



Build Intelligent apps with ML.NET and Windows Machine Learning

Ron Dagdag

@rondagdag

**Microsoft**[®]
Most Valuable
Professional

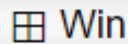

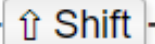

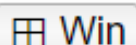

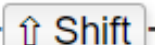


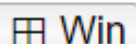

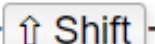


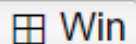




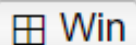

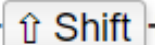


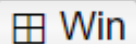




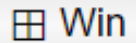

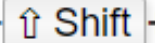


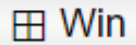

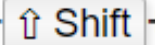


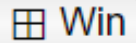

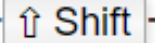


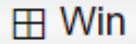






Award Categories
AI, Windows Development

First year awarded:
2017

Number of MVP Awards:
5

Have you ever felt like the
Windows Key?

-  Win +  Ctrl +  ↑ Shift +  Alt Opens website <https://www.office.com/?from=OfficeKey>.
-  Win +  Ctrl +  ↑ Shift +  Alt +  D Opens OneDrive.
-  Win +  Ctrl +  ↑ Shift +  Alt +  L Opens website LinkedIn.
-  Win +  Ctrl +  ↑ Shift +  Alt +  Y Opens website Yammer.
-  Win +  Ctrl +  ↑ Shift +  Alt +  N Opens OneNote.
-  Win +  Ctrl +  ↑ Shift +  Alt +  O Opens Outlook.
-  Win +  Ctrl +  ↑ Shift +  Alt +  P Opens PowerPoint.
-  Win +  Ctrl +  ↑ Shift +  Alt +  T Opens Teams.
-  Win +  Ctrl +  ↑ Shift +  Alt +  W Opens Word.
-  Win +  Ctrl +  ↑ Shift +  Alt +  X Opens Excel.

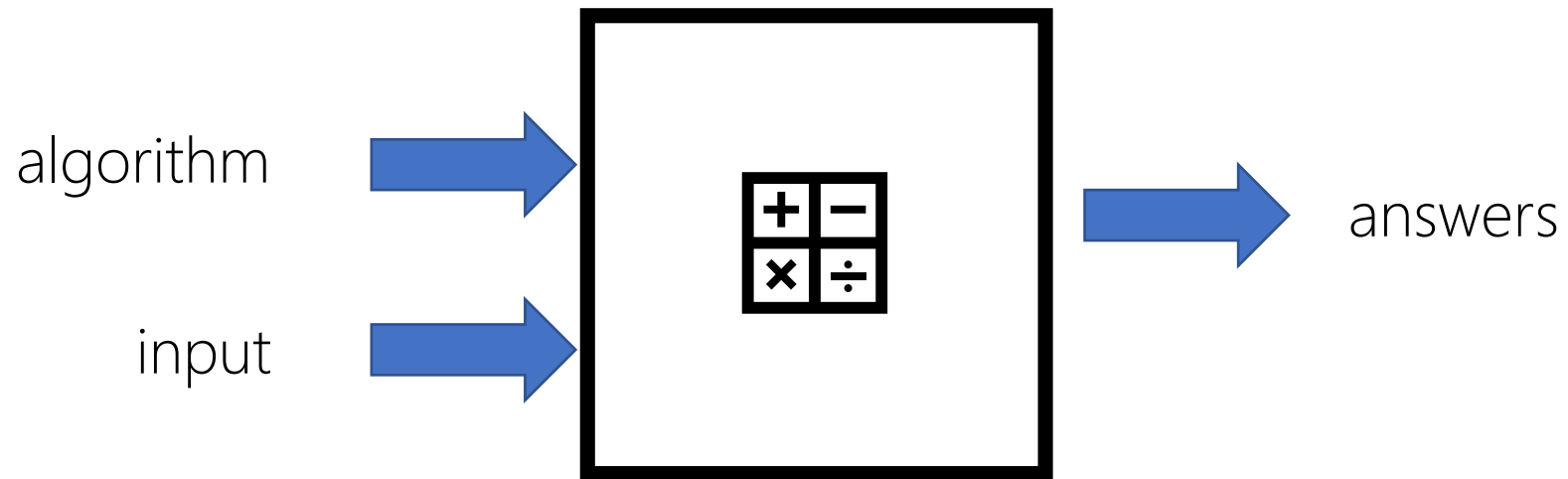


It felt like playing the    

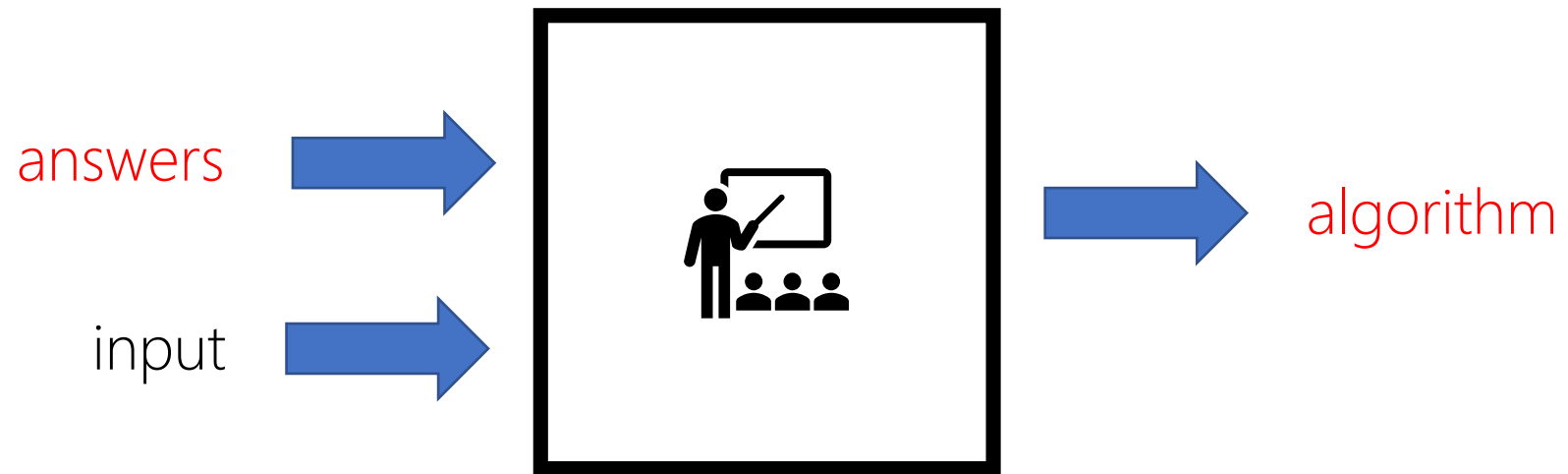
Agenda

- What is Machine Learning?
- Community Toolkit - Intelligent API
- Open Neural Network Exchange (ONNX)
- ONNX Runtime
- ML.NET Model Builder
- Windows Machine Learning
- Demo

