Digital Technologies Tech Talks

Apache Spark

An Introduction

April 13, 2022 - René Richard

IN THIS TALK …

01

INTRODUCTION

- Is this for me?
- GitHub project link
- Motivation for using Spark

02

SPARK OVERVIEW

- What is Apache Spark?
- Deployment modes & Spark SQL

03

SIMULATED CLUSTER

- Standalone cluster configuration
- Hands-on content

04

SPARK-SUBMIT

- HelloWorld batch example
- Hands-on content

05

DATA TRANSFORMATIONS

- Motivation
- Hands-on content

06

SPARK MLLIB

- Available algorithms
- Hands-on content

07

WRAP-UP

Source : https://bit.ly/3i8C4sQ

01 - INTRODUCTION

Is this for me?

- Some exposure to :
  - The command-line (on Unix-like OS)
  - Docker and docker-compose
  - **Python** or programming in general
  - Structure Query Language (SQL)
  - There will be a nod to Java and Scala jobs
  - **Interest** and/or **curiosity** (trumps all of the above)

Source : https://bit.ly/3Jgo10c

01 - INTRODUCTION

GitHub project link

- Includes :
  - Slides
  - Simulated cluster configurations
  - Source code
  - Data
  - Book

Source : https://bit.ly/3q6VsdW

01 - INTRODUCTION

Motivation for using Spark

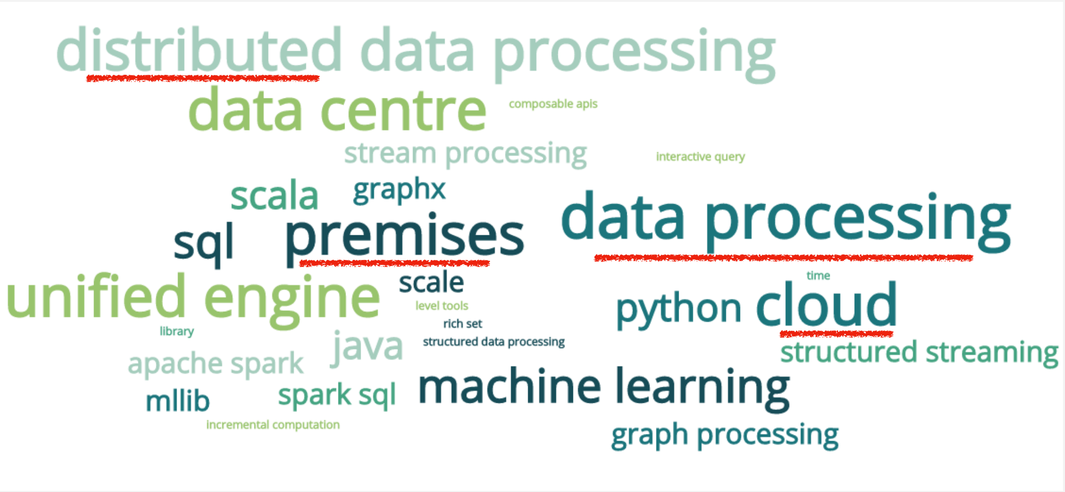
01 - INTRODUCTION

Motivation for using Spark

- Using Java, Scala, Python or R
- Need to process large amounts of data
- Interactive data exploration for larger data sets
- Seeking a developer-friendly API
- Want to leverage existing HDFS data store (although HDFS is not required for Spark)
  - Perform computations over data where it resides
- To leverage collaborative environments (e.g. Databricks)

02 - SPARK OVERVIEW

What is Apache Spark ?



