Microsoft

Open Source Summit

Data & AI

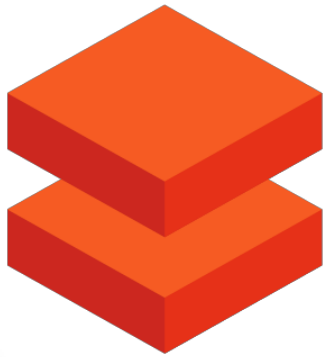
Merve Cerit

@mmervecerit

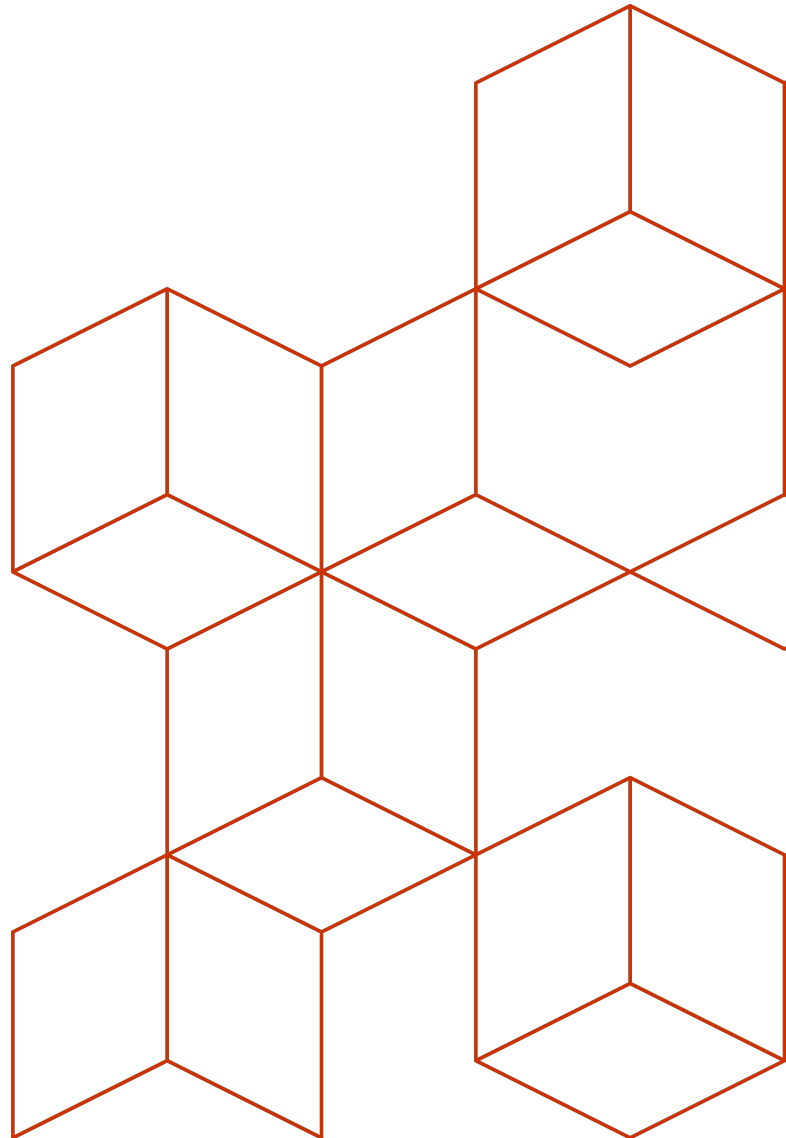
Foreword & Agenda

⩽ 50 mins

- An Introduction to ML.NET
- Deep Learning on Databricks



AI
Machine Learning
Deep Learning



Movie Recommender

Who's watching?

Cesar

Amy

Ankit

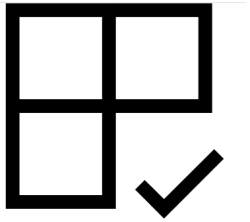

Frontend


Backend

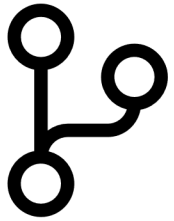
ML.NET is a machine learning framework made for .NET developers