programming



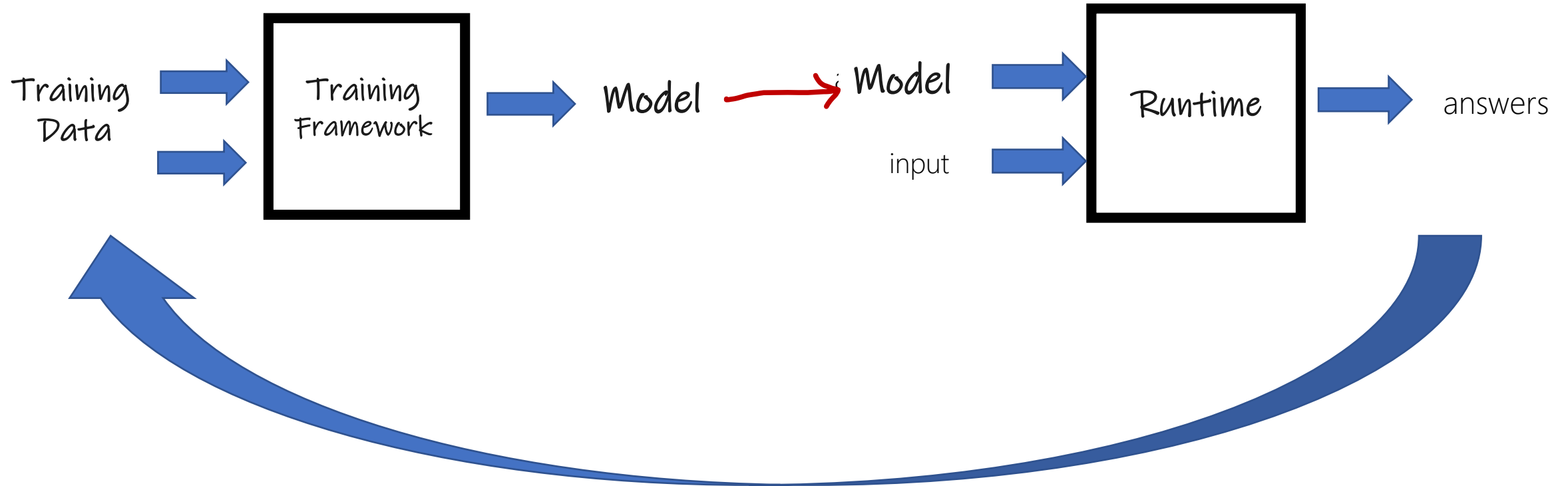
machine learning



ML Primer

machine Learning

Inferencing



Intelligent API

<https://github.com/CommunityToolkit/Labs-IntelligentAPIs>

Machine learning tasks easier for devs

No ML expertise need

Reuse existing ML models

Add Nuget package and calling a function

Inferencing machine learning models on Windows

Each APIs employs WinML



Intelligent API

<https://github.com/CommunityToolkit/Labs-IntelligentAPIs>

- Add a new nuget source with the feed URL
https://pkgs.dev.azure.com/dotnet/CommunityToolkit/_packaging/CommunityToolkit-Labs/nuget/v3/index.json
- Add nuget package to application
CommunityToolkit.Labs.Intelligent.ImageClassification
CommunityToolkit.Labs.Intelligent.ObjectDetection
CommunityToolkit.Labs.Intelligent.EmotionRecognition





Intelligent API

- Reference Library
using `CommunityToolkit.Labs.Intelligent.ImageClassification;`
- Call Classify Image
`List<ClassificationResult> list = await
SqueezeNetImageClassifier.ClassifyImage(selectedStorageFile, 3);`



Intelligent API DEMO



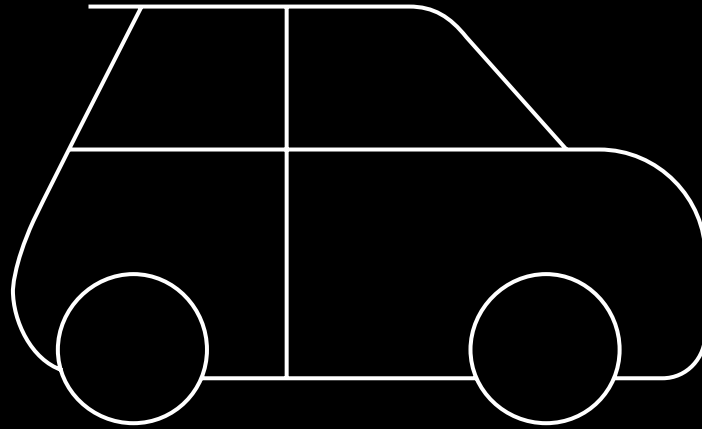
-  Win+↑ maximizes the active window.
-  Win+↓ restores the default window size or minimizes window.
-  Win+← or → align to the corresponding side of the screen.
-  Win+⇧ Shift+← or → to move the window to the next or previous monitor.

Window Shifts



AutoML with ML.NET

ML.NET CLI global tool accelerates productivity



How much is the taxi fare for 1 passenger going from Dallas to Fort Worth?

Getting started w/machine learning can be hard

ML.NET takes the guess work out of data prep, feature selection & hyperparameter tuning

Which features?

Distance

Trip time

Car type

Passengers

Time of day

...

Which algorithm?

Gradient Boosted

Nearest Neighbors

SGD

Bayesian Regression

LGBM

...

Which parameters?

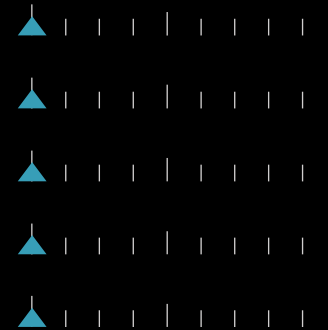
Criterion

Loss

Min Samples Split

Min Samples Leaf

XYZ



30%

Model

Getting started w/machine learning can be hard

ML.NET takes the guess work out of data prep, feature selection & hyperparameter tuning

Which features?

Distance

Trip time

Car type

Passengers

Time of day

...

Which algorithm?

Gradient Boosted

Nearest Neighbors

SGD

Bayesian Regression

LGBM

...

Which parameters?

