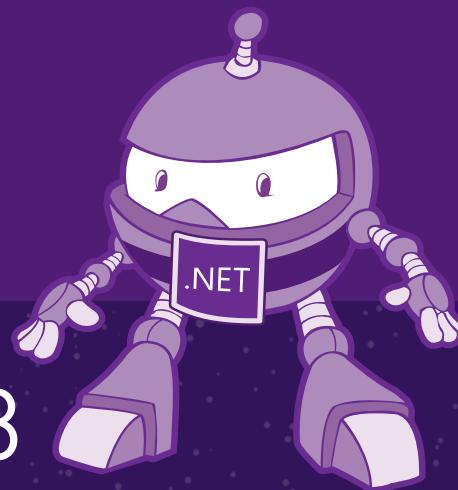


# Artificial Intelligence and Machine Learning for Every .NET Developer

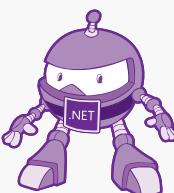
Paul Amazona  
Twitter/Github: @whatevergeek

.NET Conf SG '18

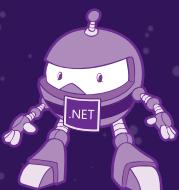


# Agenda

- Review: Machine Learning
- Pre-built AI Solutions
- Custom AI Solutions



# What is Machine Learning?

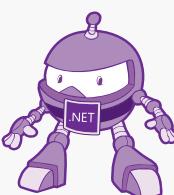


.NET Conf SG '18



# Machine Learning

## “Programming the UnProgrammable”



# Machine Learning

## “Programming the UnProgrammable”

Machine Learning creates a

$$f(x)$$

*Model*

Using this data



Face



Face



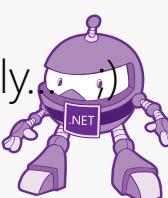
Not a face



Not a face

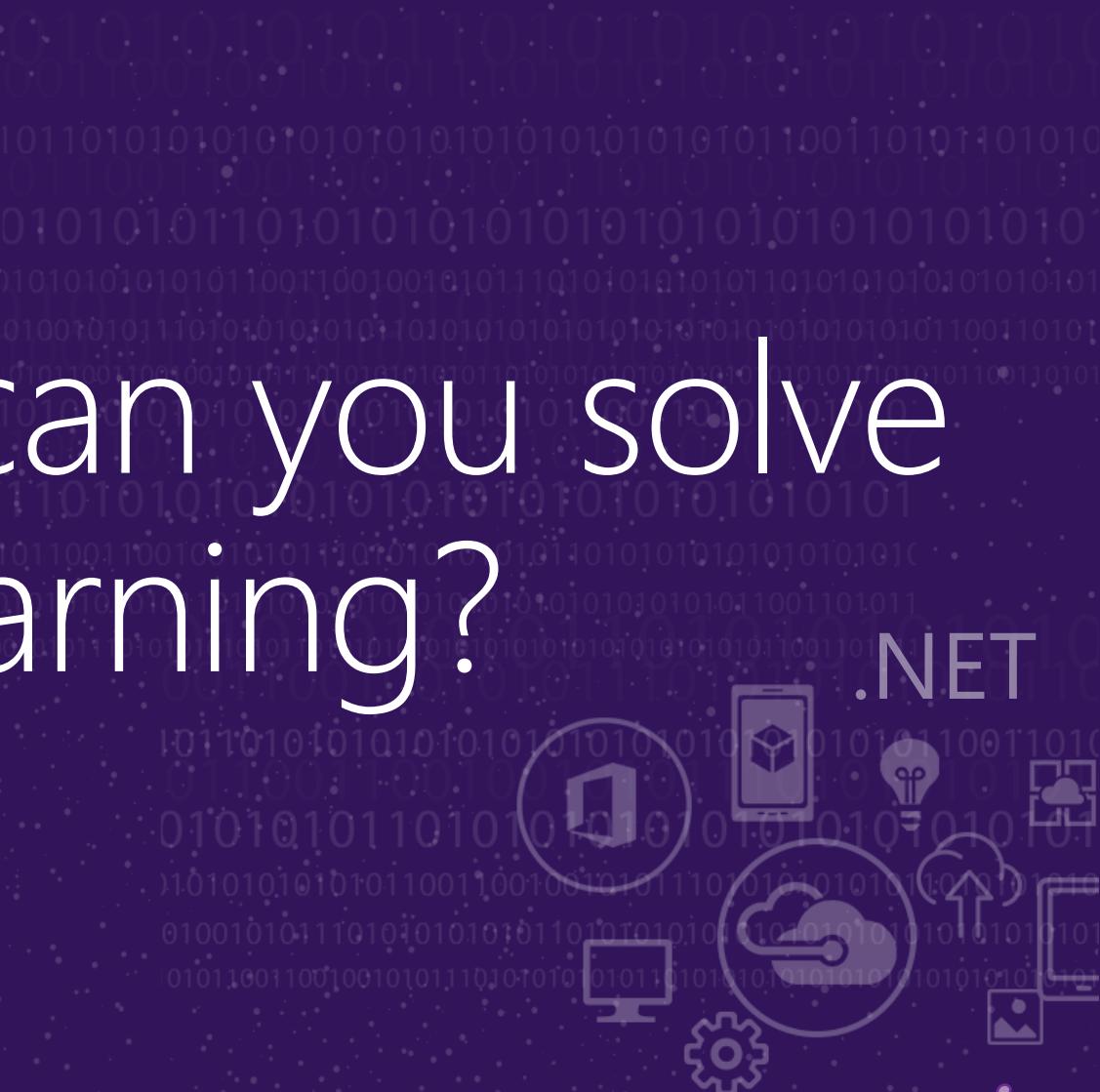
But it needs a lot of sample training data in order to predict properly.

.NET Conf SG '18

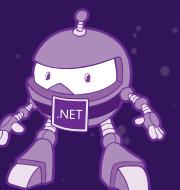


# What problems can you solve with Machine Learning?

.NET



.NET Conf SG '18



# Many Machine Learning Tasks!

**Supervised ML** (Infers label)

**Linear Discriminant Analysis**

Structured prediction

**Regression**

Naïve Bayes

Linear regression  
Logistic regression

**Decision Trees**

Binary Classification

**Multi-class Classification**

k-nearest neighbor

**Neural Networks**

(MultiLayer Perception, etc.)

Support Vector Machines

**Unsupervised ML** (Infers structure)

**Clustering**

( K-means  
Mixture models  
Hierarchical clustering)

**Topic Modeling**

Dimensionality Reduction

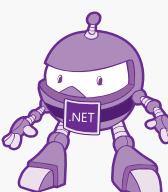
**Anomaly detection**

Latent variable models

Topic modeling

**Neural Networks**

(Autoencoders,  
Self-organizing maps, etc.)



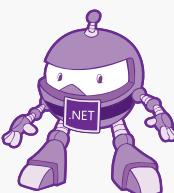
# THE GOAL

*“Democratize Machine Learning and AI in general, for developers”*

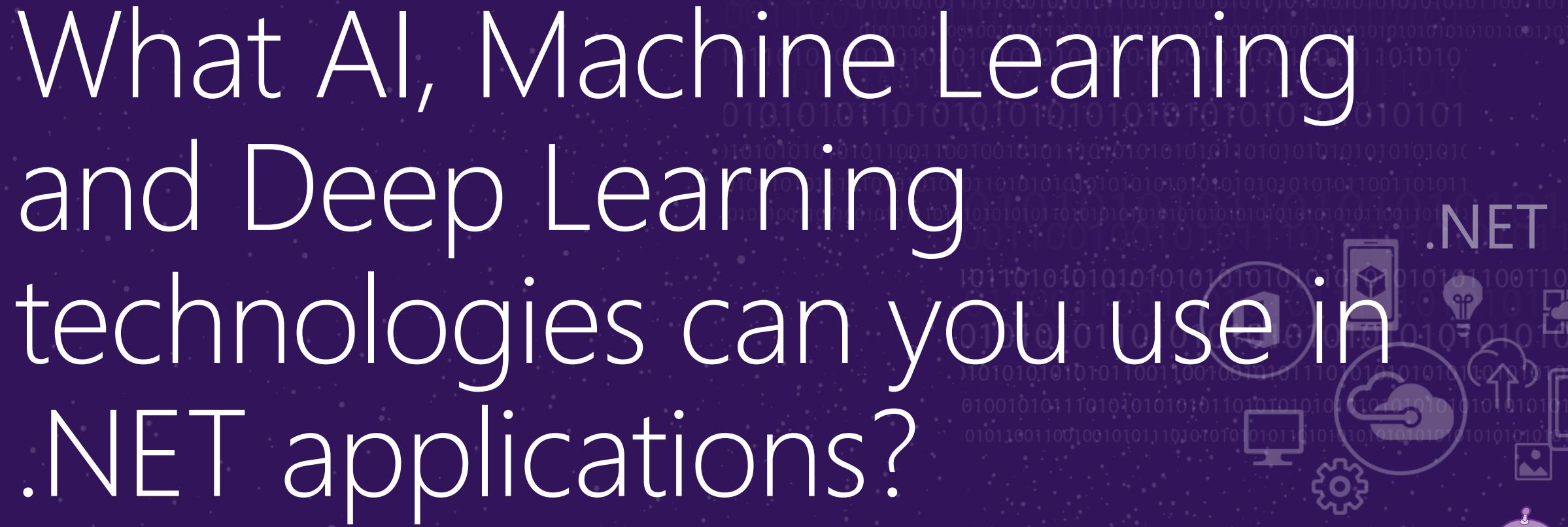
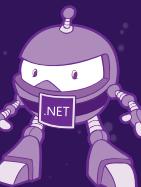
## Cost function

