



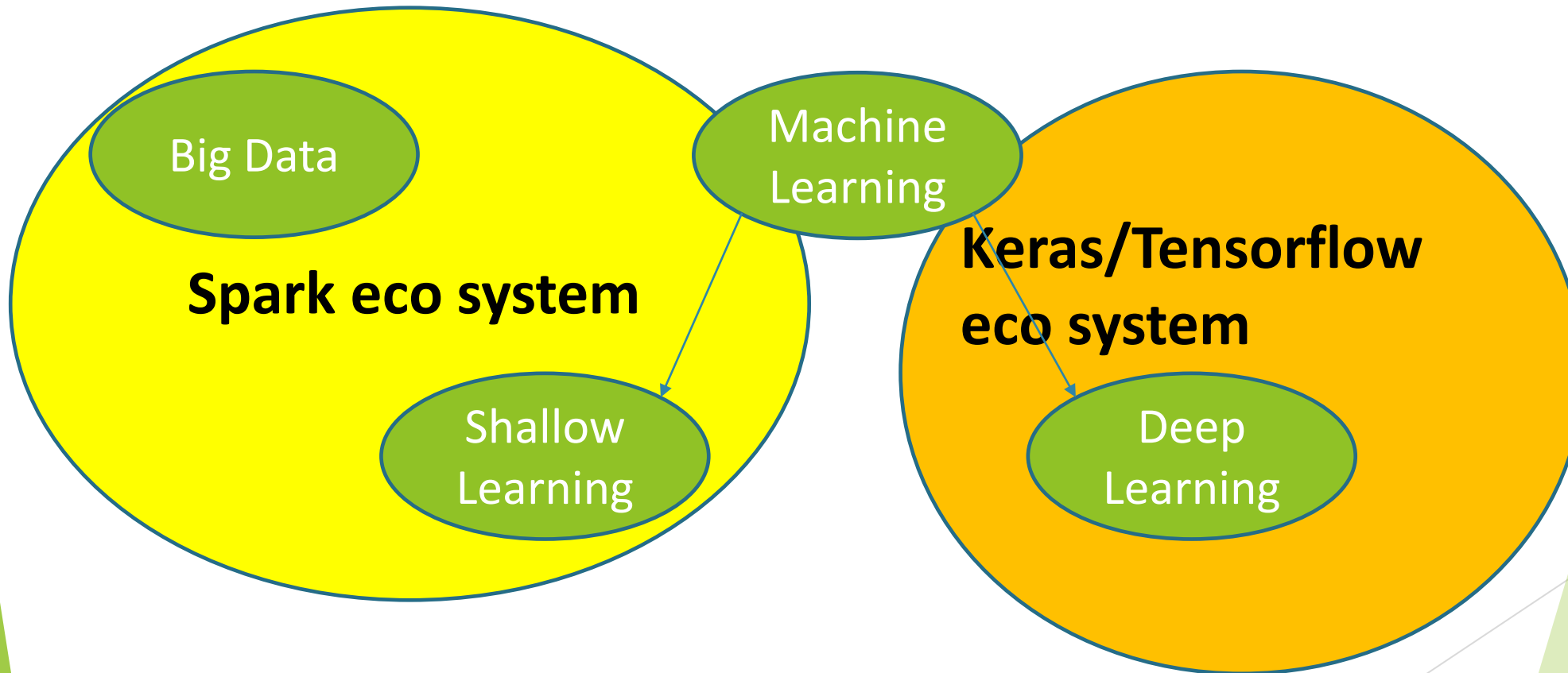
Big Data és MI eszközök, platformok

Farkas Attila

farkas.attila@sztaki.hu



MI fő területei és támogatott keretrendszerek

