MLeap + Combust.ML

Deploy Your Spark Pipelines Directly To Production

Github: https://github.com/combust-ml/mleap

Hollin Wilkins and Mikhail Semeniuk

Combust.ML

SPARK SUMMIT
EUROPE 2016

Opening Demo

How much should I rent my house for on AirBnb?

```
http://combust.ml/airbnb
```

Yes, open your cell phone and go here :)

Problem Statement: Deploying machine learning algorithms to a production environment is a lot more difficult than it has to be and is a common source of friction at data-driven organizations

Everyone wants to do better! The winning technology will be the one that enables Engineers and Data Scientists to collaborate and work across a single platform.

Outdated Research <> Engineering Dynamics

Action		Reaction
- Data scientists write data pipelines to construct research datasets	↔	- Engineers re-write the data pipelines for a production-ready system
- Engineers write scalable libraries for computing features and algorithms	↔	- Data scientists largely don't use those libraries and maintain/re-write their own copy of the code
- Data scientists largely focus on linear/logistic regressions due to engineering constraints	↔	- Talented engineers get largely tired of coding up linear regressions and updating coefficients

Hadoop and HDFS helped bridge the data gap.

Spark has bridged the language gap, by providing a common set of APIs to easly process data and train models

MLeap and Combust.ML extend Spark functionality by allowing researchers and engineers to deploy pipelines as a service

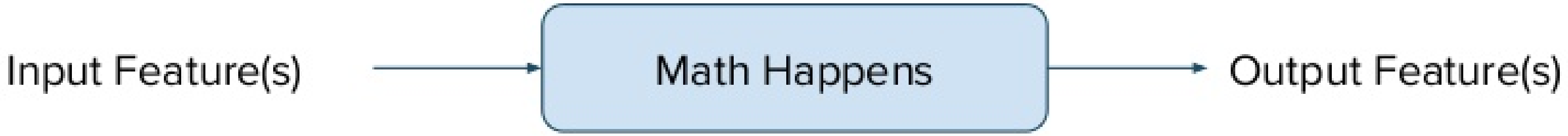
Existing Solutions: You won’t believe how many companies are still deploying algorithms in a SQL environment! And these are billion dollar operations.

	Hard-Coded Models (SQL, Java, Ruby)	PMML	Emerging Solutions (yHat, DataRobot)	Enterprise Solutions (Microsoft, IBM, SAS)	MLeap + Combust.ML
Quick to Implement	⊖	✓	⊖	⊖	✓
Open Sourced	⊖	✓	⊖	⊖	✓
Committed to Spark/Hadoop	⊖	⛔	⛔	✓	✓
API Server Infrastructure	⊖	⛔	✓	⊖	✓

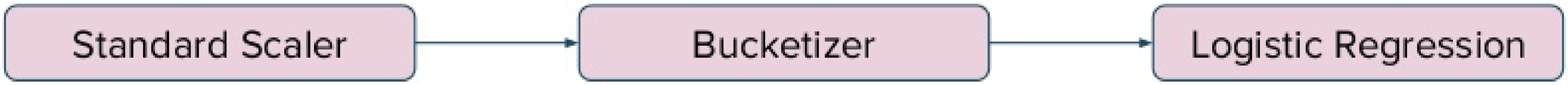
Lesson Learned: Push code down to where the data is, not the other way around!

Overview of Pipelines and Transformers

A Transformer generates a new feature or a vector of features based on an input or a vector of inputs. Some transformers need to be trained, while others are basic algebraic functions.