Logistic regression:

$$J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

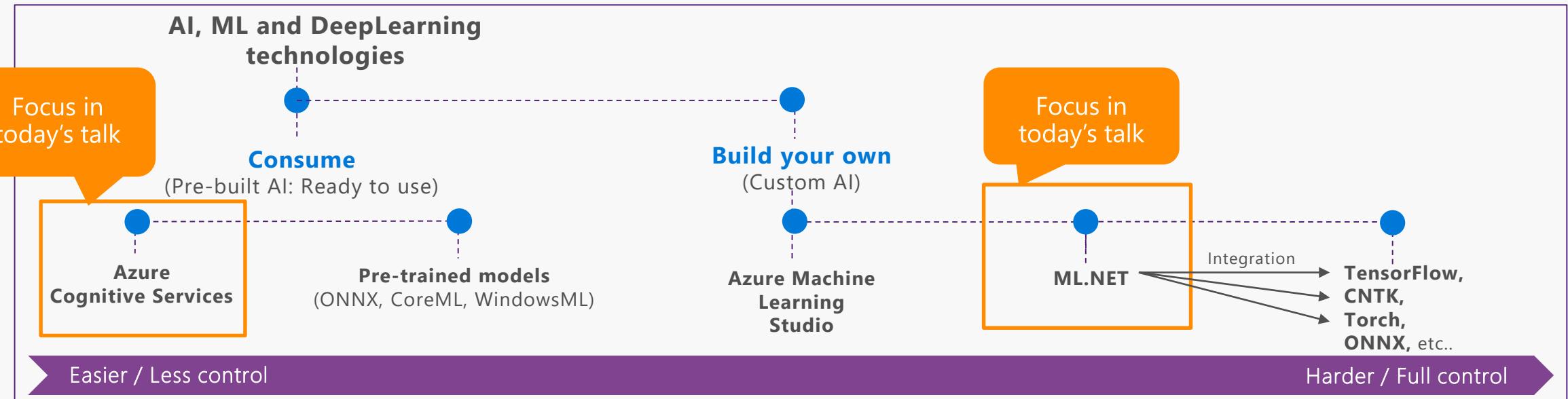


# What AI, Machine Learning and Deep Learning technologies can you use in .NET applications?

A dark purple background featuring a grid of white binary digits (0s and 1s). Overlaid on the bottom right are several white icons representing .NET technologies: a smartphone, a lightbulb, a cloud with a gear, an upload arrow, a user profile, a gear, and a monitor. The word ".NET" is partially visible on the right side of the slide.

# AI & ML portfolio for .NET applications

Consume pre-built/pre-trained models or build your own custom model?

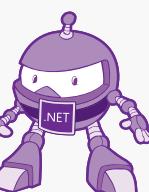


## Client apps

<b>Bots</b> (Bot Framework)	<b>Web apps</b> (ASP.NET)	<b>Mobile apps and IoT Edge devices</b> (Xamarin) (IoT Edge SDKS)
--------------------------------	------------------------------	---

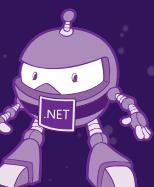
Visual Studio and .NET

.NET Conf SG '18



# Pre-build AI: Using Azure Cognitive Services in .NET applications

.NET Conf SG '18



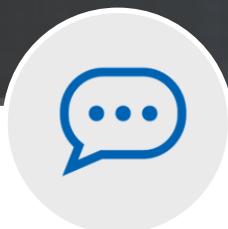
# Microsoft Cognitive Services

## Commoditized AI



### Vision

From faces to feelings, allow your apps to understand images and video



### Speech

Hear and speak to your users by filtering noise, identifying speakers, and understanding intent



### Language

Process text and learn how to recognize what users want



### Knowledge

Tap into rich knowledge amassed from the web, academia, or your own data



### Search

Access billions of web pages, images, videos, and news with the power of Bing APIs



### Labs

An early look at emerging Cognitive Services technologies: discover, try and give feedback on new technologies before general availability

# Microsoft Cognitive Services

## Commoditized AI



### Vision

Computer Vision

Content Moderator

Emotion

Face

Video Indexer

Custom Vision Service



### Speech

Bing Speech

Speaker Recognition

Custom Speech Service



### Language

Bing Spell Check

Linguistic Analysis

Text Analytics

Translator Text & Speech

Web Language Model

Language Understanding



### Knowledge

Academic Knowledge

Entity Linking

Knowledge Exploration

Recommendations

QnA Maker

Custom Decision Service



### Search

Bing Autosuggest

Bing Image Search

Bing News Search

Bing Video Search

Bing Web Search

Bing Entity Search

Bing Custom Search



### Labs

Project Prague (gesture)

Project Cuzco (events)

Project Johannesburg (routing)

Project Nanjing (isochrones)

Project Abu Dhabi (distance matrix)

Project Wollongong (location)

Demo for today

Demo for today

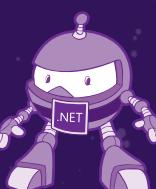
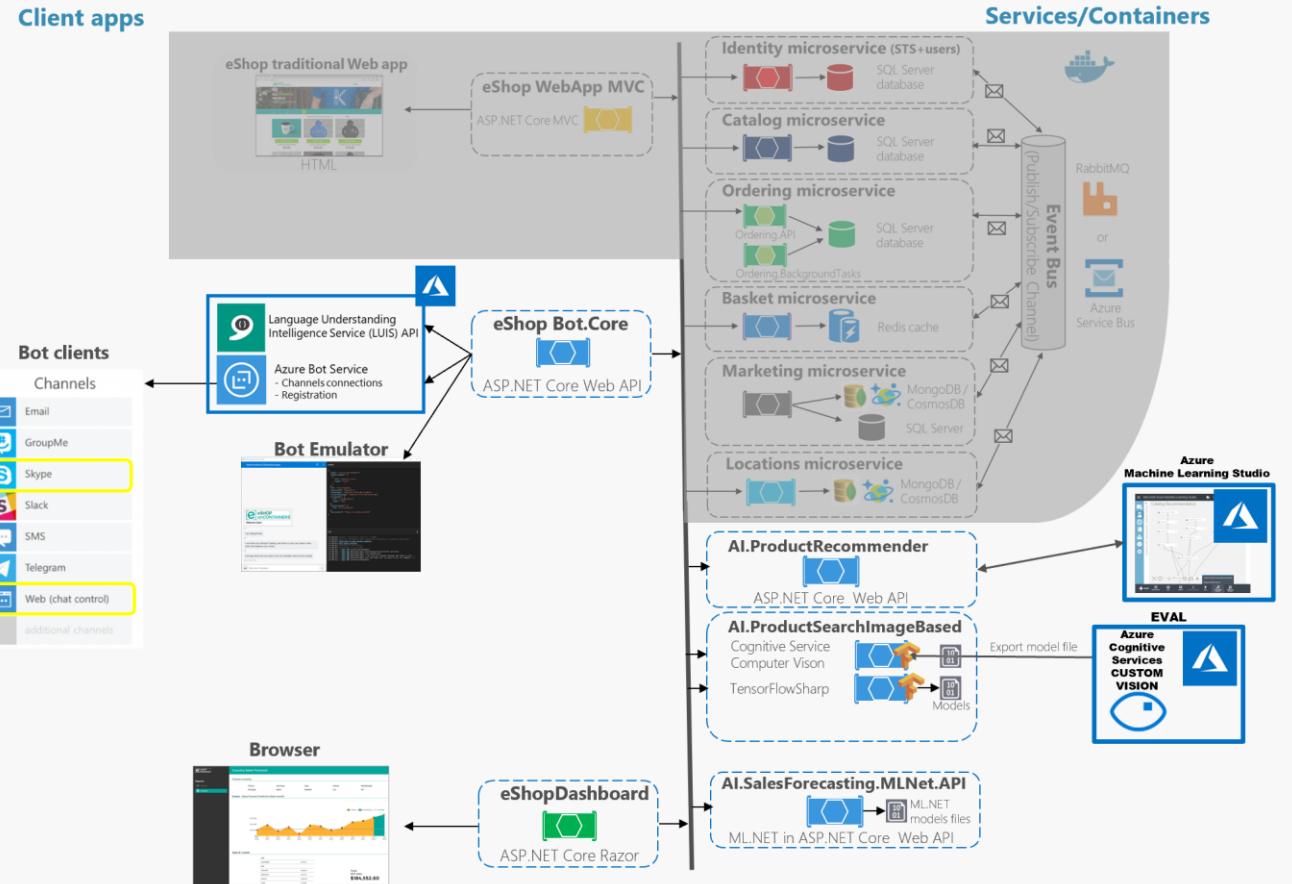
# Demo

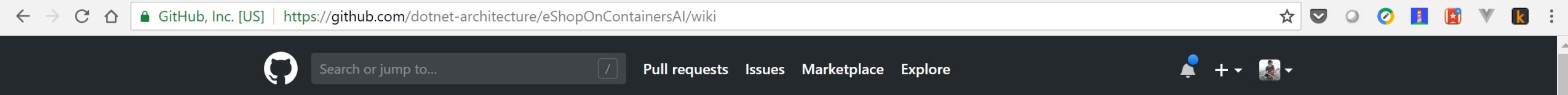
## Using Cognitive Services: Cognitive Services:

*Computer Vision &  
Custom Vision  
in eShopOnContainersAI  
app*

<https://github.com/dotnet-architecture/eShopOnContainersAI>

### eShopOnContainersAI reference application





## dotnet-architecture / eShopOnContainersAI

[Watch](#)

31

[Star](#)

99

[Fork](#)

26

[Code](#)[Issues 1](#)[Pull requests 0](#)[Projects 0](#)[Wiki](#)[Insights](#)

# Home

franperezlopez edited this page on Jul 18 · 3 revisions

## Welcome to the eShopOnContainersAI wiki!

[Pages 16](#)

### Roadmap

[Roadmap for eShopOnContainersAI \(Artificial Intelligence features\)](#)

### Setting up your development environment for eShopOnContainers

Visual Studio 2017 and Windows based This is the more straightforward way to get started: [Setting eShopOnContainersAI in a Visual Studio 2017 environment](#)

CLI and Windows based For those who prefer the CLI on Windows, using dotnet CLI, docker CLI and VS Code for Windows: [Setting up eShopOnContainersAI in a Windows CLI environment](#)

