Machine Learning with

A MACHINE LEARNING FRAMEWORK FROM MICROSOFT

Praveen Raghuvanshi

Introduction

- Cloud Architect @ Harman
- Domain: Audio, Video, Control
- Area of Expertise: Cloud, Distributed computing
- Area of Interest: AI/ML, Cloud and IoT

Prerequisites

http://bit.ly/2DijYAI

Poll

live.voxvote.com

Identify language

Bonjour

Guten Morgen

おはよ ございます

Good Morning

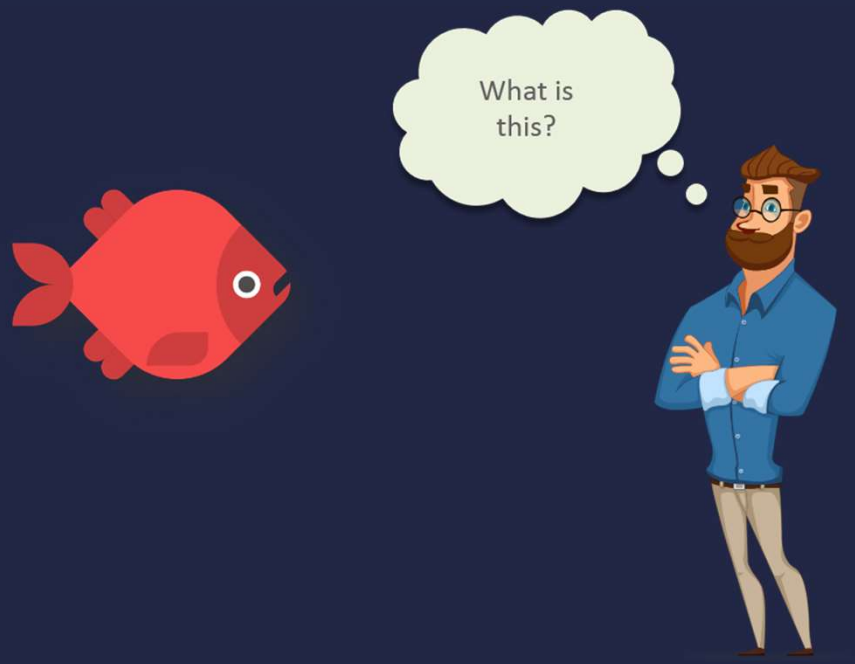
Programming Languages

Source: https://insights.dice.com/2019/10/24/tiobe-top-eight-top-ten-open/

Quiz : Type of ML

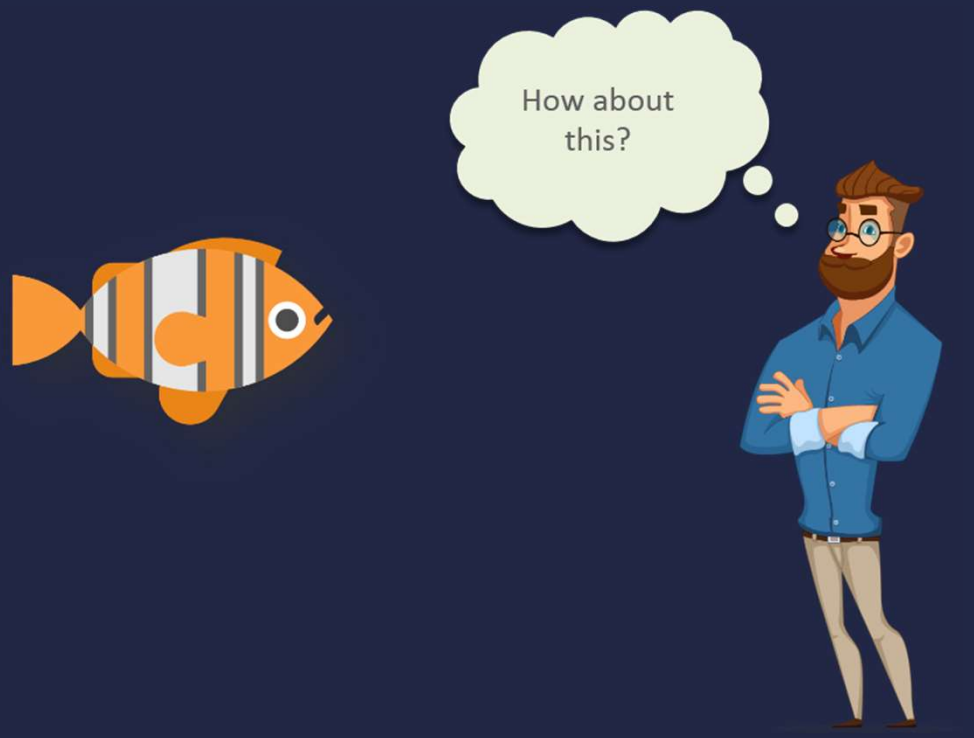
- Is this A or B?
- How much – or – How many?
- How is this organized?
- What should I do next?
- Is number a Armstrong?

Machine Learning



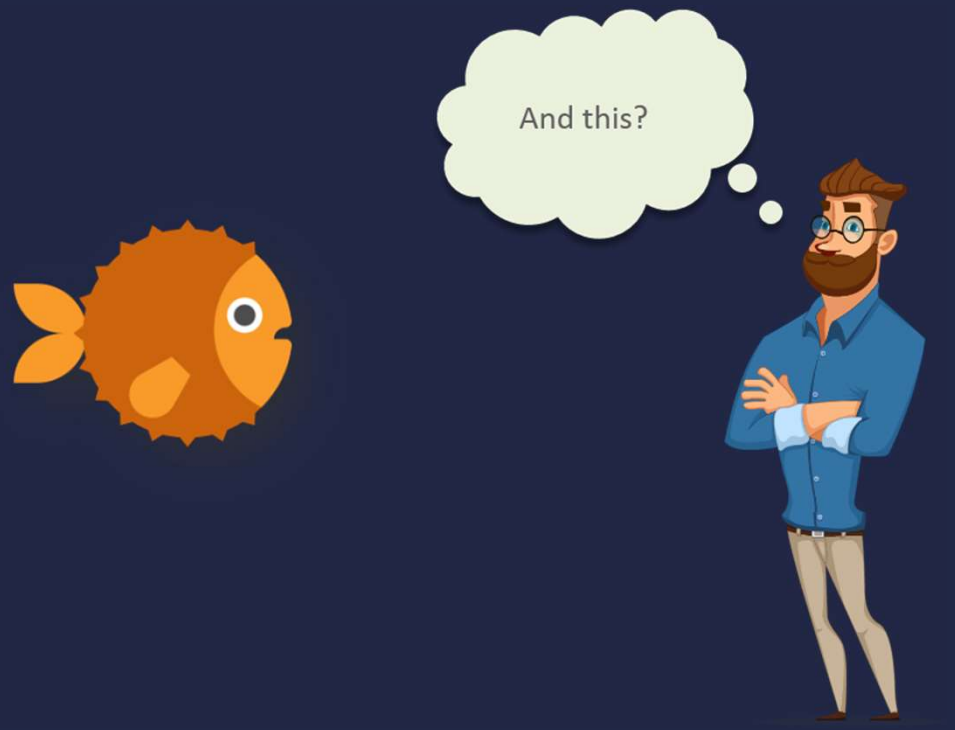
Intellpaat: https://www.quora.com/What-are-prerequisites-to-start-learning-machine-learning

