Simplifying End-to-End Big Data AI

Jason Dai

Intel Fellow

intel

Agenda

- Big Data AI
- Building Big Data AI applications
- Summary

Agenda

- **Big Data AI**
- Building Big Data AI applications
- Summary

Why Big Data AI?

Transformation of Big Data

- Storing and processing more data
- Analyzing (querying) more data
- Real-time analysis
- Modelling and prediction (ML/DL)

AI is everywhere

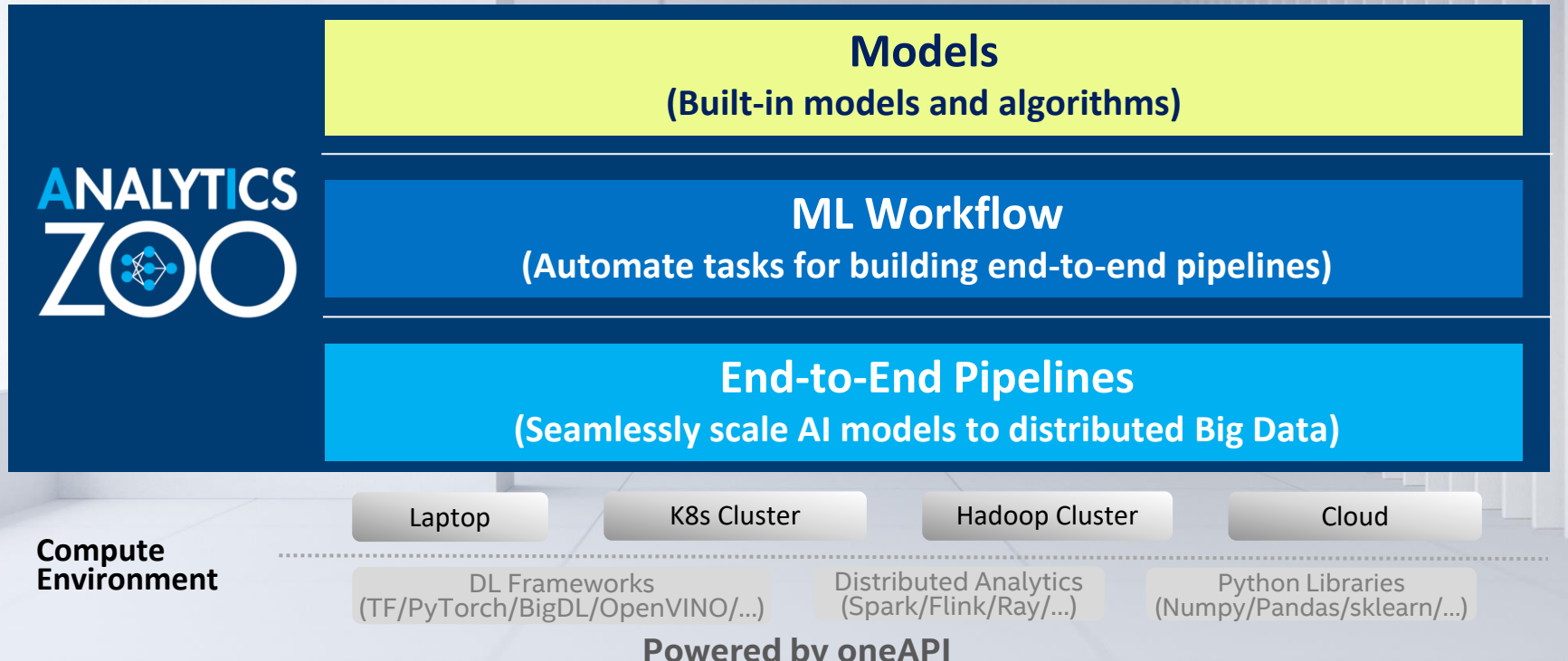
- Moving from experimentation to production
- Applying to large-scale, distributed Big Data
- End-to-end AI pipeline