CLI and Mac based For those who prefer the CLI on a Mac, using dotnet CLI, docker CLI and VS Code for Mac: [Setting up eShopOnContainersAI in a Mac development environment \(Docker CLI, dotnet CLI, VS Code and VS for Mac\)](#)

#### News

- [Roadmap](#)
- [FAQ](#)

#### Setup

- [Requirements](#)
- [Visual Studio 2017](#)
- [Windows CLI](#)
- [Linux/Mac CLI](#)

#### Scenarios

- [Recommendation systems](#)  
[Product recommendation](#)  
(ML Studio, C#)
- [Computer Vision](#)  
Image classification
  - [Cognitive Services](#)



LOGIN

ALL T-SHIRTS  
ON SALE  
THIS WEEKEND



BRAND  
All

TYPE  
All

PRODUCT SEARCH IMAGE-BASED (COMPUTERVISION)  
Choose file



Showing 12 of 3676 products - Page 1 - 307

Next



[ ADD TO CART ]

.NET BLACK & WHITE MUG

\$ 8.50



[ ADD TO CART ]

.NET BLUE HOODIE

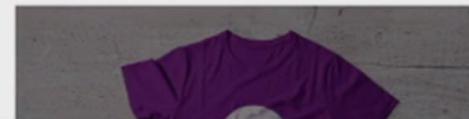
\$ 12.00



[ ADD TO CART ]

.NET BOT BLACK HOODIE

\$ 19.50





[LOGIN](#)

**ALL T-SHIRTS  
ON SALE  
THIS WEEKEND**



BRAND  
All

TYPE  
AII

PRODUCT SEARCH IMAGE-BASED / COMPUTER VISION

Choose file

Sorry, we could not find any product similar to your sample image



e-ShopOnContainers. By Microsoft Corp.

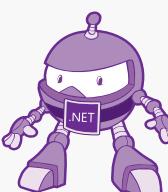
## EXAMPLE: When to use “Azure Custom Vision”

If using Azure ComputerVision (which is always online), it doesn't recognize a “Frisbee” picture. Recognizes it as “Dishware”, not a good identification → Need Custom Vision for these cases:

The screenshot shows a Visual Studio interface. On the left, there's a navigation pane with 'Run All', 'Run...', and 'Playlist : All Tests'. Below that is a tree view for 'eShopOnContainers-AI (100 tests)' containing categories like FunctionalTests, IntegrationTests, LoadTest, and UnitTest. In the center, a code editor displays C# code for processing Azure Computer Vision API responses. The code uses LINQ to filter and sort tags based on confidence levels. At the bottom, a 'Watch 1' window shows a variable named 'visionApiResponse.tags' which contains two elements. The first element has a confidence of 0.42394906282424927 and a name of "dishware". The second element has a confidence of 0.36457100510597229 and a name of "tableware".

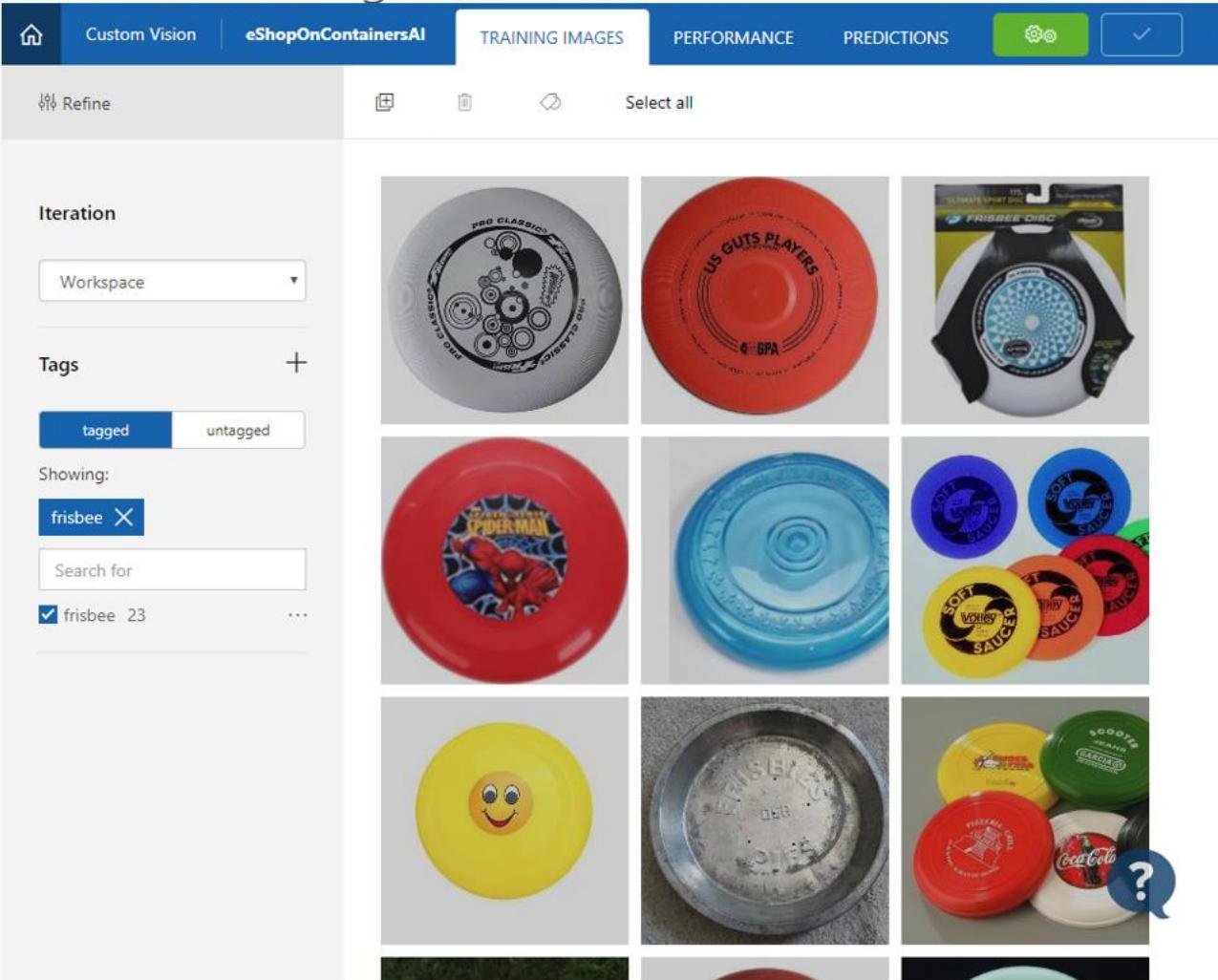
```
69 }  
70  
71     var visionApiResponse = JsonConvert.DeserializeObject<Vis  
72  
73         var query = visionApiResponse.tags.AsQueryable();  
74  
75             if (query.Any(t => t.confidence > settings.Threshold))  
76                 query = query  
77                     .Where(t => t.confidence > settings.Threshold)  
78                         .OrderByDescending(t => t.confidence)  
79                             .Take(settings.MaxLength);  
80  
81             // In case we don't find any element matching threshold  
82                 query = query  
83                     .OrderByDescending(t => t.confidence)  
84                         .Take(1);  
85
```

Name	Value
visionApiResponse.tags	{Microsoft.eShopOnContainers.Services.AI.ProductSearchImageBased.AzureComputerVisionImageProcessor+ImageTag[0]}
[0]	{Microsoft.eShopOnContainers.Services.AI.ProductSearchImageBased.AzureComputerVisionImageProcessor+ImageTag}
confidence	0.42394906282424927
name	"dishware"
[1]	{Microsoft.eShopOnContainers.Services.AI.ProductSearchImageBased.AzureComputerVisionImageProcessor+ImageTag}
confidence	0.36457100510597229
name	"tableware"



## EXAMPLE: When to use “Azure Custom Vision”

If using Azure ComputerVision (which is always online), it doesn't recognize a “Frisbee” picture. Recognizes it as “Dishware”, not a good identification → Need Custom Vision for these cases:



# customvision.ai

The screenshot shows the Microsoft Custom Vision Services interface. At the top, there's a navigation bar with the Microsoft logo, Cognitive Services, and Custom Vision. The main area features a large blue background image of a neural network. On the left, there's a stack of three labeled photos: a red cat labeled 'CAT' and '(LABLED PHOTOS)', and a brown dog labeled 'DOG'. To the right of these are four vertical rectangles representing unlabeled images, followed by a horizontal arrow pointing to the right labeled 'OUTPUT'. Below this, there are three sections: 'Upload Images' (with three small image icons), 'Train' (with a cloud icon containing a camera), and 'Evaluate' (with a grid of nine small image icons). Each section has a brief description and some statistics.

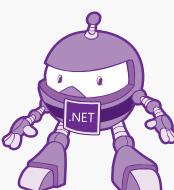
**Upload Images**  
Bring your own labeled images, or use Custom Vision to quickly add tags to any unlabeled images.

**Train**  
Use your labeled images to teach Custom Vision the concepts you care about.

**Evaluate**  
Use simple REST API calls to quickly tag images with your new custom computer vision model.

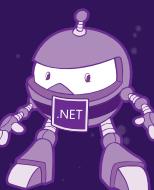
89%    93%    91%

.NET Conf SG '18



# Custom Machine Learning: Using ML.NET

.NET Conf SG '18

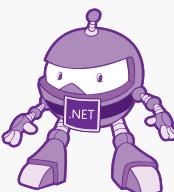


.NET

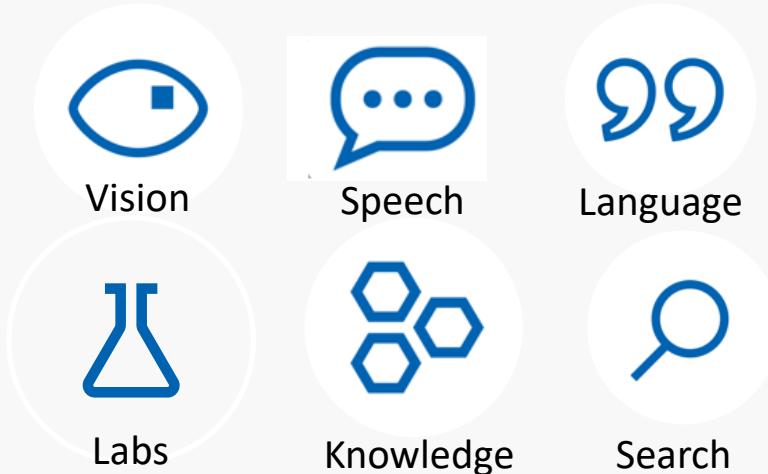


Is pre-trained/pre-built Machine Learning enough for you?  
i.e. Azure Cognitive Services, etc.