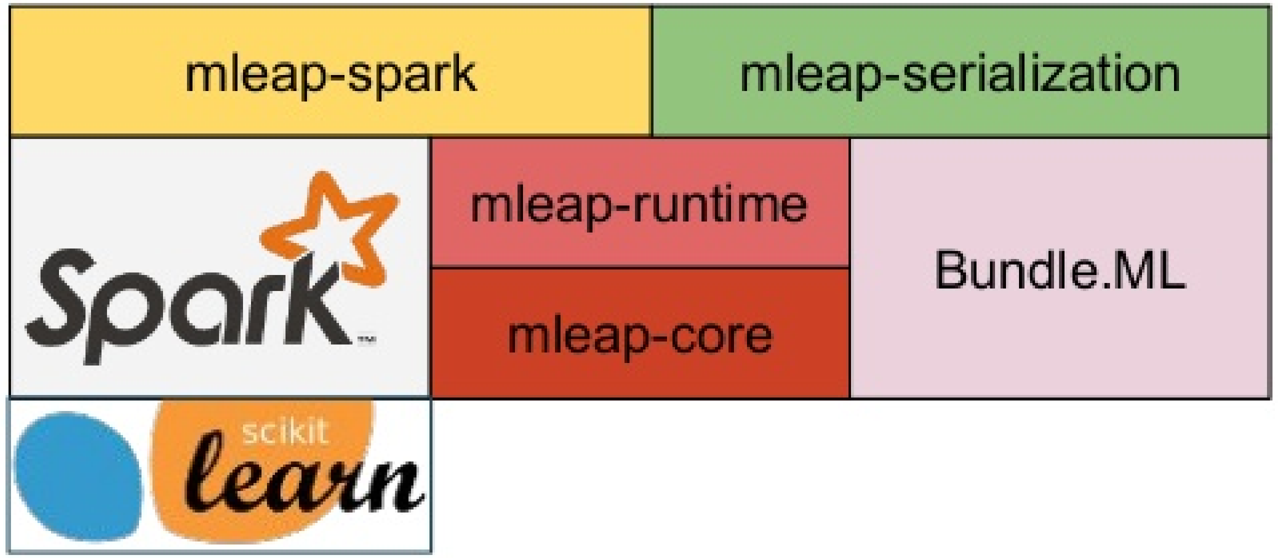


Pipelines piece together a series of transformers and generally start with feature transformers and end with a model transformer (your algorithm).



MLeap Components

- mleap-core - feature builders, regression models, classification models, clustering models, ANN
- mleap-runtime - provides DataFrame-like "LeapFrame" and MLeap transformers
- mleap-spark - serialize to Bundle.ML, execute MLeap transformers on Spark dataframes
- bundle-ml - common serialization format for Spark, MLeap, Scikit-Learn, TensorFlow
- mleap-scikit - MLeap <> Scikit-Learn transformers integration