End-to-End AI Pipeline

Growing Demand for End-to-End Big Data AI Pipeline

intel

Analytics Zoo: Software Platform for Big Data AI

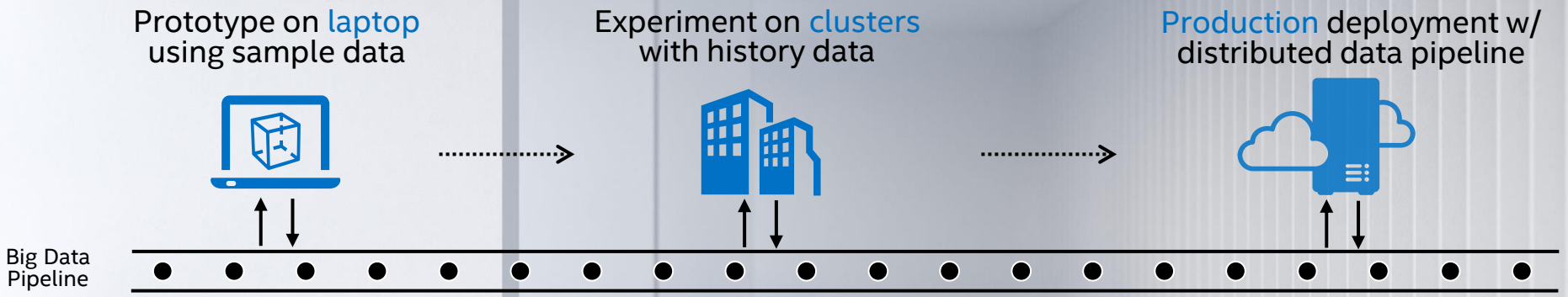


https://github.com/intel-analytics/analytics-zoo

intel

Analytics Zoo: End-to-End Big Data AI

Seamless Scaling from Laptop to Distributed Big Data



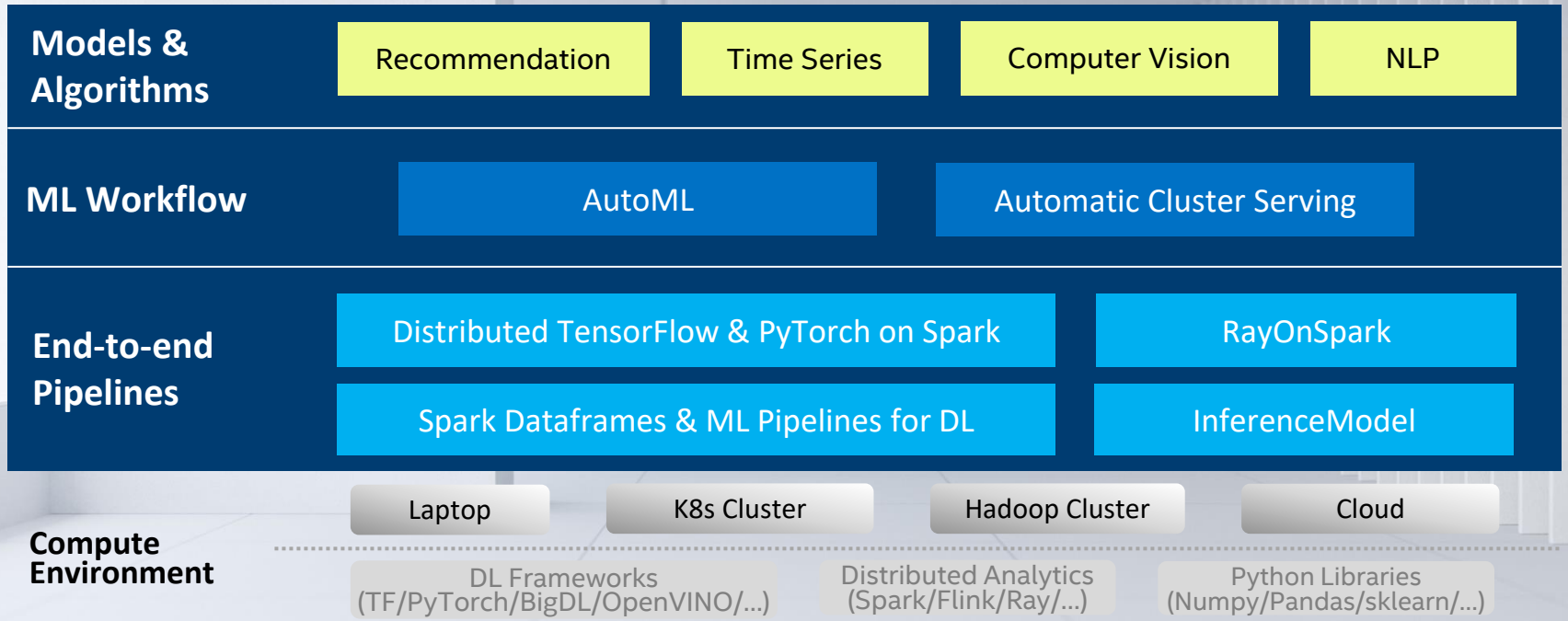
- Easily prototype end-to-end pipelines that apply AI models to big data
- "Zero" code change from laptop to distributed cluster
- Seamlessly deployed on production Hadoop/K8s clusters
- Automate the process of applying machine learning to big data

intel

Agenda

- Big Data AI
- **Building Big Data AI applications**
- Summary

Analytics Zoo: Software Platform for Big Data AI

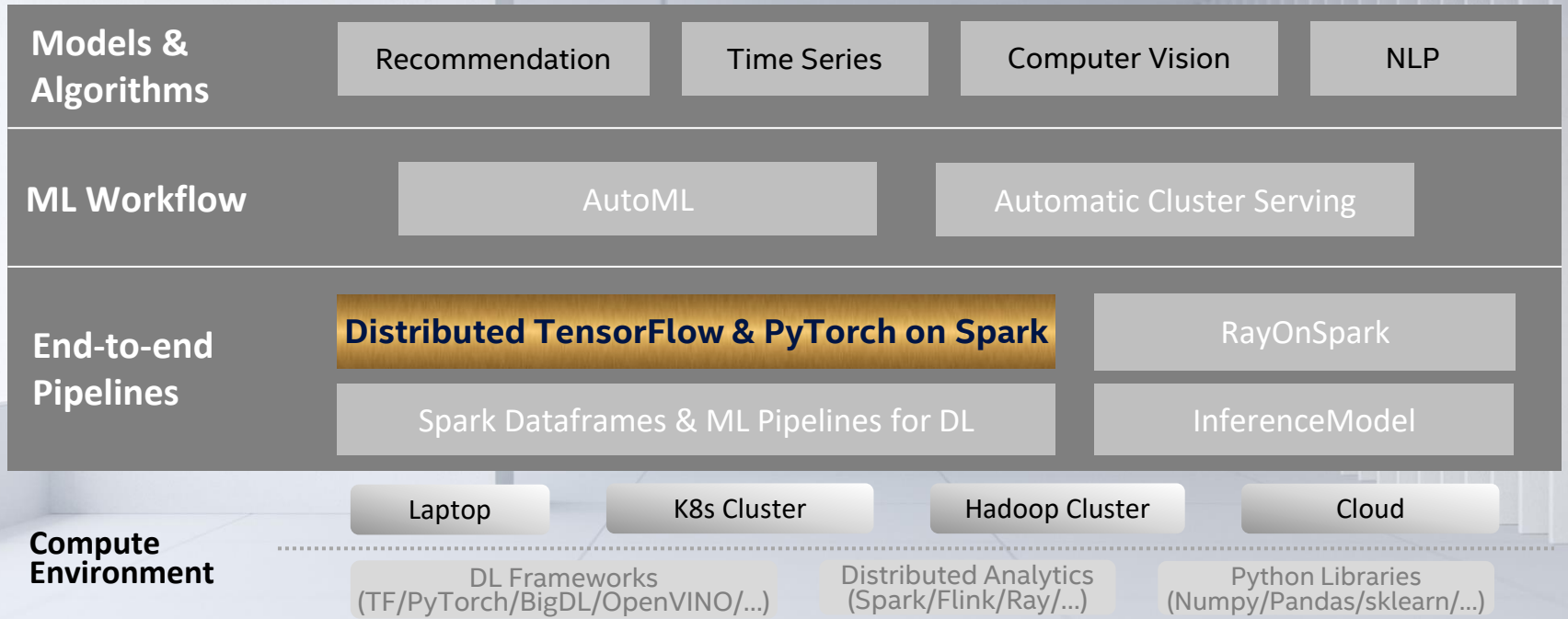


Powered by oneAPI

https://github.com/intel-analytics/analytics-zoo

intel

Analytics Zoo: Software Platform for Big Data AI



Powered by oneAPI

https://github.com/intel-analytics/analytics-zoo

intel

Distributed TensorFlow/PyTorch on Spark

Write TensorFlow/PyTorch inline with Spark code

```
#spark dataframe
train_df = spark.read.parquet(…).select(…)

#tensorflow code
from tensorflow import keras
...
model = keras.models.Model(inputs=[user, item], outputs=predictions)
model.compile(…)

#distributed training on Spark
from zoo.orca.learn.tf.estimator import Estimator
est = Estimator.from_keras(keras_model=model, …)
est.fit(data=train_df,
        feature_cols=['user', 'item'],
        label_cols=['label'],
        …)
```

