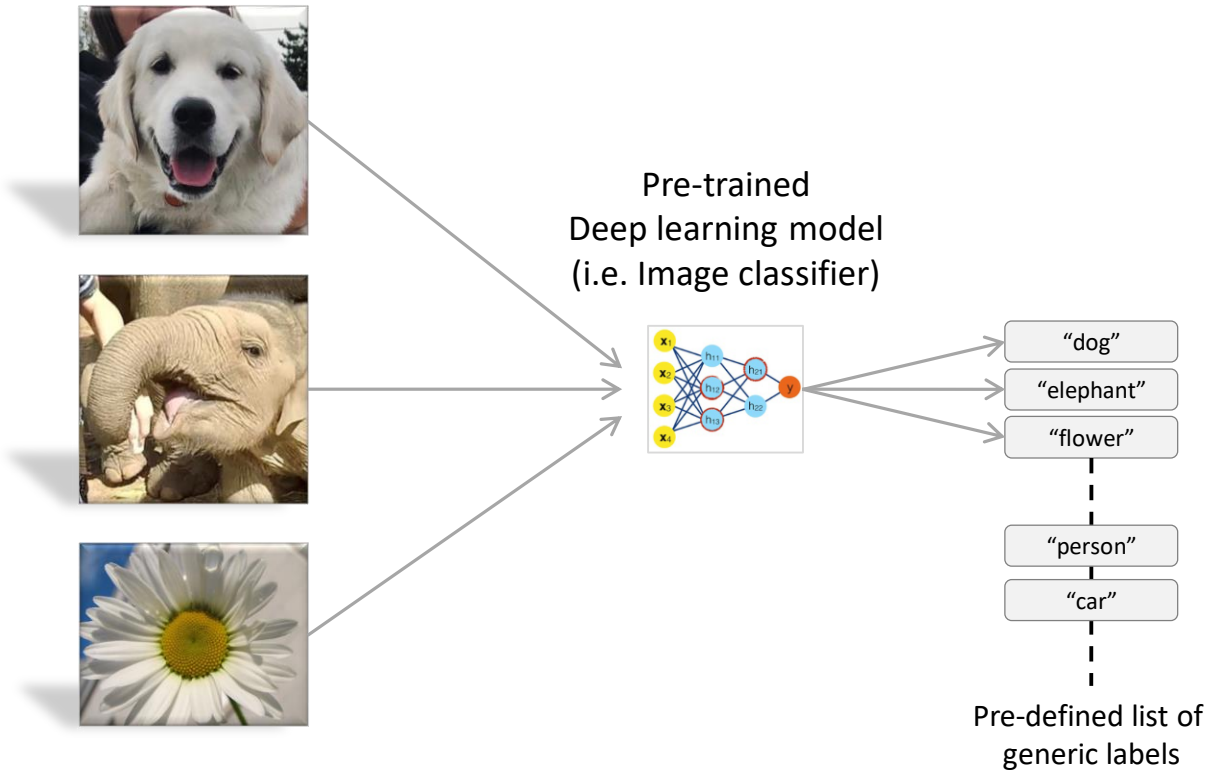
… other domains …

- Huge investment/cost in DNN architecture research plus costly training on large datasets made by organizations such as Google, Microsoft, Facebook, Universities and researchers.
- You can take advantage of it by simply consuming pre-trained models

https://modelzoo.co/categories

Consuming pre-trained deep learning models with ML.NET

Scenario: **Image classifier** (Consuming the model)



Examples of pre-trained models (Image classifiers):

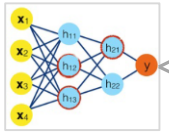
- Google **Inception v3**, **NASNet**
- Microsoft **ResNet**
- Oxford **VGG** Model, etc.

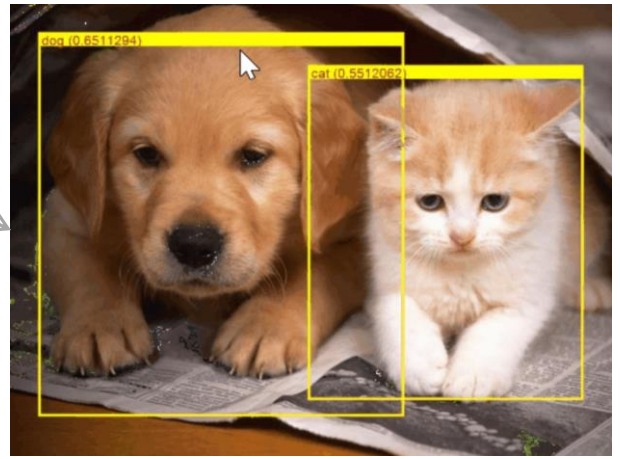
GitHub sample here

Consuming pre-trained deep learning models with ML.NET

Scenario: **Object Detection** (Consuming the model)