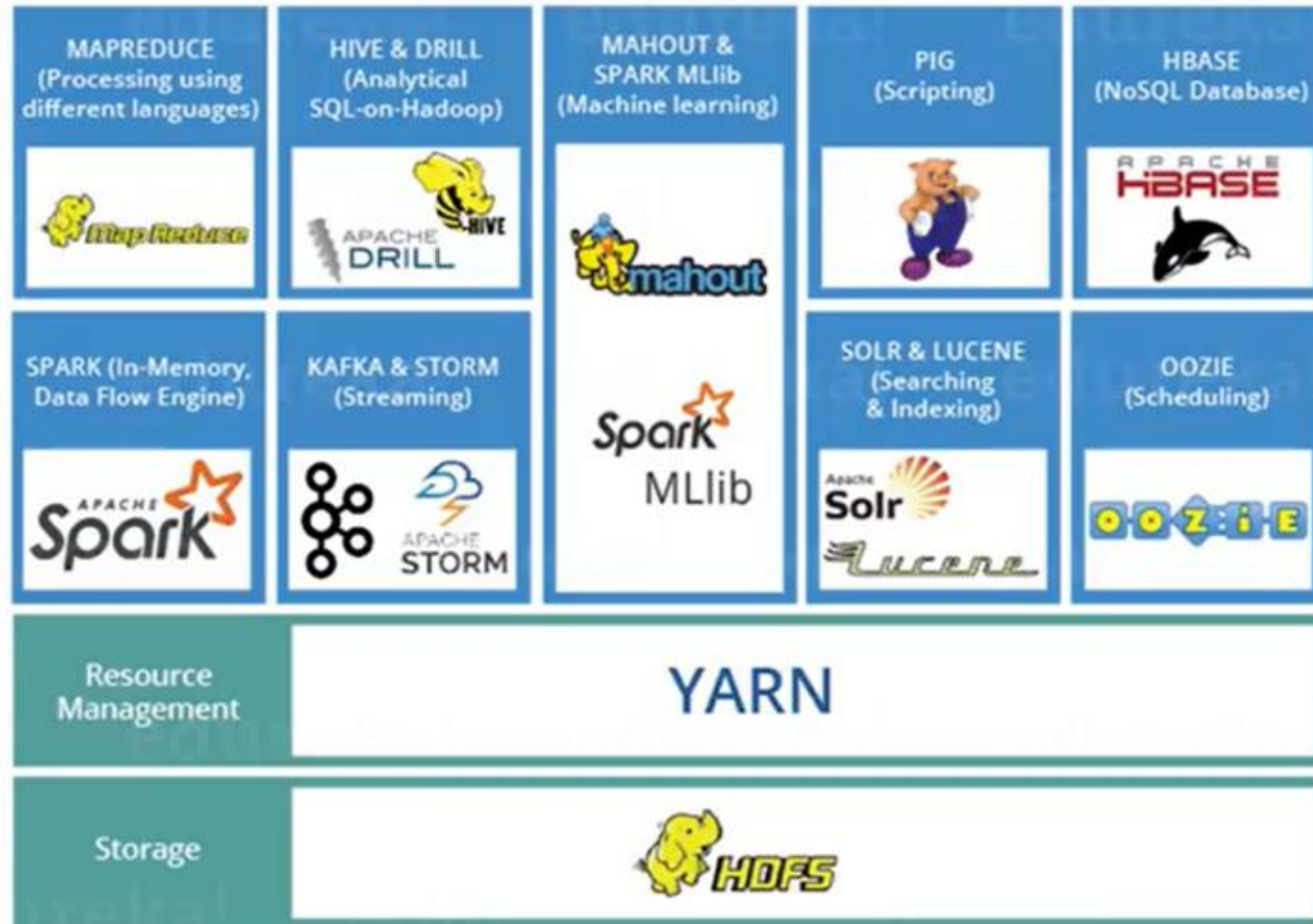


Big Data keretrendszerek

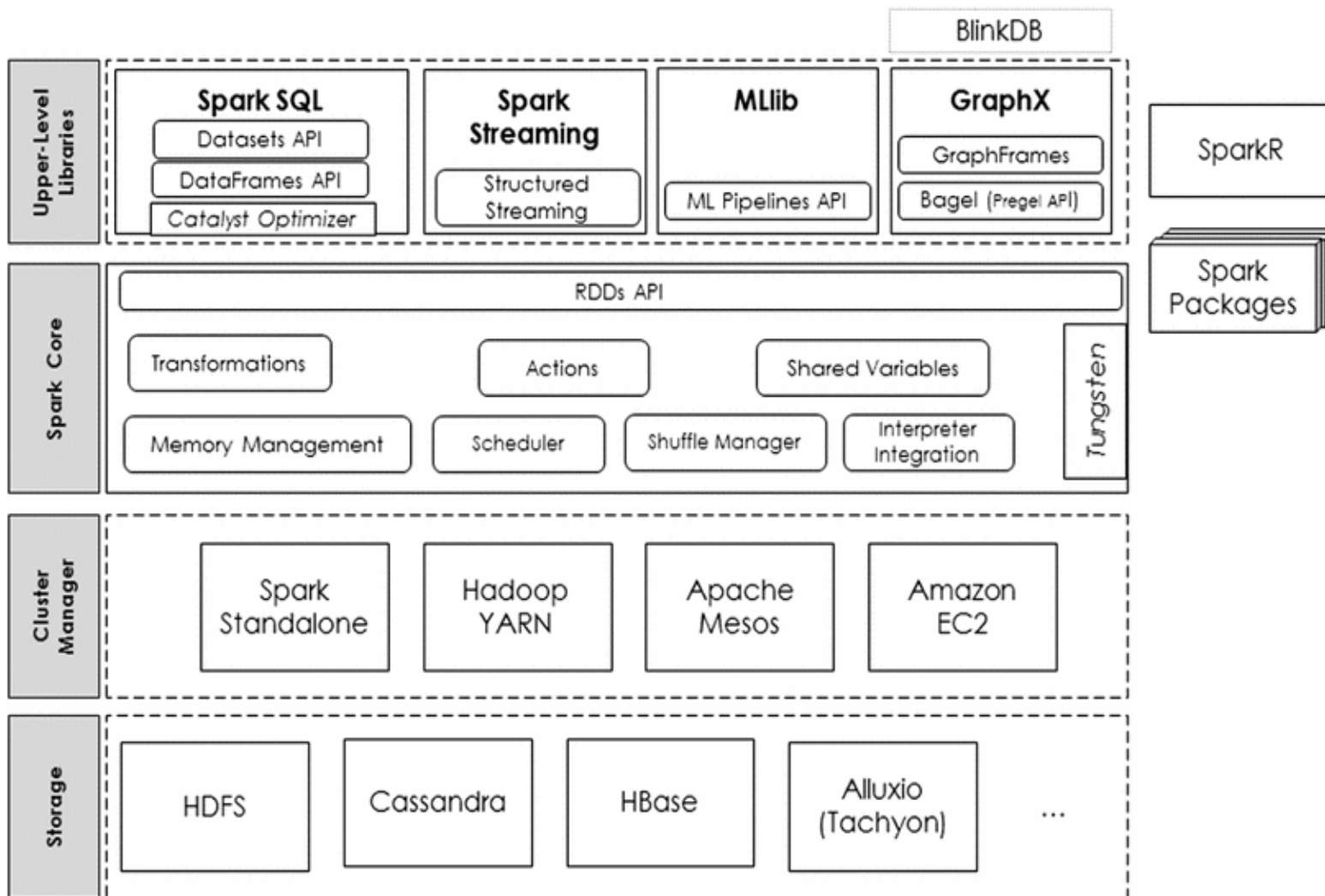
- ▶ Hadoop keretrendszer
- ▶ Spark keretrendszer