Machine Learning



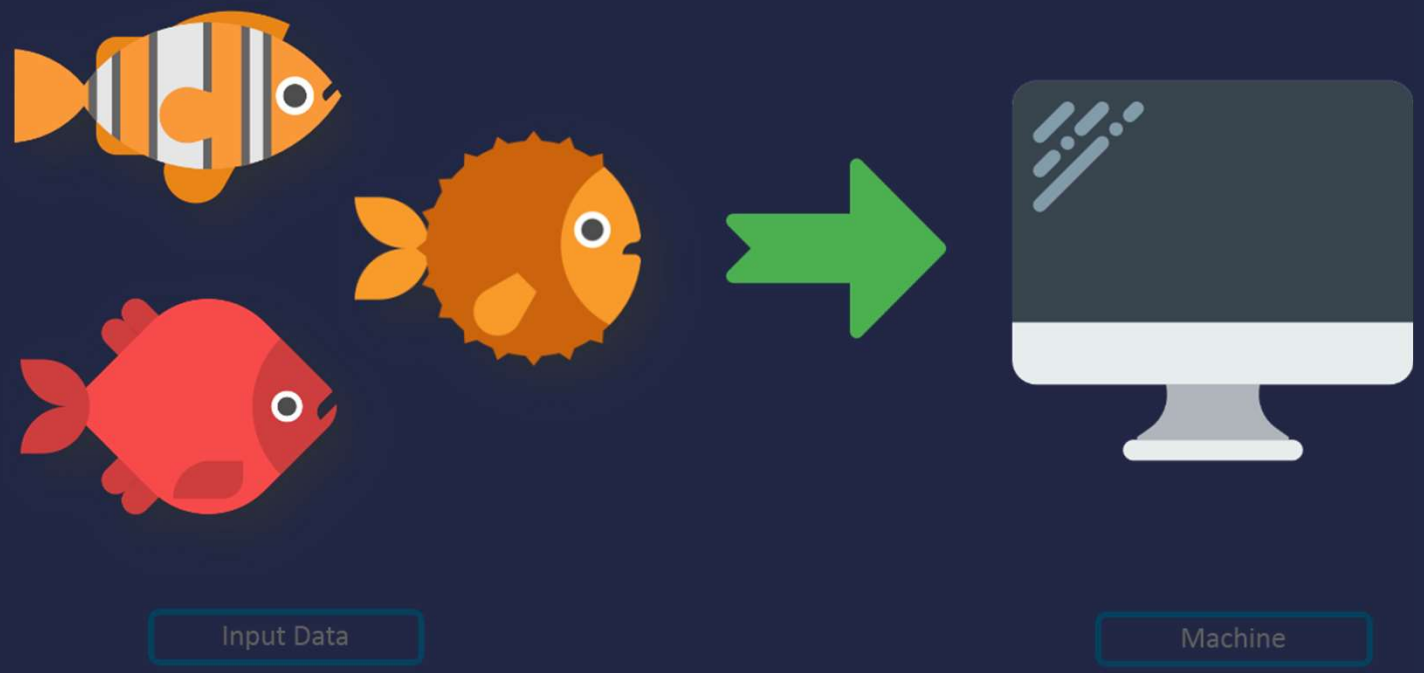
Intellipaat: https://www.quora.com/What-are-prerequisites-to-start-learning-machine-learning

Machine Learning



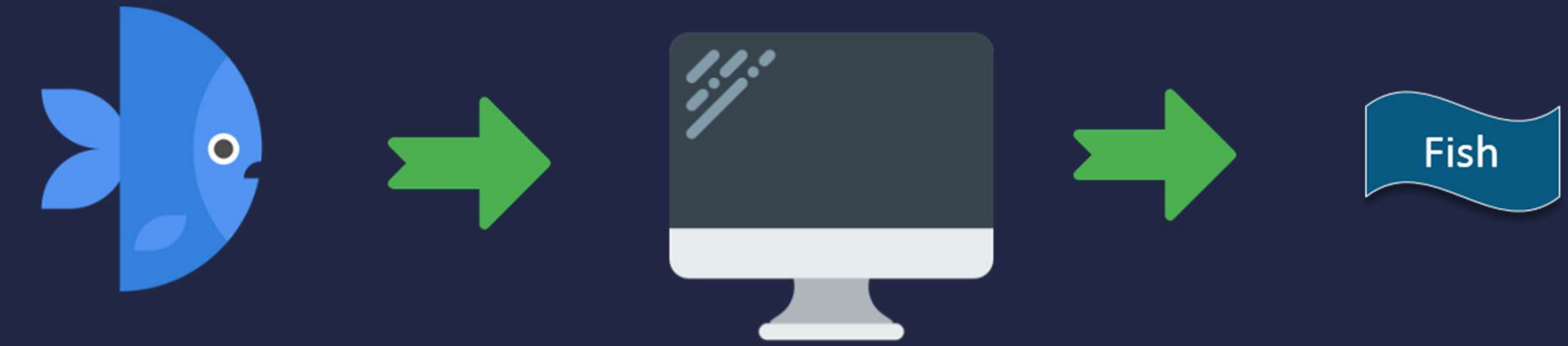
Intellipaat: https://www.quora.com/What-are-prerequisites-to-start-learning-machine-learning

Machine Learning



Intellipaat: https://www.quora.com/What-are-prerequisites-to-start-learning-machine-learning

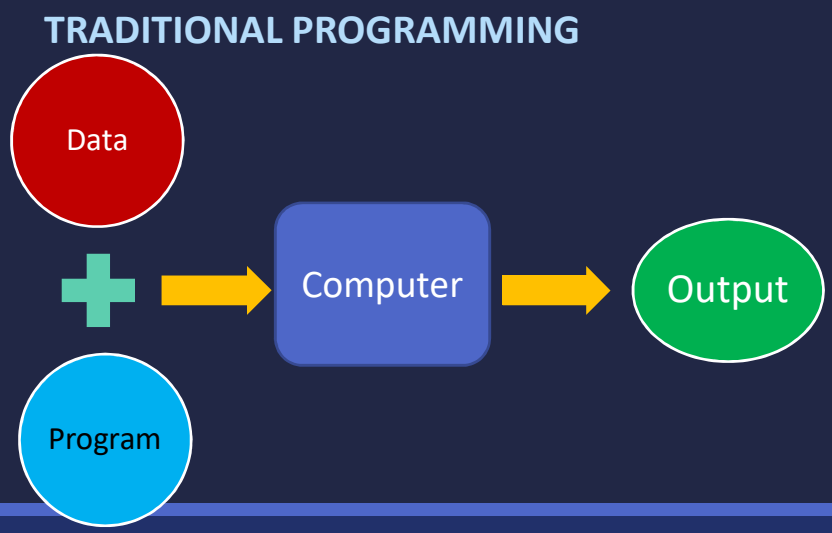
Machine Learning

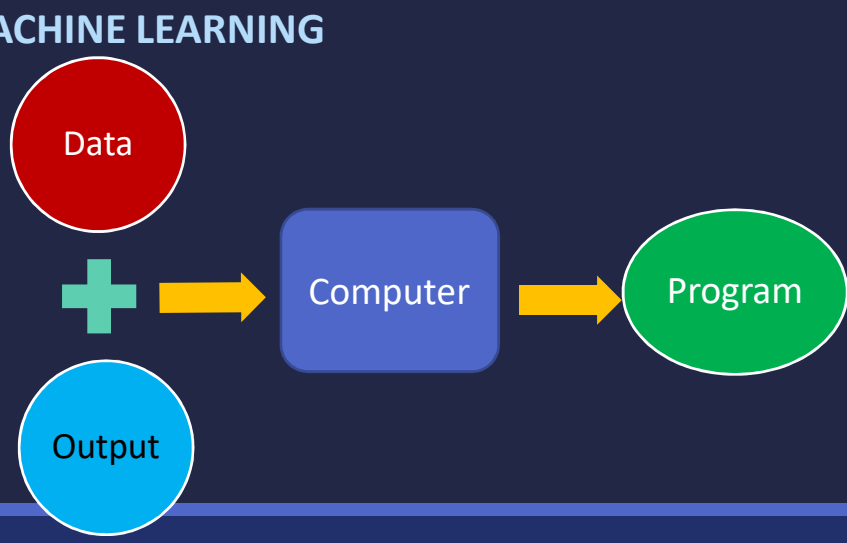


Intellpaat: https://www.quora.com/What-are-prerequisites-to-start-learning-machine-learning