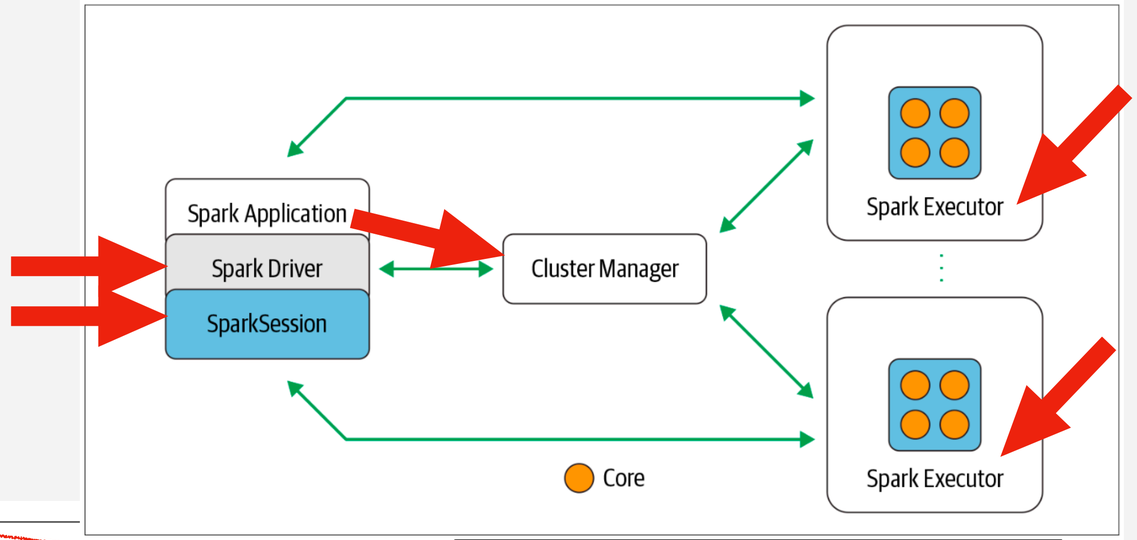


Figure from : Jules S. Damji, Brooke Wenig, Tathagata Das, and Denny Lee. Learning Spark: lightning-fast big data analytics. O'Reilly & Associates, 2020.

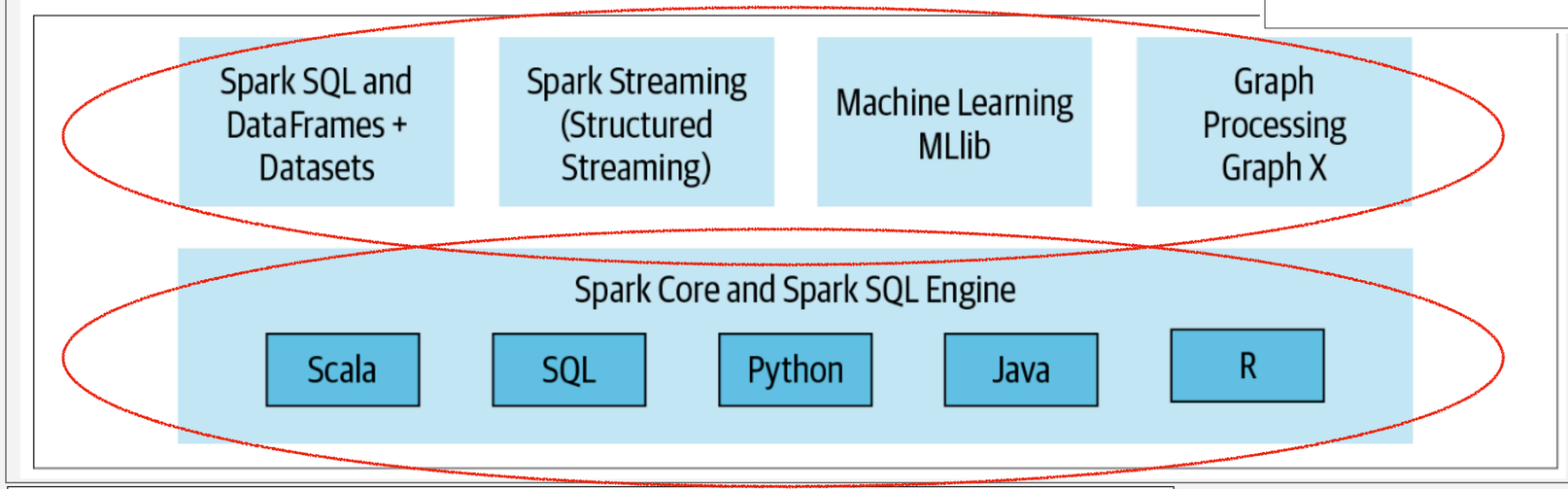


Figure from : Jules S. Damji, Brooke Wenig, Tathagata Das, and Denny Lee. Learning Spark: lightning-fast big data analytics. O'Reilly & Associates, 2020.

