AI&ML CONF

aspitalia.com

UGIdotNET

ML.NET - Machine learning for .NET developer

Marco Zamana
Technical Team Leader & Software Architect in 4ward

AI&ML CONF

Kudos

Microsoft

Agenda

- ML.NET
  - Concepts
  - Architecture
  - Components
  - The ML process
- Code Examples
  - AutoML
    - Demo
  - Code Examples
    - Demo
  - MLOps
    - Demo
  - Other features

ML.NET

Concepts

ML.NET is an open-source and cross-platform machine learning framework (Windows, Linux, macOS), created by Microsoft, for .NET developers.

Using ML.NET, developers can leverage their existing tools and skillsets to develop and infuse custom AI into their applications by creating custom machine learning models for common scenarios like Sentiment Analysis, Product Recommendation, Sales forecasts, Price Predictions, Customers Segmentation, Image Classification and many more!

Build your own

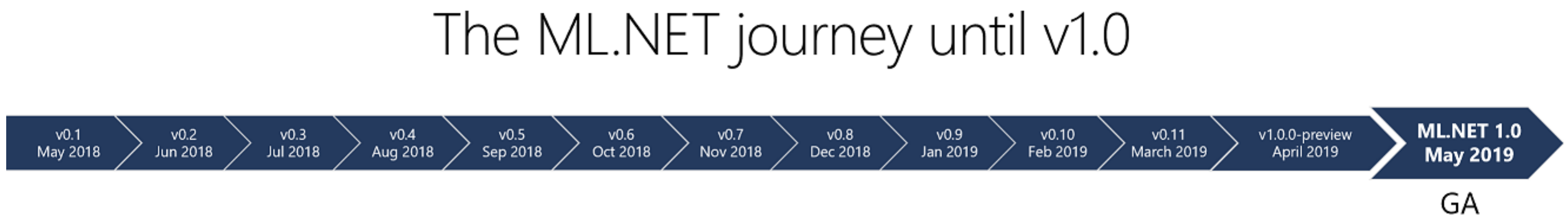
Proven & Extensible

Open Source

Developer Focused

Concepts

The journey



The ML.NET journey until v1.0

v0.1 May 2018

v0.2 Jun 2018

v0.3 Jul 2018

v0.4 Aug 2018

v0.5 Sep 2018

v0.6 Oct 2018

v0.7 Nov 2018

v0.8 Dec 2018

v0.9 Jan 2019

v0.10 Feb 2019

v0.11 March 2019

v1.0.0-preview April 2019

ML.NET 1.0 May 2019

GA

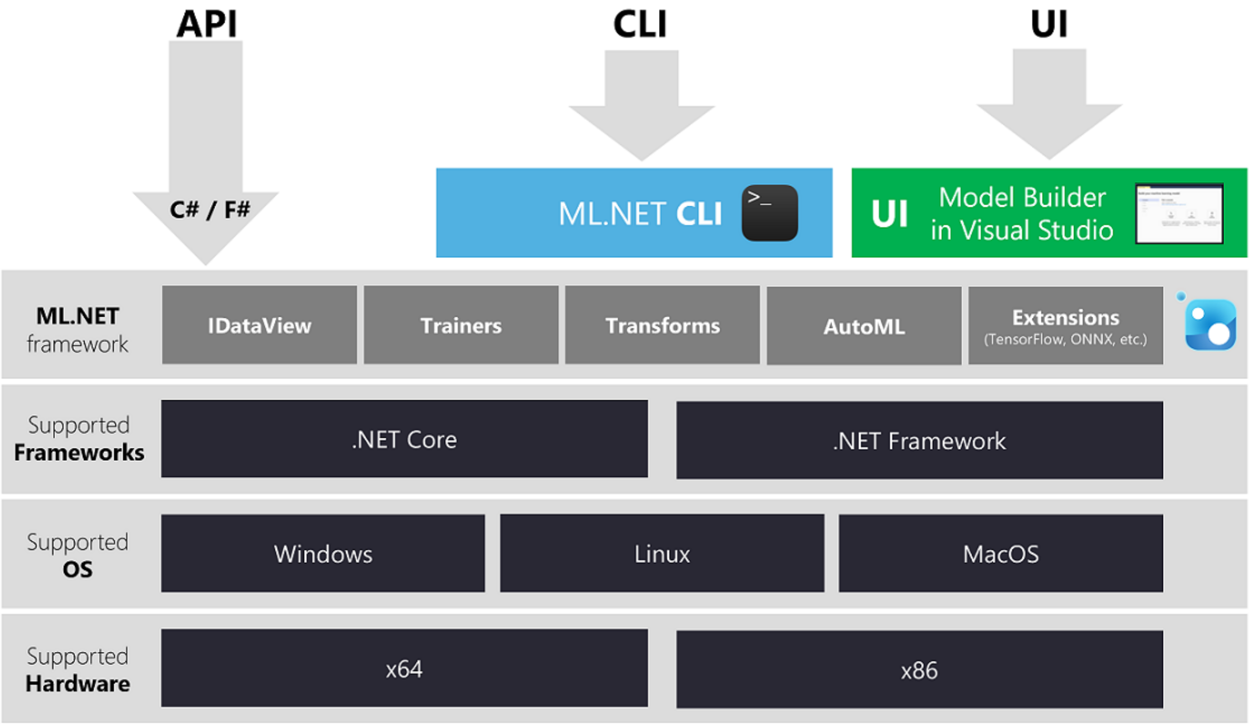
ML.NET Architecture

Hardware / Processor architecture support

- x64 bit is supported on all platforms.
- x86 is supported on Windows, except for TensorFlow, LightGBM, and ONNX related functionality.

Operating system and framework support:

- Windows, Linux, and macOS using .NET Core
- Windows using .NET Framework.



.NET versions support

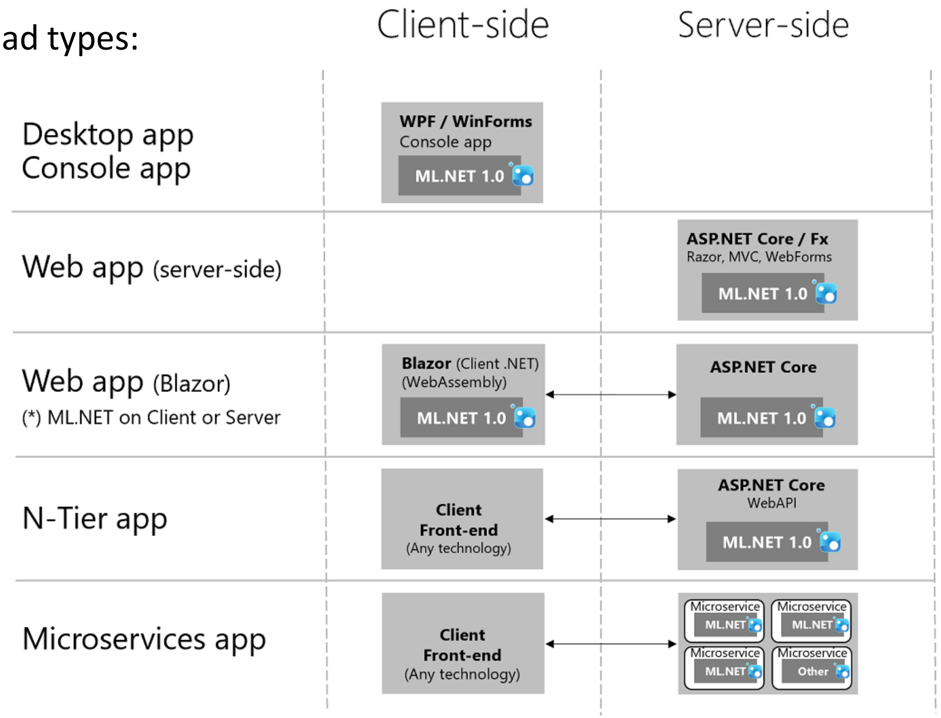
- .NET Core 2.1 or later is supported.
- .NET Framework 4.6.1 + supported, 4.7.2+ recommended.
- .NET Stardard 2.x (best choice for class libraries)