Machine learning

- Machine learning is getting computers to program themselves.
- If programming is automation, then machine learning is automating the process of automation

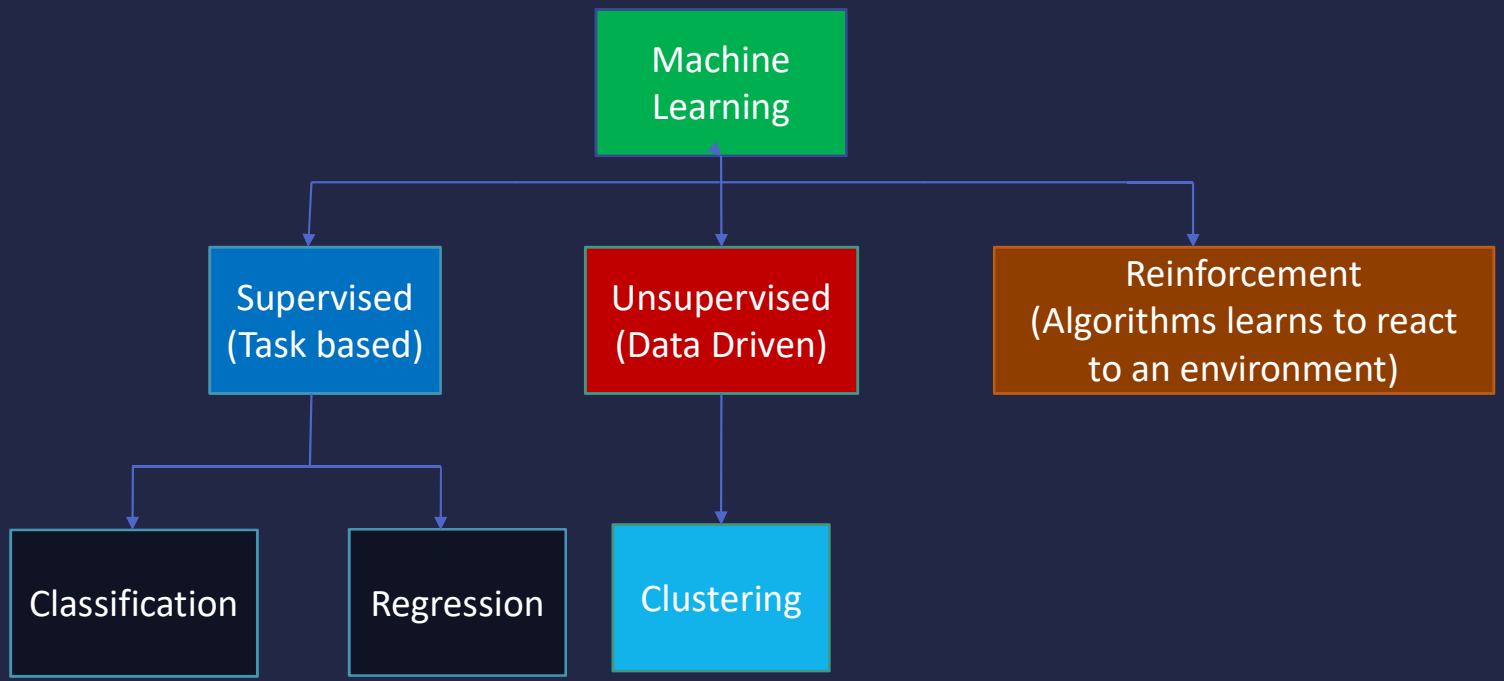




Pre-Requisites of ML

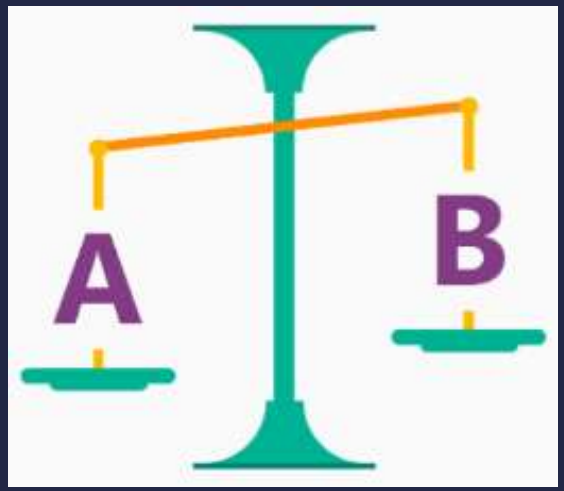
- A Pattern should exist
- Mathematical model / algorithm is unknown
- Lots of data

Types of Machine Learning



Is it A or B?

- **Classification** Algorithm
- Will this tire fail in the next 1,000 miles : Yes or No ?
- Which brings in more customers: a $5 coupon or a 25% discount ?

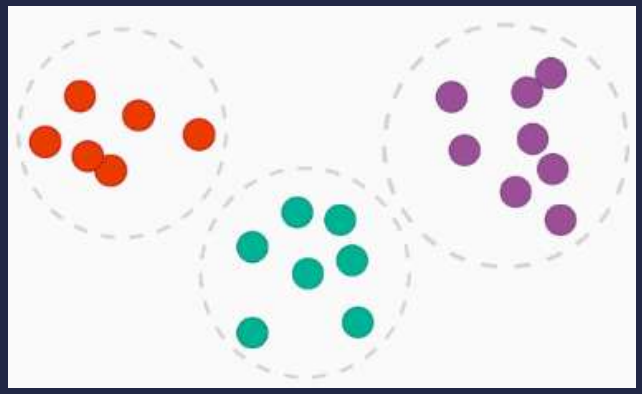


How much? How many?

- **Regression** Algorithm
- What will the temperature be next Tuesday?
- What will my fourth quarter sales be?
- They help answer any questions that asks for a number

How is this organized?

- **Clustering** Algorithm
- Which viewers like the same type of movies?
- Which printer model fail the same way?



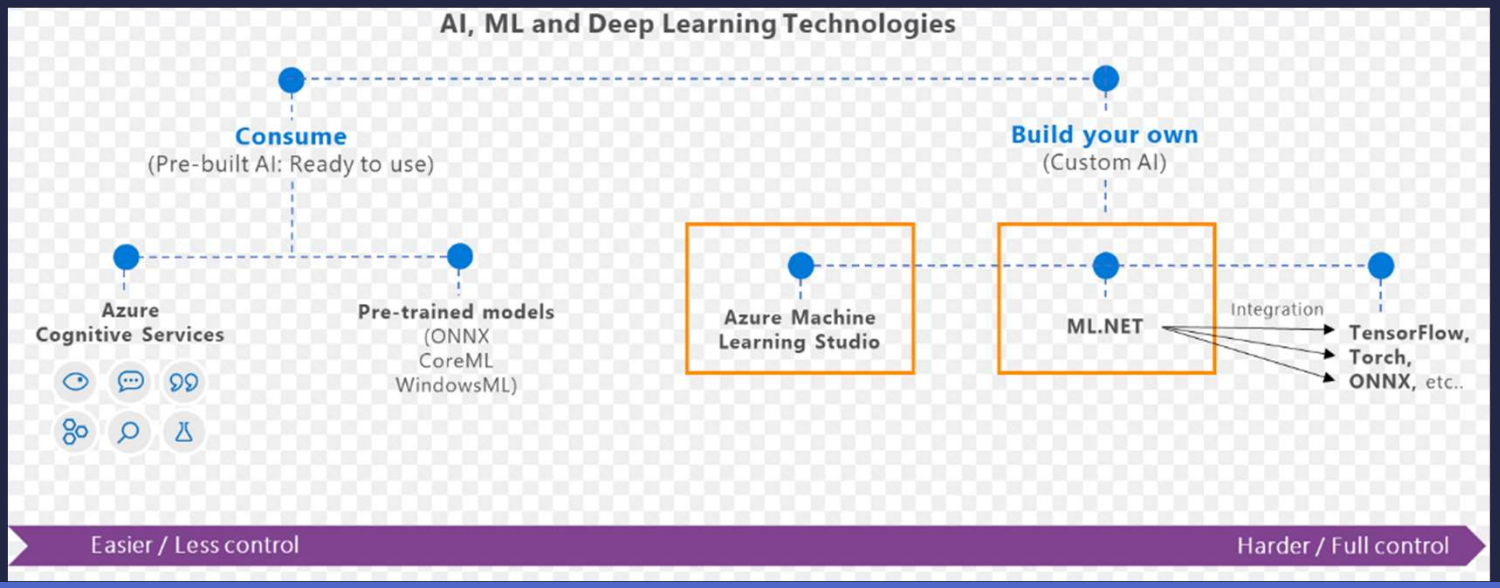
What should I Do now?