mleap-spark

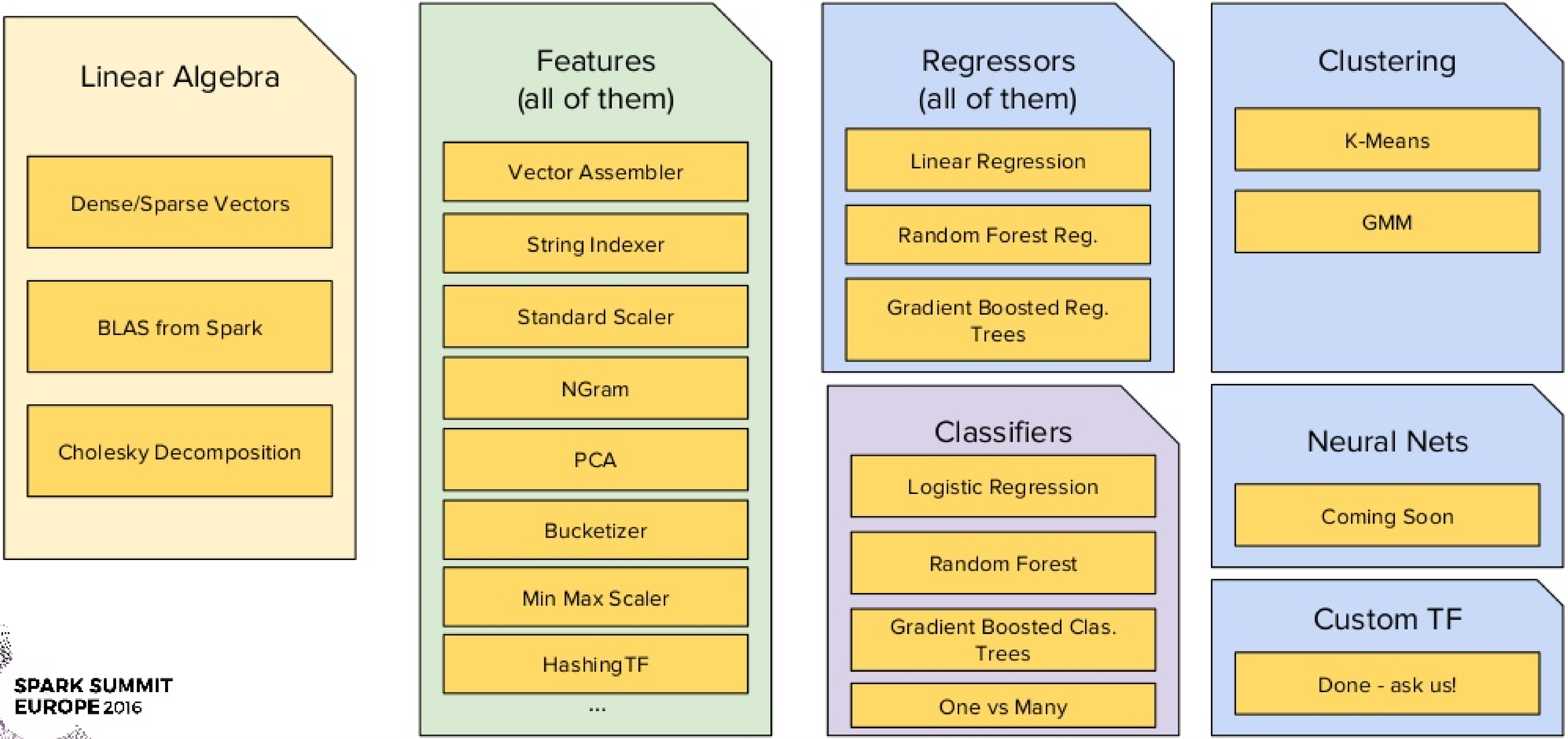
mleap-serialization

mleap-runtime

mleap-core

Bundle.ML

MLeap Core Components



Linear Algebra

- Dense/Sparse Vectors
- BLAS from Spark
- Cholesky Decomposition

Features (all of them)

- Vector Assembler
- String Indexer
- Standard Scaler
- NGram
- PCA
- Bucketizer
- Min Max Scaler
- HashingTF
- ...

Regressors (all of them)

- Linear Regression
- Random Forest Reg.
- Gradient Boosted Reg. Trees

Classifiers

- Logistic Regression
- Random Forest
- Gradient Boosted Clas. Trees
- One vs Many

Clustering

- K-Means
- GMM

Neural Nets

- Coming Soon

Custom TF

- Done - ask us!

MLeap Runtime

Power and functionality of Spark Transformers without the dependency on the Spark context. Deploy anywhere!

- Provides a LeapFrame, which stores data for transformations by MLeap transformers, which mirror the transform functionality of Spark transformers
- MLeap transformers correspond one-to-one with Spark transformers
- No dependencies on Spark
- Can implement custom transformers and serialization with a few classes worth of code