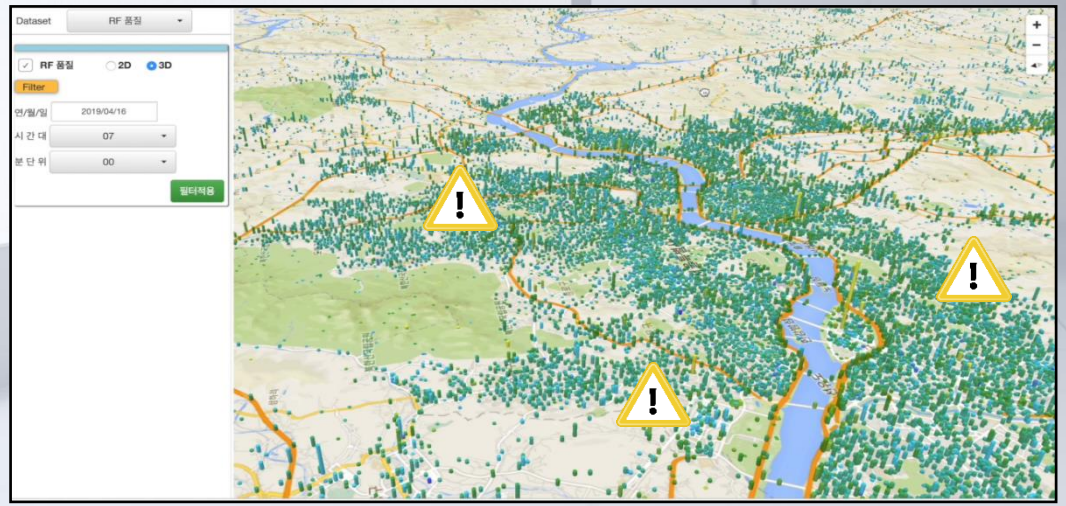
Analytics Zoo API in blue

intel

Time-Series Network Quality Prediction in SK Telecom

Distributed TensorFlow/PyTorch on Spark

- Predict Network Quality Indicators (CQI, RSRP, RSRQ, SINR, …)* for anomaly detection and real-time management

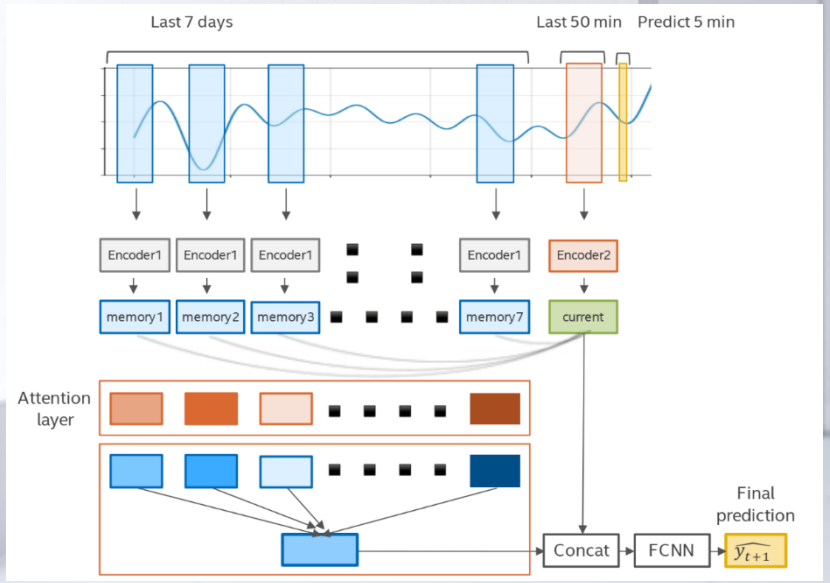


** CQI : Channel Quality Indicator*
** RSRP : Reference Signal Received Power*
** RSRQ : Reference Signal Received Quality*
** SINR :Signal to Interference Noise Ratio*

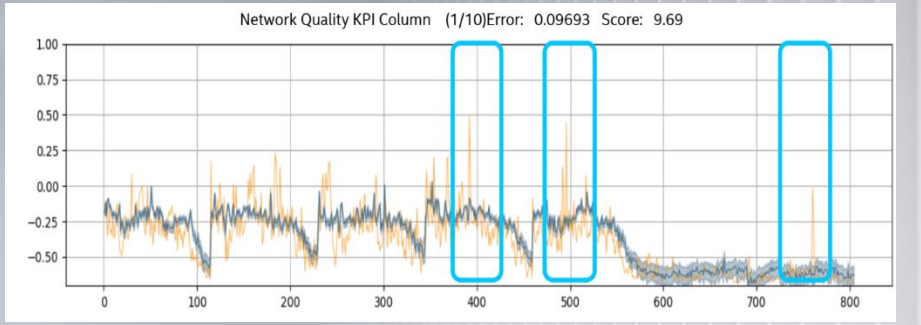
* "Vectorized Deep Learning Acceleration from Preprocessing to Inference and Training on Apache Spark in SK Telecom", Spark + AI Summit 2020
* https://networkbuilders.intel.com/solutionslibrary/sk-telecom-intel-build-ai-pipeline-to-improve-network-quality