Hadoop keretrendszer



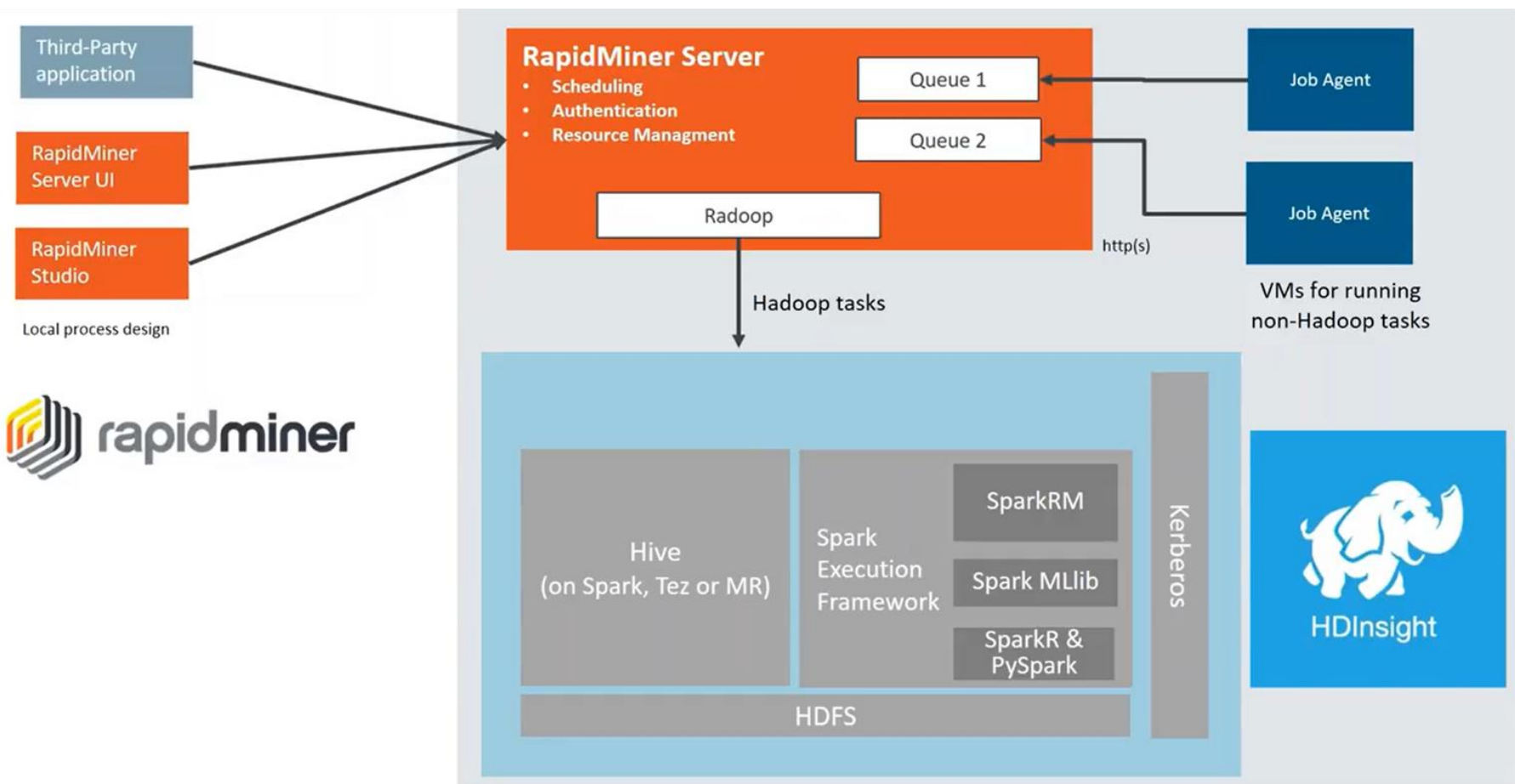
Spark keretrendszer



Rapidminer

- ▶ IDE (Integrated Development Environment)
- ▶ Funkciógazdag front-end workflow készítéshez
- ▶ Radoop backend használatával Spark és Hadoop támogatás
- ▶ Neurális hálózatok létrehozása, telepítése és menedzsmentje is támogatott