ML.NET Architecture

ML.NET 1.0 is composed by a set of NuGet packages. Therefore you can use it within your own .NET application wherever it is running (on-premises, cloud, client, server, etc.).

You can run ML.NET 1.0 NuGet packages as part of the following common application workload types:

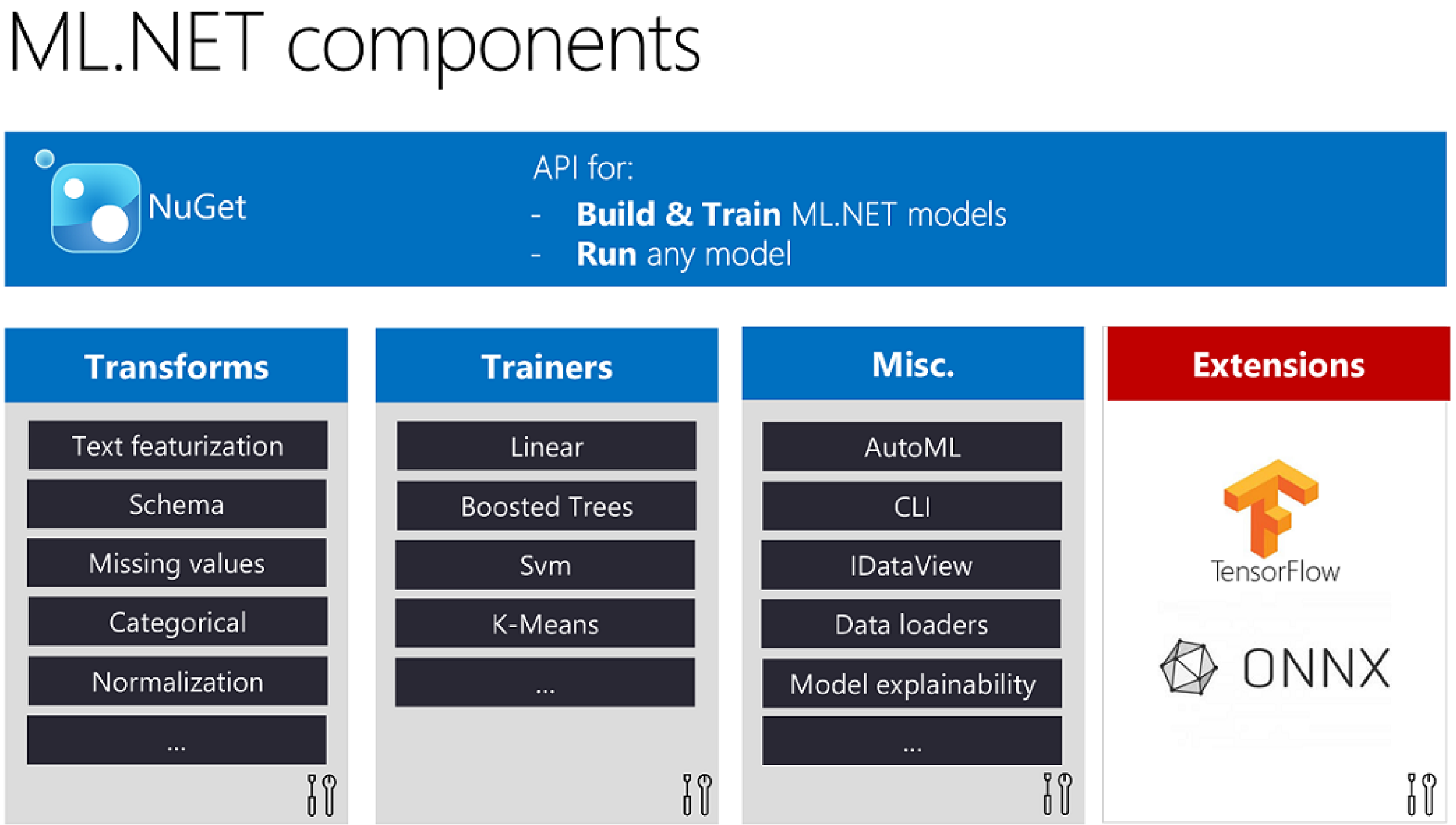
- ASP.NET Core web apps & WebAPI services
- ASP.NET web apps & WebAPI services
- Azure Functions
- Any other Azure app model app (server side)
- .NET WPF desktop app
- .NET WinForms desktop app
- .NET Core console app (usually for ML model training)
- .NET Framework console app (usually for ML model training)



	Client-side	Server-side
Desktop app Console app	**WPF / WinForms** Console app ML.NET 1.0	
Web app (server-side)		**ASP.NET Core / Fx** Razor, MVC, WebForms ML.NET 1.0
Web app (Blazor) (*) ML.NET on Client or Server	**Blazor** (Client .NET) (WebAssembly) ML.NET 1.0	**ASP.NET Core** ML.NET 1.0
N-Tier app	**Client** **Front-end** (Any technology)	**ASP.NET Core** WebAPI ML.NET 1.0
Microservices app	**Client** **Front-end** (Any technology)	Microservice ML.NET Microservice ML.NET Microservice ML.NET Microservice Other

ML.NET Components

ML.NET is being launched as a part of the .NET Foundation.
It has .NET API(s) for both model training and consumption, along with a variety of transforms and learners required for many popular ML tasks like regression and classification



ML.NET components

NuGet

API for:

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

Transforms

- Text featurization
- Schema
- Missing values
- Categorical
- Normalization
- …

Trainers

- Linear
- Boosted Trees
- Svm
- K-Means
- …

Misc.