intel

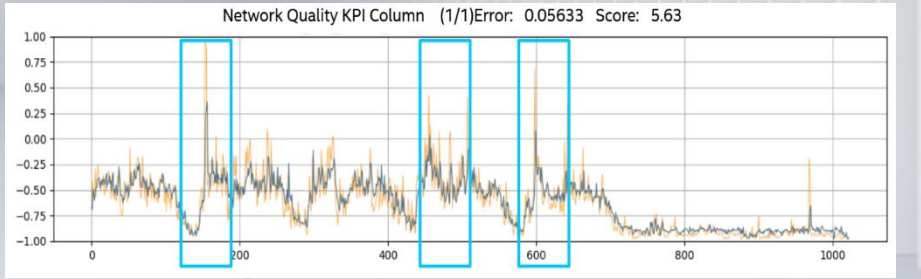
Model: Memory Augmented Network



seq2seq



Mem-network



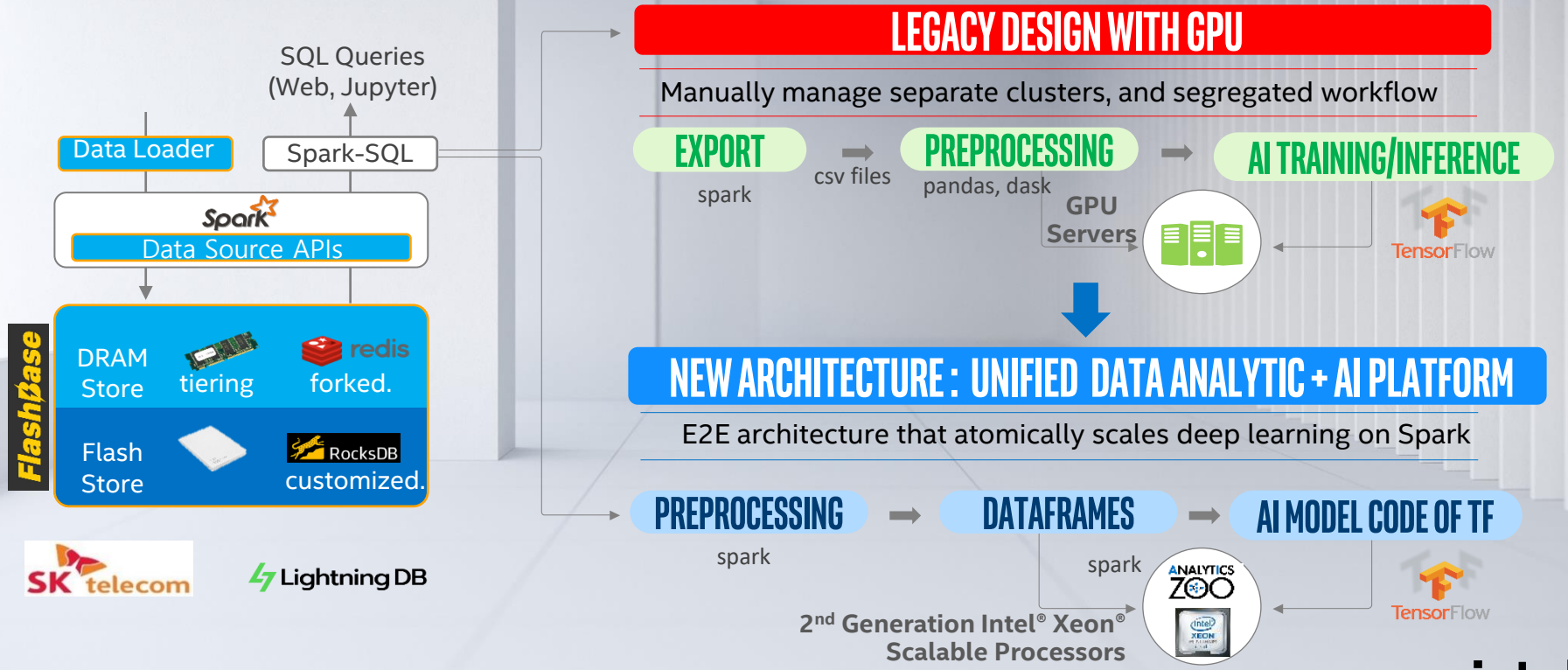
Improved predictions for sudden change!

https://networkbuilders.intel.com/solutionslibrary/sk-telecom-intel-build-ai-pipeline-to-improve-network-quality