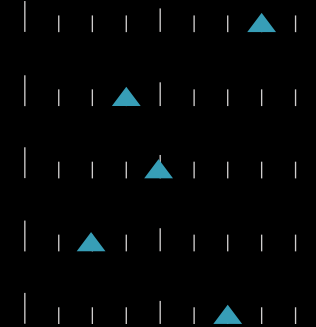
Neighbors

Weights

Min Samples Split

Min Samples Leaf

XX



30%

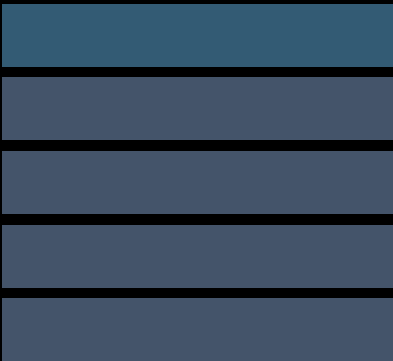
Model

Iterate

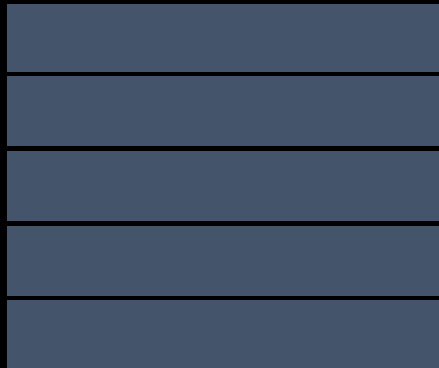
Getting started w/machine learning can be hard

ML.NET takes the guess work out of data prep, feature selection & hyperparameter tuning

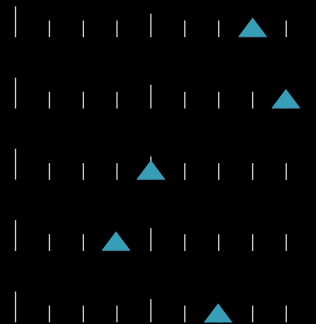
Which features?



Which algorithm?



Which parameters?



30%

15%

Iterate

ML.NET accelerates model development

Input

101010
010101
101010

Enter data

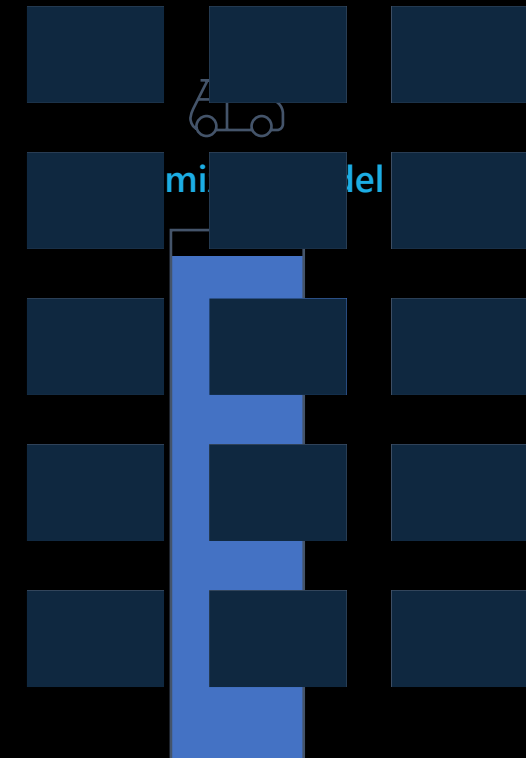
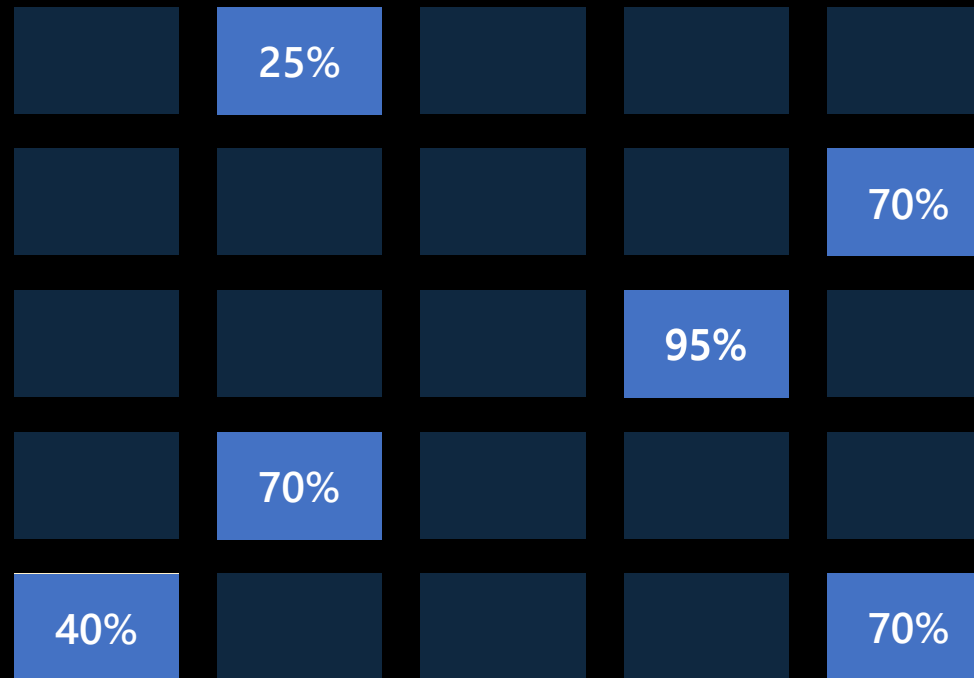