*As always.. the answer is... : "It depends..." ;)*



# Pre-built ML Models (Azure Cognitive Services)



Consume (C#, VB, F#)

e.g. Sentiment Analysis using Azure Cognitive Services

```
TextAnalyticsAPI client = new TextAnalyticsAPI();
client.AzureRegion = AzureRegions.WestUs;
client.SubscriptionKey = "1bf33391DeadFish";

client.Sentiment(
    new MultiLanguageBatchInput(
        new List<MultiLanguageInput>()
    {
        new MultiLanguageInput("en", "0",
            "This is a great vacuum cleaner")
    }));

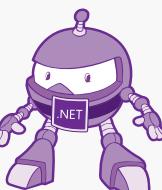
```



96% positive

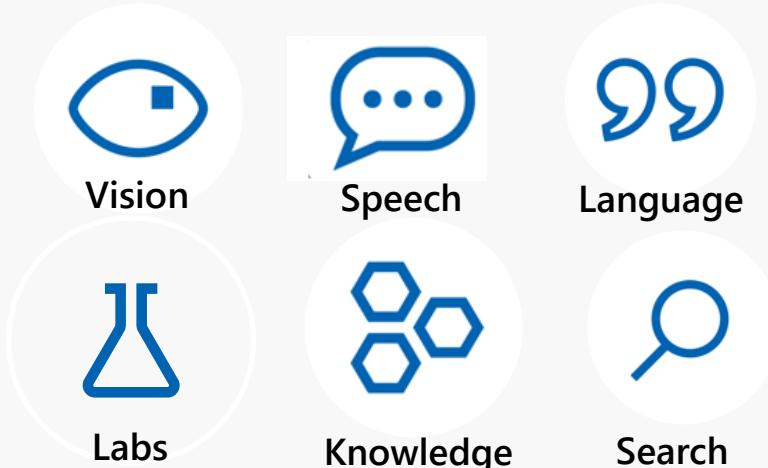
Easy / Less Control

Full Control / Harder



.NET Conf SG '18

# Pre-built ML Models (Azure Cognitive Services)



e.g. Sentiment Analysis using Azure Cognitive Services

```
TextAnalyticsAPI client = new TextAnalyticsAPI();
client.AzureRegion = AzureRegions.WestUs;
client.SubscriptionKey = "1bf33391DeadFish";

client.Sentiment(
    new MultiLanguageBatchInput(
        new List<MultiLanguageInput>()
    {
        new MultiLanguageInput("en", "0",
            "This vacuum cleaner sucks so much dirt")
    }));

```

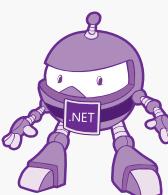


9% positive

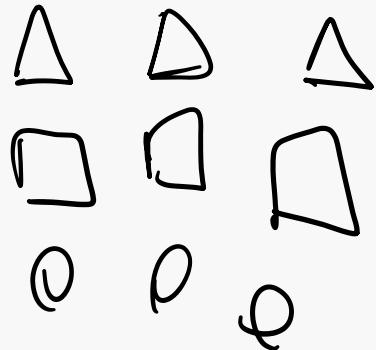
Easy / Less Control

Full Control / Harder

.NET Conf SG '18

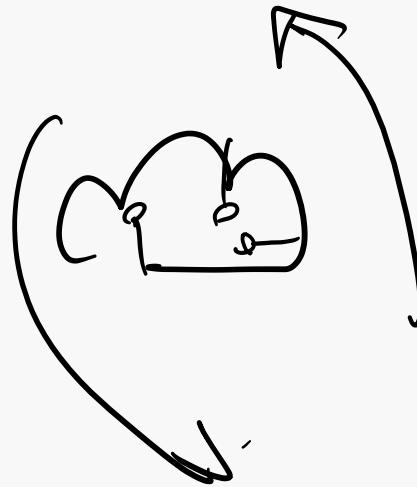


# Build your own (custom) ML Models

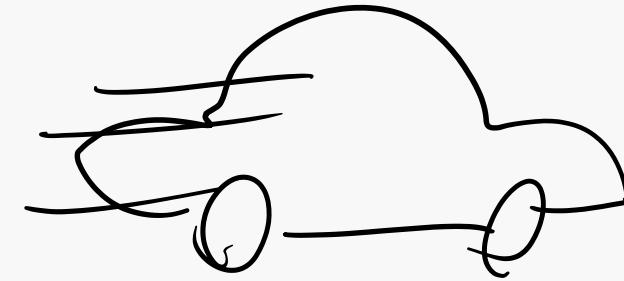


**Prepare Your Data**

Easy / Less Control



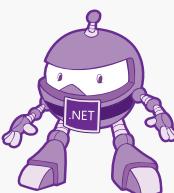
**Build, Train  
& Evaluate**



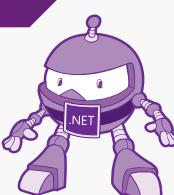
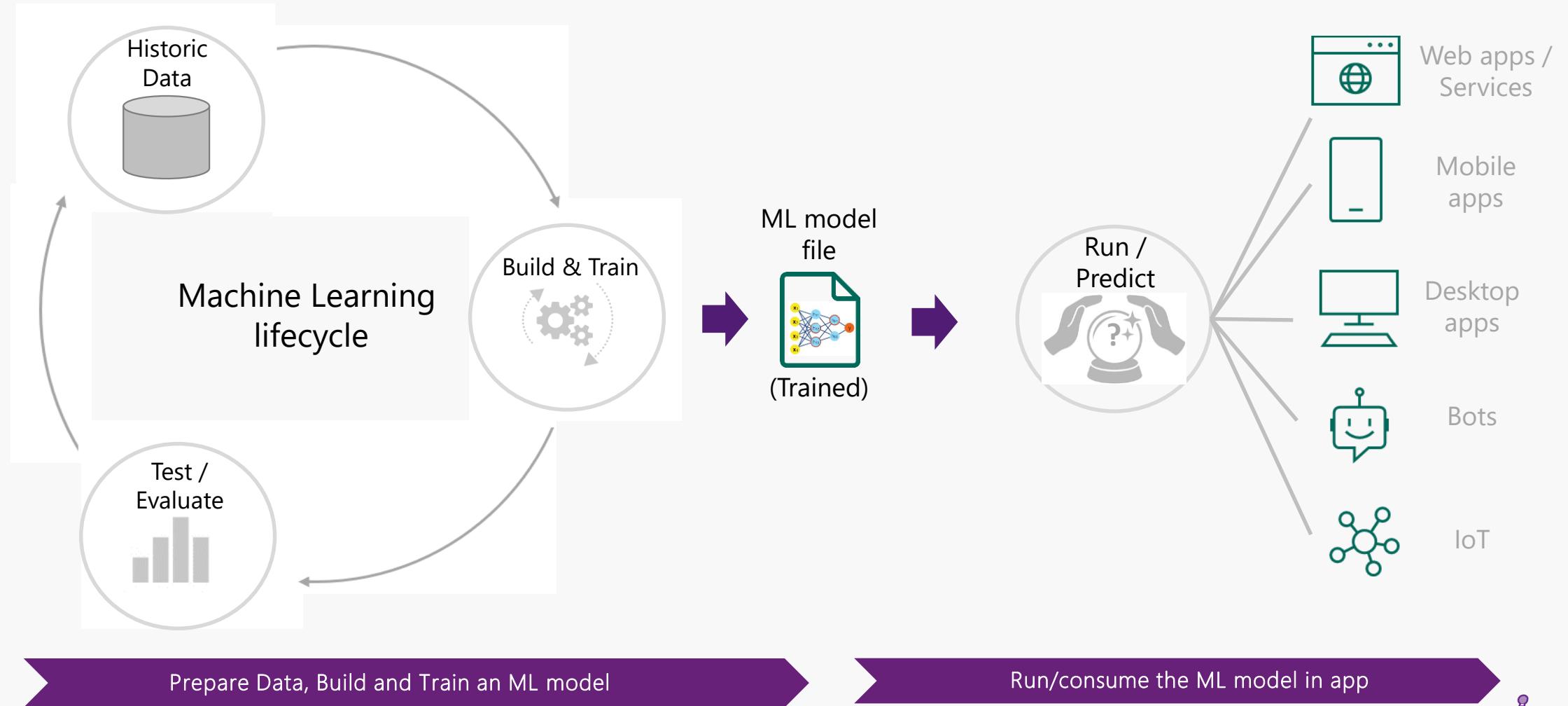
**Run**

Full Control / Harder

.NET Conf SG '18



# Processes for Building your own (custom) ML Models



# Introducing ML.NET

Currently in [v0.5](#)  
preview Sept-2018

Machine Learning **framework** made for .NET developers

(Supported on Windows, Linux, and macOS)