intel

Architecture: Migrating to Analytics Zoo

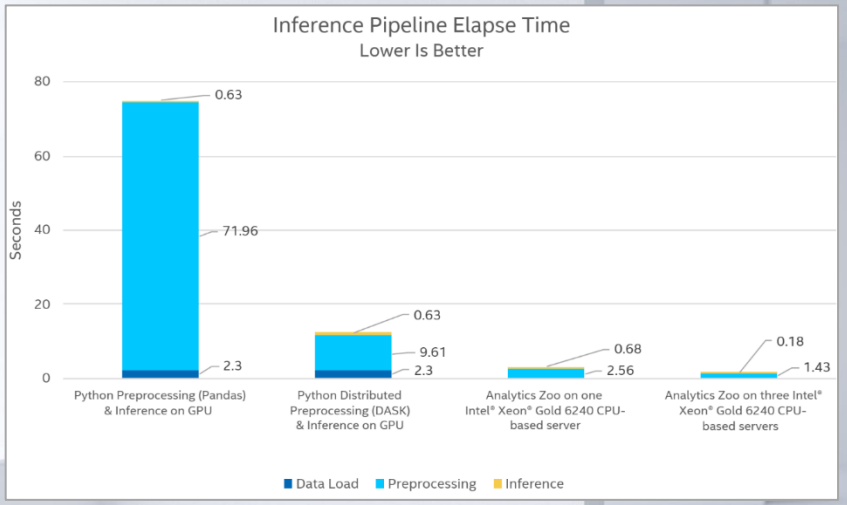
Distributed TensorFlow/PyTorch on Spark

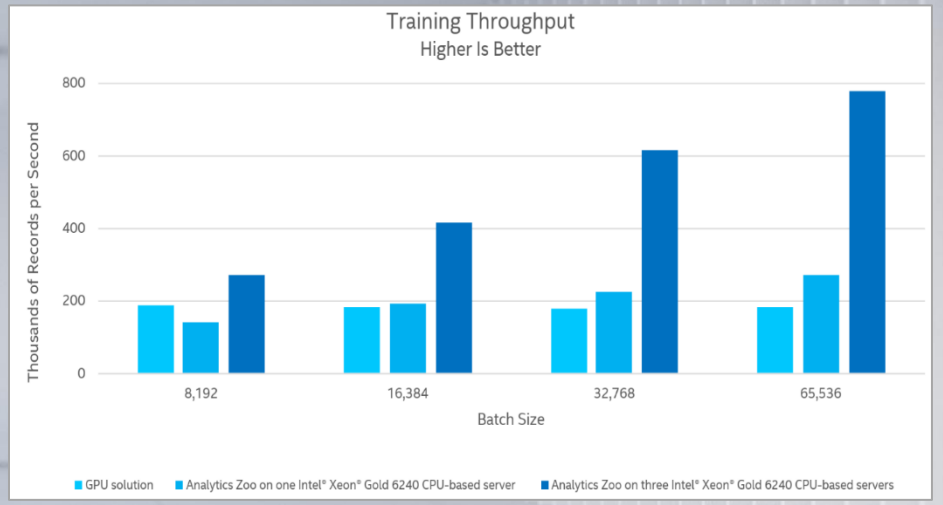


https://networkbuilders.intel.com/solutionslibrary/sk-telecom-intel-build-ai-pipeline-to-improve-network-quality

intel

Inference and Training Speed-up with Analytics Zoo





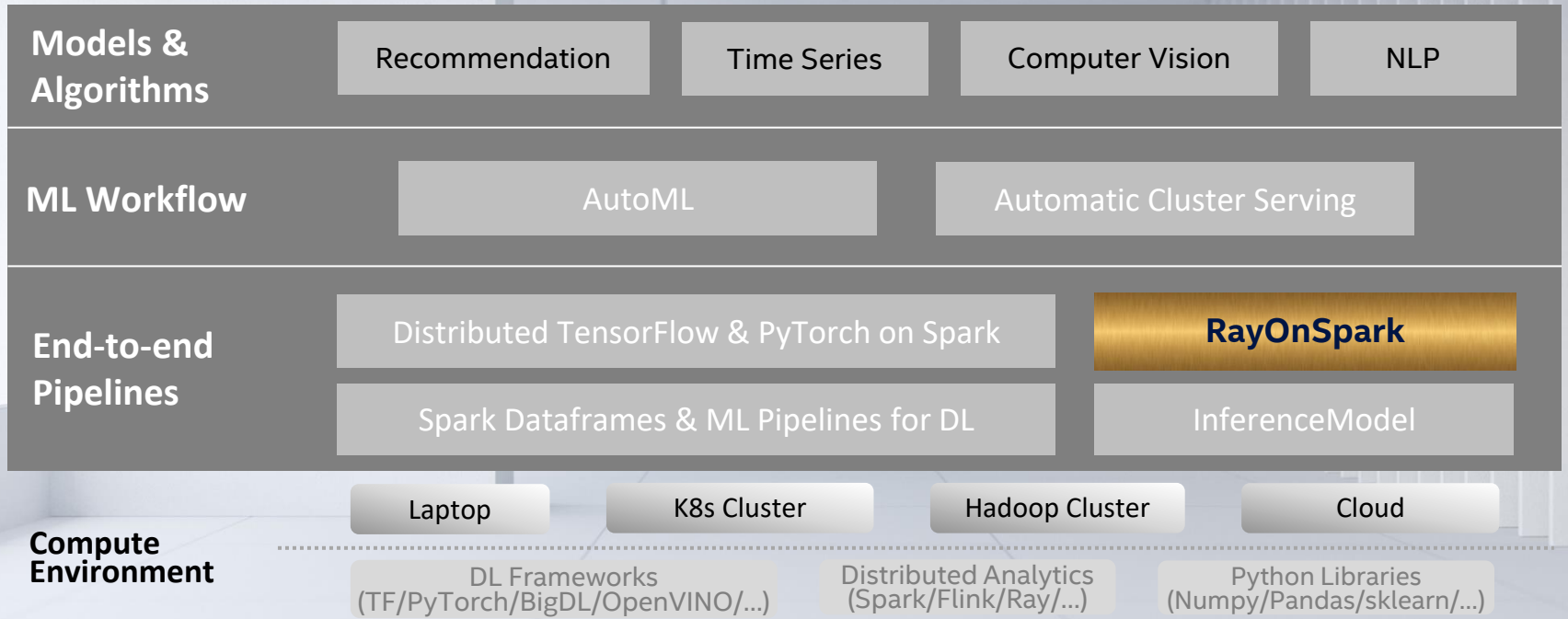
Up-to 6x speedup for end-to-end inference running Analytics Zoo on Xeon in SK Telecom*

Up-to 4x speedup for end-to-end training running Analytics Zoo on Xeon in SK Telecom*

* https://networkbuilders.intel.com/solutionslibrary/sk-telecom-intel-build-ai-pipeline-to-improve-network-quality

intel

Analytics Zoo: Software Platform for Big Data AI



Powered by oneAPI

https://github.com/intel-analytics/analytics-zoo

intel

RayOnSpark

Run Ray Programs Directly on Big Data Platform

```
from zoo.orca import init_orca_context, stop_orca_context

init_orca_context(cluster_mode="yarn", …, init_ray_on_spark=True)

#Ray code
@ray.remote
class TestRay():
    def hostname(self):
        import socket
        return socket.gethostname()

actors = [TestRay.remote() for i in range(0, 100)]
print([ray.get(actor.hostname.remote()) for actor in actors])

stop_orca_context()
```

https://medium.com/riselab/rayonspark-running-emerging-ai-applications-on-big-data-clusters-with-ray-and-analytics-zoo-923e0136ed6a

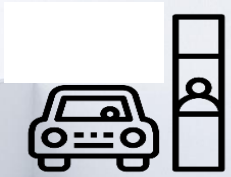
Analytics Zoo API in blue

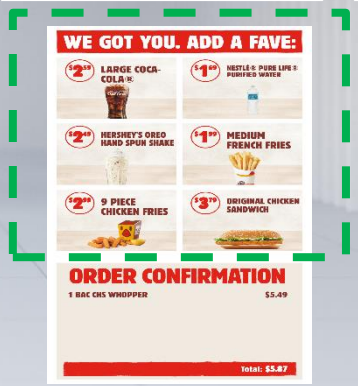
intel

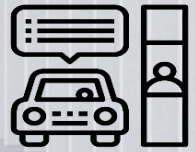
Fast Food Recommendation in Burger King

End-to-End Training Pipeline w/ RayOnSpark

Guest arrives ODMB → Checks Menu Board → Cashier enters order → Checks Menu Board → Guest completes order