Define goals



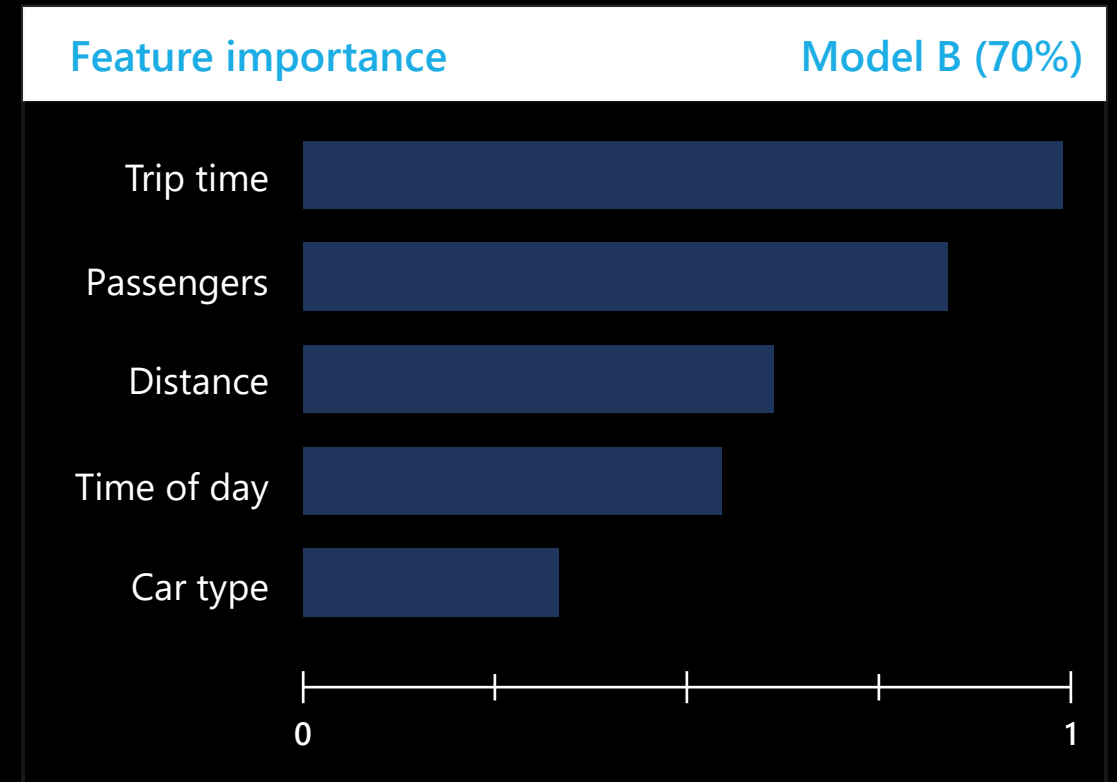
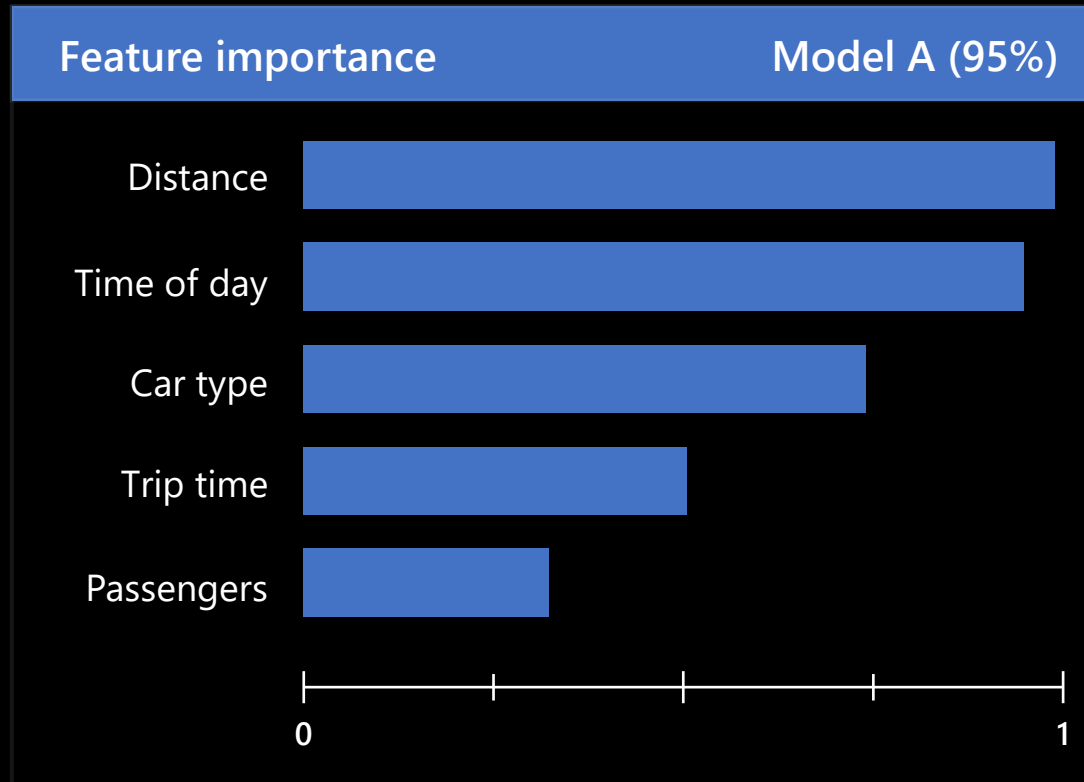
Apply constraints







Intelligently test multiple models in parallel






ML.NET accelerates model development

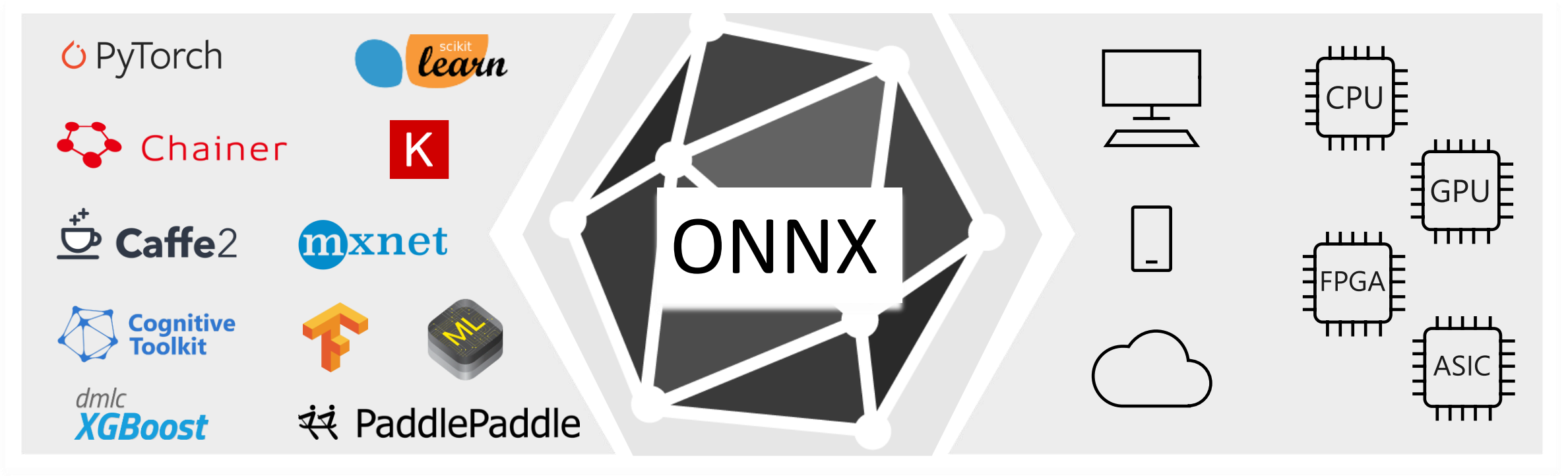
with **model explainability**



-  Win opens the Start Menu
-  Win+D hide/shows the desktop
-  Win+E opens Windows Explorer
-  Win+F opens Find files and folders
-  Win+M minimizes all windows
-  Win+L locks the desktop

Windows   

Open and Interoperable AI



When to use ONNX?