Build your own



Developer Focused



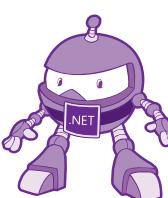
Proven & Extensible



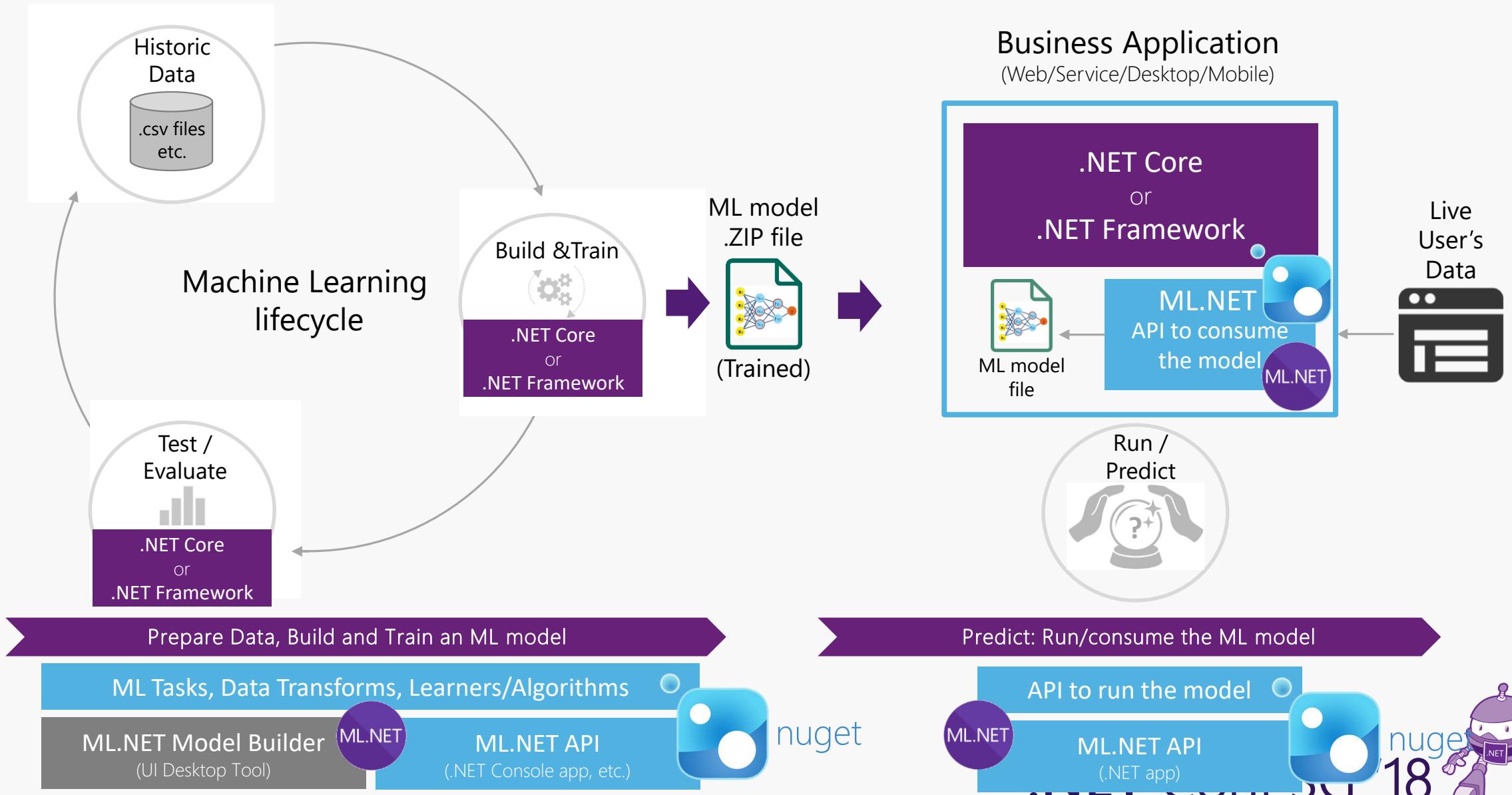
Open Source  
&  
Cross Platform

<https://github.com/dotnet/machinelearning>

.NET Conf SG '18

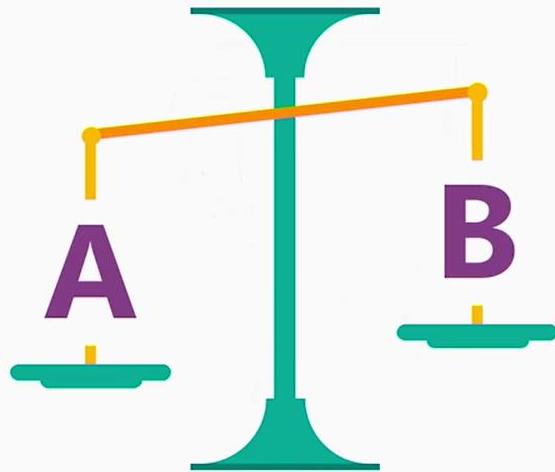


# ML.NET is a framework for custom ML

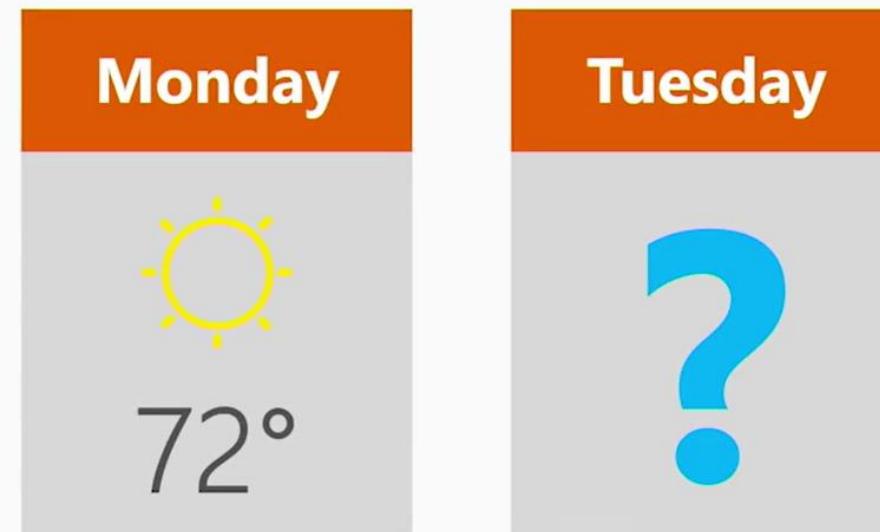


# A few problems you can solve with ML.NET

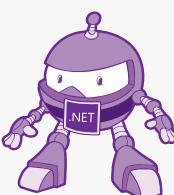
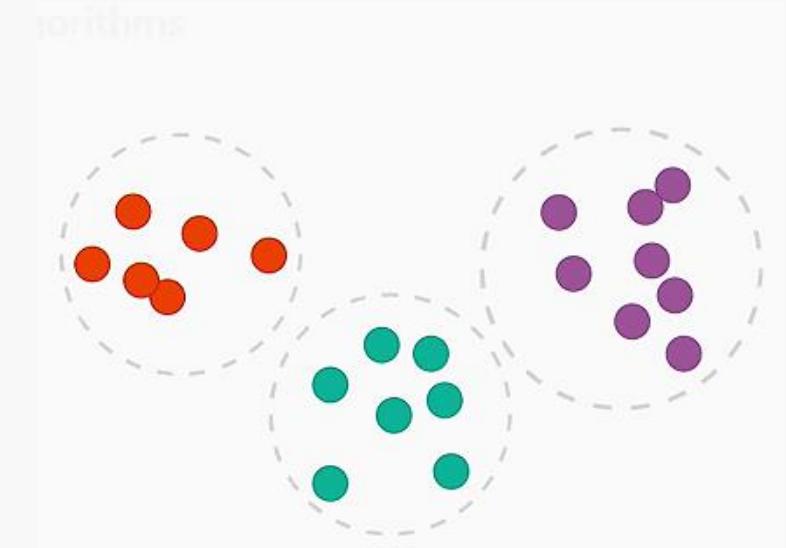
Is this A or B?



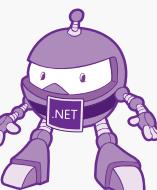
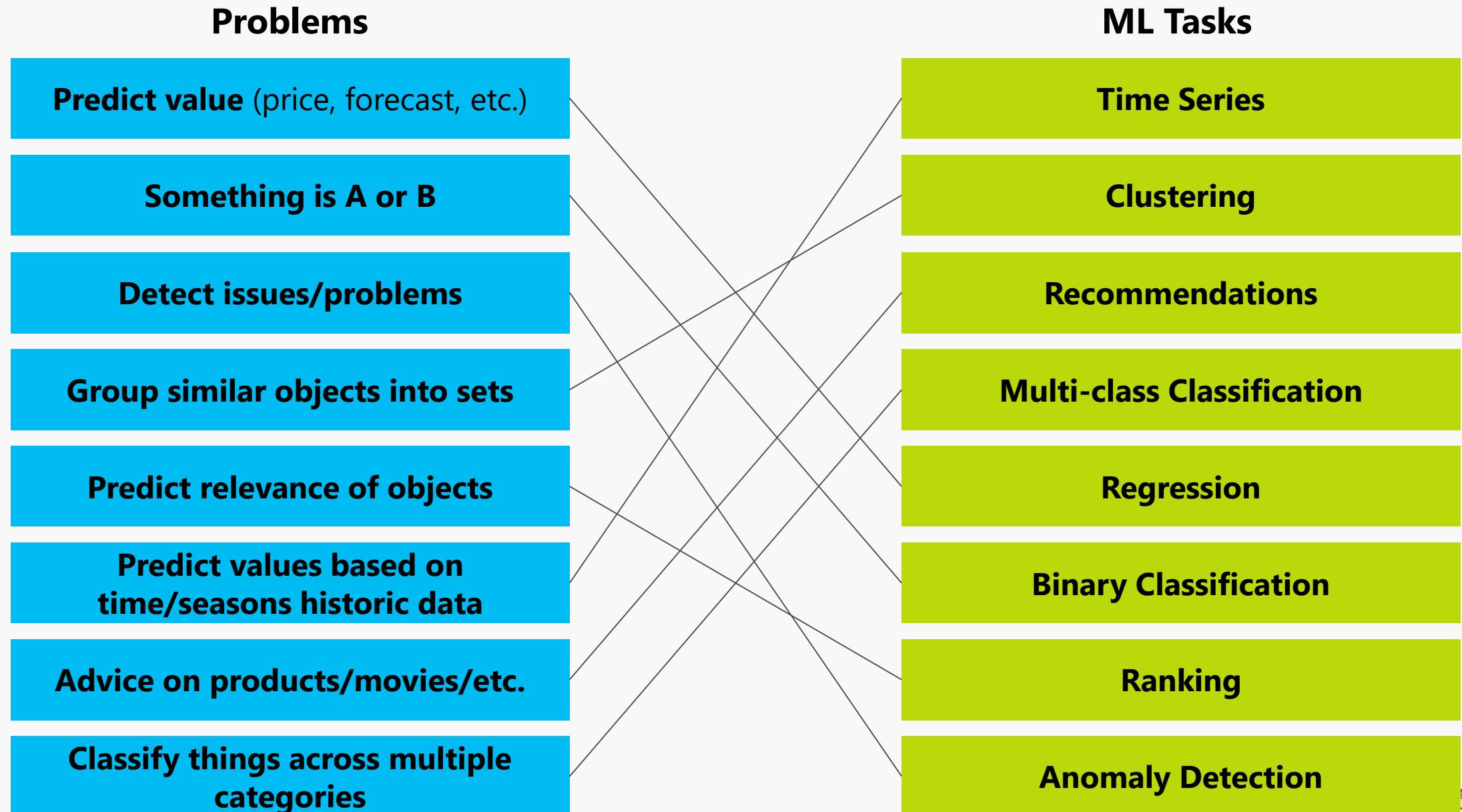
How much? How many?