02 - SPARK OVERVIEW

Deployment Modes

Mode	Spark driver	Spark executor	Cluster manager
★ Local	Runs on a single JVM, like a laptop or single node	Runs on the same JVM as the driver	Runs on the same host
★ Standalone	Can run on any node in the cluster	Each node in the cluster will launch its own executor JVM	Can be allocated arbitrarily to any host in the cluster
YARN (client)	Runs on a client, not part of the cluster	YARN's NodeManager's container	YARN's Resource Manager works with YARN's Application Master to allocate the containers on NodeManagers for executors
YARN (cluster)	Runs with the YARN Application Master	Same as YARN client mode	Same as YARN client mode
Kubernetes	Runs in a Kubernetes pod	Each worker runs within its own pod	Kubernetes Master

Figure from : Jules S. Damji, Brooke Wenig, Tathagata Das, and Denny Lee. Learning Spark: lightning-fast big data analytics. O'Reilly & Associates, 2020.

02 - SPARK OVERVIEW

Local Deployment Mode

- Easiest way to try out Apache Spark
- All processing is done on single machine
- Still benefit from parallelized processing across all the cores on single machine, but not across several servers

```
from pyspark.sql import SparkSession

spark = SparkSession\
            .builder\
            .master("local[20]")\
            .appName("pyspark_local")\
            .getOrCreate()
```

Local Spark thread-based cluster using 20 cores

02 - SPARK OVERVIEW

Popular Spark use cases

- Parallelize computations, hide complexity, enjoy distribution and fault tolerance
- Perform **ad-hoc queries** to explore and visualize data sets
- Combine data from multiple sources
- Prepare data for downstream **ML modelling**
- Training and evaluating ML models (MLlib)
- End-to-end pipelines from **streaming** data (Structured Streaming)

02 - SPARK OVERVIEW

Spark SQL

- DataFrame and dataset abstractions
- Data read/write (JSON, CSV, Avro, Parquet, ORC, etc.)
- Bridge to standard tools via JDBC/ODBC connectors
- Support for ANSI SQL:2003-compliant commands
- Generates optimized query plans for final execution

Transformations	Actions
orderBy()	show()
groupBy()	take()
filter()	count()
select()	collect()
join()	save()

02 - SPARK OVERVIEW

GPU-Accelerated ETL - (new in Spark 3.x)