Rapidminer és Spark integráció



Rapidminer Studio

Views: Design Results Auto Model

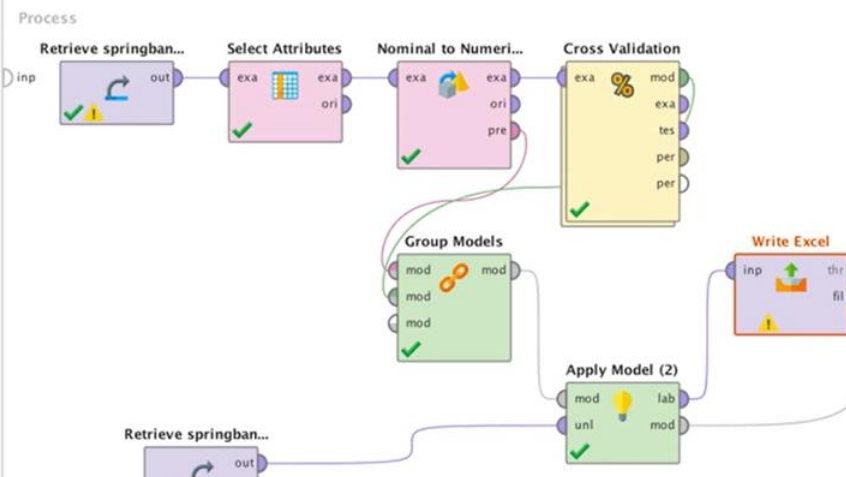
Find data, operators...etc All Studio

Repository

+ Add Data

- Kaggle (florian)
 - KNN (florian)
 - KNN noise (florian - v1, 5/22/17 1:01 PM)
 - KNN SpringBank Drive (florian - v1, 5/22/17 2:37)
 - knn-demo-1 (florian - v1, 5/22/17 2:37)
 - knn-demo-2 (florian - v1, 5/21/17 2:42)
 - SpringBank Drive Linear Regression
 - SpringBank Drive Linear Regression
 - springbankdrive new data (florian - v1, 5/22/17 2:37)
 - springbankdrive sample (florian - v1, 5/22/17 2:37)
 - springbankdrive-text-enriched (florian - v1, 5/22/17 2:37)
 - ManuCore (florian)

Process



```

graph LR
    Inp((inp)) --> RSR[Retrieve springban...]
    RSR --> SA[Select Attributes]
    SA --> NN[Nominal to Numerical]
    NN --> CV[Cross Validation]
    CV --> GM[Group Models]
    GM --> AM[Apply Model (2)]
    AM --> WE[Write Excel]
    RSR --> WE
    
```

Parameters

Write Excel

excel file

file format: **xlsx**

sheet name: **RapidMiner Data**

date format: **HH:mm:ss**

number format: **#.0**

Write Excel.through (through)
 Meta data: Data Table
 • Source: //Classes/KNN/springbankdrive-text-enriched

Number of examples = 10
 27 attributes:
 Generated by: [Write Excel.through](#) ← [Apply Model \(2\).labelled data](#) ← [Retrieve springbankdrive new data.output](#)

| Role | Name | Type | Range | Missin... |
|------|--------------|------------|---------------|-----------|
| | StreetNum... | integer | = [251 - 1... | = 0 |
| | Street | polynomial | = [Boler R... | = 0 |
| | Address | polynomial | = [104 Ox... | = 0 |
| | ZipCode | polynomial | = [NSV, N... | = 0 |
| | Longitude | real | = [-81.33... | = 0 |
| | Latitude | real | = [42.951... | = 0 |

Press "F3" for focus.

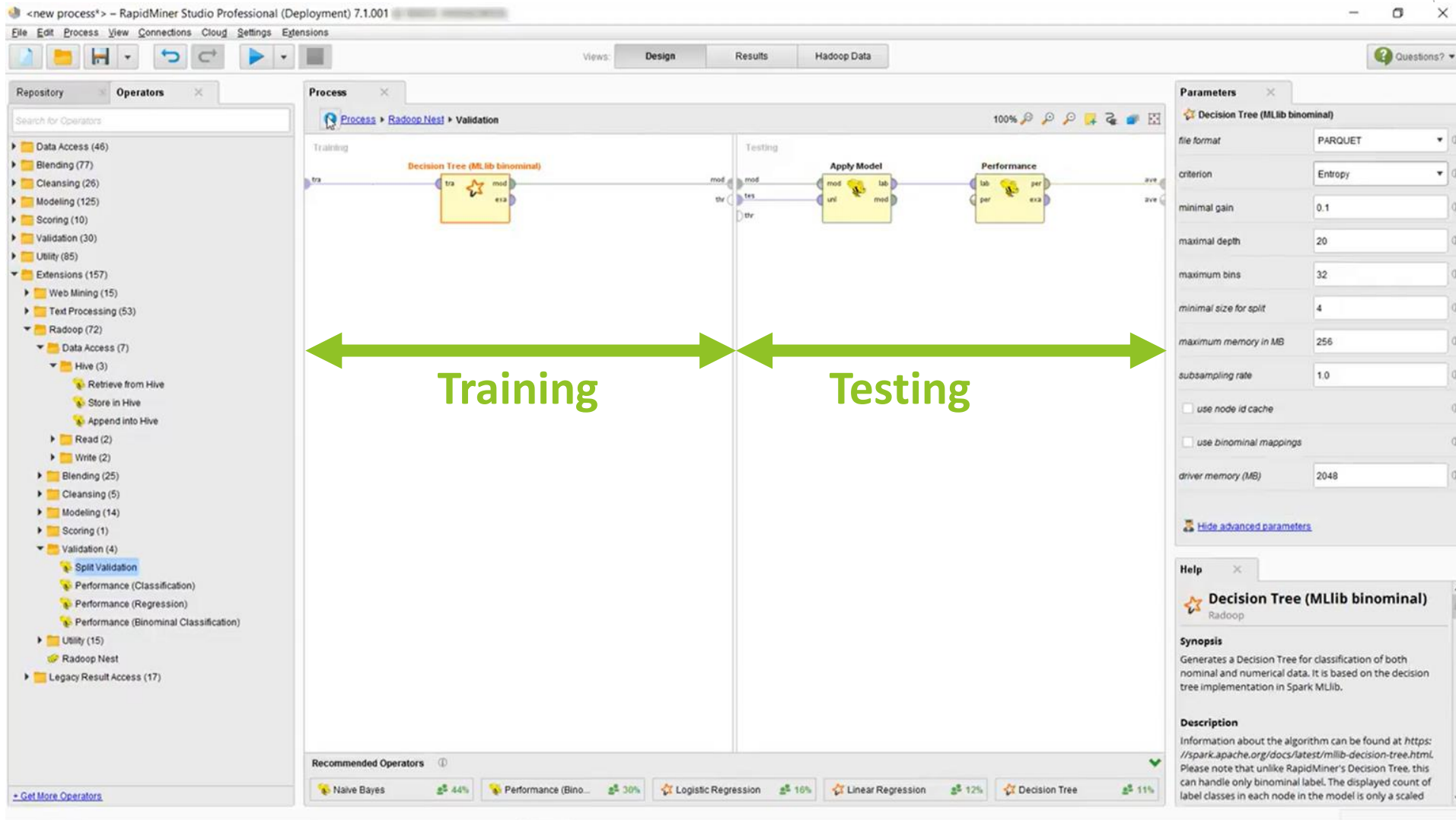
This operator writes an ExampleSet to an Excel spreadsheet file.
[Jump to Tutorial Process](#)