Supported on Windows, Linux, and macOS

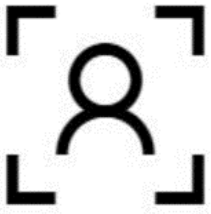
Build your own



Proven & Extensible



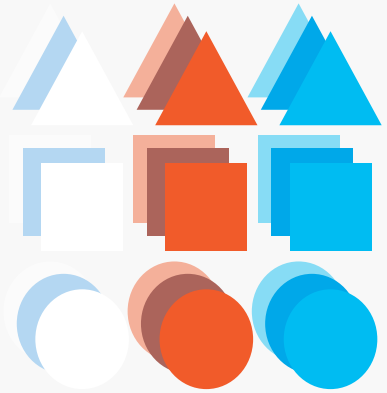
Open Source

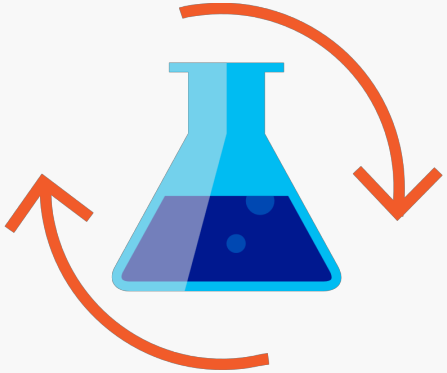


Developer Focused

https://github.com/dotnet/machinelearning

Machine learning workflow





Prepare Your Data

Build & Train

Run

Load Data → Extract Features → Train Model → Evaluate Model → Model Consumption

Introduction to ML.NET – Command Cheatsheet

- Install .NET SDK

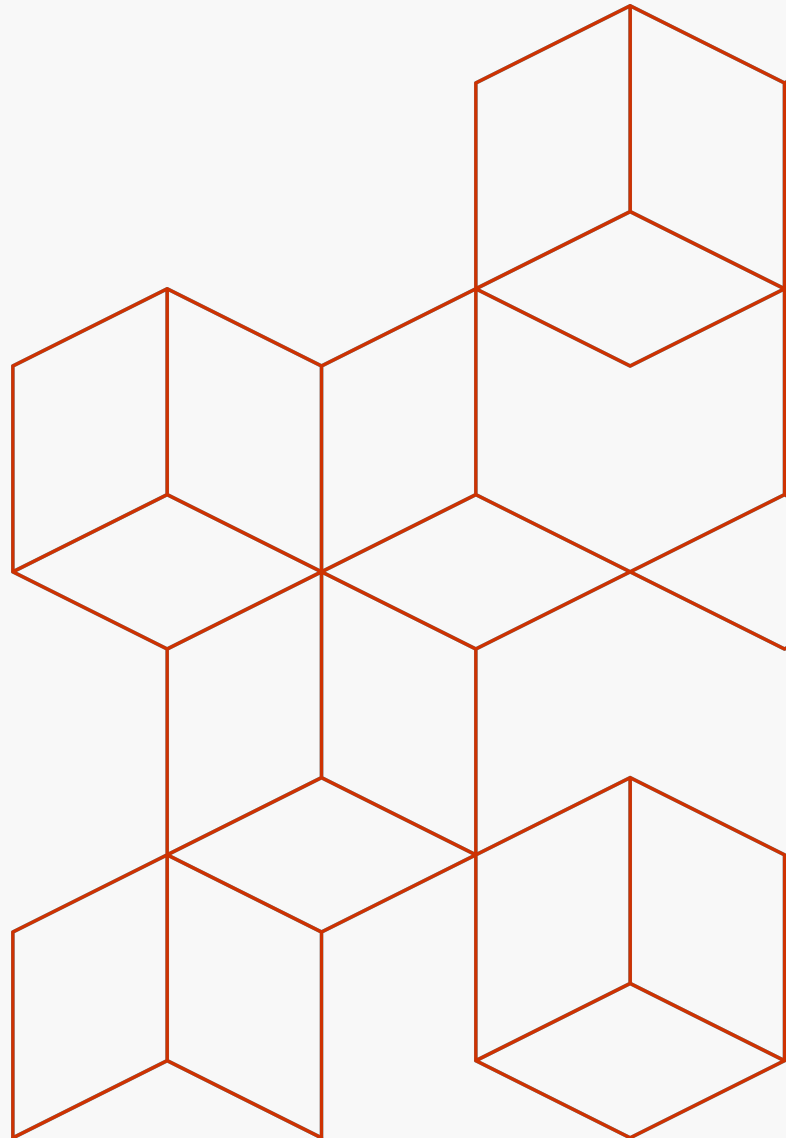
dotnet new console –o newapp
cd newapp
dotnet add package Microsoft.ML --version 0.8.0

curl https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data -o iris-data.txt
curl https://raw.githubusercontent.com/mmervecerit/ML.NET-trials/master/Program.cs -o Program.cs
dotnet run

https://github.com/dotnet/machinelearning.git

Movie Recommendations

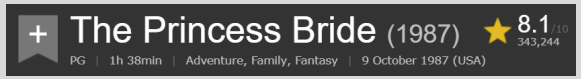
Demo: ML.NET + Factorization Machines



Approaches to Movie Recommendations

1. Population Averages

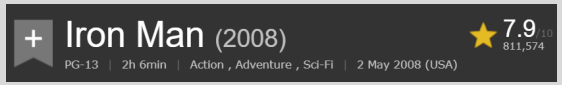
e.g. IMDB, Yelp



IMDB Rating > 8.0, Recommend Movie!

2. Content Based Filtering

e.g. Movie genre



Genre = Superhero movies, Recommend!

3. Collaborative Filtering

Approaches to Movie Recommendations

3. Collaborative filtering

If a person A (e.g. Gal) has the same opinion as a person B (e.g. Cesar) on an issue,
A (Gal) is more likely to have B's (Cesar) opinion on a different issue than that of a random person

	Heat	Mission Impossible	Home Alone	Terminator 2	Casino Royale
Ankit	×	×	✓	✓	×
Gal	✓	✓	×	✓	?
Cesar	✓	✓	×	✓	✓

What's the probability of Gal liking Casino Royale?