Pre-trained
Deep learning model
(i.e. Object Detection)





Examples of pre-trained models for obj. detection:

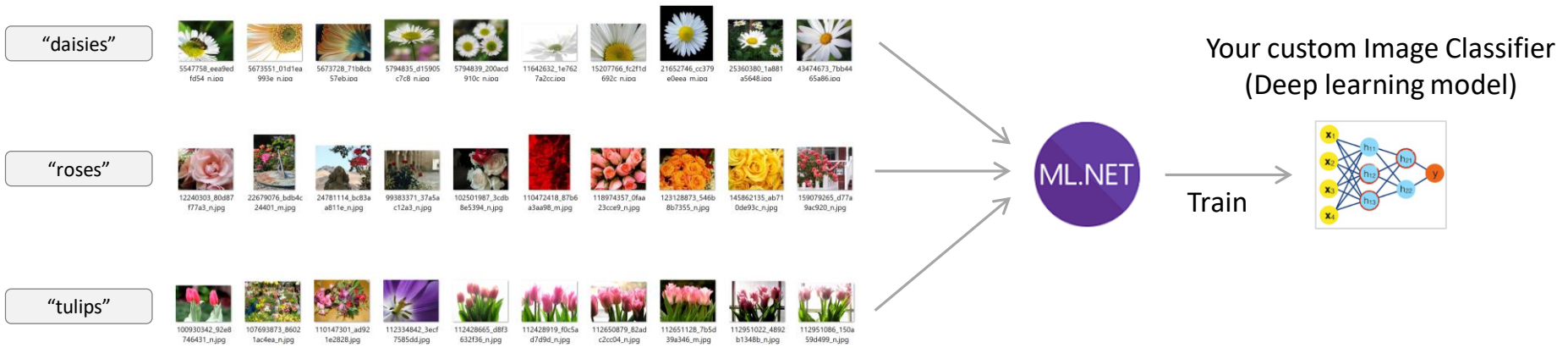
- **YOLO** (You Only Look Once)
- **Faster R-CNN**
- **SSD**, etc.

GitHub sample here

What if I want to classify based on my own custom domain?

Scenario: **Image classifier** (Training)

Then you train your own custom deep learning model with ML.NET



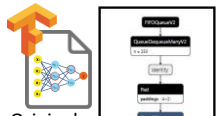
- When training, ML.NET is based on **TensorFlow** underneath (Transfer Learning).
- ML.NET currently supports "**ImageClassifier** training" (Preview)
  - **Object Detection** training will come soon

Transfer learning in TensorFlow with ML.NET **ImageClassifier** (Preview)

Simplification benefit: High level API oriented to ML Task: 'ImageClassification'

Pre-trained
TensorFlow model
DNN graph
(i.e. Inception, ResNet, etc.)

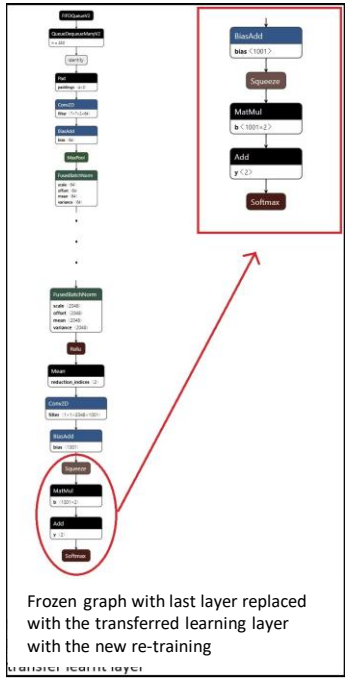
Final trained custom
TensorFlow model
DNN graph
(With Transfer learning)



Original
Pre-trained
.meta file

Original
frozen graph

Transfer
Learning
Re-train



Frozen graph with last layer replaced
with the transferred learning layer
with the new re-training

Re-trained
graph

ML.NET pipeline creating a new re-trained TensorFlow Model
through Transfer Learning within TensorFlow