Trained in Python - deploy into a C#/Java/Javascript app

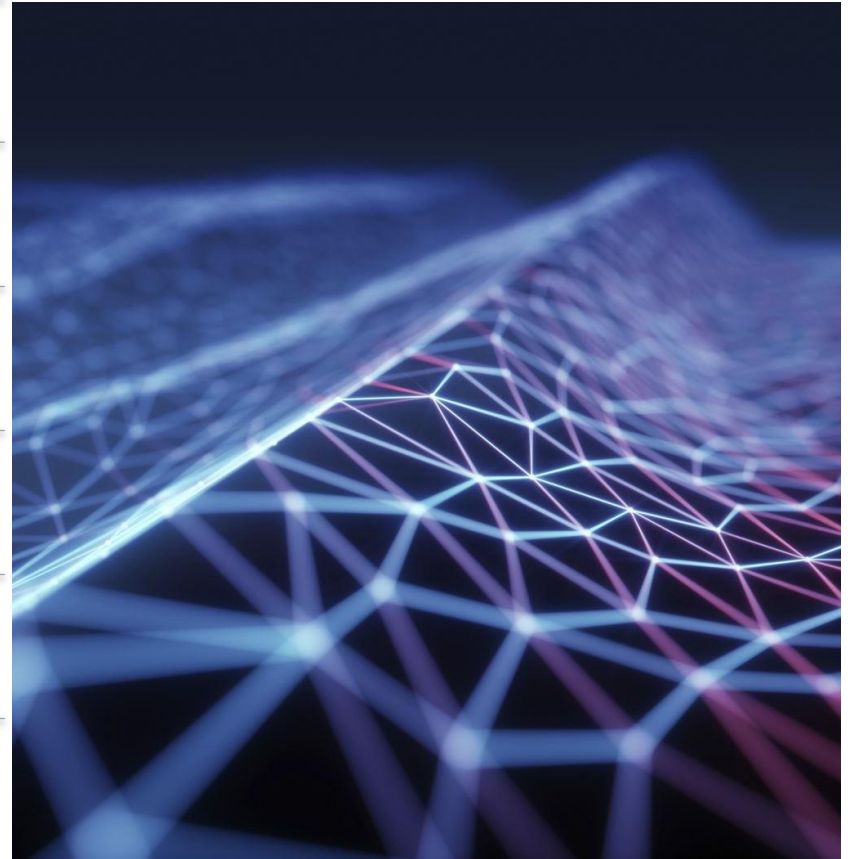
High Inferencing latency for production use

Model to run resource on IoT/edge devices

Model to run on different OS or Hardware

Combine models created from different frameworks

Training takes too long (transformer models)



ONNX Runtime

onnxruntime.ai

Optimize Inferencing

Optimize Training

Platform

Windows

Linux

Mac

Android

iOS

Web Browser
(Preview)

API

Python

C++

C#

C

Java

JS

Obj-C

WinRT

Architecture

X64

X86

ARM64

ARM32

IBM Power

Hardware Acceleration

Default CPU

CoreML

CUDA

DirectML

oneDNN

OpenVINO

TensorRT

NNAPI

ACL (Preview)

ArmNN
(Preview)

MIGraphX
(Preview)

TVM (Preview)

Rockchip NPU
(Preview)

Vitis AI (Preview)

Installation Instructions

Install Nuget package [Microsoft.ML.OnnxRuntime.DirectML](#)

- ☐ Win+P brings up projection settings
- ☐ Win++ to zoom into the screen at the mouse cursor position
- ☐ Win+- to zoom out if the Magnifier Utility is running.
- ☐ Win+Esc to exit zoom.

🔍 Zooming In and Out 🔍



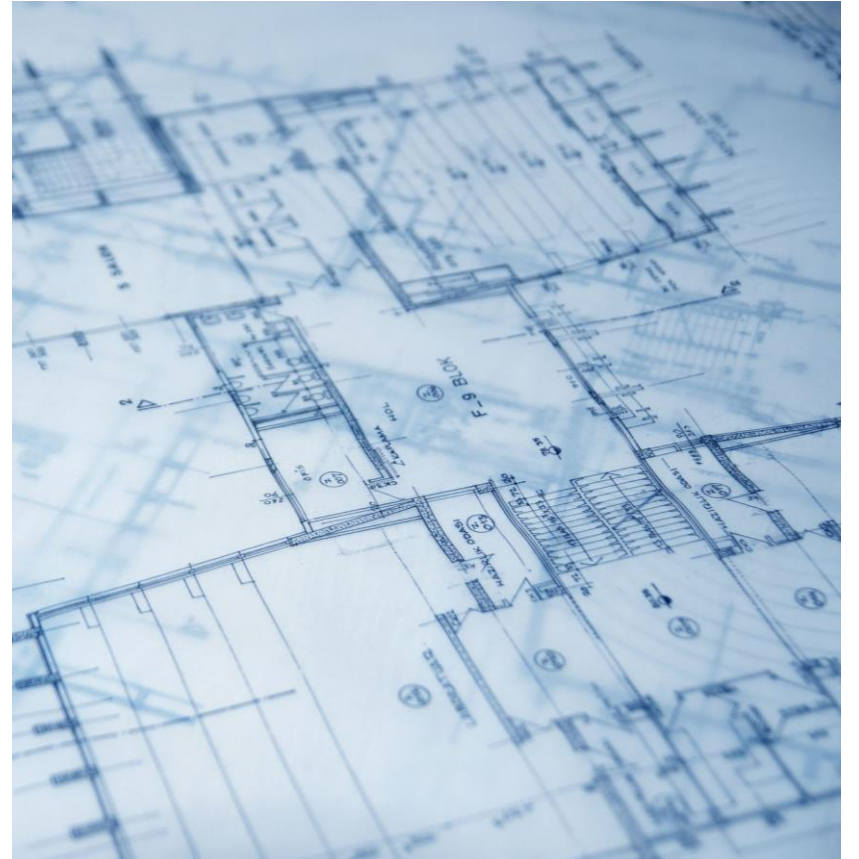
ML.NET

- machine learning to .NET applications
- Add automatic predictions to apps
- online or offline
- ML.NET can generate machine learning **model**.
- model - steps to transform input data into a prediction
- import pre-trained TensorFlow and ONNX models
- Supports Windows, Linux, and macOS