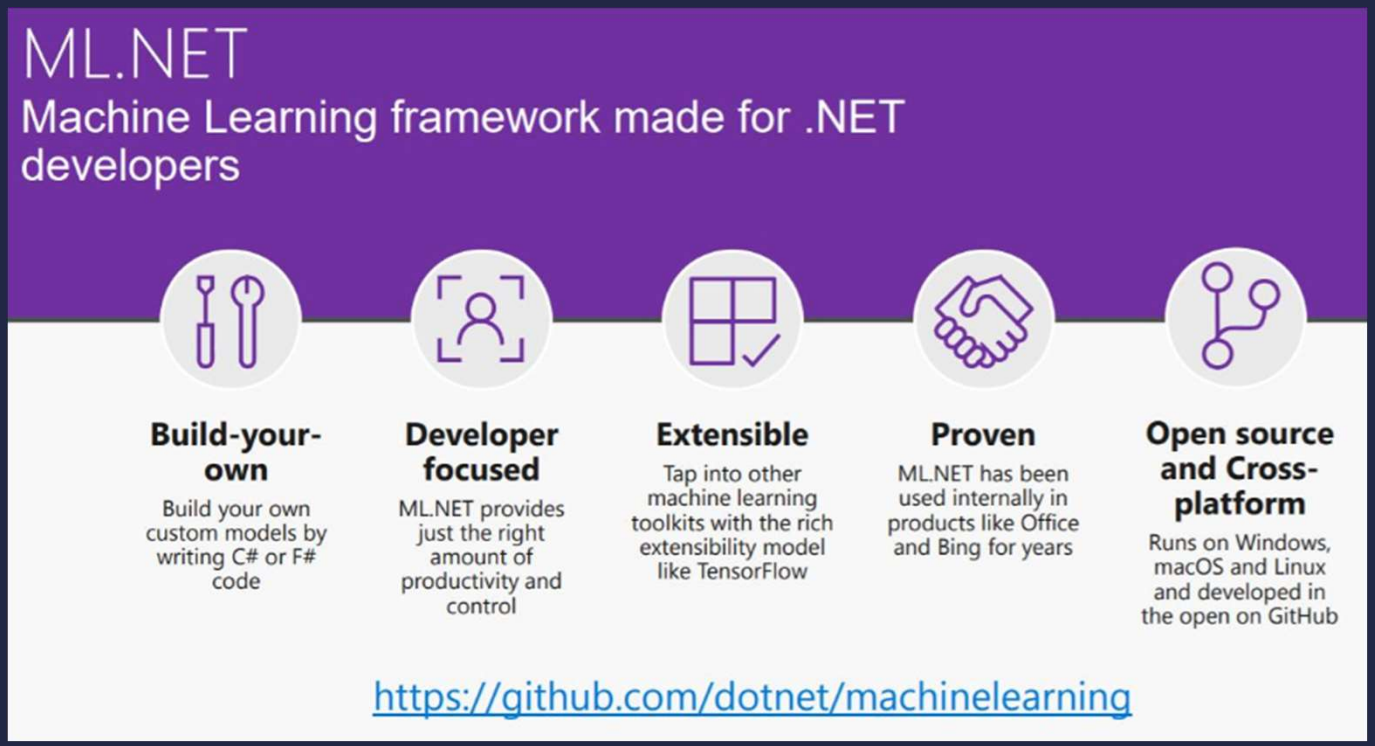
- **Reinforcement** learning Algorithm
- If I'm a self-driving car: At a yellow light, brake or accelerate?
- For a robot vacuum: Keep vacuuming, or go back to the charging station?

ML Frameworks

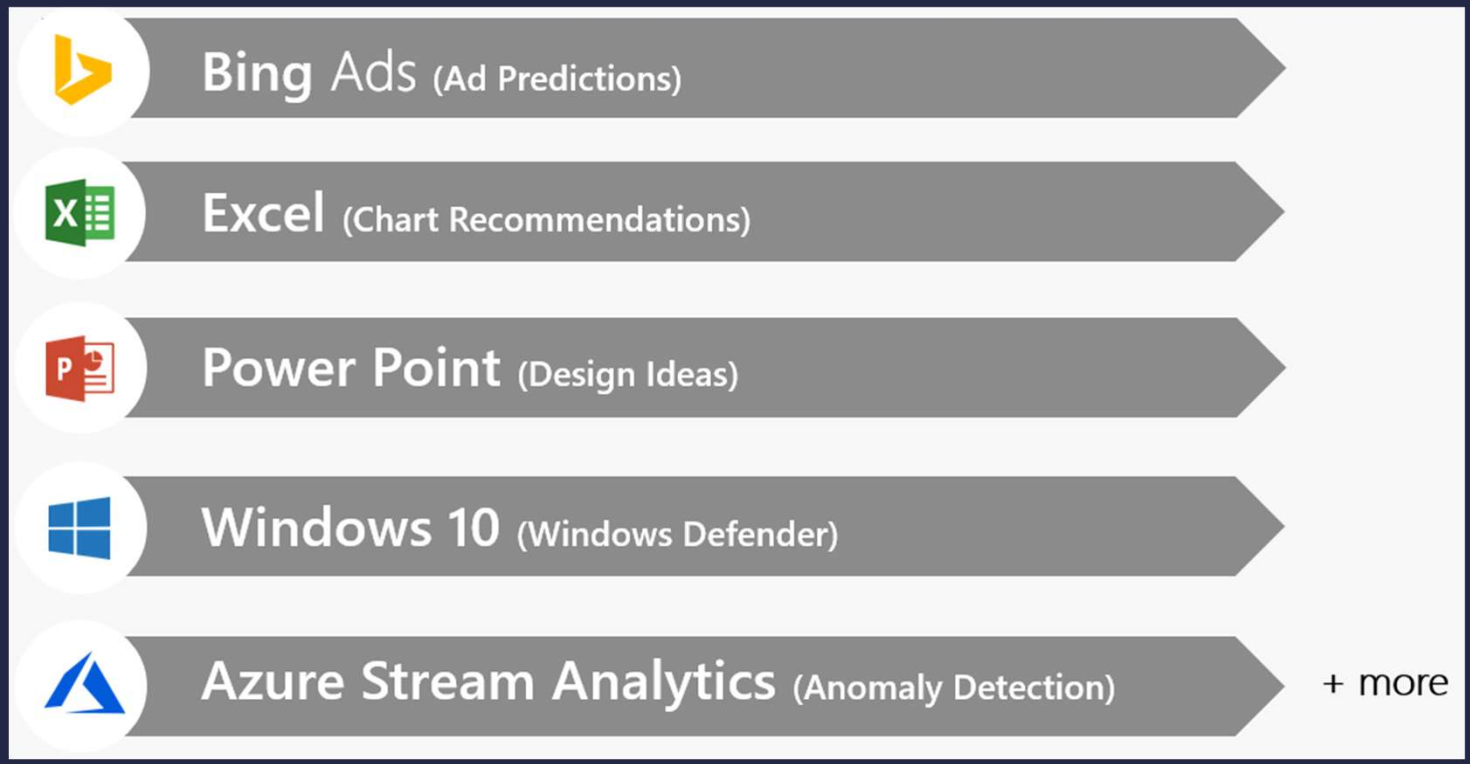
What is ML.Net?