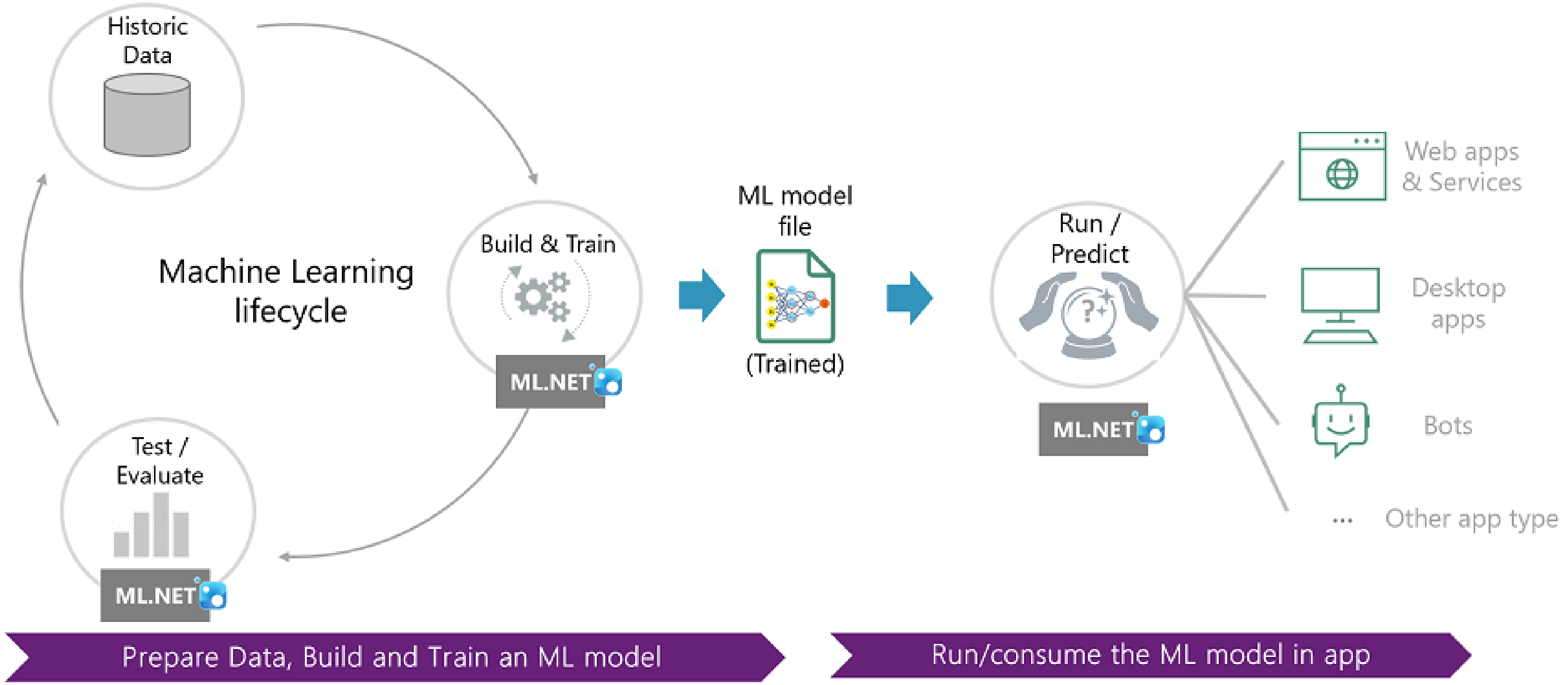
- AutoML
- CLI
- IDataView
- Data loaders
- Model explainability
- …

Extensions

- TensorFlow
- ONNX

The ML process

Process to Build & Run your own (custom) ML Model



AutoML

C#

ML.NET
API
(Code)

ML.NET
Model Builder
(Visual Studio UI)

>_

ML.NET
CLI
(Command-Line Interface)

Three ways to use ML.NET

AutoML

Approachable machine learning in Visual Studio

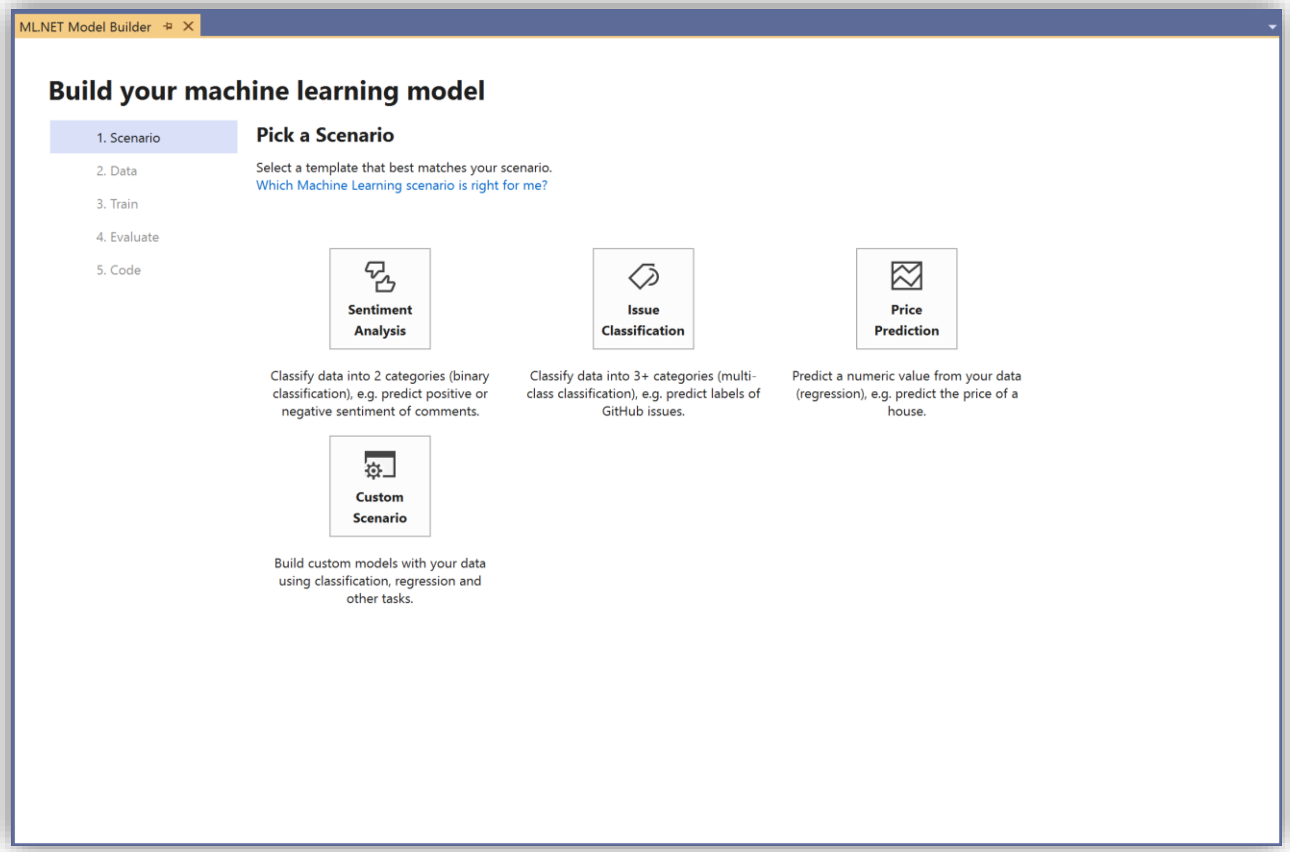
A simple UI to easily build custom ML models with Automated ML

Load from files and databases

Generate code for training and consumption

Run everything local

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



Build your machine learning model

Pick a Scenario

Select a template that best matches your scenario.
Which Machine Learning scenario is right for me?