Description

Recommended Operators

- Set Role 37%
- Split Data 28%
- Normalize 28%
- Filter Examples 27%

Rapidminer Studio #2



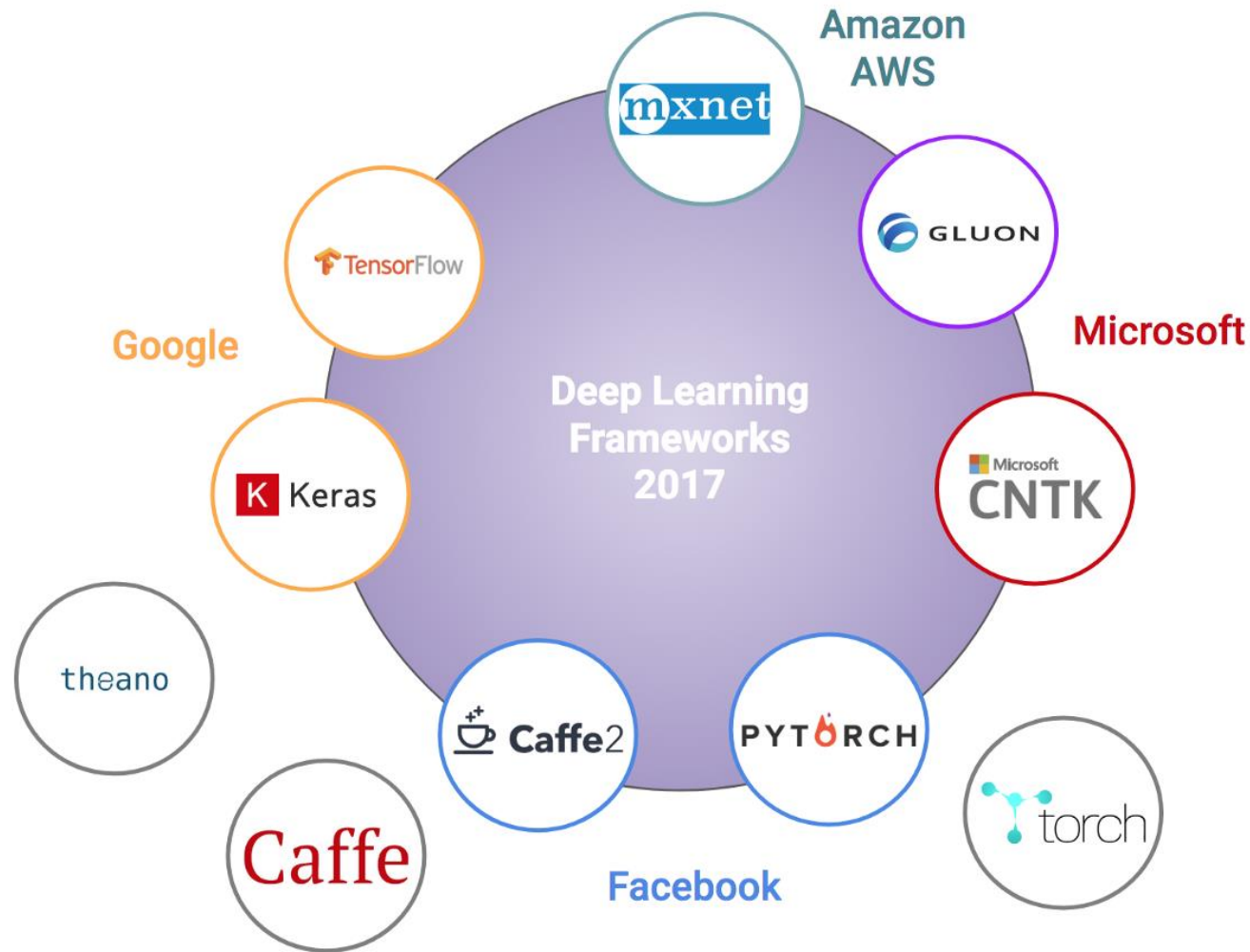
The screenshot displays the Rapidminer Studio Professional (Deployment) 7.1.001 interface. The main workspace is divided into two sections: Training and Testing. In the Training section, a 'Decision Tree (MLlib binominal)' operator is connected to a 'Split Validation' operator. In the Testing section, the 'Decision Tree (MLlib binominal)' operator is connected to an 'Apply Model' operator, which is then connected to a 'Performance' operator. A large green double-headed arrow spans across the Training and Testing sections, with the word 'Training' on the left and 'Testing' on the right. The right sidebar shows the 'Parameters' for the 'Decision Tree (MLlib binominal)' operator, including settings for file format (PARQUET), criterion (Entropy), minimal gain (0.1), maximal depth (20), maximum bins (32), minimal size for split (4), maximum memory in MB (256), and subsampling rate (1.0). The bottom of the interface shows a 'Recommended Operators' section with a list of operators and their usage percentages: Naive Bayes (44%), Performance (Bino... (30%), Logistic Regression (16%), Linear Regression (12%), and Decision Tree (11%).