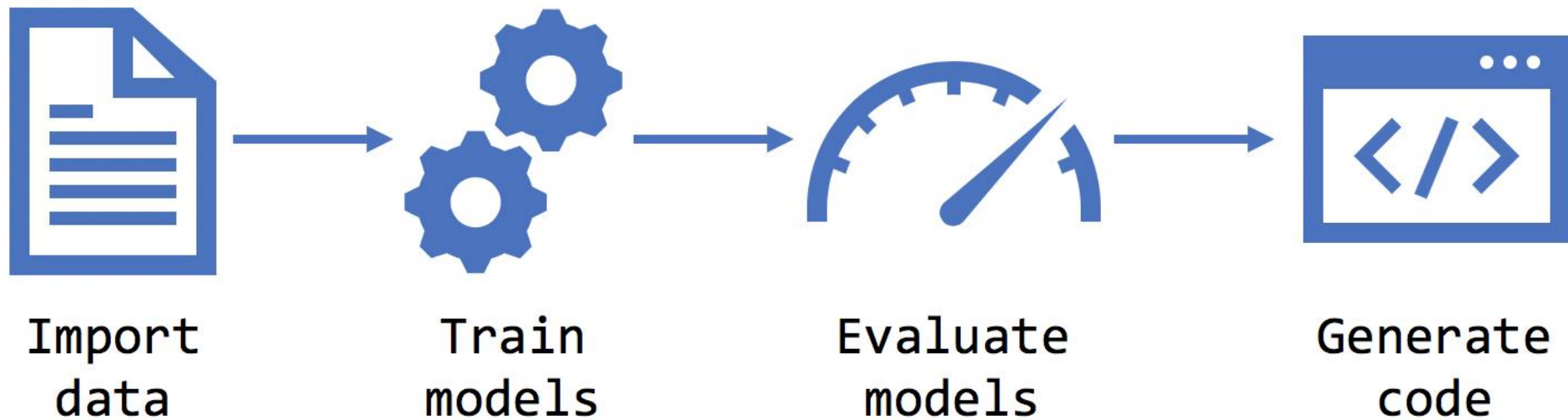


ML.NET Model Builder

- Simple UI tool in Visual Studio
- Runs locally to build, train and ship ML projects
- build/train in Azure
- Generates Custom ML models



Model Builder



Model Builder

Model Builder supports the following environment options:

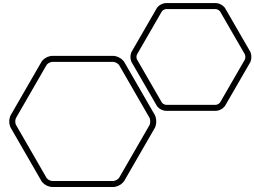
Scenario	Local CPU	Local GPU	Azure GPU
Data classification	✓	✗	✗
Value prediction	✓	✗	✗
Image classification	✓	✓	✓
Recommendation	✓	✗	✗
Object detection	✗	✗	✓

Model Builder

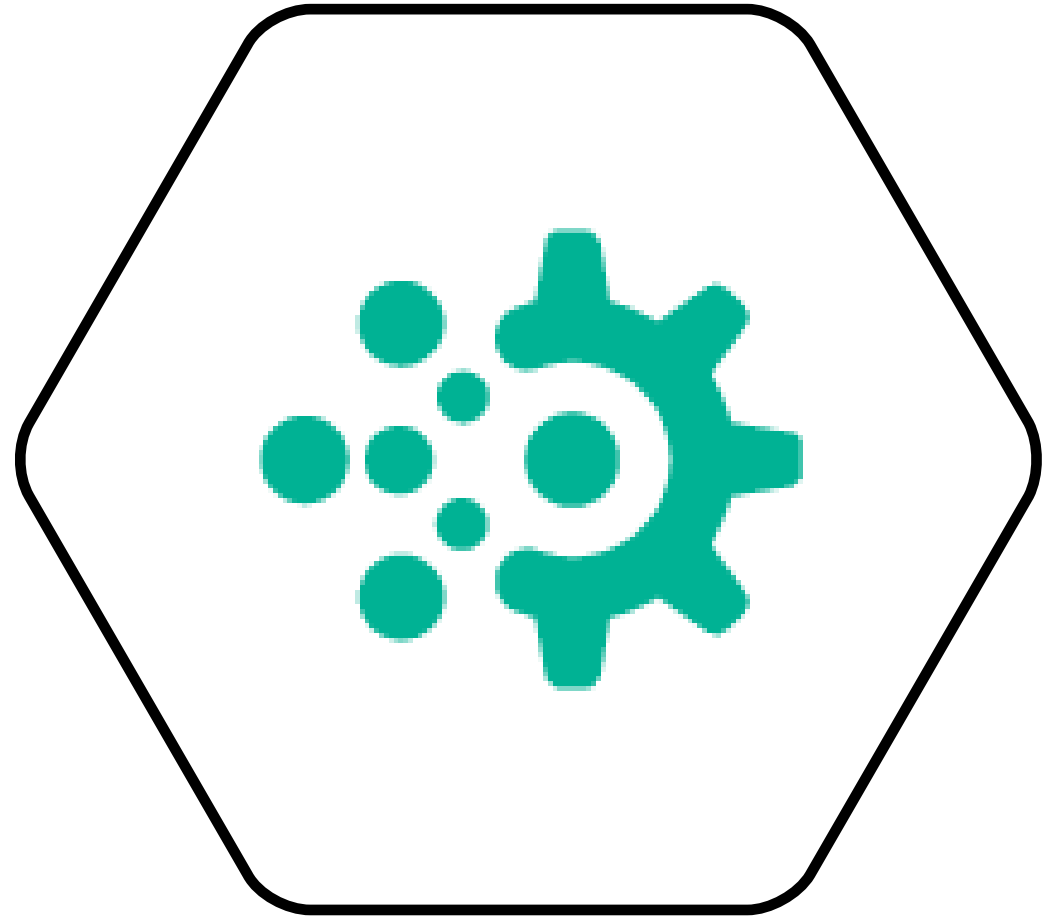
Dataset size	Average time to train
0 - 10 MB	10 sec
10 - 100 MB	10 min
100 - 500 MB	30 min
500 - 1 GB	60 min
1 GB+	3+ hours




These numbers are a guide only. The exact length of training is dependent on:

- the number of features (columns) being used to as input to the model
- the type of columns
- the ML task
- the CPU, disk, and memory performance of the machine used for training