MLeap TransformBuilder

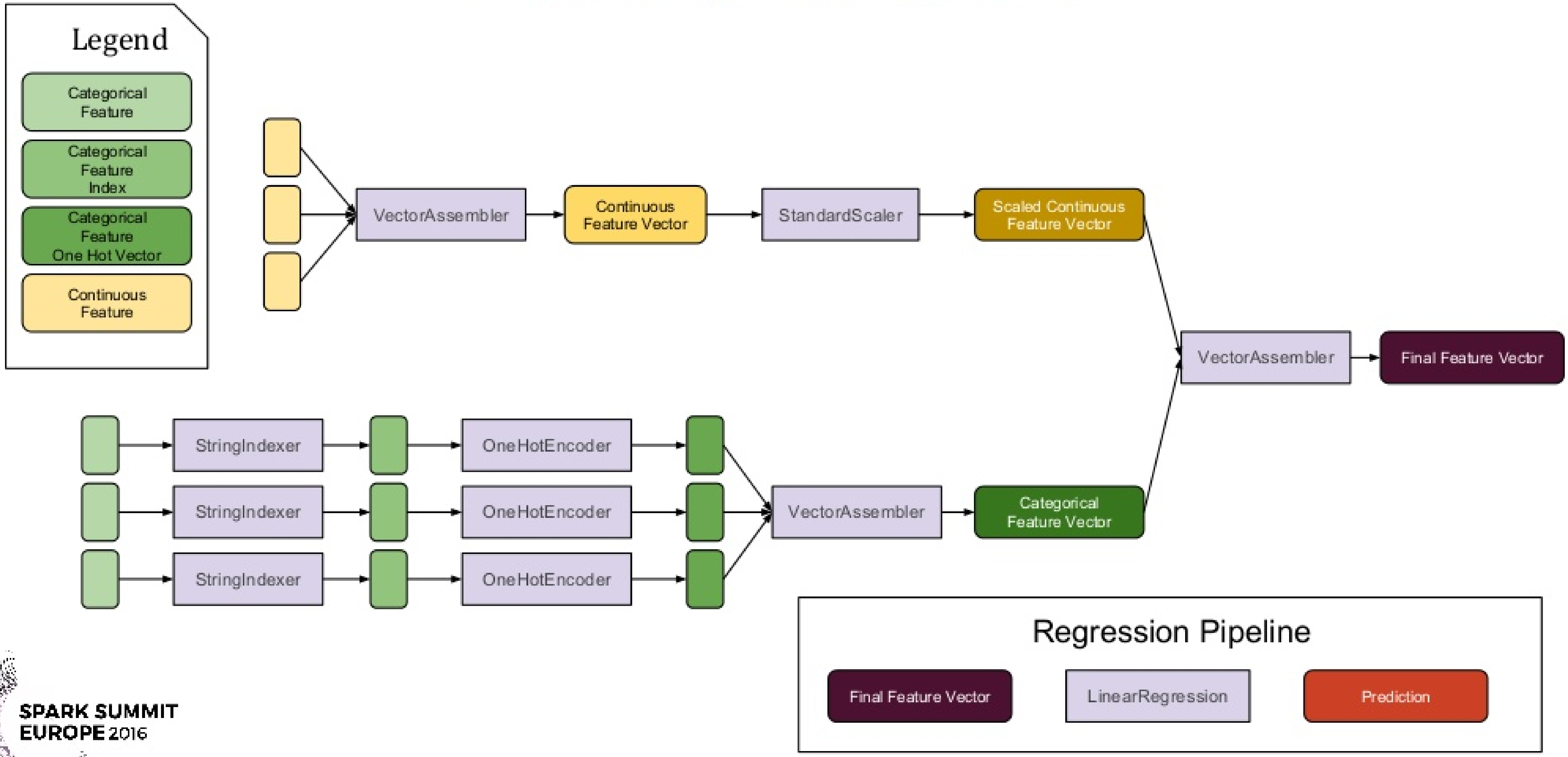
A **TransformBuilder** is used by **Mleap Transformer**s to transform an arbitrary context. The context can be:

1. **A LeapFrame**, this will immediately transform the LeapFrame using a transformer pipeline

2. **A Spark DataFrame**, this will convert Mleap UDFs from Mleap Transformers to Spark UDFs used to transform the Spark DataFrame

3. **A TransformCompiler**, this is a planned feature to allow compilation of your pipeline for ultra-fast execution on our model servers and **C libraries**

Demo Pipeline Upclose



SPARK SUMMIT EUROPE 2016

Serialize to Bundle.ML

The goal of MLeap and Bundle.ML is to let you serialize and deserialize your entire pipeline and not just the algorithm portion.

- Provides common serialization for both Spark and MLeap transformers
- 100% protobuf/JSON based for easy reading, compact data, and portability
- Store as a zip file for easy transport
- Scikit-Learn Support (In Development): Scikit and Spark share a common set of transformers and models already, and are both focused on transformer-based pipelines. The goal is to build a common serialization format between the two.

