MLlib: Main Guide (ml-guide.html) »

- Basic statistics (ml-statistics.html)
- Data sources (ml-datasource.html)
- Pipelines (ml-pipeline.html)
- Extracting, transforming and selecting features (ml-features.html)
- Classification and Regression (ml-classification-regression.html)
- Clustering (ml-clustering.html)
- Collaborative filtering (ml-collaborative-filtering.html)
- Frequent Pattern Mining (ml-frequent-pattern-mining.html)
- Model selection and tuning (ml-tuning.html)
- Advanced topics (ml-advanced.html)

MLlib: RDD-based API Guide (mllib-guide.html)

- Data types (mllib-data-types.html)
- Basic statistics (mllib-statistics.html)
- Classification and regression (mllib-classification-regression.html)

Machine Learning Library (MLlib) Guide

MLlib is Spark's machine learning (ML) library. Its goal is to make practical machine learning scalable and easy. At a high level, it provides tools such as:

- ML Algorithms: common learning algorithms such as classification, regression, clustering, and collaborative filtering
- Featurization: feature extraction, transformation, dimensionality reduction, and selection
- Pipelines: tools for constructing, evaluating, and tuning ML Pipelines
- Persistence: saving and load algorithms, models, and Pipelines
- Utilities: linear algebra, statistics, data handling, etc.

Announcement: DataFrame-based API is primary API

The MLlib RDD-based API is now in maintenance mode.

As of Spark 2.0, the RDD (rdd-programming-guide.html#resilient-distributed-datasets-rdds)-based APIs in the `spark.mllib` package have entered maintenance mode. The primary Machine Learning API for Spark is now the DataFrame (sql-programming-guide.html)-based API in the `spark.ml` package.

What are the implications?

- MLlib will still support the RDD-based API in `spark.mllib` with bug fixes.
- MLlib will not add new features to the RDD-based API.
- In the Spark 2.x releases, MLlib will add features to the DataFrames-based API to reach feature parity with the RDD-based API.

Why is MLlib switching to the DataFrame-based API?

- DataFrames provide a more user-friendly API than RDDs. The many benefits of DataFrames include Spark Datasources, SQL/DataFrame queries, Tungsten and Catalyst optimizations, and uniform APIs across languages.
- The DataFrame-based API for MLlib provides a uniform API across ML algorithms and across multiple languages.
- DataFrames facilitate practical ML Pipelines, particularly feature transformations. See the Pipelines guide (ml-pipeline.html) for details.

What is "Spark ML"?

- "Spark ML" is not an official name but occasionally used to refer to the MLlib DataFrame-based API. This is majorly due to the `org.apache.spark.ml` Scala package name used by the DataFrame-based API, and the "Spark ML Pipelines" term we used initially to emphasize the pipeline concept.

Is MLlib deprecated?

- No. MLlib includes both the RDD-based API and the DataFrame-based API. The RDD-based API is now in maintenance mode. But neither API is deprecated, nor MLlib as a whole.

Dependencies

MLlib uses linear algebra packages Breeze (http://www.scalanlp.org/) and dev.ludovic.netlib (https://github.com/luhenry/netlib) for optimised numerical processing[1 (#fn:1)]. Those packages may call native acceleration libraries such as Intel MKL (https://software.intel.com/content/www/us/en/develop/tools/math-kernel-library.html) or OpenBLAS (http://www.openblas.net) if they are available as system libraries or in runtime library paths.

However, native acceleration libraries can't be distributed with Spark. See MLlib Linear Algebra Acceleration Guide (ml-linalg-guide.html) for how to enable accelerated linear algebra processing. If accelerated native libraries are not enabled, you will see a warning message like below and a pure JVM implementation will be used instead:

```
WARNING: Failed to load implementation from:dev.ludovic.netlib.blas.JNIBLAS
```

To use MLlib in Python, you will need NumPy (http://www.numpy.org) version 1.4 or newer.

Highlights in 3.0

The list below highlights some of the new features and enhancements added to MLlib in the `3.0` release of Spark:

- Multiple columns support was added to `Binarizer` (SPARK-23578 (https://issues.apache.org/jira/browse/SPARK-23578)), `StringIndexer` (SPARK-11215 (https://issues.apache.org/jira/browse/SPARK-11215)), `StopWordsRemover` (SPARK-29808 (https://issues.apache.org/jira/browse/SPARK-29808)) and PySpark `QuantileDiscretizer` (SPARK-22796 (https://issues.apache.org/jira/browse/SPARK-22796)).
- Tree-Based Feature Transformation was added (SPARK-13677 (https://issues.apache.org/jira/browse/SPARK-13677)).
- Two new evaluators `MultilabelClassificationEvaluator` (SPARK-16692 (https://issues.apache.org/jira/browse/SPARK-16692)) and `RankingEvaluator` (SPARK-28045 (https://issues.apache.org/jira/browse/SPARK-28045)) were added.
- Sample weights support was added in `DecisionTreeClassifier/Regressor` (SPARK-19591 (https://issues.apache.org/jira/browse/SPARK-19591)), `RandomForestClassifier/Regressor` (SPARK-9478 (https://issues.apache.org/jira/browse/SPARK-9478)), `GBTClassifier/Regressor` (SPARK-9612 (https://issues.apache.org/jira/browse/SPARK-9612)), `MulticlassClassificationEvaluator` (SPARK-24101 (https://issues.apache.org/jira/browse/SPARK-24101)), `RegressionEvaluator` (SPARK-24102 (https://issues.apache.org/jira/browse/SPARK-24102)), `BinaryClassificationEvaluator` (SPARK-24103 (https://issues.apache.org/jira/browse/SPARK-24103)), `BisectingKMeans` (SPARK-30351 (https://issues.apache.org/jira/browse/SPARK-30351)), `KMeans` (SPARK-29967 (https://issues.apache.org/jira/browse/SPARK-29967)) and `GaussianMixture` (SPARK-30102 (https://issues.apache.org/jira/browse/SPARK-30102)).
- R API for `PowerIterationClustering` was added (SPARK-19827 (https://issues.apache.org/jira/browse/SPARK-19827)).
- Added Spark ML listener for tracking ML pipeline status (SPARK-23674 (https://issues.apache.org/jira/browse/SPARK-23674)).
- Fit with validation set was added to Gradient Boosted Trees in Python (SPARK-24333 (https://issues.apache.org/jira/browse/SPARK-24333)).
- `RobustScaler` (ml-features.html#robustscaler) transformer was added (SPARK-28399 (https://issues.apache.org/jira/browse/SPARK-28399)).
- `Factorization Machines` (ml-classification-regression.html#factorization-machines) classifier and regressor were added (SPARK-29224 (https://issues.apache.org/jira/browse/SPARK-29224)).
- Gaussian Naive Bayes Classifier (SPARK-16872 (https://issues.apache.org/jira/browse/SPARK-16872)) and Complement Naive Bayes Classifier (SPARK-29942 (https://issues.apache.org/jira/browse/SPARK-29942)) were added.
- ML function parity between Scala and Python (SPARK-28958 (https://issues.apache.org/jira/browse/SPARK-28958)).
- `predictRaw` is made public in all the Classification models. `predictProbability` is made public in all the Classification models except `LinearSVCModel` (SPARK-30358 (https://issues.apache.org/jira/browse/SPARK-30358)).

Migration Guide

The migration guide is now archived on this page (ml-migration-guide.html).

[1] To learn more about the benefits and background of system optimised natives, you may wish to watch Sam Halliday's ScalaX talk on High Performance Linear Algebra in Scala (http://fommil.github.io/scalax14/#/). ↩ (#fnref:1)