Classify data into 2 categories (binary classification), e.g. predict positive or negative sentiment of comments.

Classify data into 3+ categories (multi-class classification), e.g. predict labels of GitHub issues.

Predict a numeric value from your data (regression), e.g. predict the price of a house.

Build custom models with your data using classification, regression and other tasks.

AutoML

ML Tasks supported by the CLI are the same than with AutoML:

- Regression
- Binary Classification
- Multi-class Classification

```
PS C:\>
```

Demo

AUTOML

Code Examples

Github labeler

Code Examples

Object detection example

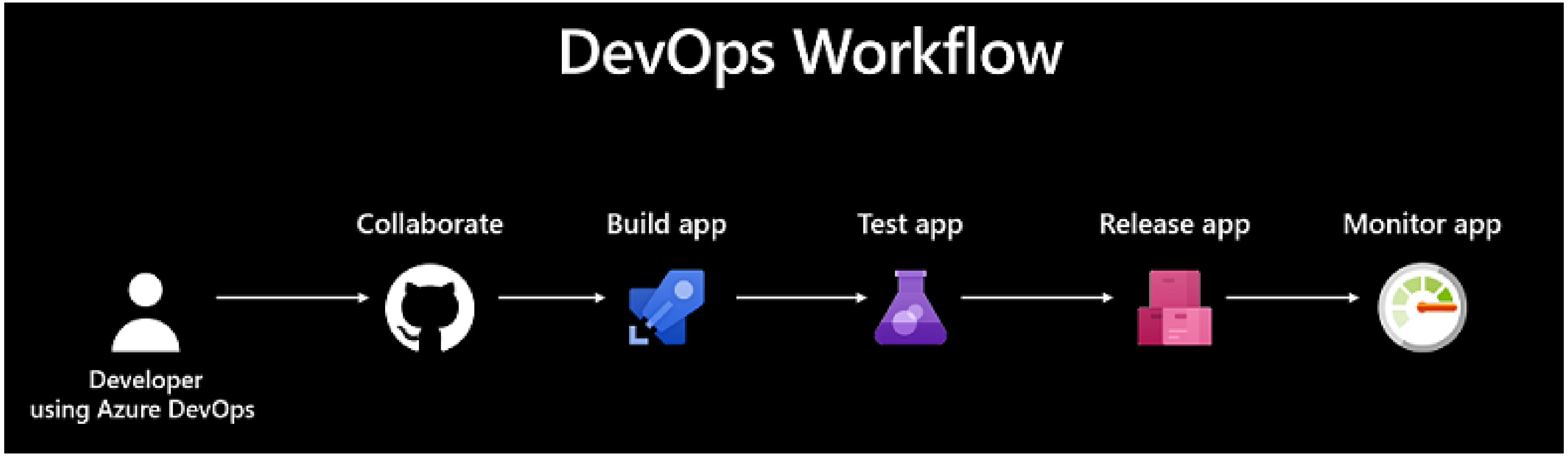
MLOps

Object detection example

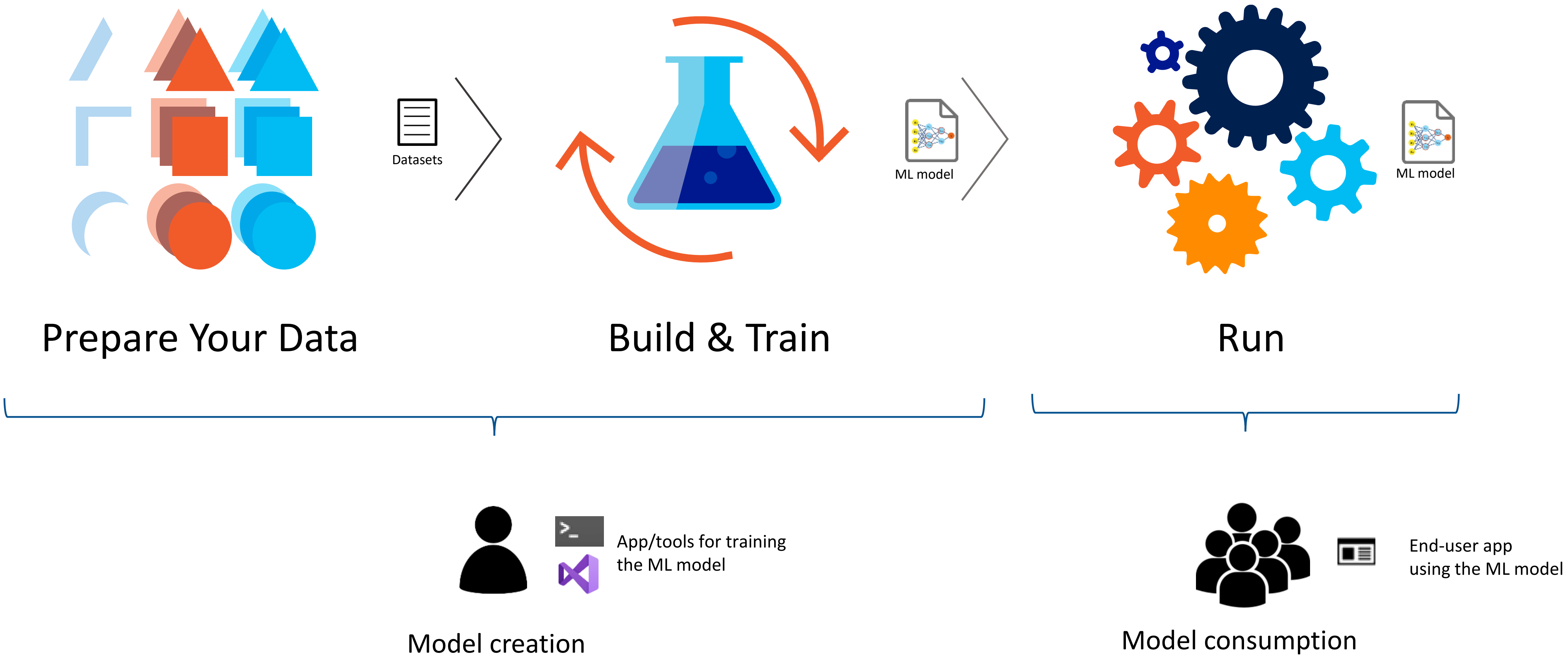
The MLOPS process

Normal Application LifeCycle

- Building
- Maintaining
- Continuously updating the end-user business application