Deep Learning for Java (DL4J)

- ▶ Támogatott nyelvek:
 - ▶ Java and Scala
- ▶ Apache Hadoop és Spark támogatás elosztott CPU vagy GPU környezetben
- ▶ Széleskörű neurális háló modell támogatás
- ▶ Java nem elterjedt eszköz mély tanulás területén, így nehezen integrálható más keretrendszerekkel
- ▶ A legfőbb előnye, hogy Java alapú rendszerekhez jól illeszthető gépi tanulási feladatokhoz

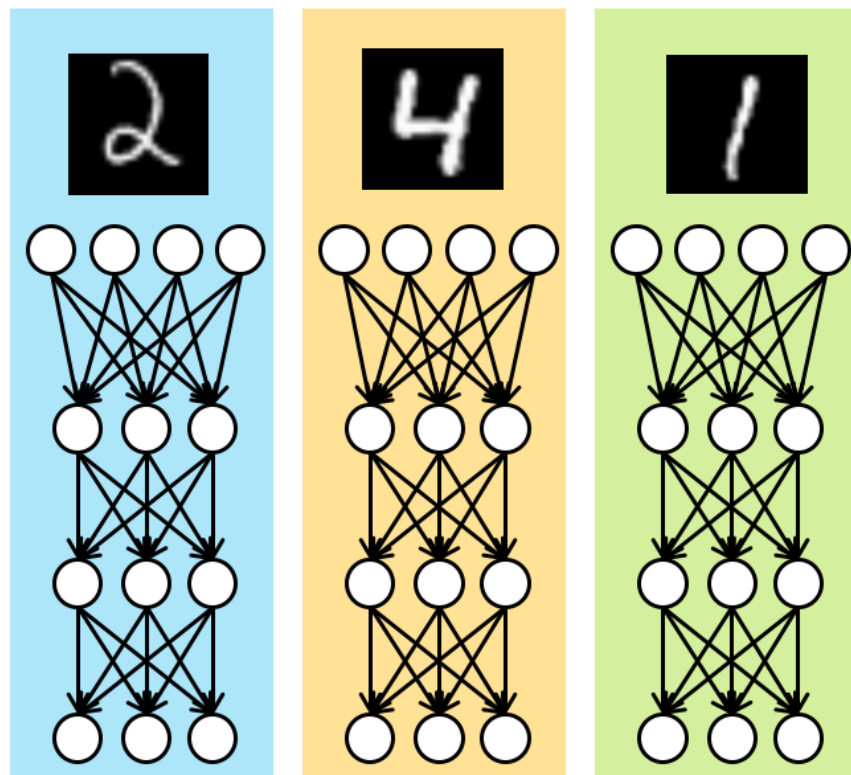


Mély tanulási keretrendszerek

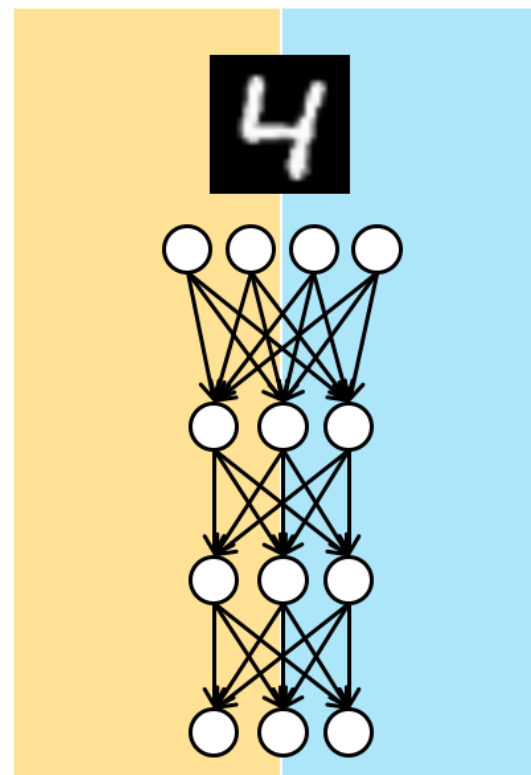


Párhuzamosítási stratégiák

Adat párhuzamos



Modell párhuzamos



Apache MXNet

- ▶ Nyílt forráskódú mély tanulási keretrendszer
- ▶ Széleskörű programozási nyelv támogatás
- ▶ Átjárhatóságot biztosít prototípus és produkciós rendszerek között
- ▶ NVIDIA CUDA alapú GPU támogatás
- ▶ Adat és modell párhuzamos elosztott tanítás
- ▶ Központosított kulcs-érték tár az elosztott tanítás szinkronizálásához