ML.NET 0.3 provides support for Collaborative Filtering (Factorization Machines)

Approaches to Movie Recommendation

Which Dataset to use?

Movie Lens Dataset

20 million ratings, 27,000 movies across 138,000 users.

https://grouplens.org/datasets/movielens/

Ratings.csv

UserId	movieId	Rating	TimeStamp
1	1	2	1094785740
1	2	4.0	1094785734
1	6	4.0	1112485573
1	10	4.0	1112484703

...

Movies.csv

movieID	Title	Genre
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
6	Heat (1995)	Action|Crime|Thriller
10	GoldenEye (1995)	Action|Adventure|Thriller

Features (input)

Ratings

UserId	movieId	Rating	TimeStamp
1	1	0	1094785740
1	2	1	1094785734
1	6	1	1112485573
1	10	1	1112484703

Movies

movieID	Title	Genre
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
6	Heat (1995)	Action|Crime|Thriller
10	GoldenEye (1995)	Action|Adventure|Thriller

Label (output)

Ratings

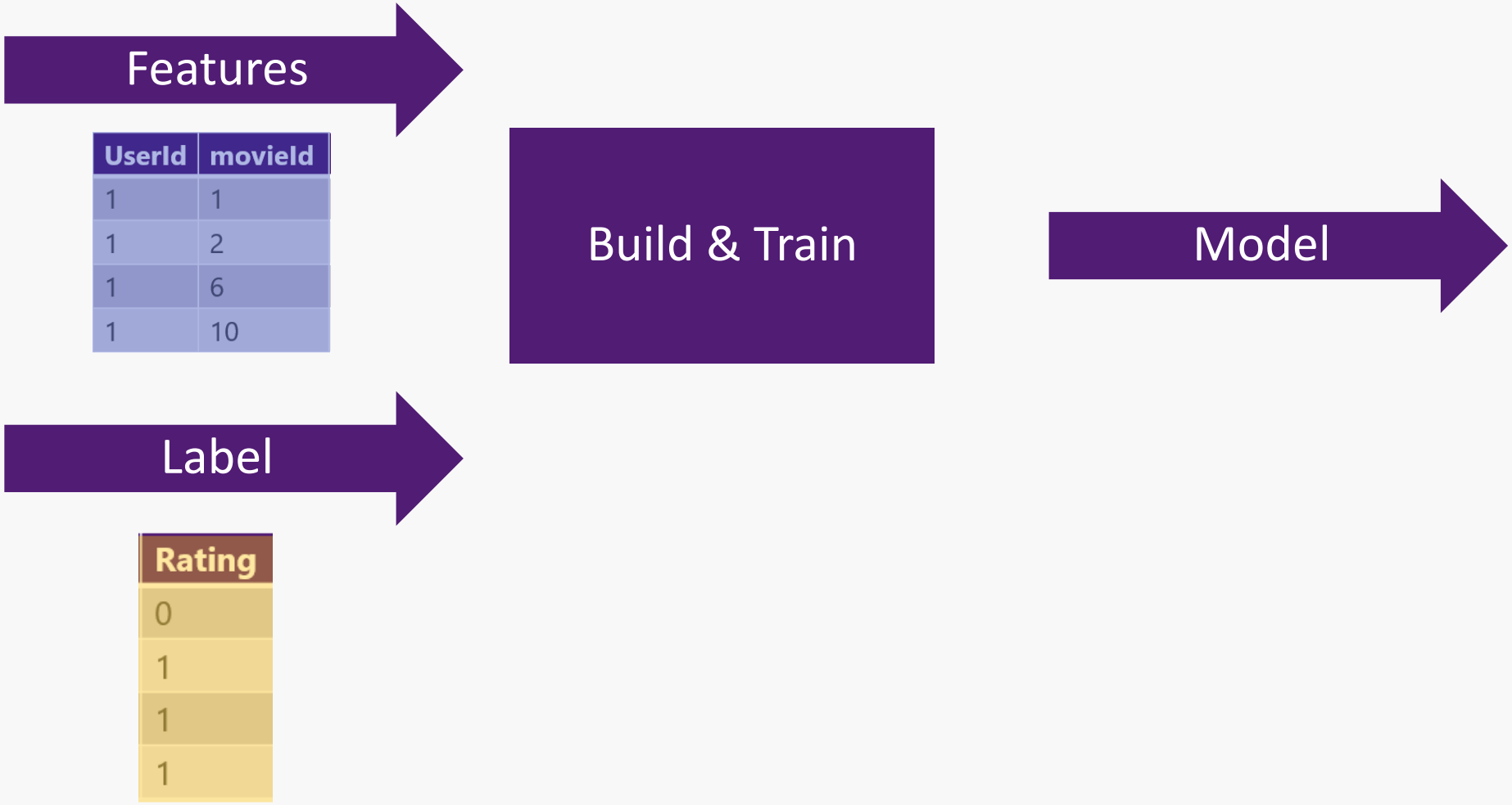
UserId	movieId	Rating	TimeStamp
1	1	0	1094785740
1	2	1	1094785734
1	6	1	1112485573
1	10	1	1112484703