Normal ML Workflow



Datasets

ML model

Prepare Your Data

Build & Train

Run

ML model

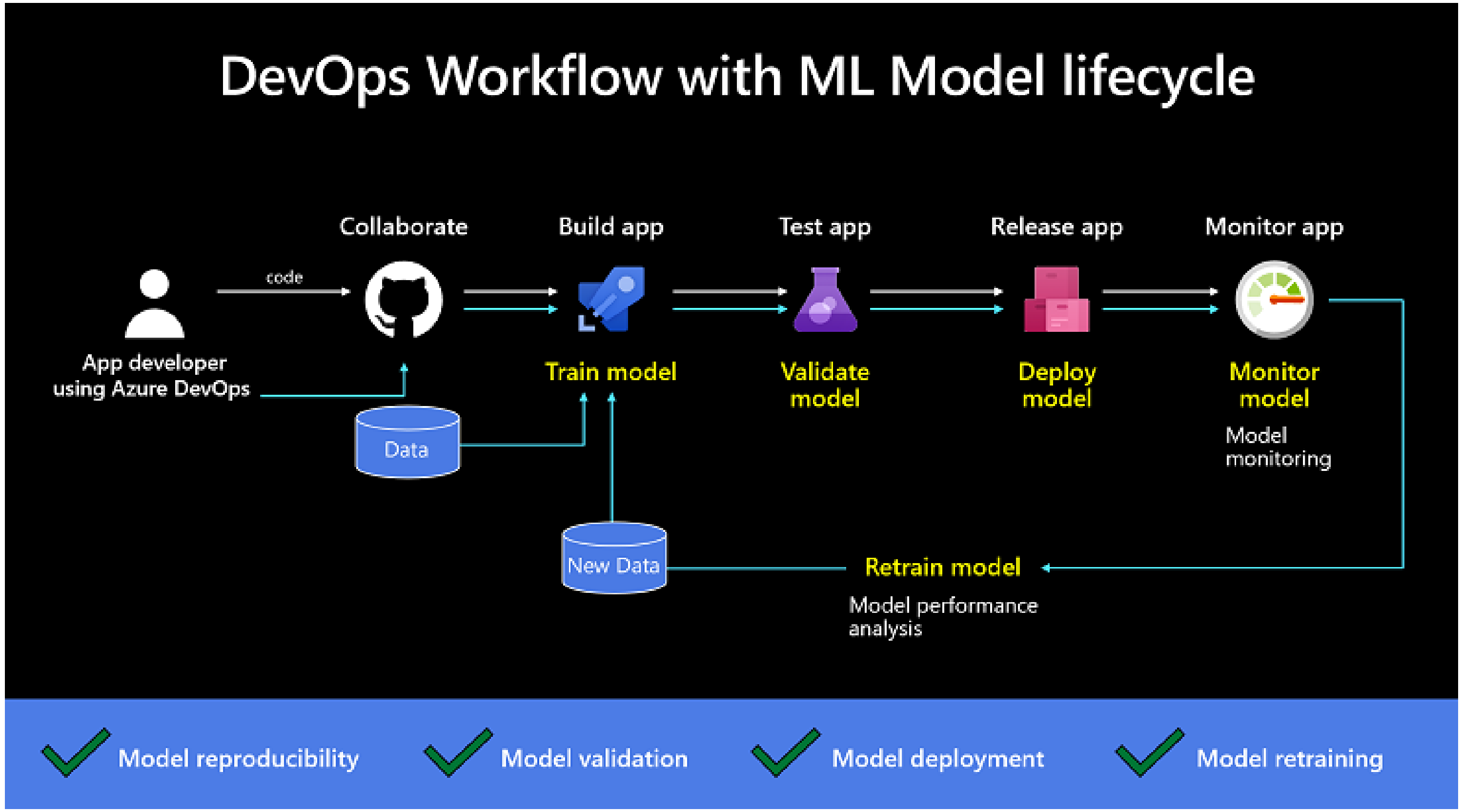
App/tools for training
the ML model

Model creation

End-user app
using the ML model

Model consumption

The MLOPS process



DevOps Workflow with ML Model lifecycle

App developer using Azure DevOps

code

Collaborate

Build app

Train model

Test app

Validate model

Release app

Deploy model

Monitor app

Monitor model

Model monitoring

Data

New Data

Retrain model

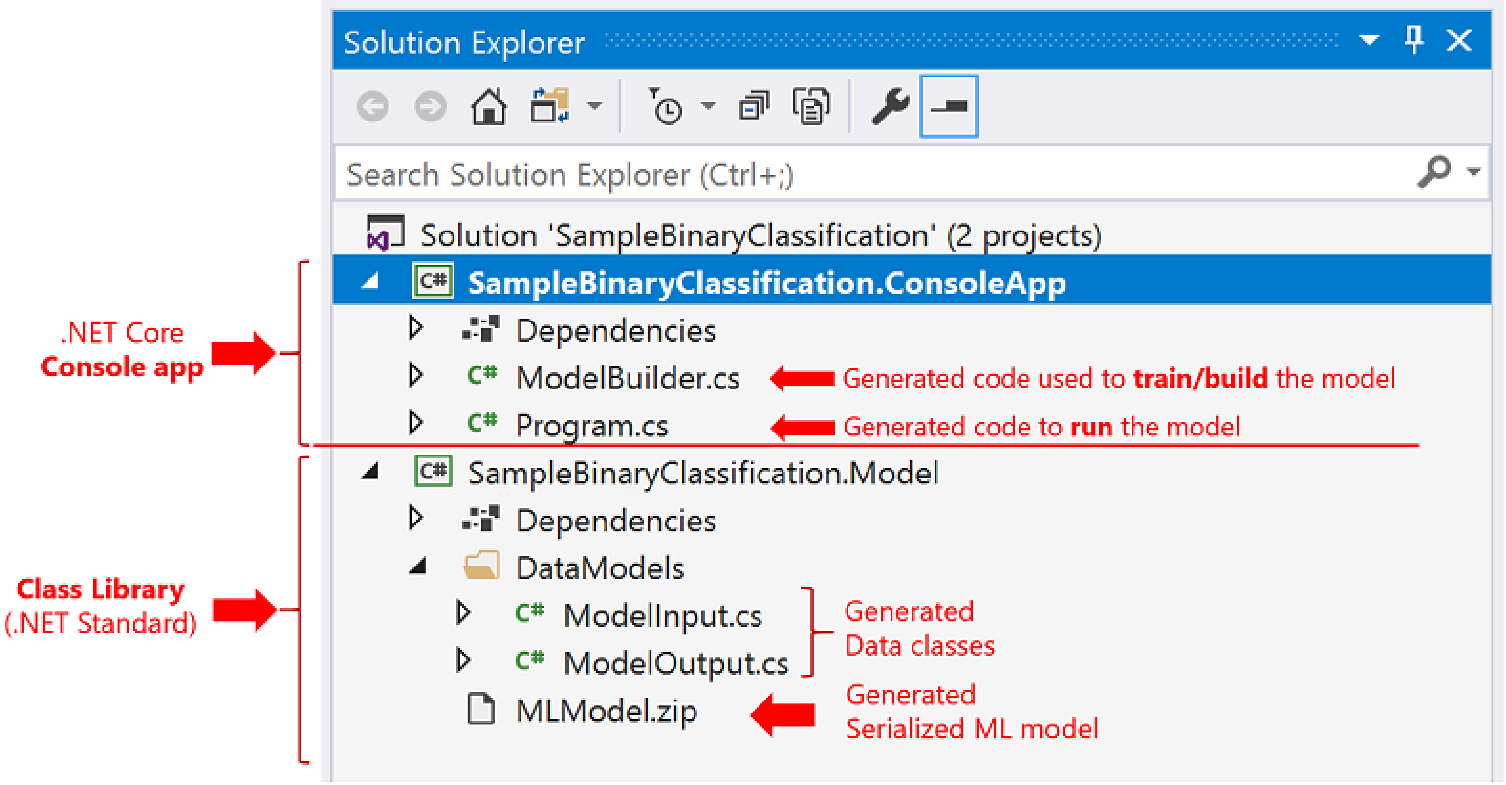
Model performance analysis

✓ Model reproducibility ✓ Model validation ✓ Model deployment ✓ Model retraining

DevOps Workflow with ML Model lifecycle

The MLOPS process

This is our sample ML model and trainer console app for the CI pipeline



Solution Explorer

Search Solution Explorer (Ctrl+;)

```
Solution 'SampleBinaryClassification' (2 projects)
  SampleBinaryClassification.ConsoleApp
    Dependencies
    ModelBuilder.cs
    Program.cs
  SampleBinaryClassification.Model
    Dependencies
    DataModels
      ModelInput.cs
      ModelOutput.cs
    MLModel.zip
```

.NET Core **Console app**

Class Library (.NET Standard)

Generated code used to **train/build** the model

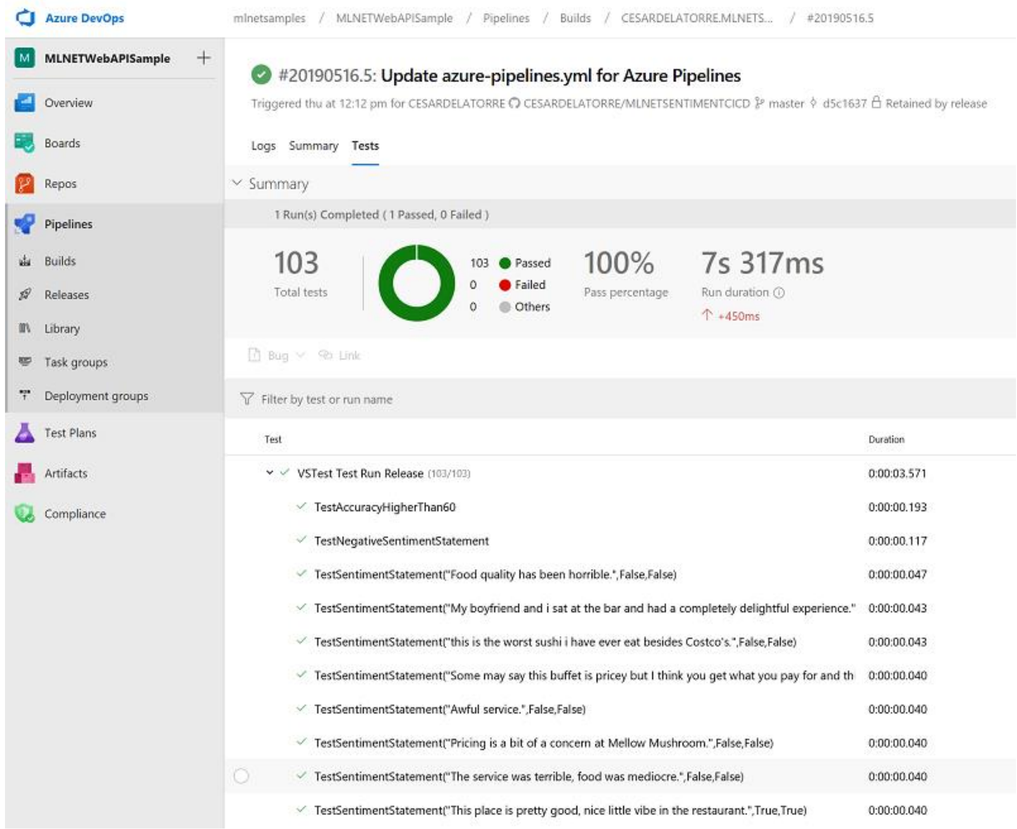
Generated code to **run** the model

Generated Data classes

Generated Serialized ML model

The MLOPS process