MLeap Spark

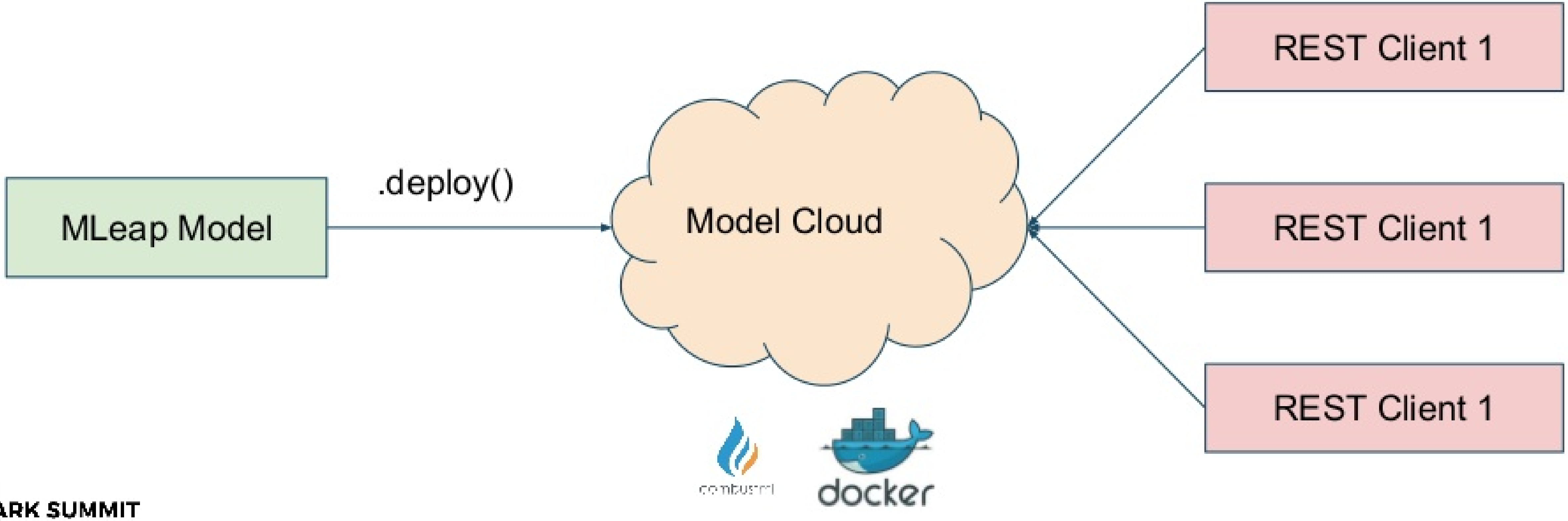
MLeap-Spark provides serialization of spark ml piplines to/from Bundle.ML

- Provides several extensions and modifications to Spark transformers
  - SVM - Support Vector Machine estimator/model (uses MLlib)
  - OneHotEncoder - Custom implementation to get around Spark's reliance on metadata
  - OneVsRest - Custom implementation to allow output of probabilities

- Allows execution of Mleap Transformers on Spark DataFrames

Combust Model Cloud

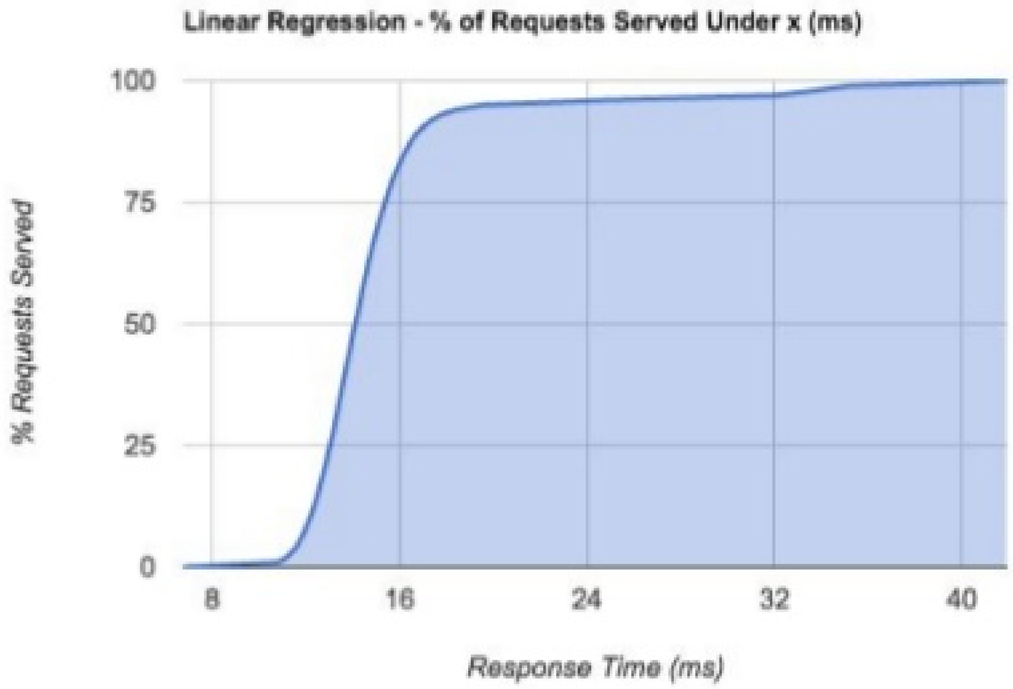
Provides **RESTful** endpoints to your **MLeap models**. Highly-optimized for throughput. **14ms** average response time for the example pipeline, can be optimized even further for serialization of LeapFrames across the network.