Movies

movieID	Title	Genre
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
6	Heat (1995)	Action|Crime|Thriller
10	GoldenEye (1995)	Action|Adventure|Thriller

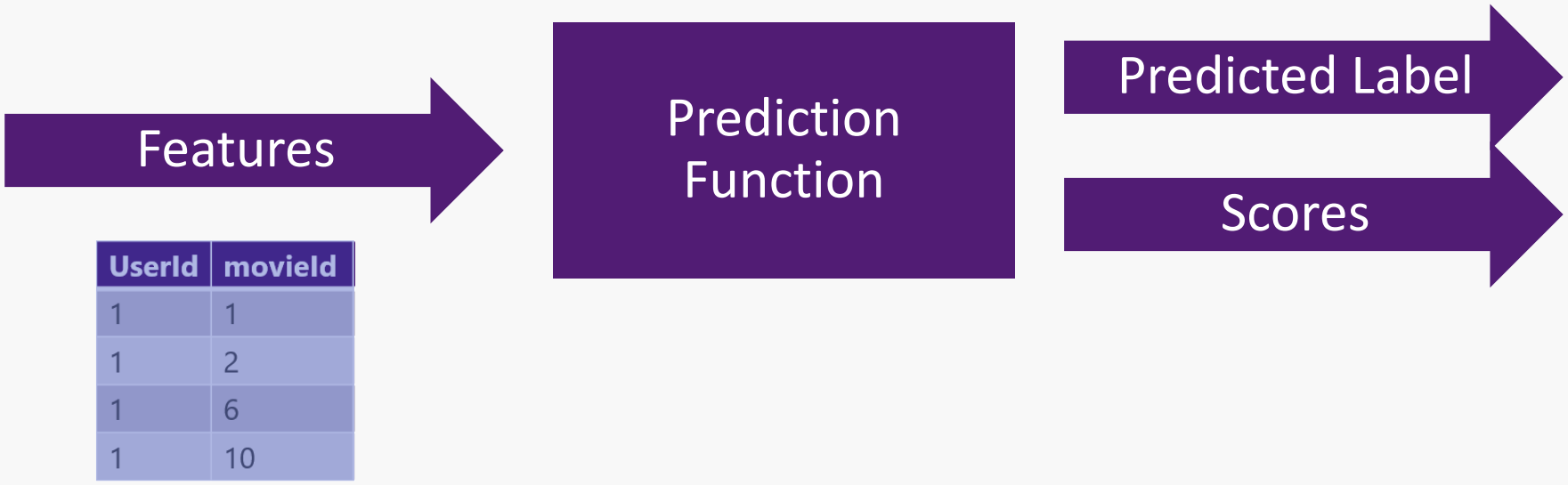
Movie Recommendation with ML.NET

Training



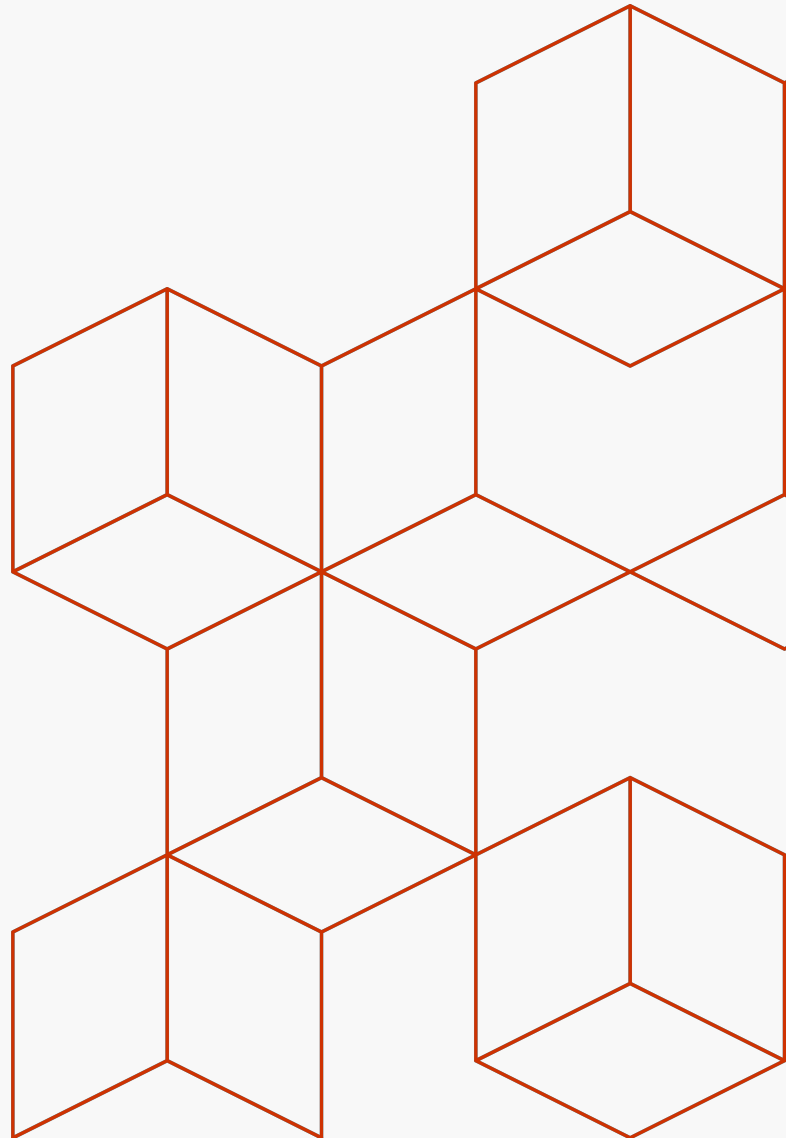
Movie Recommendation with ML.NET

Prediction



UserId	movieId
1	1
1	2
1	6
1	10

Deep Learning

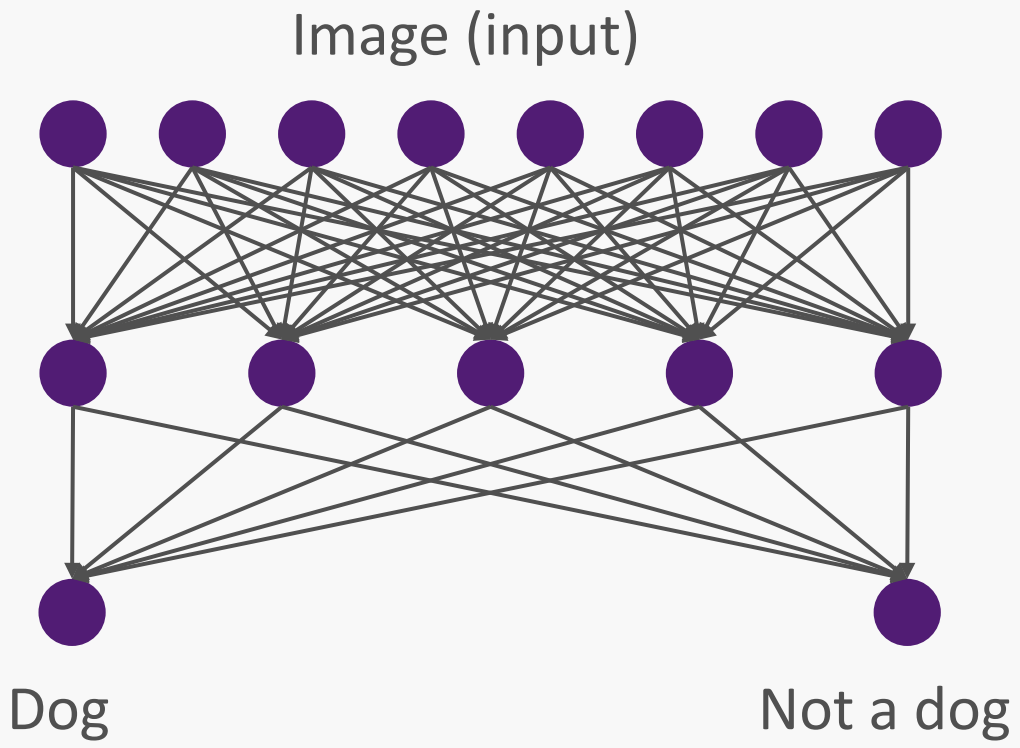


Deep learning