- ML framework from Microsoft for developing Custom AI/ML applications
- Originated in 2002 as part of Microsoft Research project

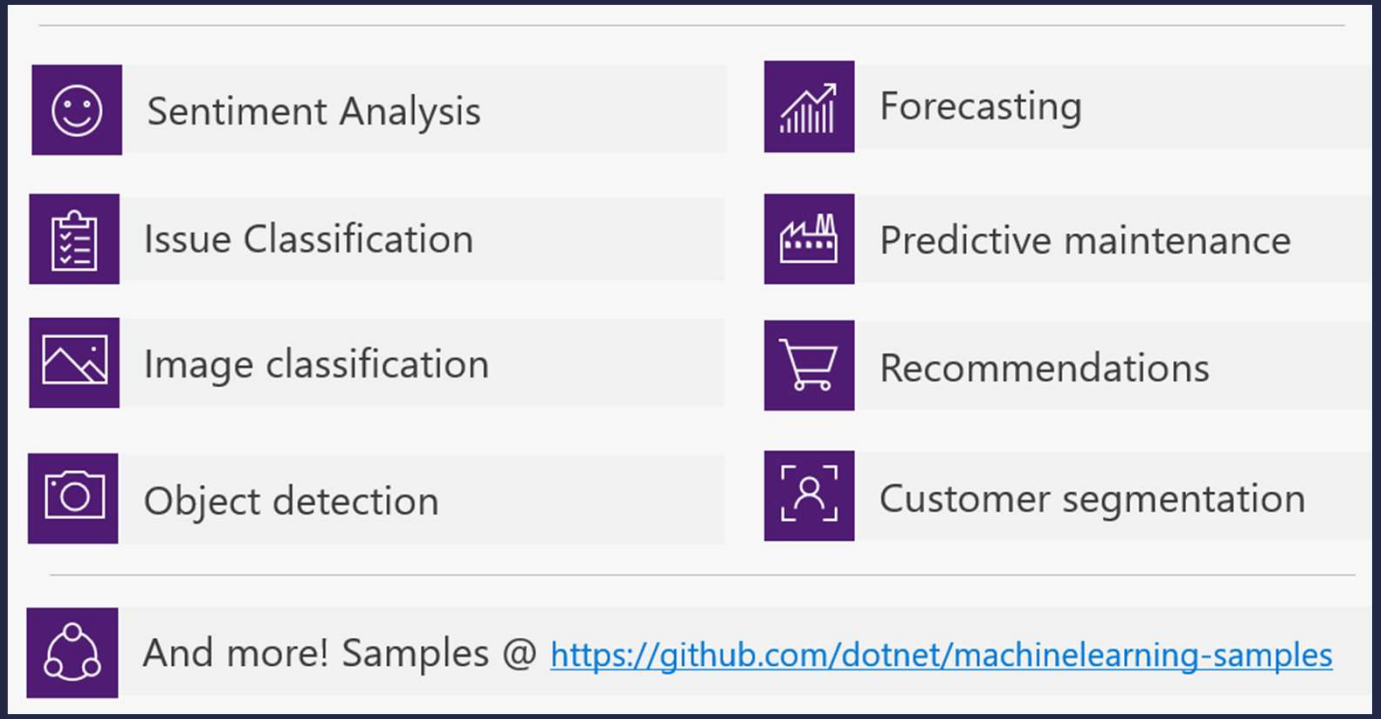



ML.NET
Machine Learning framework made for .NET developers
Build-your-own
Build your own custom models by writing C# or F# code
Developer focused
ML.NET provides just the right amount of productivity and control
Extensible
Tap into other machine learning toolkits with the rich extensibility model like TensorFlow
Proven
ML.NET has been used internally in products like Office and Bing for years
Open source and Cross-platform
Runs on Windows, macOS and Linux and developed in the open on GitHub
https://github.com/dotnet/machinelearning