If your build pipeline passes all the tests you defined, than you are good to go and deploy the just trained/built ML.NET model.



#20190516.5: Update azure-pipelines.yml for Azure Pipelines

Triggered thu at 12:12 pm for CESARDELATORRE CESARDELATORRE/MLNETSENTIMENTCICD master d5c1637 Retained by release

Summary

1 Run(s) Completed (1 Passed, 0 Failed)

103 Total tests

103 ● Passed
0 ● Failed
0 ● Others

100% Pass percentage

7s 317ms Run duration

↑ +450ms

Test	Duration
✓ VSTest Test Run Release (103/103)	0:00:03.571
✓ TestAccuracyHigherThan60	0:00:00.193
✓ TestNegativeSentimentStatement	0:00:00.117
✓ TestSentimentStatement("Food quality has been horrible.",False,False)	0:00:00.047
✓ TestSentimentStatement("My boyfriend and i sat at the bar and had a completely delightful experience."	0:00:00.043
✓ TestSentimentStatement("this is the worst sushi i have ever eat besides Costco's.",False,False)	0:00:00.043
✓ TestSentimentStatement("Some may say this buffet is pricey but I think you get what you pay for and th	0:00:00.040
✓ TestSentimentStatement("Awful service.",False,False)	0:00:00.040
✓ TestSentimentStatement("Pricing is a bit of a concern at Mellow Mushroom.",False,False)	0:00:00.040
✓ TestSentimentStatement("The service was terrible, food was mediocre.",False,False)	0:00:00.040
✓ TestSentimentStatement("This place is pretty good, nice little vibe in the restaurant.",True,True)	0:00:00.040