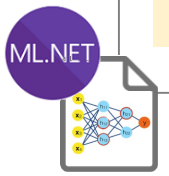
```
var pipeline = mlContext.Transforms.Conversion.MapValueToKey("Label")
    .Append(mlContext.Model.ImageClassification("Image", "Label",
                                                arch: DnnEstimator.Architecture.InceptionV3));

var trainedModel = pipeline.Fit(trainDataset);
```

New TensorFlow model
Retrained with Transfer
Learning
(.pb file)

Final
re-trained
.pb file

New ML.NET model
As simple wrapper for
execution from .NET
(.zip file)

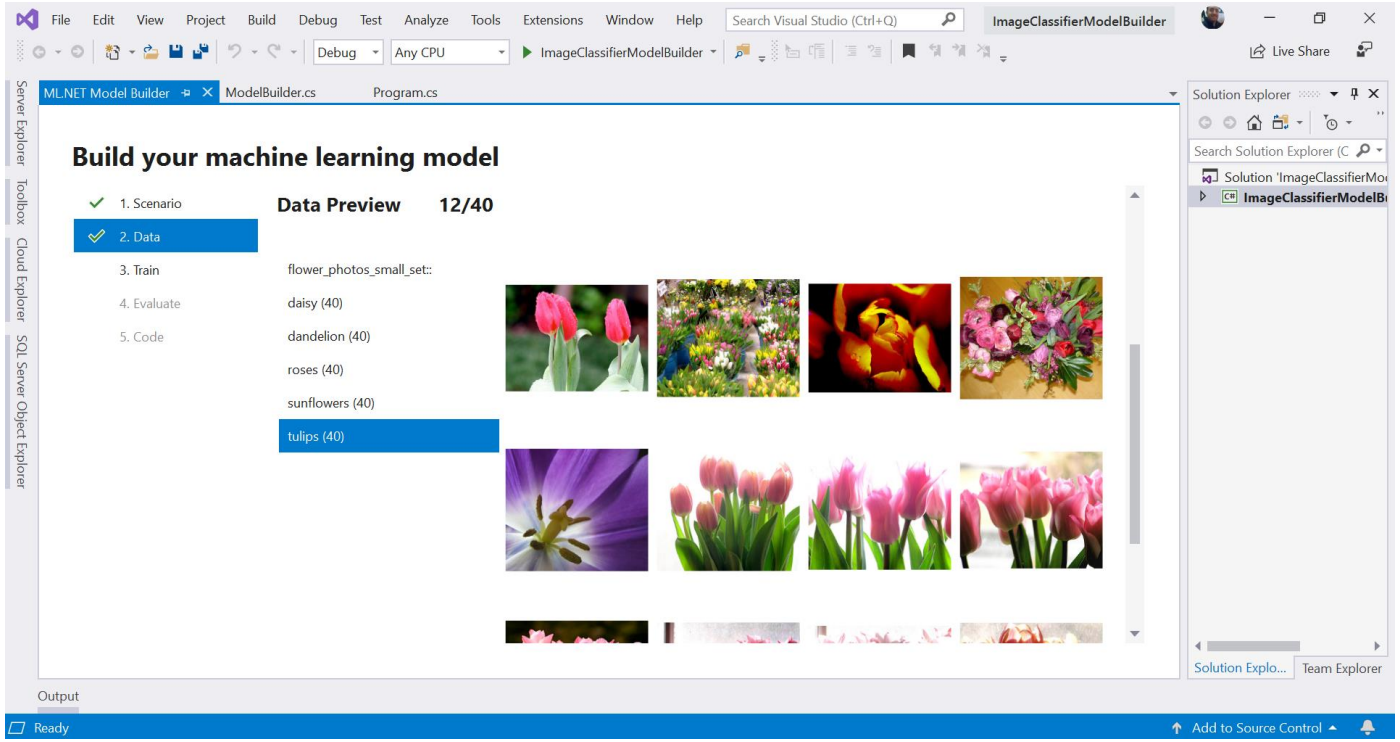


End-user C# app
Scoring/predicting
with model

Blog Post: Training Image Classification/Recognition models based on Deep Learning & Transfer Learning with ML.NET

Image Classification in ML.NET Model Builder (VS extension)

- A simple UI to easily build custom ML models with Automated ML
- Directly load image folders
- Generate code for training and consumption
- Run everything local



(*) The Image Classifier feature in Model Builder will be public preview in upcoming releases.
As of Sept. 23rd 2019 this feature is still not publicly released in Model Builder, yet.
You can already use the API in public preview, though.