- Revolutionizing areas like computer vision and speech recognition
- Takes advantage of large amounts of data and compute

Dog

Not a dog

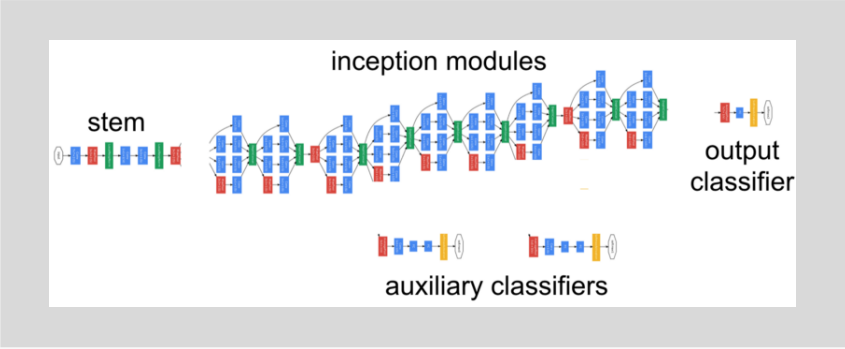


Deep learning with ML.NET

+

Predicting Image classes with ML.NET + TensorFlow

- TensorFlow: popular deep learning framework
- Pretrained model: Inception
  - 1000 classes e.g. cats, cars, breeds of dogs
- Implemented as a transformer in ML.NET