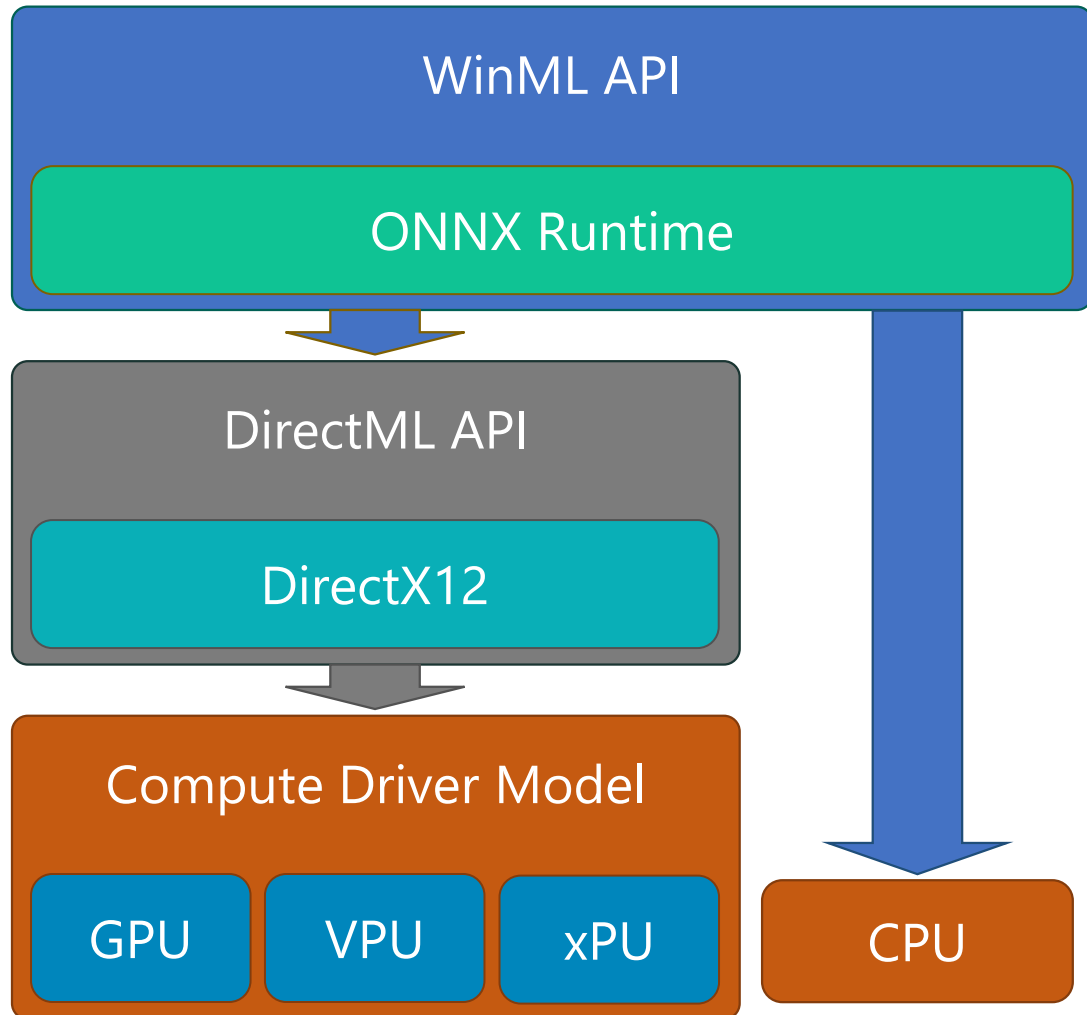
Model Builder DEMO



-  Win+Ctrl+D creates a new virtual desktop.
-  Win+Ctrl+F4 closes the active virtual desktop.
-  Win+Ctrl+← or → switches between virtual desktops.

What happened to my window   

Windows AI platform



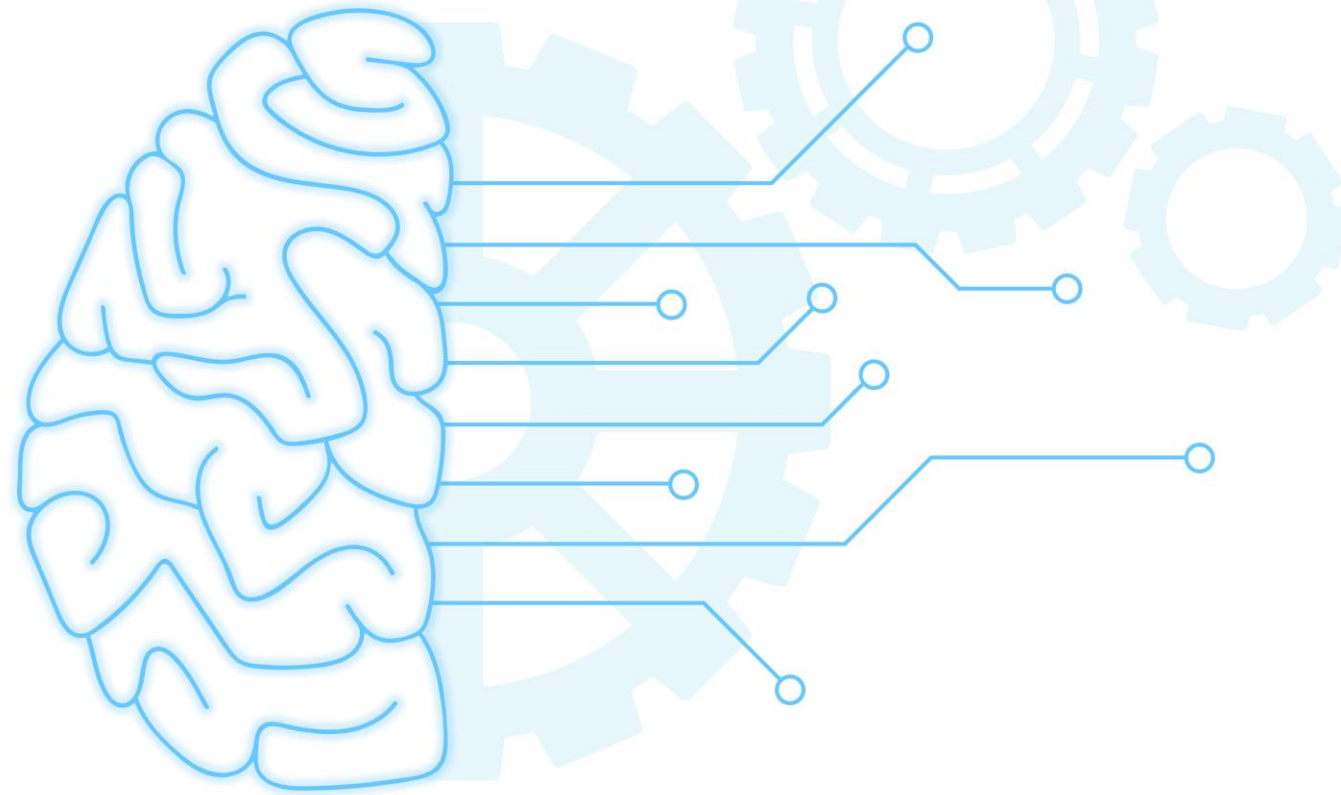
- WinML
 - **Practical**, simple model-based API for ML inferencing on Windows
- DirectML
 - **Realtime, high control** ML operator API; part of DirectX family
- Compute Driver Model
 - Robust **hardware reach**/abstraction layer for compute and graphics silicon

