



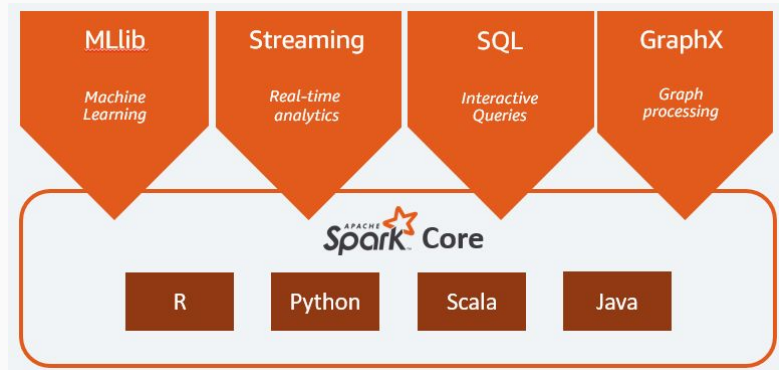
# Apache Spark MLlib Tool Talk

By: Sadiq Salewala, Grayson Sanders, Chris Gauthier

# Introduction

Apache Spark is one of the on-demand big data tools which is being used by many companies around the world. Its ability to do In-Memory computation and Parallel-Processing are the main reasons for the popularity of this tool.

MLlib is a package from Apache Spark.



# Overview

## Ease of Use

Usable in Java, Scala, Python, and R

## Performance


High-quality algorithms, 100x faster than MapReduce

## Runs Everywhere

Spark runs on Hadoop, Apache Mesos, Kubernetes, standalone, or in the cloud, against diverse data sources



```
pip install pyspark
```



APACHE  
**Spark**<sup>™</sup>

Lightning-fast unified analytics engine

Download

Libraries ▾

Documentation ▾

Examples

Community ▾

Developers ▾

Apache Software Foundation ▾

## Download Apache Spark<sup>™</sup>

- Choose a Spark release: 3.1.1 (Mar 02, 2021) ▾
- Choose a package type: Pre-built for Apache Hadoop 2.7 ▾
- Download Spark: [spark-3.1.1-bin-hadoop2.7.tgz](#)
- Verify this release using the 3.1.1 [signatures](#), [checksums](#) and [project release KEYS](#).

Note that, Spark 2.x is pre-built with Scala 2.11 except version 2.4.2, which is pre-built with Scala 2.12. Spark 3.0+ is pre-built with Scala 2.12.

### Latest Preview Release

Preview releases, as the name suggests, are releases for previewing upcoming features. Unlike nightly packages, preview releases have been audited by the project's management committee to satisfy the legal requirements of Apache Software Foundation's release policy. Preview releases are not meant to be functional, i.e. they can and highly likely will contain critical bugs or documentation errors. The latest preview release is Spark 3.0.0-preview2, published on Dec 23, 2019.

#### Latest News


Spark 3.1.1 released (Mar 02, 2021)

Spark 3.0.2 released (Feb 19, 2021)

Next official release: Spark 3.1.1 (Jan 07, 2021)

Spark 2.4.7 released (Sep 12, 2020)

[Archive](#)



APACHECON  
September 21-23  
[www.apachecon.com](http://www.apachecon.com) 2021

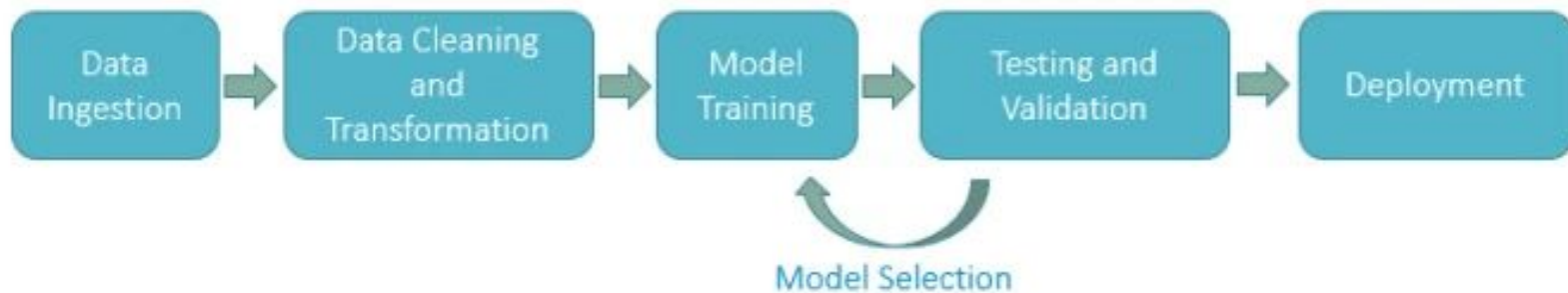
Download Spark

```
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to

  ____      __
 / _ )__  / /_
/_  \__/_/ __/
/___/___/_/

 version 2.4.5

Using Python version 3.6.9 (default, Nov  7 2019 10:44:02)
SparkSession available as 'spark'.
>>>
```



# Applications

MLlib contains many algorithms and utilities.