How is this organized?

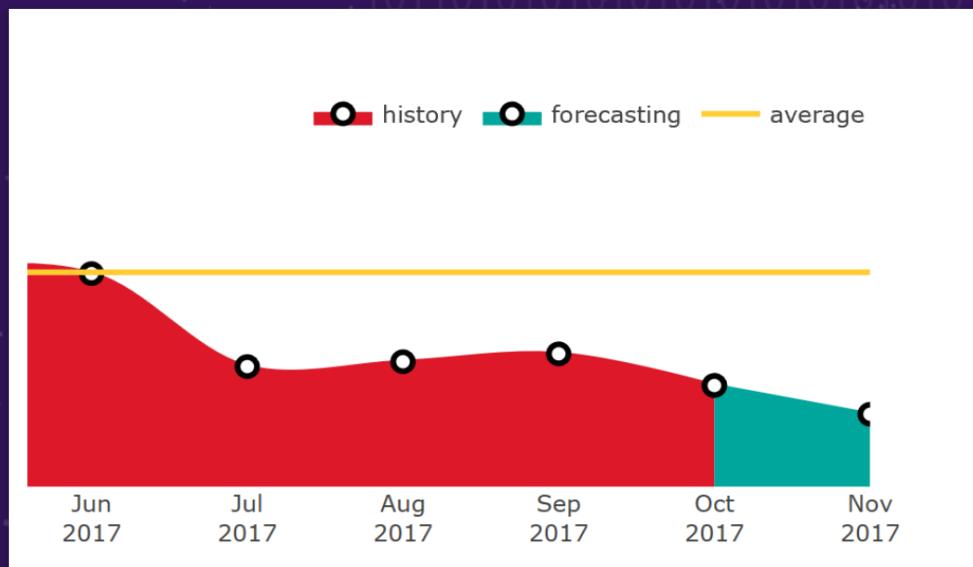


# Mapping from Problems to ML Tasks



# Demo

Sales forecast – eShopDashboardML  
(Regression problem predicting sales forecast)

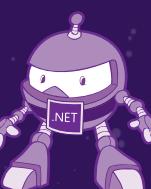


Sales Forecasting



<https://github.com/dotnet/machinelearning-samples>

.NET Conf SG '18



# ML.NET is a framework first

nuget

Developer-friendly ML APIs to:

- **Build & Train** ML.NET models
- **Run** any model

.NET Standard  
.NET Core  
.NET Framework

Transforms	Learners	Misc.
Text	Linear	ML Data framework
Schema	Boosted Trees	Evaluators
Missing values	Svm	Calibrators
Categorical	K-Means	Data loaders
Normalization		
Feature Selection		

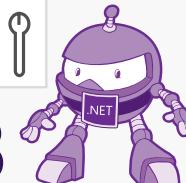
Preliminary support of TensorFlow scoring in ML.NET is available since v0.5

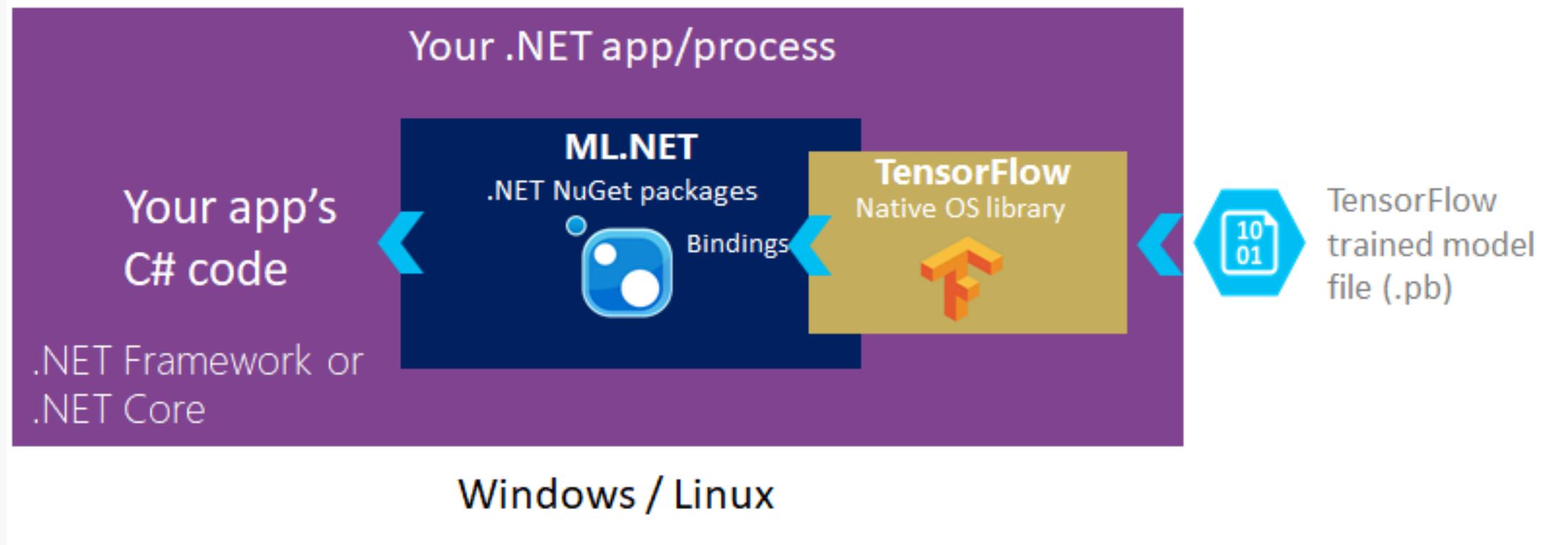
 TensorFlow

 torch

 Microsoft CNTK

 Accord.NET



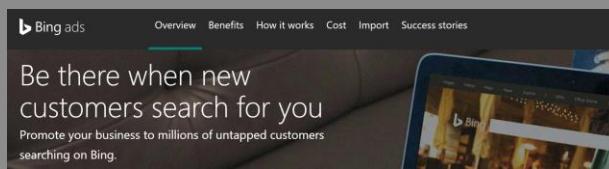


<https://blogs.msdn.microsoft.com/dotnet/2018/09/12/announcing-ml-net-0-5/>

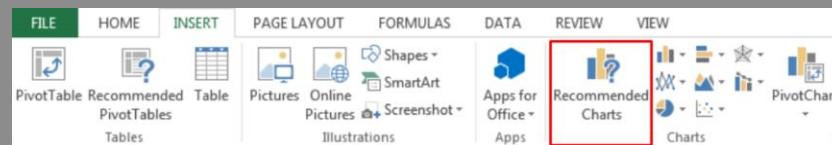
# ML.NET: Proven at large scale in Microsoft



Bing Ads



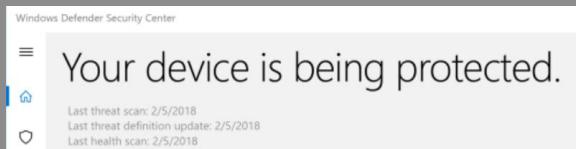
Excel



Power Point



Windows 10



# The Goal for ML.NET ?

*Democratize Machine Learning custom models  
for .NET developers with a framework and  
tools especially tailored for developers*

**Cost function**

Logistic regression:

$$J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$



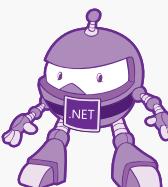
The screenshot shows the ML.NET studio interface. At the top, there is a code editor window containing C# code for defining a learning pipeline:

```
var pipeline = new LearningPipeline();
pipeline.Add(new TextLoader<TaxiTrip>
    (DataPath, useHeader: true, separator: ","));
pipeline.Add(new CategoricalOneHotVectorizer
    ("vendor_id",
     "rate_code",
     "payment_type"));
pipeline.Add(new ColumnConcatenator("Features", "vendor_id", "rate_code", ...));
pipeline.Add(new FastTreeRegressor());
pipeline.Train<TaxiTrip, TaxiTripFarePrediction>();
```