Windows Machine Learning (WinML)

- Ease of development
- Abstract model-specific code away
- Broad hardware support
- Performs hardware optimizations
- Implement Machine Learning in Windows apps using Windows ML

Windows Machine Learning (WinML)

- Improve performance significantly on Windows
- high-performance
- Low latency, real-time results
- Increased flexibility
- Reduced operational costs
- Reliable API for deploying hardware-accelerated ML inferences on Windows devices





DEMO

Summary

- What is Machine Learning?
 - training and inferencing
- Open Neural Network Exchange (ONNX)
 - ML Model file like pdf
- ONNX Runtime
 - API to use onnx models into apps

Summary

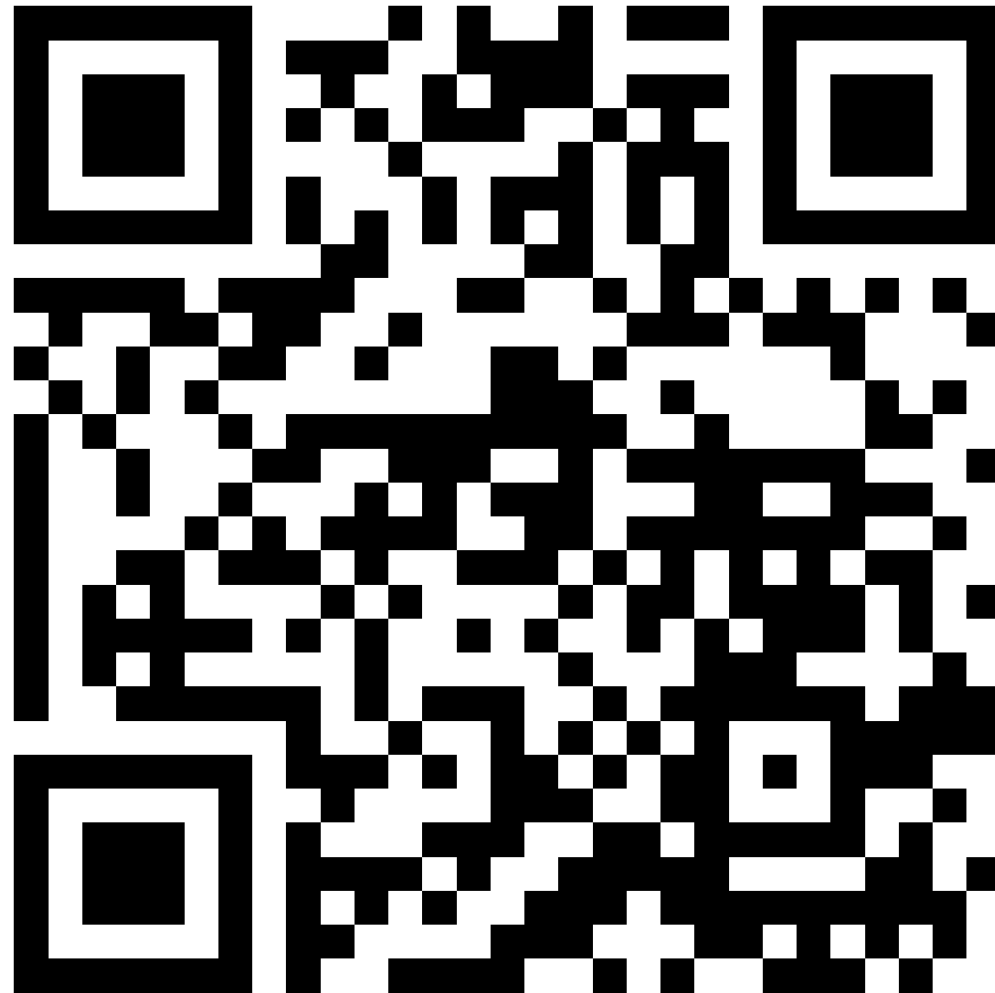
- Community Toolkit - Intelligent API
 - Nuget package to add computer vision models to win apps
- ML.NET Model Builder
 - Generates Custom ML models in Visual Studio
- Windows Machine Learning
 - Implement ML in Windows apps

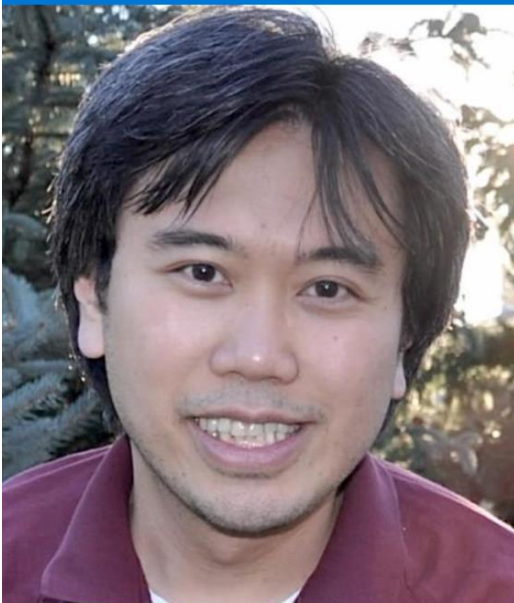
-  Win+. or ; opens the emoji panel while typing
-  Win+↑ Shift+S - Opens Snip & Sketch tool to capture screen selection and puts into clipboard.

Have Fun With Emoji 🎉 🎈



<https://github.com/rondagdag/mlnet-modelbuilder-talk>





Award Categories

AI, Windows Development

First year awarded:

2017

Number of MVP Awards:

5

About Me

Ron Dagdag

Lead Software Engineer at Spacee

5th year Microsoft MVP awardee

www.dagdag.net

ron@dagdag.net
[@rondagdag](https://twitter.com/rondagdag)

Linked In
www.linkedin.com/in/rondagdag/

Thanks for geeking out with me about Windows Keys, ML.NET, Windows AI

[@rondagdag](https://twitter.com/rondagdag)

<https://linktr.ee/rondagdag>