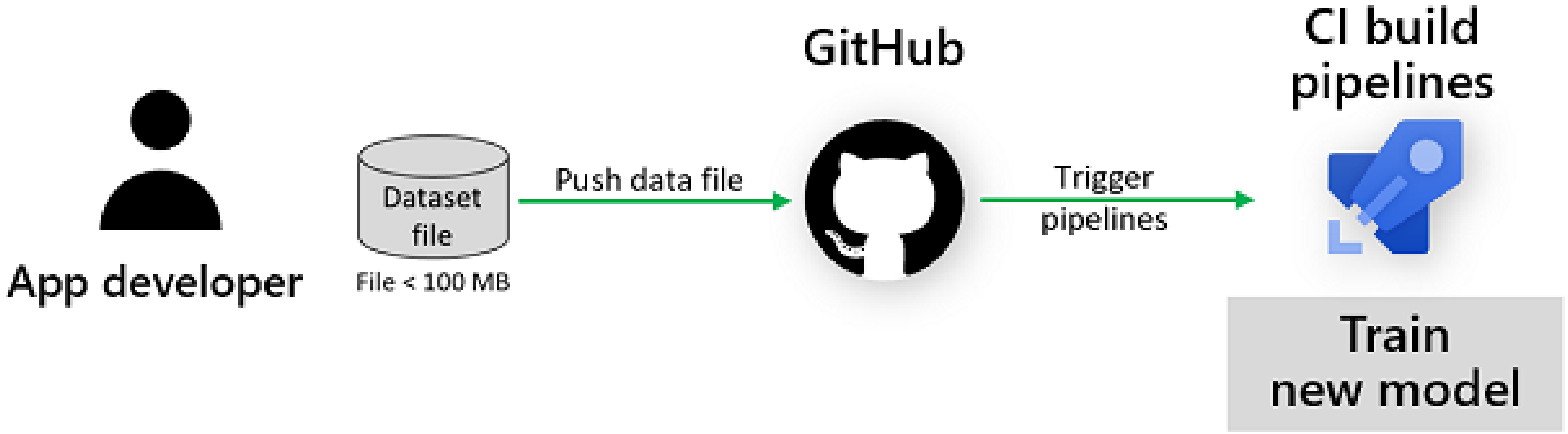
Unit tests from Azure DevOps dashboard

Demo

MLOPS pipeline examples

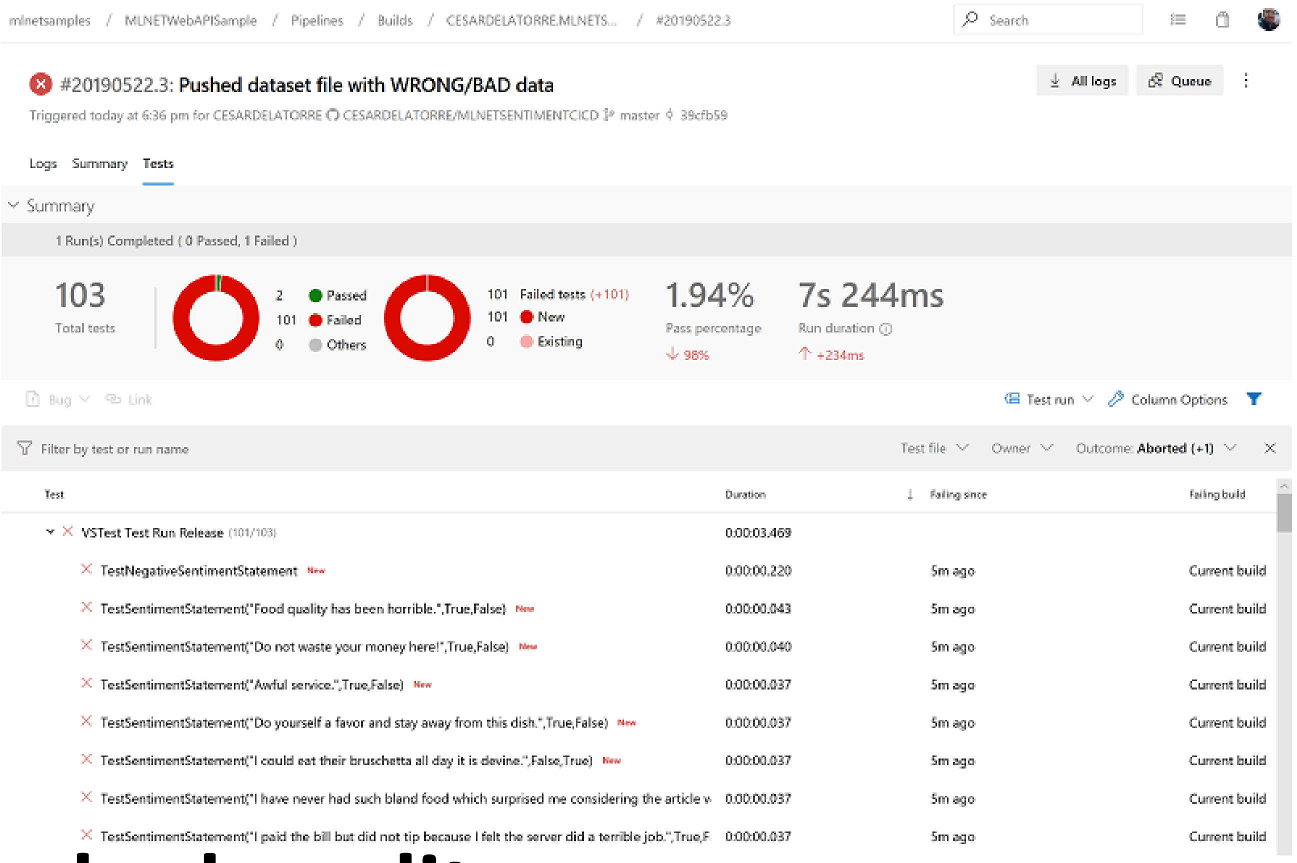
AI&ML CONF

The MLOPS process



«Providing new data» is the trigger

The MLOPS process



mlnetsamples / MLNETWebAPISample / Pipelines / Builds / CESARDELATORRE.MLNETS... / #20190522.3

#20190522.3: Pushed dataset file with WRONG/BAD data

Triggered today at 6:36 pm for CESARDELATORRE CESARDELATORRE/MLNETSENTIMENTCICD master 39cfb59