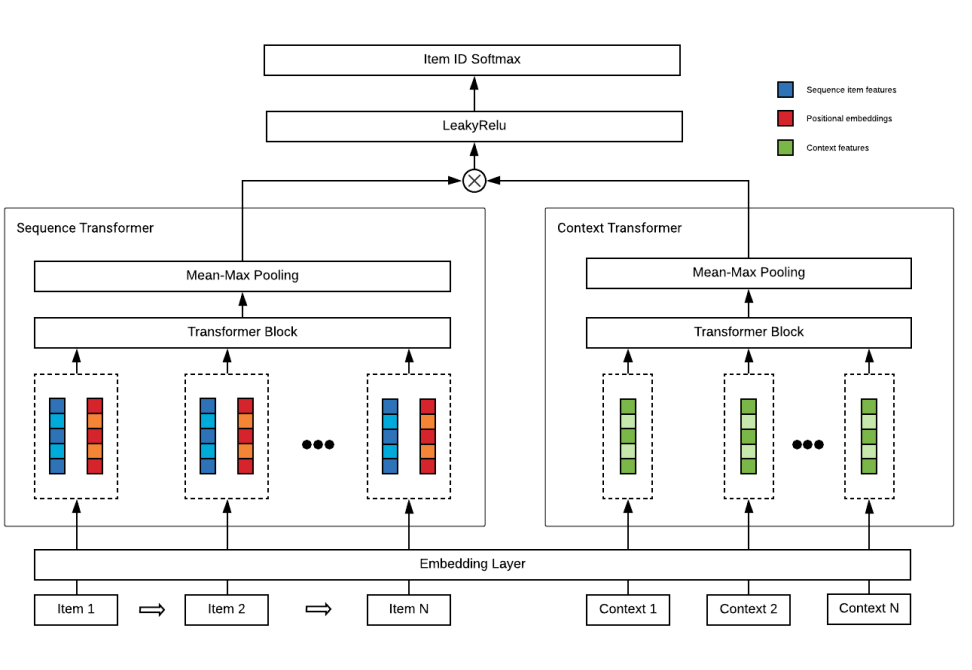




* https://medium.com/riselab/context-aware-fast-food-recommendation-at-burger-king-with-rayonspark-2e7a6009dd2d
* *"Context-aware Fast Food Recommendation with Ray on Apache Spark at Burger King", Data + AI Summit Europe 2020*

intel

Model: Transformer Cross Transformer (TxT) Model



Model Components

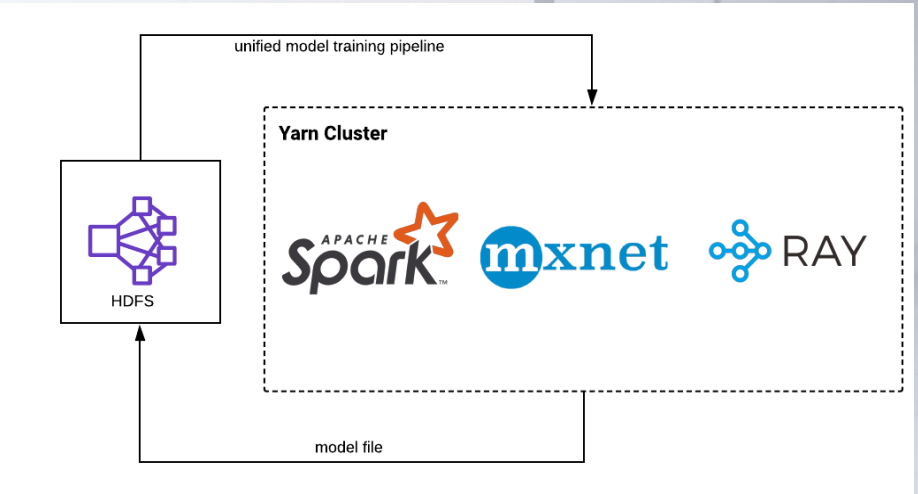
- **Sequence Transformer**
  - Taking item order sequence as input
- **Context Transformer**
  - Taking multiple context features as input
- **Latent Cross Joint Training**
  - Element-wise product for both transformer outputs

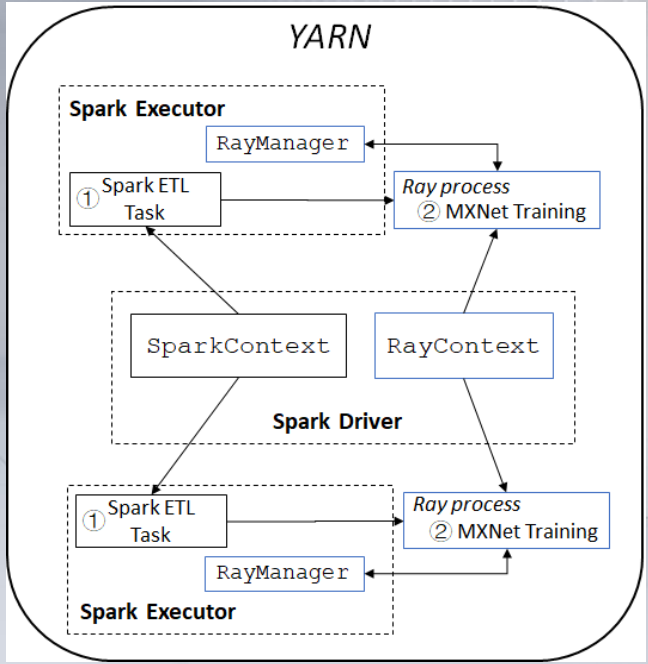
* https://medium.com/riselab/context-aware-fast-food-recommendation-at-burger-king-with-rayonspark-2e7a6009dd2d
* *"Context-aware Fast Food Recommendation with Ray on Apache Spark at Burger King", Data + AI Summit Europe 2020*