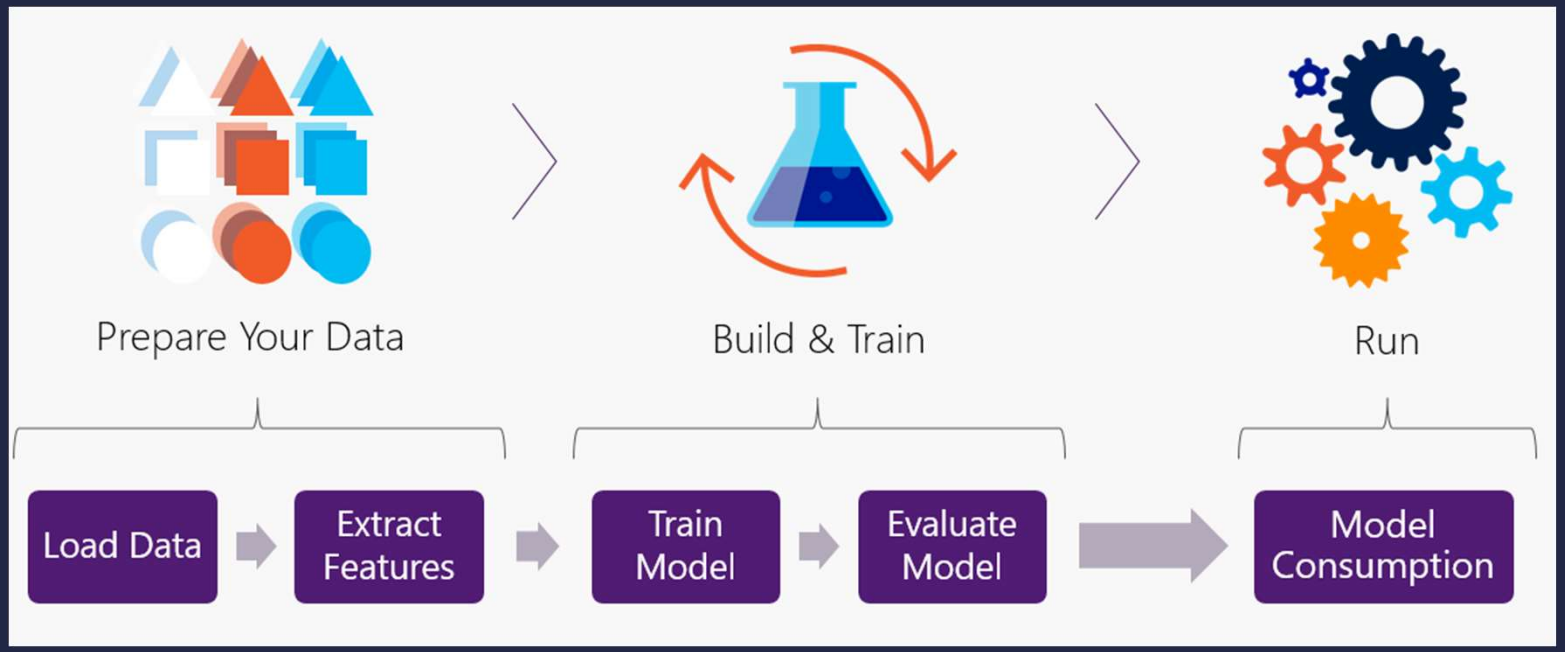
Proven at scale, Enterprise ready



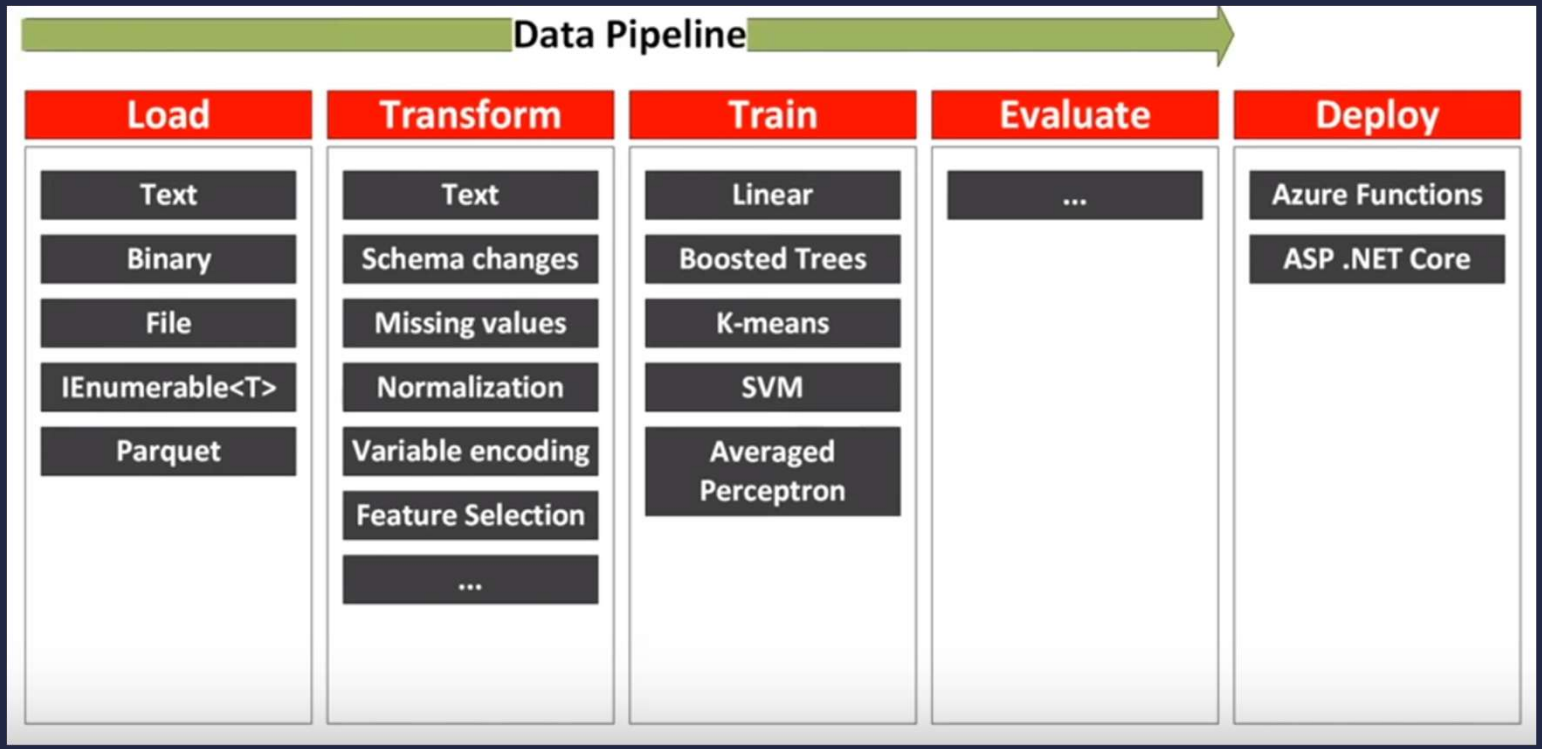
Possibilities



Workflow

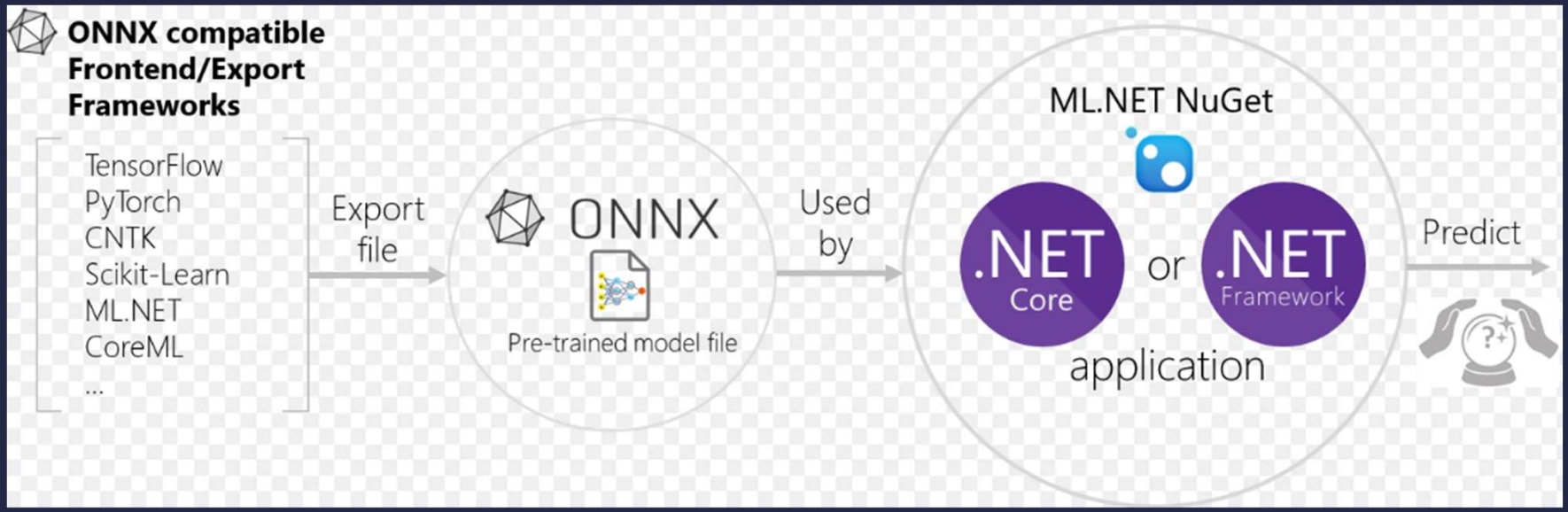

Prepare Your Data
Build & Train
Run
Load Data
Extract Features
Train Model
Evaluate Model
Model Consumption

Data pipeline



Interoperability: ONNX
Open Neural Network Exchange Format

- ONNX is developed and maintained by a community of partners such as Microsoft, Facebook.
- ONNX files could be viewed using Netron