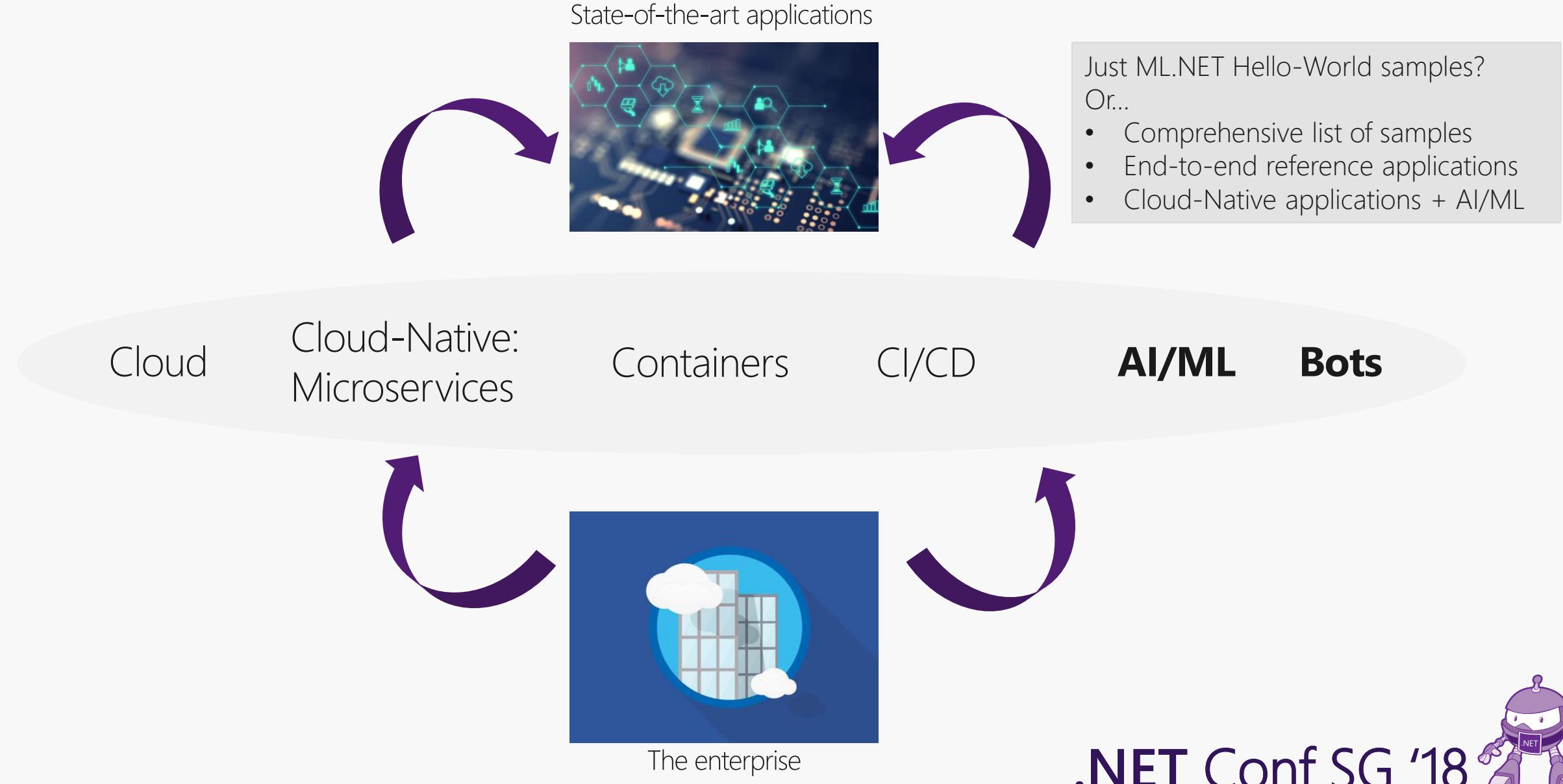
To the right of the code editor is the Microsoft Visual Studio logo. Below the code editor is a data preview window titled "Data Preview after applying the first transform". It shows a table with three rows of data, each with columns for ID, Area, Title, Description, and Label.

.NET code-first  
approach to build  
& train custom  
models

UI tool,  
easy to get started  
for .NET developers  
(\*) To be released

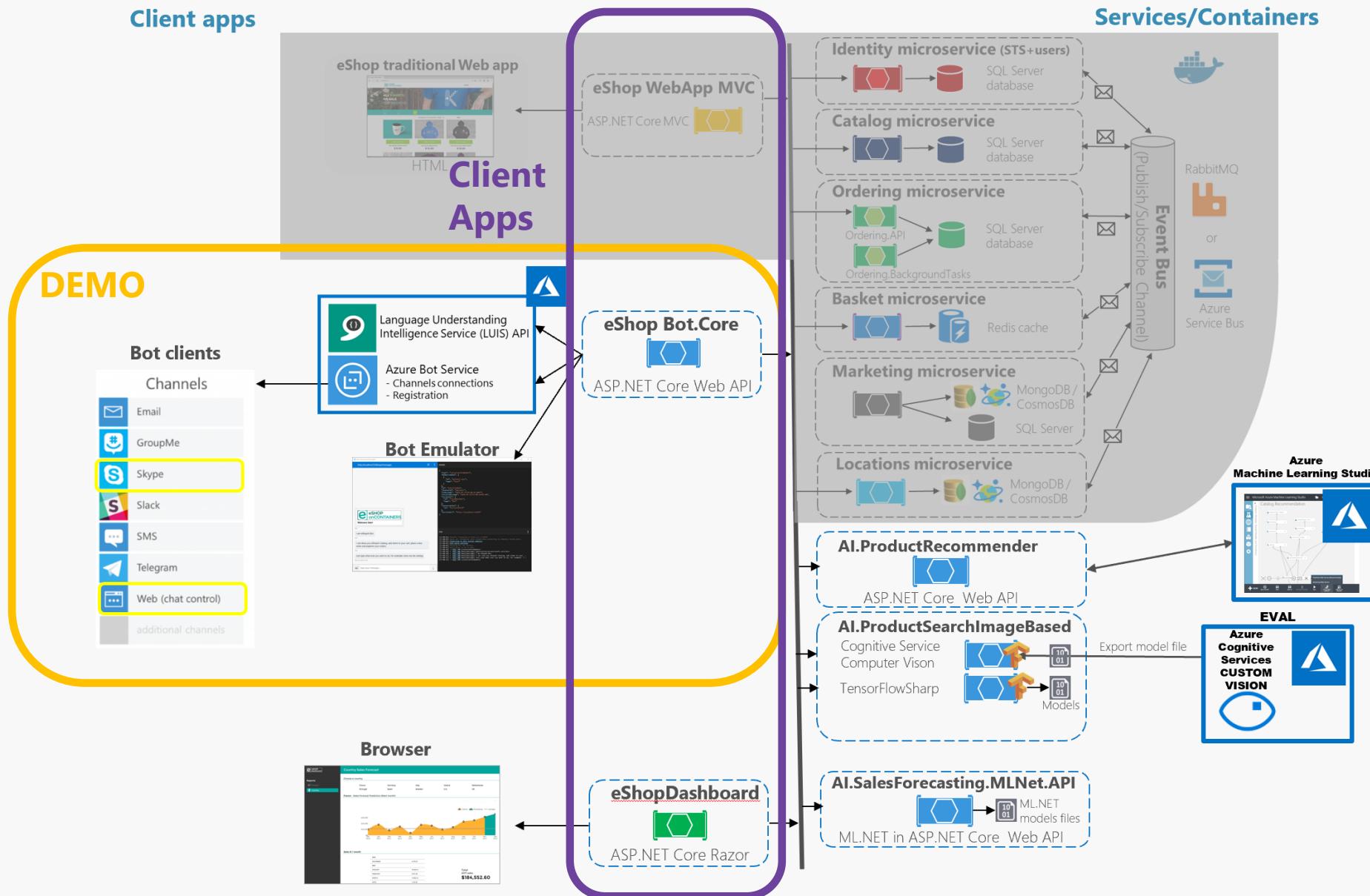


# Enterprise innovation goes across all technologies, not just AI...

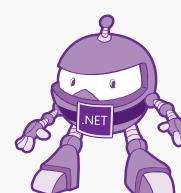


# Demo: Surfacing AI with Microsoft Bot Framework + L.U.I.S.

<https://github.com/dotnet-architecture/eShopOnContainersAI/>



'18



# Demo

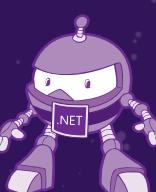
## eShopOnContainersAI:

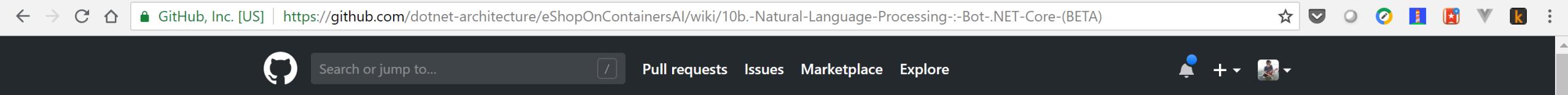
End-to-end AI scenario: Cloud-Native + AI/ML

- + ML.NET
- + Cognitive Services
  - Custom Vision
- + **Bot Framework**
- + **Luis (Language Understanding)**



<https://github.com/dotnet-architecture/eShopOnContainersAI/>





dotnet-architecture / eShopOnContainersAI

Watch 31

Star 99

Fork 26

Code

Issues 1

Pull requests 0

Projects 0

Wiki

Insights

# 10b. Natural Language Processing : Bot .NET Core (BETA)

Cesar De la Torre edited this page on Jul 13 · 12 revisions

## Introduction

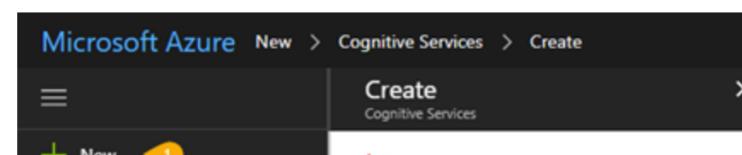
► Pages 16

The current Bot.API application is based in the v.4.0.1-preview nuget packages of the Bot Framework, which is currently under development. As result, the application is restricted to following requirements:

- Web and Skype channel support
- user welcome scenario and product browsing / filtering is available (authentication, basket and ordering scenarios expected when official release)
- docker deployment using .Net Core 2.1 is supported

## Step 1: Luis registration

You need to create a LUIS resource using the Azure Portal. In the search box used for creating new resources, look up for a resource of type Cognitive Services. The Create Cognitive Service blade will look like the following screenshot:



### News

- [Roadmap](#)
- [FAQ](#)

### Setup

- [Requirements](#)
- [Visual Studio 2017](#)
- [Windows CLI](#)
- [Linux/Mac CLI](#)

### Scenarios

- [Recommendation systems](#)  
[Product recommendation](#)  
(ML Studio, C#)
- [Computer Vision](#)  
Image classification
- [Cognitive Services](#)

**BOT EXPLORER**

**ENDPOINT** + Start Over Save Transcript As...

Hi. How can I help you?