intel

Architecture: Unified Data Processing and Training

End-to-End Pipeline w/ RayOnSpark

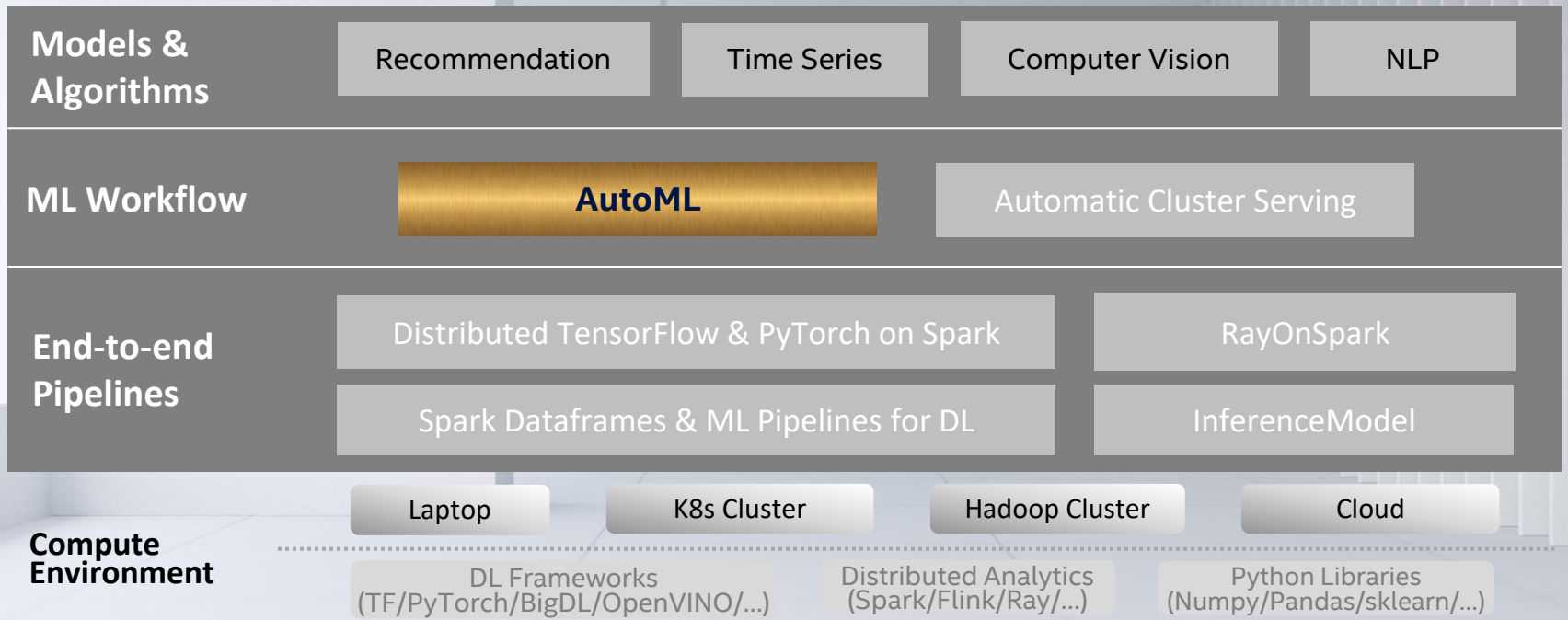




* https://medium.com/riselab/context-aware-fast-food-recommendation-at-burger-king-with-rayonspark-2e7a6009dd2d
* *"Context-aware Fast Food Recommendation with Ray on Apache Spark at Burger King", Data + AI Summit Europe 2020*

intel

Analytics Zoo: Software Platform for Big Data AI



https://github.com/intel-analytics/analytics-zoo

intel

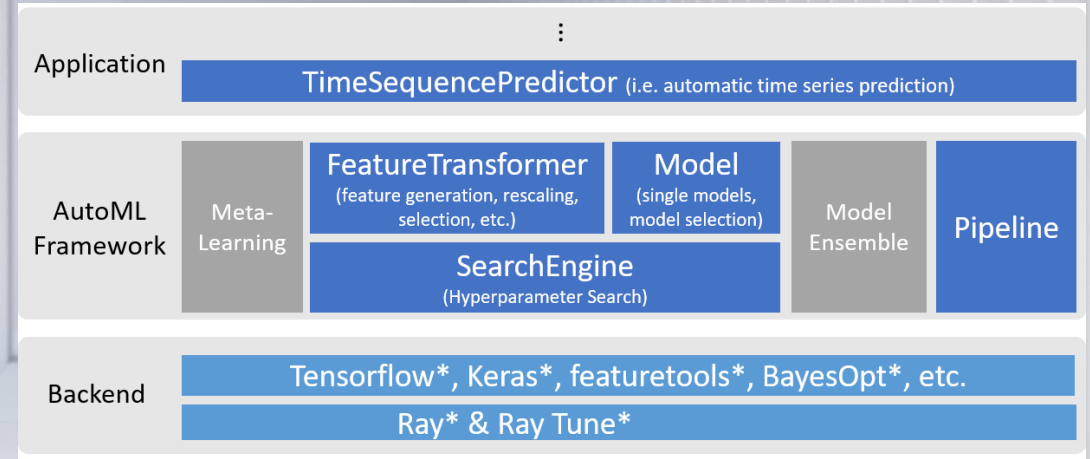
Scalable AutoML for Time Series Prediction

Automated feature generation, model selection and hyper parameter tuning

```
tsp = TimeSequencePredictor( \
            dt_col="datetime",
            target_col="value")

pipeline = tsp.fit(train_df,
             val_df, metric="mse",
             recipe=RandomRecipe())

pipeline.predict(test_df)
```



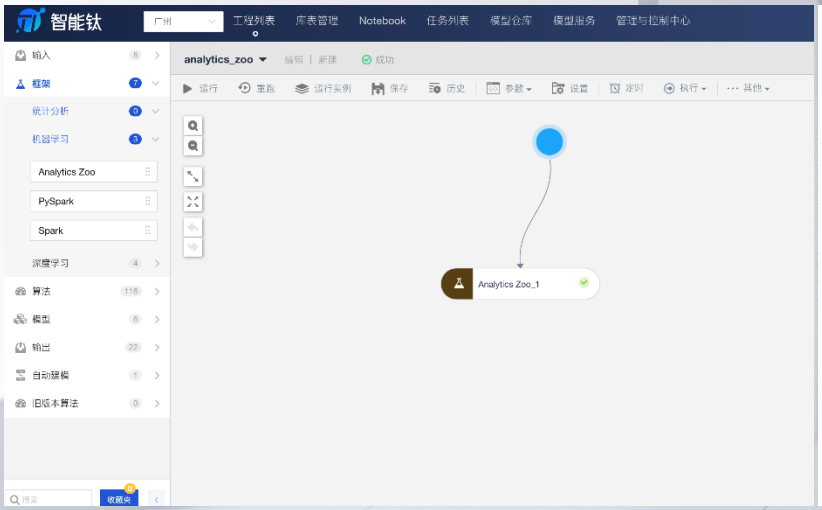
https://medium.com/riselab/scalable-automl-for-time-series-prediction-using-ray-and-analytics-zoo-b79a6fd08139