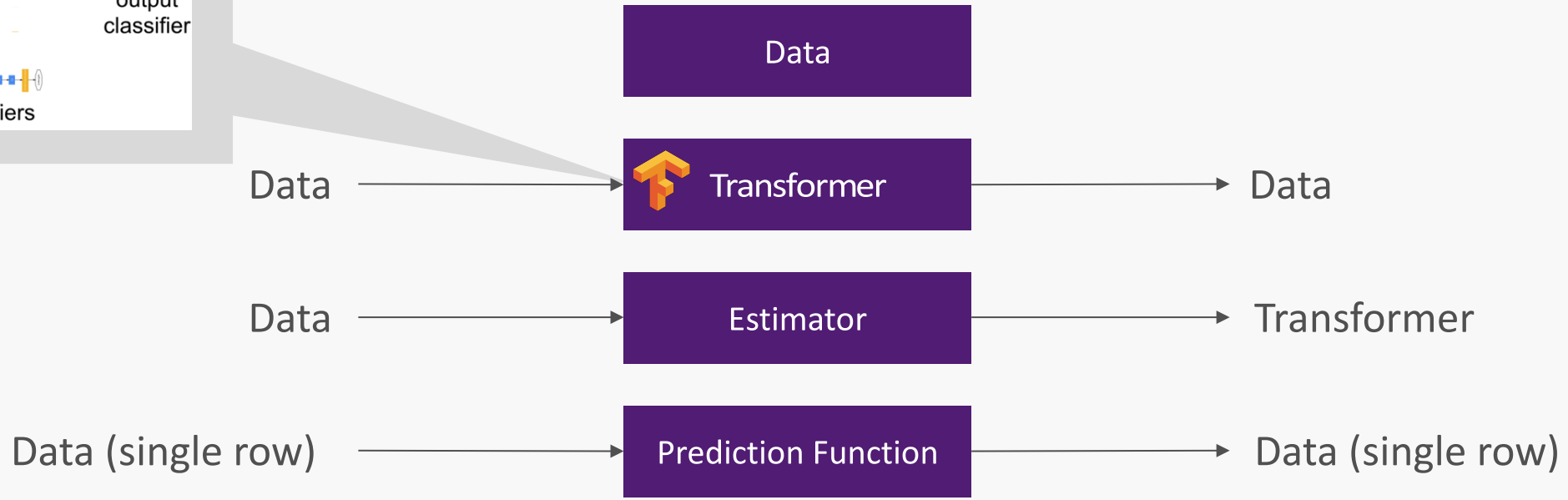
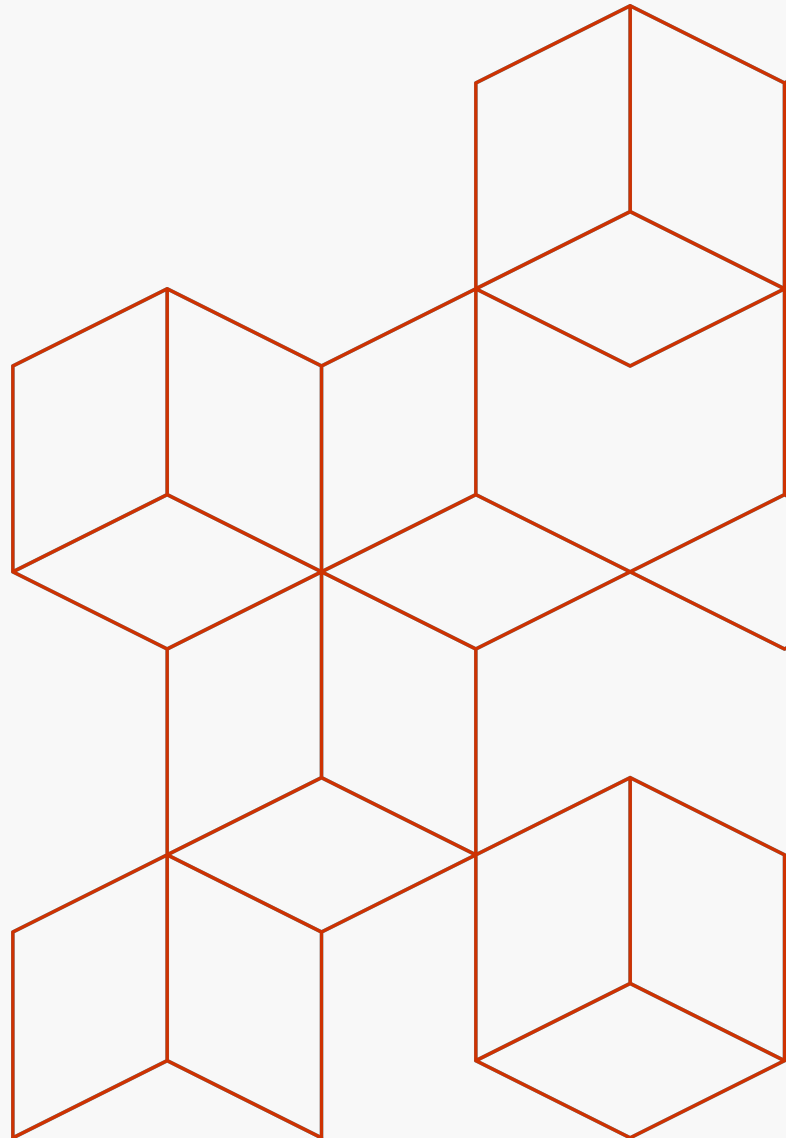