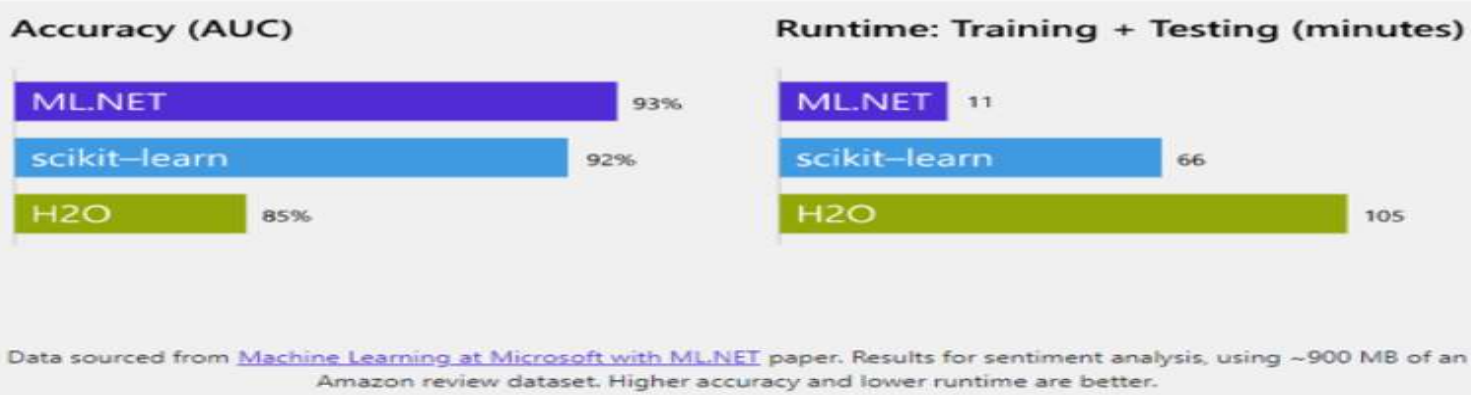


Performance

High performance and accuracy

Using a 9GB Amazon review data set, ML.NET trained a sentiment analysis model with 95% accuracy. Other popular machine learning frameworks failed to process the dataset due to memory errors. Training on 10% of the data set, to let all the frameworks complete training, ML.NET demonstrated the highest speed and accuracy.

The performance evaluation found similar results in other machine learning scenarios, including click-through rate prediction and flight delay prediction.



Data sourced from Machine Learning at Microsoft with ML.NET paper. Results for sentiment analysis, using ~900 MB of an Amazon review dataset. Higher accuracy and lower runtime are better.

Code Demo(s)

Sentiment Analysis

- Positive(+ve) , Negative(-ve)
- Label and Features
- Train model using ML.Net
- Train using AutoML
- Code generation
- Accuracy improvement
- https://docs.microsoft.com/en-us/dotnet/machine-learning/tutorials/sentiment-analysis

Text	Sentiment
Wow... Loved this place.	1
The fries were great too.	1
Not tasty and the texture was just nasty.	0
A great touch.	1
Waitress was a little slow in service.	0

Dataset - Yelp reviews

Features	Label
Wow... Loved this place.	1
The fries were great too.	1
Not tasty and the texture was just nasty.	0
A great touch.	1
Waitress was a little slow in service.	0

Load

```
// Initialize ML Context
MLContext mlContext = new MLContext();

// Load : Split it into 80% training and 20% test data
var trainTestSplit = 0.2;
IDataView dataView = mlContext.Data.LoadFromTextFile<SentimentData>
(_dataPath,  hasHeader: false);

TrainTestData splitDataView = mlContext.Data.TrainTestSplit(dataVie
w, testFraction: trainTestSplit);
```

Load -> Transform

```
 // Transform : Converts the text column(SentimentText) into numeric t
ype Features column using FeaturizeText

var estimator = mlContext.Transforms.Text.FeaturizeText(outputColumnNa
me: "Features", inputColumnName: nameof(SentimentData.SentimentText))


    .Append(mlContext.BinaryClassification.Trainers.SdcaLogisticRegres
sion(labelColumnName: "Label", featureColumnName: "Features"));
```

Load -> Transform -> Train

```
// Train/Fit model

var model = estimator.Fit(splitDataView.TrainSet);
```

Load -> Transform -> Train -> Evaluate

```
// Evaluate : Evaluate performance of the model using Test set
IDataView predictions = model.Transform(splitDataView.TestSet);


CalibratedBinaryClassificationMetrics metrics = mlContext.BinaryCla
ssification.Evaluate(predictions, "Label");
```

Load -> Transform -> Train -> Evaluate -> Predict

```
// Create PredictionEngine passing above model
PredictionEngine<SentimentData, SentimentPrediction> predictionFunction = mlContex
t.Model.CreatePredictionEngine<SentimentData, SentimentPrediction>(model);

// Create sample text
SentimentData sampleStatement = new SentimentData
{
     SentimentText = "This was a very bad steak"
};

// Predict
var resultPrediction = predictionFunction.Predict(sampleStatement);
```

Load -> Transform -> Train -> Evaluate -> Predict -> Save

```
// Save Model
mlContext.Model.Save(model, splitDataView.TrainSet.Schema,
"model.zip");
```

Binary Classification

- Titanic - Survival Prediction
- Label and Features
- Train model using ML.Net
- Train using AutoML
- Code generation
- Accuracy improvement
- Data: https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/problem12.html

Titanic Data

Label	Features						
Survived	Pclass	Name	Sex	Age	Siblings Aboard	Parents Aboard	Fare
0	3	Mr. Owen	male	22	1	0	7.25
1	1	Mrs. John	female	38	1	0	71.2833
1	3	Miss. Lain	female		0	0	7.925
1	1	Mrs. Jacqu	female	35	1	0	53.1
0	3	Mr. Willia	male	35	0	0	8.05
0	3	Mr. James	male		0	0	8.4583

Load

```
// Create MLContext
var mlContext = new MLContext(seed: 1);

// Load
Console.WriteLine("Loading...");
var data = mlContext.Data.LoadFromTextFile<Passenger>(DataPath, hasHeader: true, separatorChar: ',');
var trainTestData = mlContext.Data.TrainTestSplit(data, 0.2); // Training/Test : 80/20
```

Load -> Transform

```
// Trasform
Console.WriteLine("Transforming...");
var dataProcessPipeline = mlContext.Transforms.Categorical.OneHotEncoding("Sex", "Sex")
    .Append(mlContext.Transforms.ReplaceMissingValues("Age", replacementMode: ReplacementMode.Mean))
    .Append(mlContext.Transforms.Concatenate("Features", "PClass", "Sex", "SiblingsAboard", "ParentsAboard"));
```

Load -> Transform -> Train

```
// Train
Console.WriteLine("Training...");
var trainingPipeline = dataProcessPipeline.Append(mlContext.BinaryClassification.
    Trainers.LbfgsLogisticRegression("Survived"));

var trainedModel = trainingPipeline.Fit(trainTestData.TrainSet);
```

Load -> Transform -> Train -> Evaluate

```
// Evaluate
Console.WriteLine("Evaluating...");
var predictions = trainedModel.Transform(trainTestData.TrainSet);
var metrics = mlContext.BinaryClassification.Evaluate(predictions, "Survived", "Score");

Console.WriteLine($"  Accuracy:          {metrics.Accuracy:P2}");
```

Load -> Transform -> Train -> Evaluate -> Save

```
// Save
Console.WriteLine("Saving...");
var savedPath = Path.Combine(Directory.GetCurrentDirectory(), "model.zip");
mlContext.Model.Save(trainedModel, trainTestData.TrainSet.Schema, savedPath);
```

Load -> Transform -> Train -> Evaluate -> Save -> Predict

```
// Predict
Console.WriteLine("********** Predict...");
var predictionEngine = mlContext.Model.CreatePredictionEngine<Passenger,
                                        PassengerPrediction>(trainedModel);
var passenger = new Passenger()
{
    PClass = 1,
    Name = "Mark Farragher",
    Sex = "male",
    Age = 48,
    SiblingsAboard = 0,
    ParentsAboard = 0
};

// make the prediction
var prediction = predictionEngine.Predict(passenger);

// report the results
Console.WriteLine($"Passenger:   {passenger.Name} ");
Console.WriteLine($"Prediction:  {(prediction.Prediction ? "survived" : "perished")} ");
```

Load a model

```
ITransformer _model;

using (var stream = System.IO.File.OpenRead(modelPath))
{
    _model = _context.Model.Load(stream);
}
```