The Microsoft Cognitive Toolkit (CNTK)

- ▶ Nyílt forráskódú mély tanulási keretrendszer produkciós szintű elosztott mély tanuláshoz
- ▶ Microsoft fejlesztés - Azure felhő támogatás
- ▶ A neurális hálózat számítási gráfként való reprezentálása
- ▶ Széleskörű programozási nyelv támogatás
- ▶ NVIDIA CUDA alapú GPU támogatás
- ▶ MPI használata állomások közötti kommunikációra





ELKH Cloud

PyTorch

- ▶ Nyílt forráskódú gépi tanulási keretrendszer prototípus és produkciós rendszerekhez
- ▶ Facebook fejlesztés
- ▶ Python alapú implementáció
- ▶ Tensort definiál a mátrix műveletek GPU-n történő végrehajtásához
- ▶ NVIDIA CUDA alapú GPU támogatás
- ▶ Adat és modell párhuzamos elosztott tanítás
- ▶ MPI és Gloo alapú állomások közötti kommunikáció



TensorFlow

- ▶ Nyílt forráskódú gépi tanulási keretrendszer
- ▶ Google fejlesztés
- ▶ Számítási gráf alapú műveletvégzés
- ▶ Tensort (n-dimenziós mátrix) definiál a matematikai műveletekhez
- ▶ CPU, GPU és TPU támogatás
- ▶ NVIDIA CUDA alapú GPU támogatás
- ▶ TensorFlow 2.0 verziótól a Keras integrálásra került
 - ▶ Magasabb szintű programozási környezet
 - ▶ Zökkenőmentes CPU és GPU közötti váltás
- ▶ gRPC protokollt használ állomások közötti kommunikációhoz