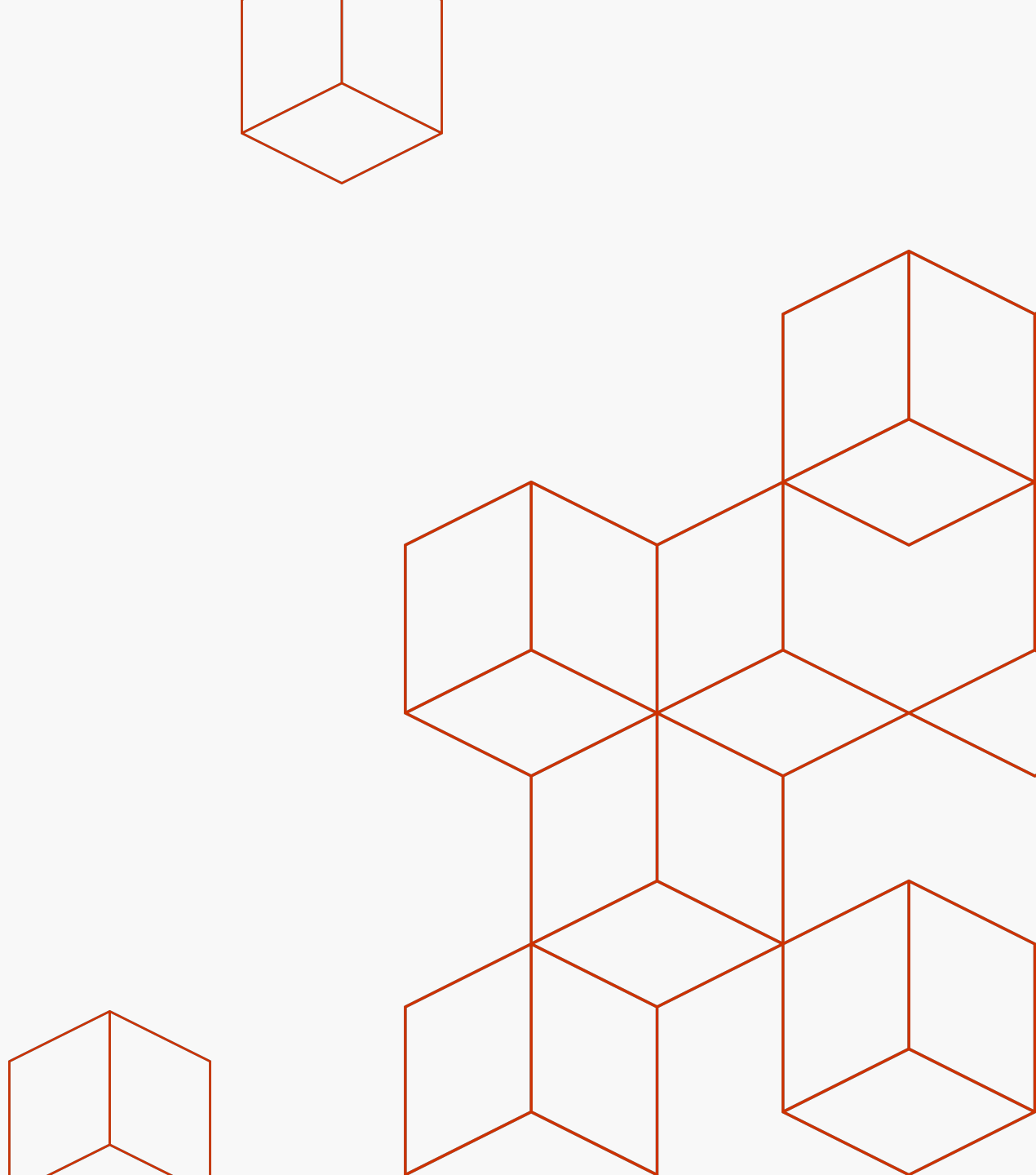
Image Classification Scorer

Demo: ML.NET + TensorFlow

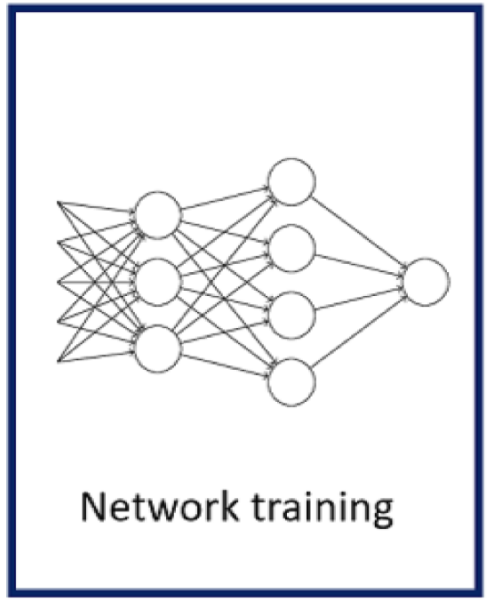


ONNX: Open and interoperable AI

Bir kat aşağı?