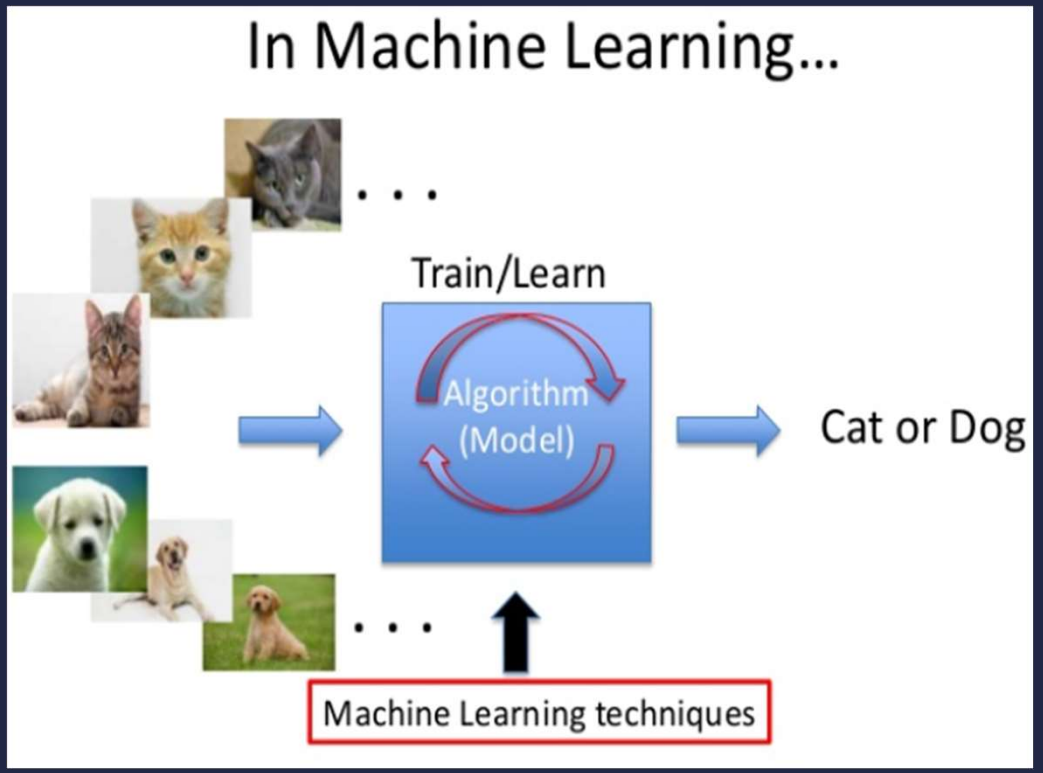
AutoML – In Preview

S.No	Custom ML	AutoML
1.	Manually experiment with different algorithms	Runs different algorithms
2	Time consuming in identifying best algorithm	Runs many algorithm without code changes
3	Provides better control of algorithms	No control over algorithm selection
4	Manually write code	Auto-generates code

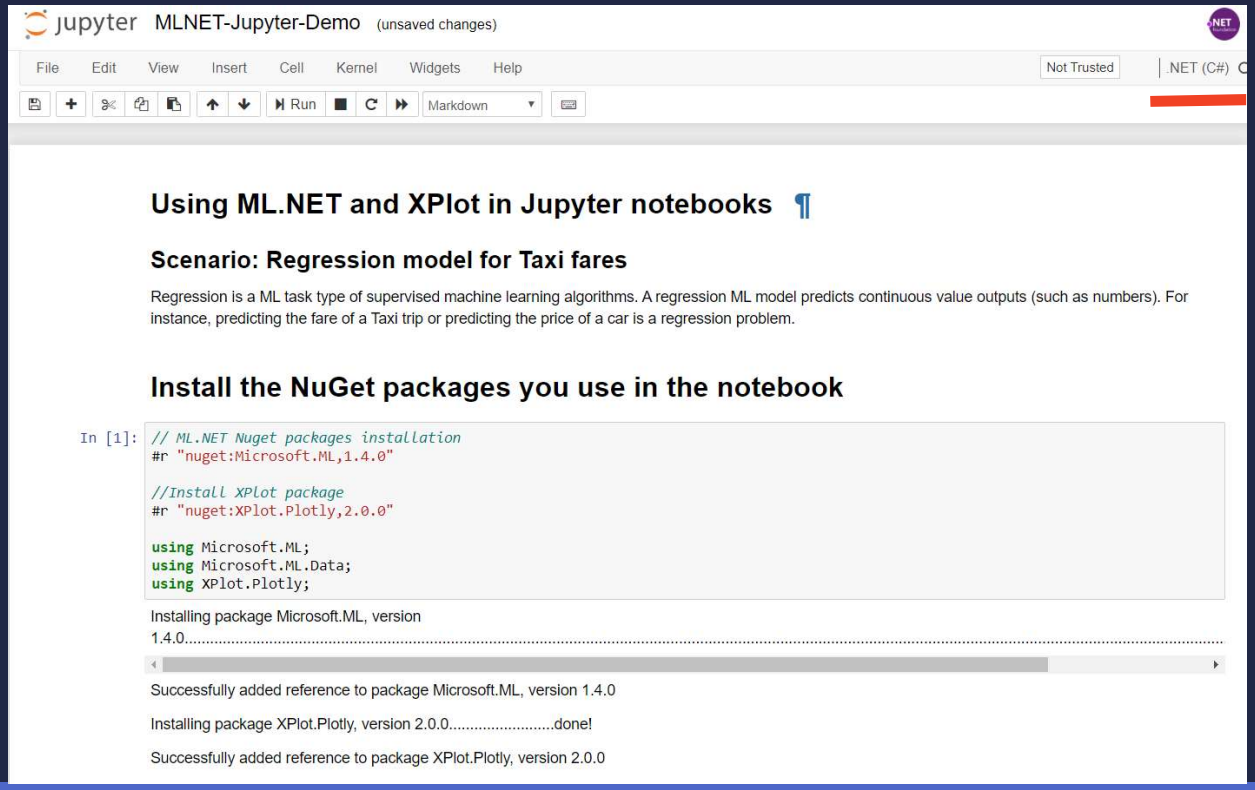
```
mlnet auto-train --task binary-classification --dataset "titanic.csv" --label-column-index 0 --has-header true --max-exploration-time 30
```

Interoperability

- Photo-Search
- ML.Net + ONNX
- Keras CNN Model
- No rewrite of algorithms/model
- Save effort and time



Jupyter Notebook – C#/F#



https://devblogs.microsoft.com/cesardelatorre/using-ml-net-in-jupyter-notebooks/

References

- ML.Net : https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet
- ML.Net: https://devblogs.microsoft.com/cesardelatorre/what-is-ml-net-1-0-machine-learning-for-net/
- ONNX : https://onnx.ai/
- Photo-Search (ONNX) : https://github.com/Tak-Au/Photo-Search
- Music Repair : https://www.youtube.com/watch?v=nnV-1q-z9uE
- ML Cookbook : https://github.com/dotnet/machinelearning/blob/master/docs/code/MlNetCookBook.md
- Deploy to Azure functions : http://luisquintanilla.me/2018/08/21/serverless-machine-learning-mlnet-azure-functions/
- https://rubikscode.net/2019/02/18/ultimate-guide-to-machine-learning-with-ml-net/
- https://www.youtube.com/watch?v=dojO4zEL9sg
- https://www.youtube.com/watch?v=zy7Y9CHji2k

Thank you

Q & A

 praveenraghuvanshi@gmail.com

 https://in.linkedin.com/in/praveenraghuvanshi

 https://github.com/praveenraghuvanshi1512

 @praveenraghuvan