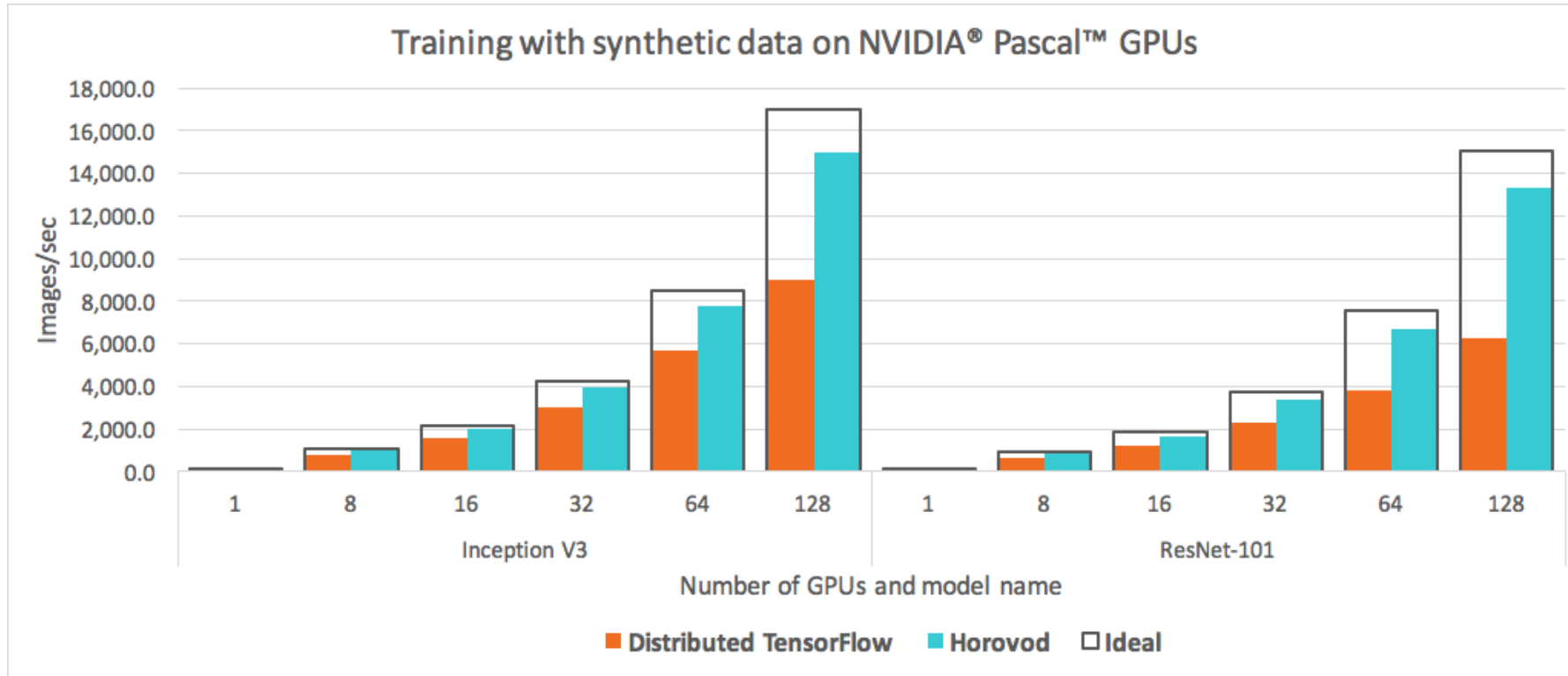
TensorFlow

Horovod

- ▶ Nyílt forráskódú elosztott mély tanulási keretrendszer
- ▶ Uber fejlesztés
- ▶ TensorFlow, Keras, PyTorch, Apache MXNet és Spark támogatás
- ▶ Könnyen alkalmazható elosztott tanítás
- ▶ Adat párhuzamos végrehajtás
 - ▶ Modell párhuzamos végrehajtás egy állomáson belül
- ▶ MPI alapú állomások közötti kommunikáció
- ▶ Baidu eredeti megoldásának továbbfejlesztése
- ▶ Ring-allreduce stratégia
 - ▶ $2*(N-1)$ kommunikáció minden tanítási lépés után
 - ▶ Allreduce $N*(N-1)$ kommunikációt generál lépésenként

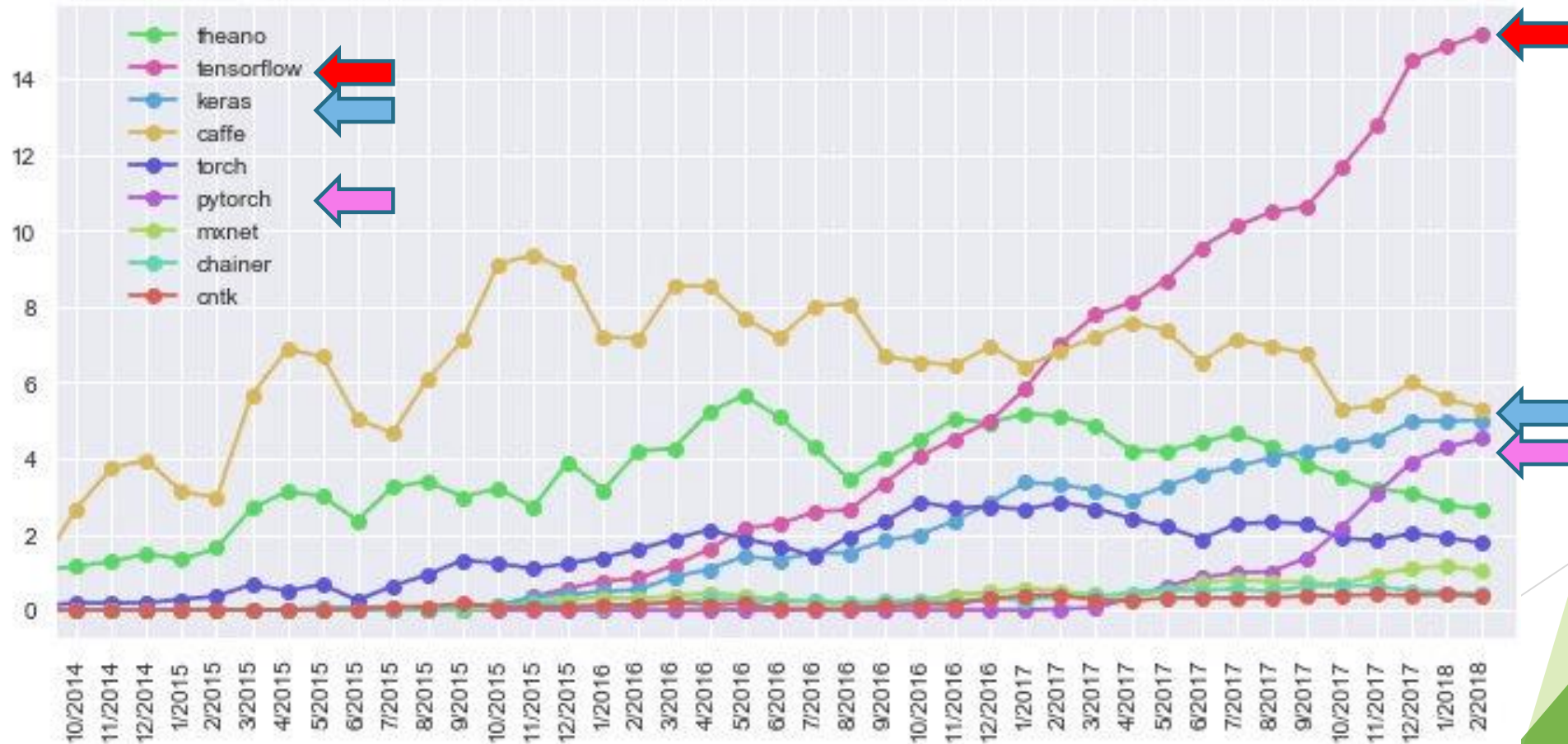


Horovod elosztott tanítás



Mély tanulási keretrendszerek népszerűsége

Percent of ML papers that mention...



Összegzés



- ▶ Adat feldolgozás és Big Data feladatok
 - ▶ Hadoop (disk intenzív műveletek)
 - ▶ Spark (in-memory műveletek)
- ▶ Mély tanulási feladatok
 - ▶ TensorFlow (Keras)
 - ▶ PyTorch
- ▶ Elosztott mély tanulási feladatok
 - ▶ Horovod



ELKH Cloud

Köszönöm a figyelmet!