1 Run(s) Completed (0 Passed, 1 Failed)

103 Total tests

2 ● Passed
101 ● Failed
0 ● Others

101 Failed tests (+101)
101 ● New
0 ● Existing

1.94% Pass percentage ↓ 98%

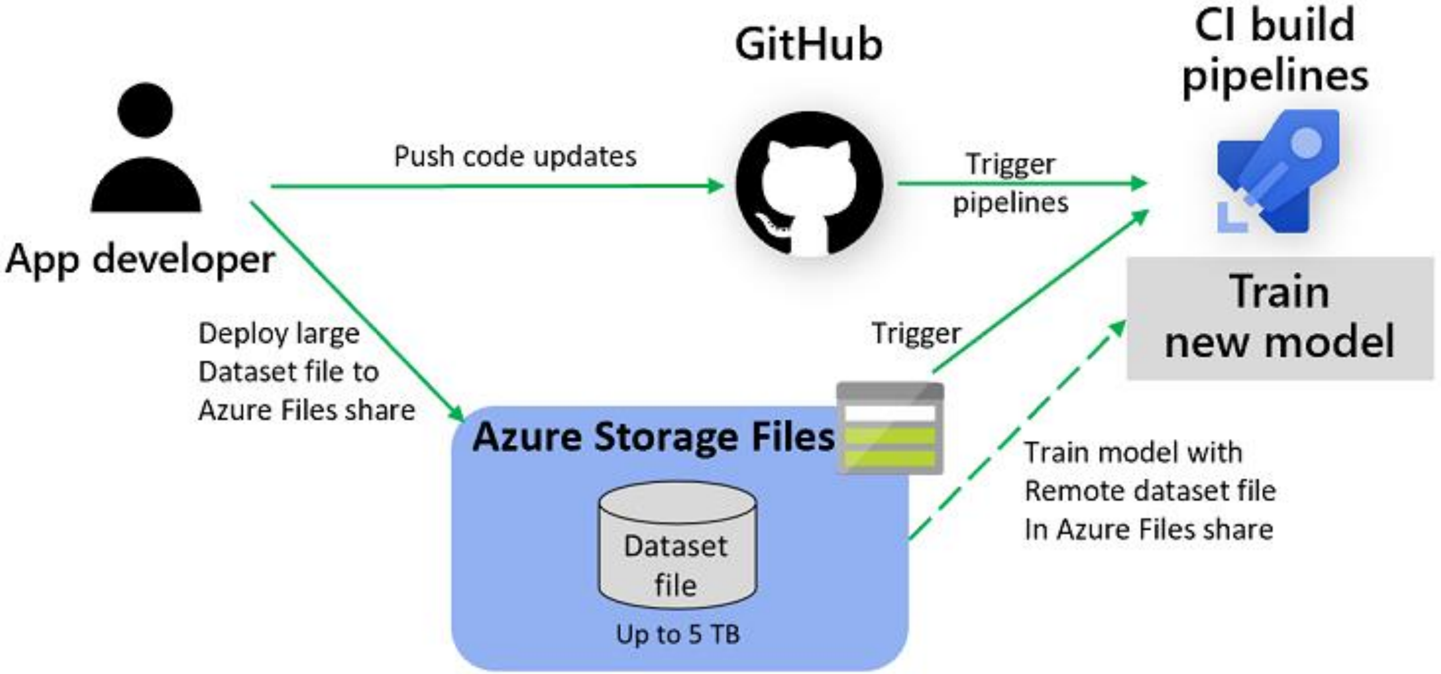
7s 244ms Run duration ↑ +234ms

Outcome: Aborted (+1)

Test	Duration	Failing since	Failing build
VSTest Test Run Release (101/103)	0:00:03.469		
TestNegativeSentimentStatement New	0:00:00.220	5m ago	Current build
TestSentimentStatement("Food quality has been horrible.",True,False) New	0:00:00.043	5m ago	Current build
TestSentimentStatement("Do not waste your money here!",True,False) New	0:00:00.040	5m ago	Current build
TestSentimentStatement("Awful service.",True,False) New	0:00:00.037	5m ago	Current build
TestSentimentStatement("Do yourself a favor and stay away from this dish.",True,False) New	0:00:00.037	5m ago	Current build
TestSentimentStatement("I could eat their bruschetta all day it is devine.",False,True) New	0:00:00.037	5m ago	Current build
TestSentimentStatement("I have never had such bland food which surprised me considering the article w	0:00:00.037	5m ago	Current build
TestSentimentStatement("I paid the bill but did not tip because I felt the server did a terrible job.",True,F	0:00:00.037	5m ago	Current build

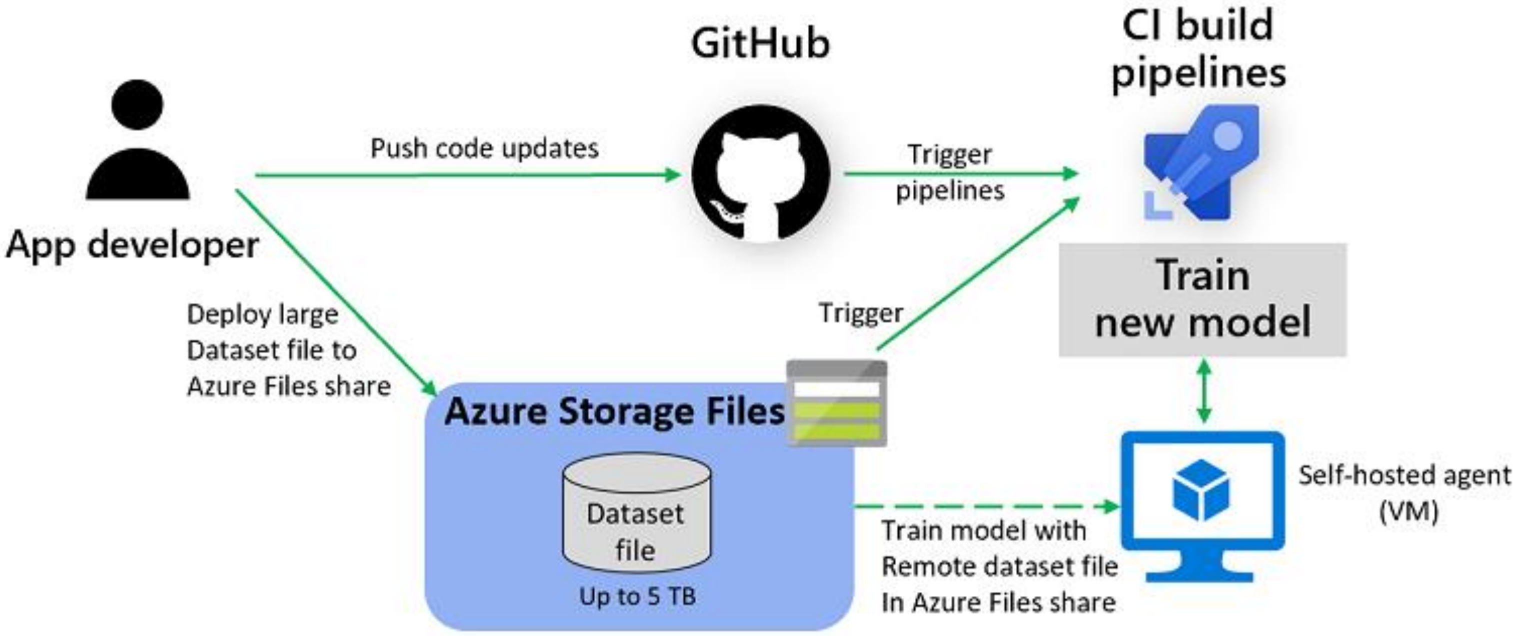
“bad data” == bad quality

The MLOPS process



Azure files can help us for large dataset

The MLOPS process



Azure DevOps self-hosted agent is better

Grazie!

marco.zamana1@gmail.com
@marco_zamana