Bot

**Shoe me products!**

User

Please select a brand you'd like to search for

Bot

You can also upload an image/photo to search for similar products

Bot



User

Page 1 of 1 (3 items)

 Cup Sheet 8.50 \$ Cup Sheet	 Roslyn Red Sheet 8.50 \$ Roslyn Red Sheet
--	--

Bot

choose one option

Bot at 10:29:44 AM

**Home**

Type your message...

**INSPECTOR - JSON**

```
{
  "attachmentLayout": "carousel",
  "attachments": [
    {
      "content": {
        "buttons": [],
        "images": [
          {
            "url": "http://localhost:5202/api/v1/c/catalog/items/10/pic/"
          }
        ],
        "subtitle": "12.00 $",
        "text": ".NET Foundation Sheet",
        "title": ".NET Foundation Sheet"
      },
      "contentType": "application/vnd.microsoft.card.thumbnail"
    },
    {
      "content": {
        "buttons": [],
        "images": [
          {
            "url": "http://localhost:5202/api/v1/c/catalog/items/11/pic/"
          }
        ],
        "subtitle": "8.50 $",
        "text": "Cup<T> Sheet",
        "title": "Cup<T> Sheet"
      }
    }
  ]
}
```

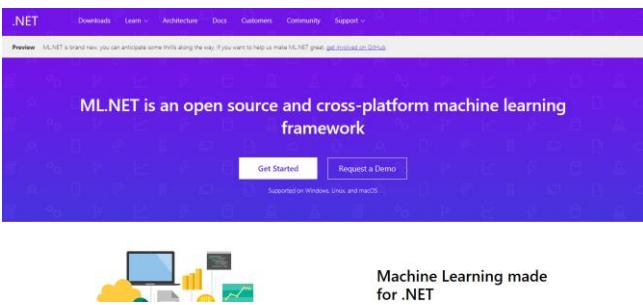
**LOG**

```
[10:28:16] ngrok listening on https://6a50c653.ngrok.io
[10:28:16] ngrok traffic inspector: http://127.0.0.1:4040
[10:28:16] Will use ngrok for local addresses
[10:28:16] POST 201 directline.startConversation
[10:28:18] POST 200 conversations.replyToActivity
[10:28:18] < message application/vnd.microsoft.card.hero
[10:28:19] POST 200 conversations.replyToActivity
[10:28:19] < message Howdy! - I am eShopAI-Bot.
[10:28:19] POST 200 conversations.replyToActivity
[10:28:19] < message I can show you the eShopAI Catalog, add items to y...
[10:28:19] POST 200 conversations.replyToActivity
[10:28:19] < message Just type whatever you want to do, for example: *s...
[10:28:41] > message Hi buddy!
[10:28:42] POST 200 conversations.replyToActivity
[10:28:42] < message Hi. How can I help you?
[10:28:42] POST 200 directline.postActivity
[10:29:00] > message Shoe me products!
[10:29:00] POST 200 conversations.replyToActivity
[10:29:00] < message Please select a brand you'd like to search for
[10:29:01] POST 200 conversations.replyToActivity
[10:29:01] < message You can also upload an image/photo to search for ...
[10:29:01] POST 200 directline.postActivity
[10:29:43] > message image/png
[10:29:44] POST 200 conversations.replyToActivity
[10:29:44] < message Page 1 of 1 ( 3 items )
[10:29:44] POST 200 conversations.replyToActivity
[10:29:44] < message choose one option
[10:29:45] POST 200 directline.upload
```

# Resources

## Get Started

[dot.net/ml](http://dot.net/ml)



## ML.NET Samples

(eShopDashboard, etc.)

<http://github.com/dotnet/machinelearning-samples>

A screenshot of the GitHub repository for "dotnet/machinelearning-samples". The repository has 40 stars, 28 forks, and 1 issue. It contains 27 commits, 1 branch, 0 releases, and 6 contributors. The code is licensed under MIT. The repository description is "Samples for ML.NET, an open source and cross-platform machine learning framework for .NET. http://dot.net/ml". The codebase includes branches for machine-learning, algorithms, dotnet, csharp, and ml.

## End-to-end Native App eShopOnContainersAI:

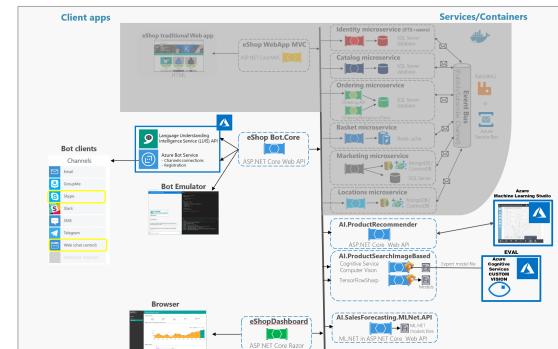
<https://github.com/dotnet-architecture/eShopOnContainersAI>

## Get Involved in OSS

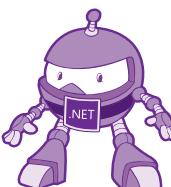
<http://github.com/dotnet/machinelearning/>

<http://aka.ms/newapifeedback/>

A screenshot of the GitHub repository for "dotnet/machinelearning". The repository has 426 stars, 303 forks, and 122 issues. It contains 105 commits, 3 branches, 2 releases, and 35 contributors. The code is licensed under MIT. The repository description is "ML.NET is an open source and cross-platform machine learning framework for .NET. http://dot.net/ml". The codebase includes branches for build, docs, pkg, src, and test.



.NET Conf SG '18



# Related sessions in .NET Conf 2018

- Machine Learning in .NET (ML.NET) - [Ankit Asthana](#) [Gal Oshri](#)
- Cognitive Services in Xamarin Applications - [Veronika Kolesnikova](#)
- Artificial Intelligence and Machine Learning for Every .NET Developer - [Cesar De la Torre Llorente](#)

Platinum Sponsors



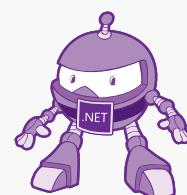
Gold Sponsors



Other Sponsors

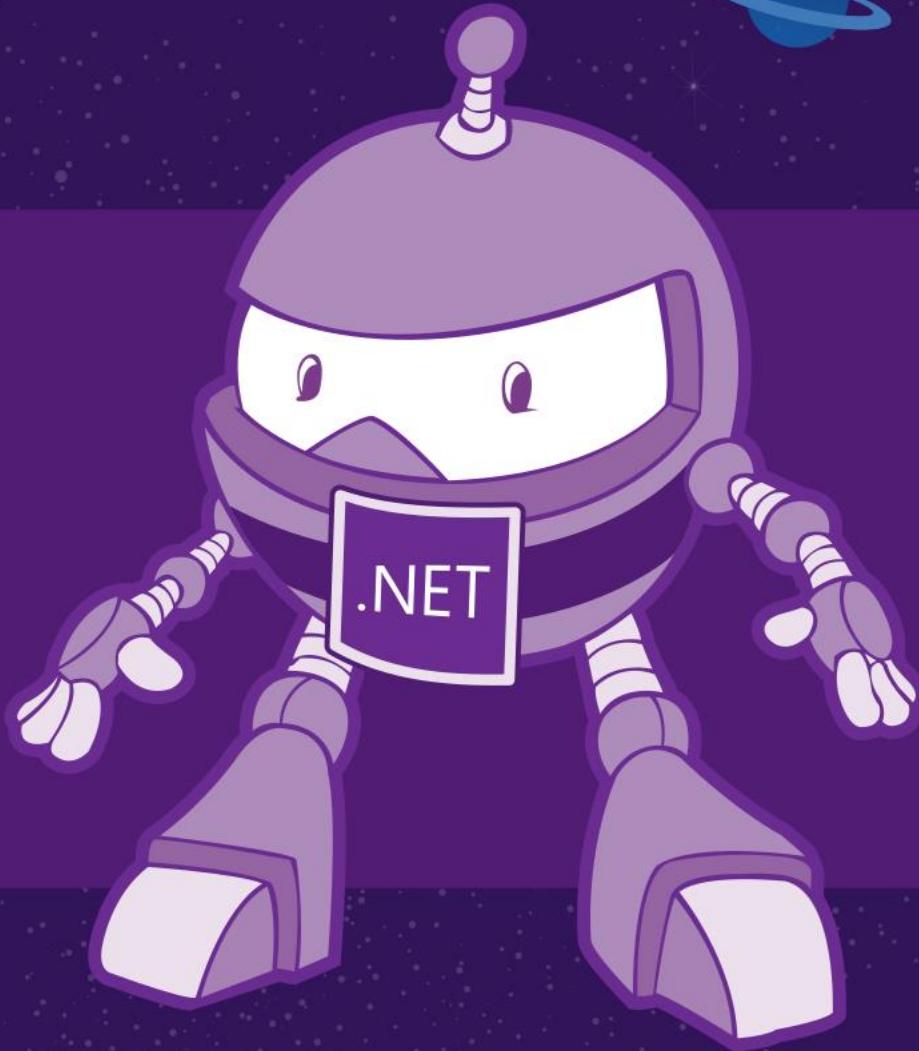


.NET Conf SG '18



# .NET Conf SG '18

## Discover the world of .NET



[www.dotnetconf.net](http://www.dotnetconf.net)

[www.dotnetconfsg.com](http://www.dotnetconfsg.com)