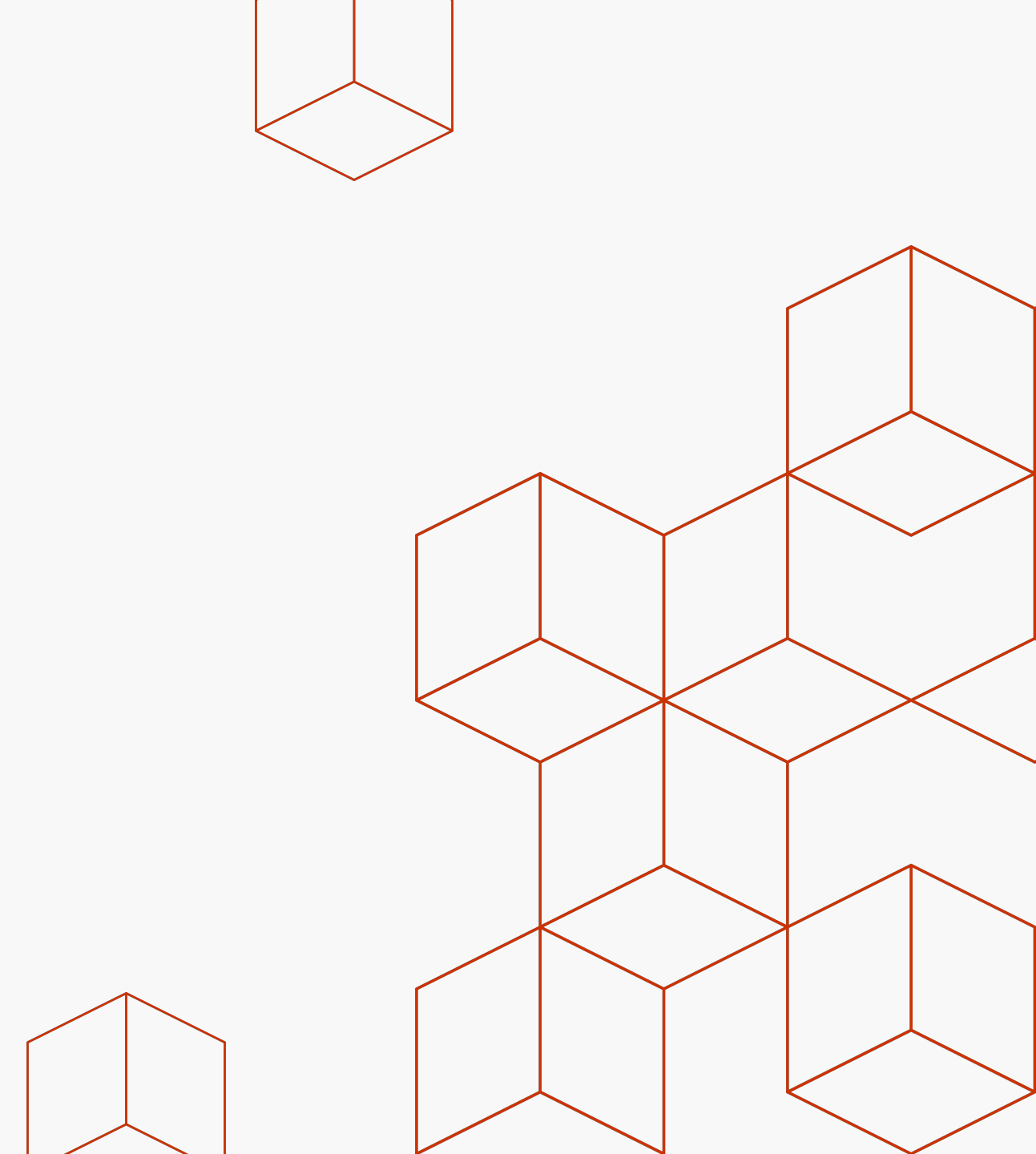
Data & Labels



0
1
2
3
4
5
6
7
8
9

Databricks’te bir örnek

Images
Labels
Loss Function
Optimizer



Cheatsheet

```
import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
x = tf.placeholder(tf.float32, [None, 784])
W = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))
y = tf.matmul(x, W) + b
y_ = tf.placeholder(tf.float32, [None, 10])
cross_entropy = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits=y, labels=y_))
train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)
correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_, 1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
summary = tf.summary.scalar("accuracy", accuracy)
sess = tf.InteractiveSession()
tf.global_variables_initializer().run()
for batch in range(1000):
        batch_xs, batch_ys = mnist.train.next_batch(100)
        _, batch_summary = sess.run([train_step, summary], feed_dict={x: batch_xs,  y_: batch_ys})
print(sess.run(accuracy, feed_dict={x: mnist.test.images, y_: mnist.test.labels}))
```

ML.NET

https://github.com/dotnet/machinelearning.git

Databricks

https://docs.azuredatabricks.net

Deep Learning VM + Kubernetes

https://github.com/sozercan/tensorflow-object-detection

The Final Say & Thank you!

@mmervecerit
t-meceri@microsoft.com
mmervecerit@gmail.com