Analytics Zoo API in blue

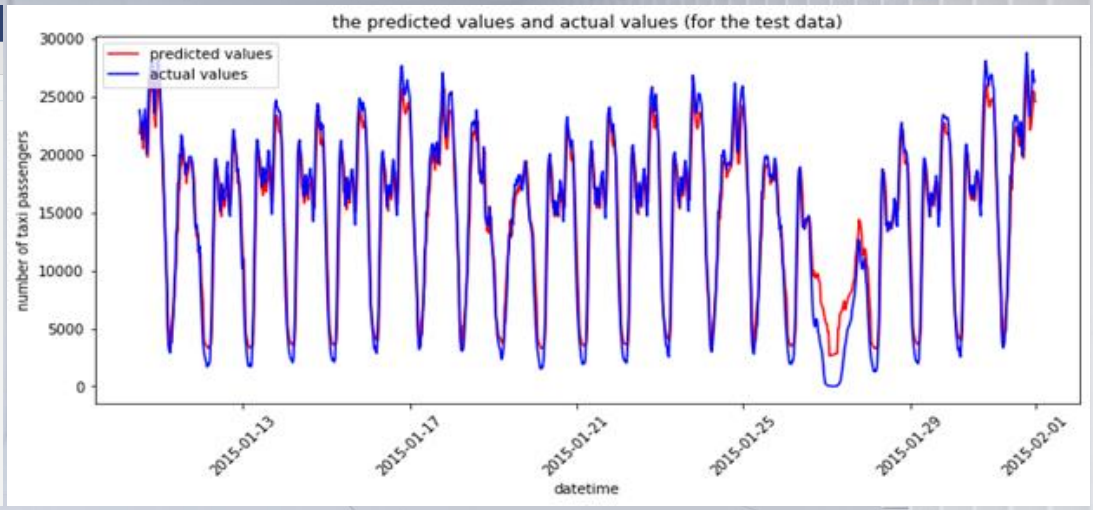
intel

TI-One ML Platform in Tencent Cloud

Scalable AutoML for Time Series Prediction



Using Analytics Zoo in Tencent Cloud TI-One ML Platform



Predicting NYC Taxi Passengers Using AutoML

https://software.intel.com/content/www/us/en/develop/articles/tencent-cloud-leverages-analytics-zoo-to-improve-performance-of-ti-one-ml-platform.html

intel

Summary

INDUSTRY INFLECTIONS ARE FUELING THE GROWTH OF DATA

5G Network Transformation, Artificial Intelligence, Intelligent Edge, Cloudification

AI & ANALYTICS ARE THE DEFINING WORKLOADS OF THE NEXT DECADE

with growing demand for end-to-end AI pipeline

UNMATCHED PORTFOLIO BREADTH AND ECOSYSTEM SUPPORT

Intel delivers a silicon & software foundation designed for the diverse range of use cases from the cloud to the edge

ANALYTICS ZOO OPEN-SOURCE SOFTWARE PLATFORM FOR BIG DATA AI

Simplifies End-to-End Big Data AI pipeline solutions development

intel

Reference

Analytics Zoo: Software Platform for Big Data AI

- E2E Big Data & AI pipeline (distributed TF/PyTorch/OpenVINO/Ray on Spark)
- Advanced AI workflow (AutoML, Time-Series, Cluster Serving, etc.)

Github

- Project repo: https://github.com/intel-analytics/analytics-zoo

  Documentation: https://analytics-zoo.readthedocs.io/
- Use cases: https://analytics-zoo.readthedocs.io/en/latest/doc/Application/powered-by.html

Technical paper/tutorials

- CVPR 2020 tutorial: https://jason-dai.github.io/cvpr2018/
- ACM SoCC 2019 paper: https://arxiv.org/abs/1804.05839
- AAAI 2019 tutorial: https://jason-dai.github.io/aaai2019/
- CVPR 2018 tutorial: https://jason-dai.github.io/cvpr2018/


Thank You
intel
intel

Notices & Disclaimers

- Performance varies by use, configuration and other factors. Learn more at www.intel.com/performanceIndex
- Performance may vary based on the specific game title and server configuration. To reference the full list of Intel Server GPU platform measurements, please refer to http://www.intel.com/content/www/us/en/benchmarks/server/graphics/IntelServerGPU
- All product plans and roadmaps are subject to change without notice.
- Intel technologies may require enabled hardware, software or service activation.
- No product or component can be absolutely secure.
- Your costs and results may vary.
- Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.
- All product plans and roadmaps are subject to change without notice.
- © Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others. Intel Server GPU TCO analysis is based on internal Intel research. Pricing as of 10/01/2020. Analysis assumes standard serving pricing, GPU list pricing, and software pricing based on estimated Nvidia software license costs of $1 per year for 5 years.
- Intel Server GPU Performance may vary based on the specific game title and server configuration. To reference the full list of Intel Server GPU platform measurements, please refer to http://www.intel.com/content/www/us/en/benchmarks/server/graphics/IntelServerGPU
- Video game footage courtesy of Tencent Games and Gamestream.
- LEGO STAR WARS TITLES : © Lucasfilm Entertainment Company Ltd. or Lucasfilm Ltd. & ® or TM as indicated. All rights reserved.
- LEGO, the LEGO logo and the Minifigure are trademarks of The LEGO Group. © The LEGO Group. All rights reserved.
- “DiRT4”™ : © 2017 The Codemasters Software Company Limited ("Codemasters"). All rights reserved. "Codemasters"®, “EGO”®, the Codemasters logo, and “DiRT”® are registered trademarks owned by Codemasters. “DiRT4”™ and “RaceNet”™ are trademarks of Codemasters. All rights reserved. Under licence from International Management Group (UK) Limited. All other copyrights or trademarks are the property of their respective owners and are being used under license. Developed by Codemasters.