More things you can do with ML.NET ...

Sentiment analysis

Analyze the sentiment of customer reviews using a binary classification algorithm.

Sentiment analysis sample >

Product recommendation

Recommend products based on purchase history using a matrix factorization algorithm.

Product recommendation sample >

Price prediction

Predict taxi fares based on distance traveled etc. using a regression algorithm.

Price prediction sample >

Customer segmentation

Identify groups of customers with similar profiles using a clustering algorithm.

Customer segmentation sample >

GitHub labeler

Suggest the GitHub label for new issues using a multi-class classification algorithm.

GitHub labeler sample >

Fraud detection

Detect fraudulent credit card transactions using a binary classification algorithm.

Fraud detection sample >

Spam detection

Flag text messages as spam using a binary classification algorithm.

Spam detection sample >

Image classification

Classify images (e.g. broccoli vs pizza) using a TensorFlow deep learning algorithm.

Image classification sample >

Sales forecasting

Forecast future sales for products using a regression algorithm.

Sales forecasting sample >

And more! Samples @ https://github.com/dotnet/machinelearning-samples

ML.NET Resources

Get started at **http://dot.net/ml**

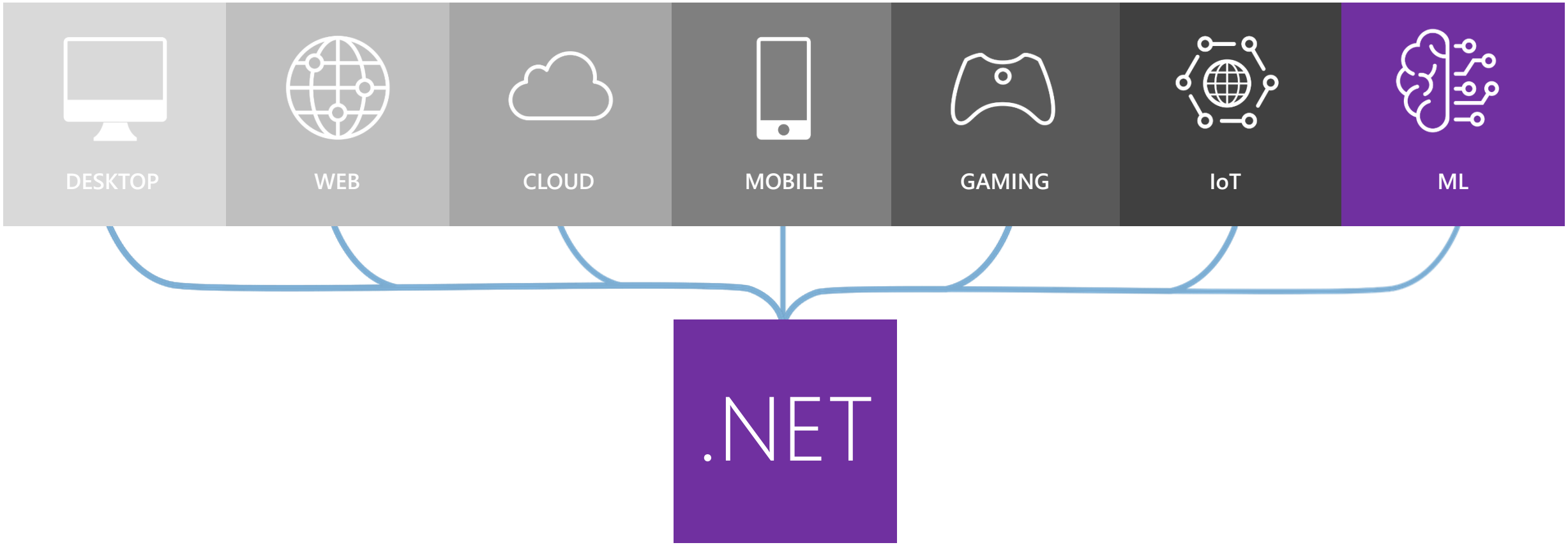
Try the samples at **http://aka.ms/mlnetsamples**

Read the docs at **http://aka.ms/mlnetdocs**

Watch ML.NET videos at **https://aka.ms/mlnetyoutube**

Request features or contribute at **http://aka.ms/mlnet**

Your platform for building **anything**



Arigato! :-]