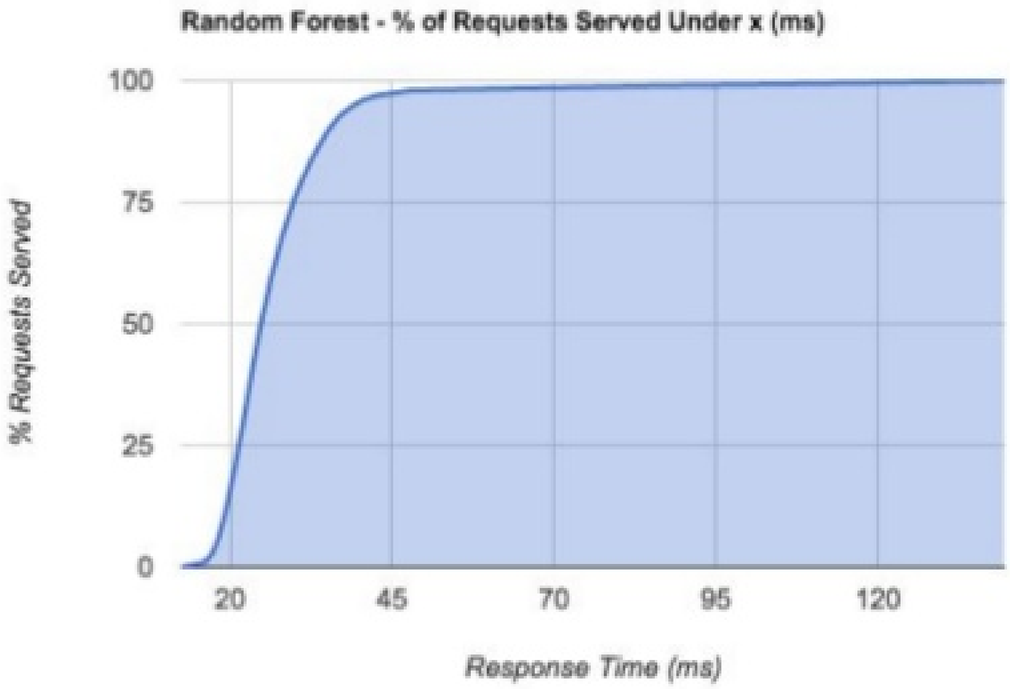
Benchmarks: Combust.ml Model Server

MacBook Pro, Uncompiled Models, JSON Serialization, Airbnb Models

Linear Regression (14ms)



Random Forest Regression (24ms)



combuts.ml Overview

combust.ml is built on (soon-to-be) **open-sourced** API servers, optimized for executing MLeap pipelines from Spark, Scikit-Learn transformers

1. Public hosting for trying combust services, deploy limited number of models, no support for custom transformers
2. Private hosting allows for automatic model scaling, custom transformers, non-public REST servers, compile models to C libraries, scaling with mesos on AWS or private data center
3. Training platform for non-technical audiences

Train **Store** **Deploy**

Future of MLeap

- Complete set of Spark/Scikit-Learn Transformers
- Unify core model libraries with Spark
- Python interface for PySpark users (export Spark pipelines)

THANK YOU.

Mikhail Semeniuk
email: mikhail@combust.ml
github: https://github.com/combust-ml/mleap
twitter: https://twitter.com/MikhailSemeniuk
linkedin: https://www.linkedin.com/in/semeniuk

Hollin Wilkins
email: hollin@combust.ml
github: https://github.com/combust-ml/mleap
twitter: https://twitter.com/HollinWilkins
linkedin: https://www.linkedin.com/in/hollinwilkins