ML algorithms include:

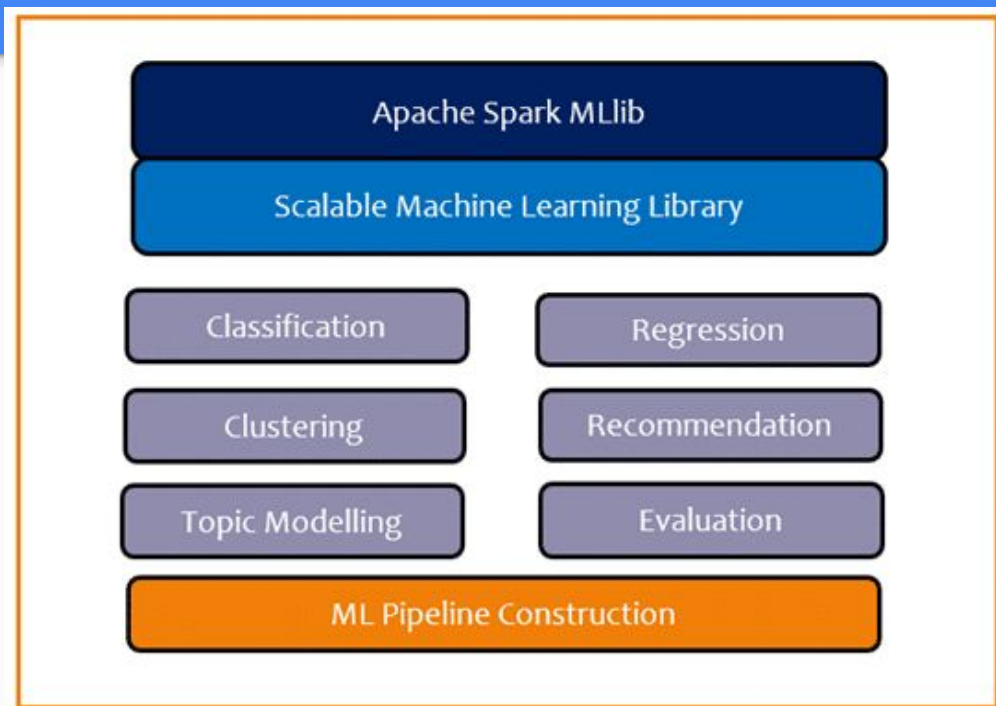
- Classification: logistic regression, naive Bayes,...
- Regression: generalized linear regression, survival regression,...
- Decision trees, random forests, and gradient-boosted trees
- Recommendation: alternating least squares (ALS)
- Clustering: K-means, Gaussian mixtures (GMMs),...
- Topic modeling: latent Dirichlet allocation (LDA)
- Frequent itemsets, association rules, and sequential pattern mining

ML workflow utilities include:

- Feature transformations: standardization, normalization, hashing,...
- ML Pipeline construction
- Model evaluation and hyper-parameter tuning
- ML persistence: saving and loading models and Pipelines

Other utilities include:

- Distributed linear algebra: SVD, PCA,...
- Statistics: summary statistics, hypothesis testing



# Linear Regression Example

Linear regression is another classical supervised learning setting. In this problem, each entity is associated with a real-valued label (as it would be in binary classification).

We want to predict labels as closely as possible given numerical features representing entities. MLlib supports linear regression as well as L1 (lasso) and L2 (ridge) regularized variants. The regression algorithms in MLlib also leverage the underlying gradient descent primitive (described below), and have the same parameters as the binary classification algorithms..

Available algorithms for linear regression:

- `LinearRegressionWithSGD`
- `RidgeRegressionWithSGD`
- `LassoWithSGD`

# Recommendation Example

Dataset: Rating data from the MovieLens web site

Goal: Recommend movies to users using collaborative filtering (done through Alternating Least Squares) based on a user's movie watching history



# Recommendation Example

	item1	item2	item3	item4
user1	2	5	1	3
user2	4	?	?	1
user3	?	4	2	?
user4	2	4	3	1
user5	1	3	2	?

# Clustering Example

MLlib supports a variety of different clustering algorithms (K-means, Latent Dirichlet allocation, etc.), which seek to group data based on the similar qualities of the observations.

K-means is a clustering algorithm that has the goal of partitioning data ( $n$  observations) into  $k$  clusters, in which the clusters are centered around a mean (or centroid). These clusters can then be used in a variety of applications.

# Sources

- <https://spark.apache.org/mllib/>
- <https://spark.apache.org/docs/0.9.0/mllib-guide.html>
- <http://spark.apache.org/docs/latest/ml-clustering.html>
- <https://towardsdatascience.com/build-recommendation-system-with-pyspark-using-alternating-least-squares-als-matrix-factorisation-eb1ad2e7679>
- <https://towardsdatascience.com/machine-learning-linear-regression-using-pyspark-9d5d5c772b42>

Thank you! 🥰