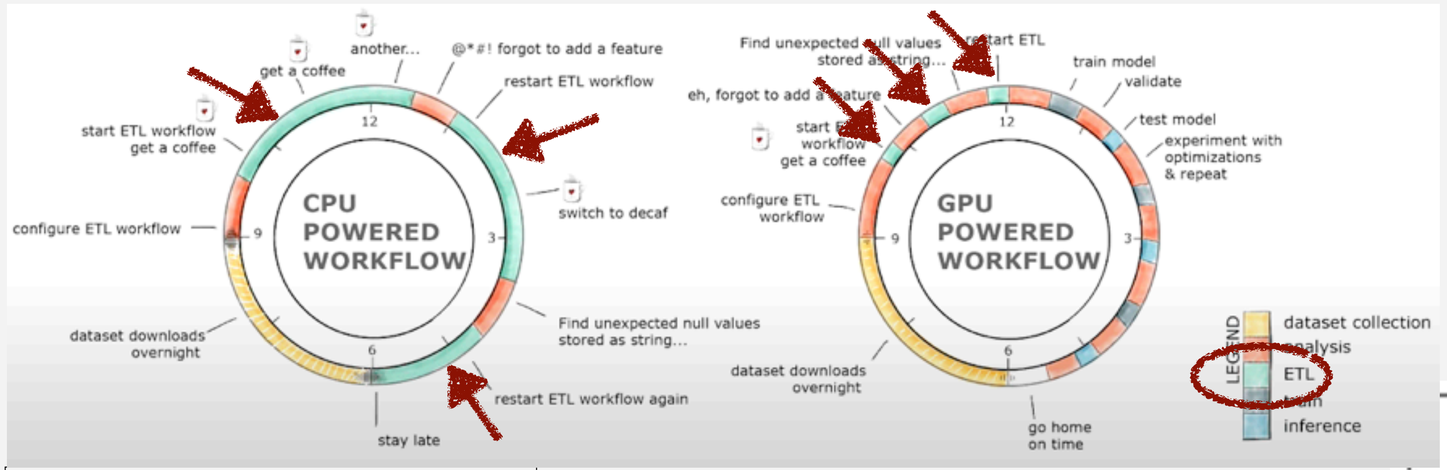


Figure from : https://bit.ly/3wmUTkv

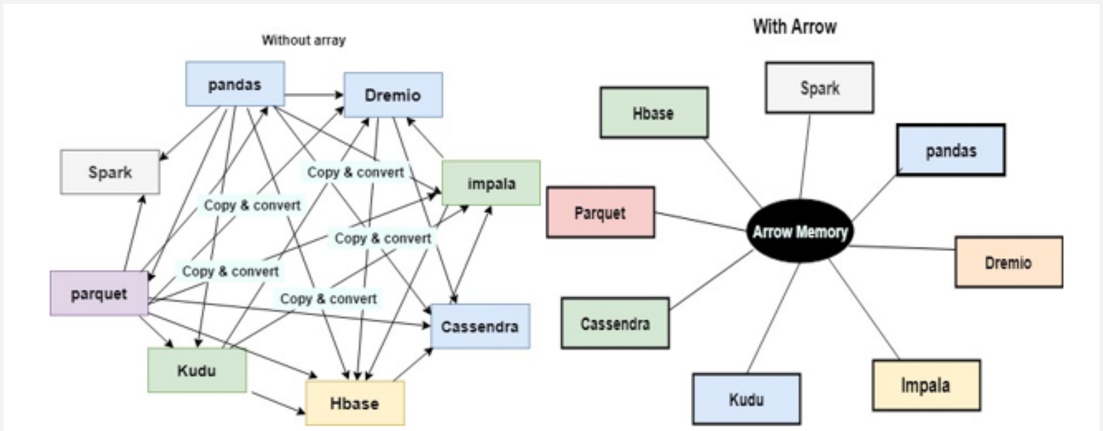


Figure from : https://www.tutorialandexample.com/apache-arrow-tutorial

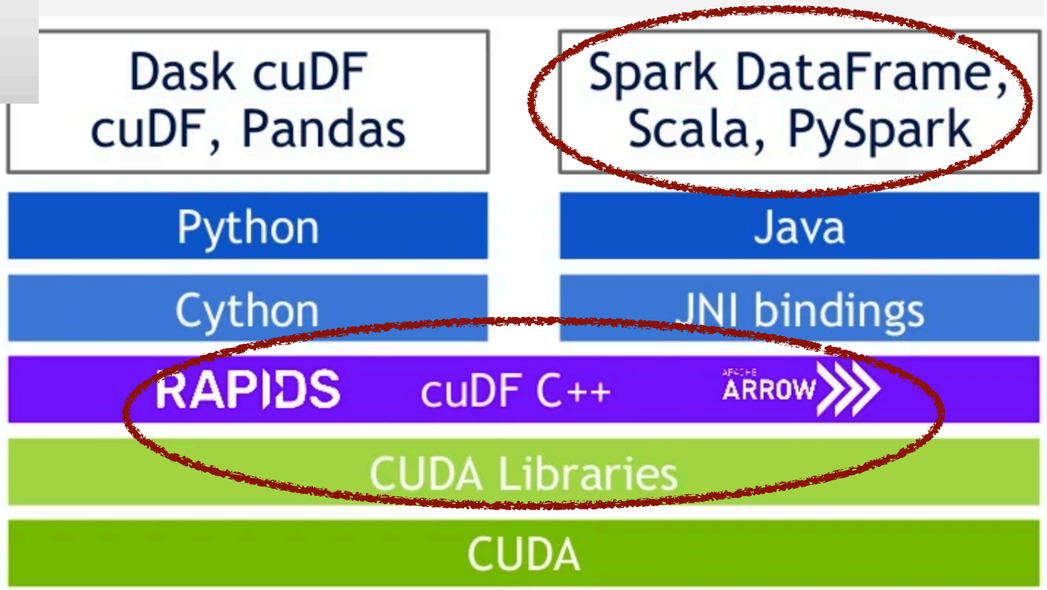


Figure from : https://bit.ly/3IfIXmx

03 - SIMULATED CLUSTER

Standalone cluster configuration

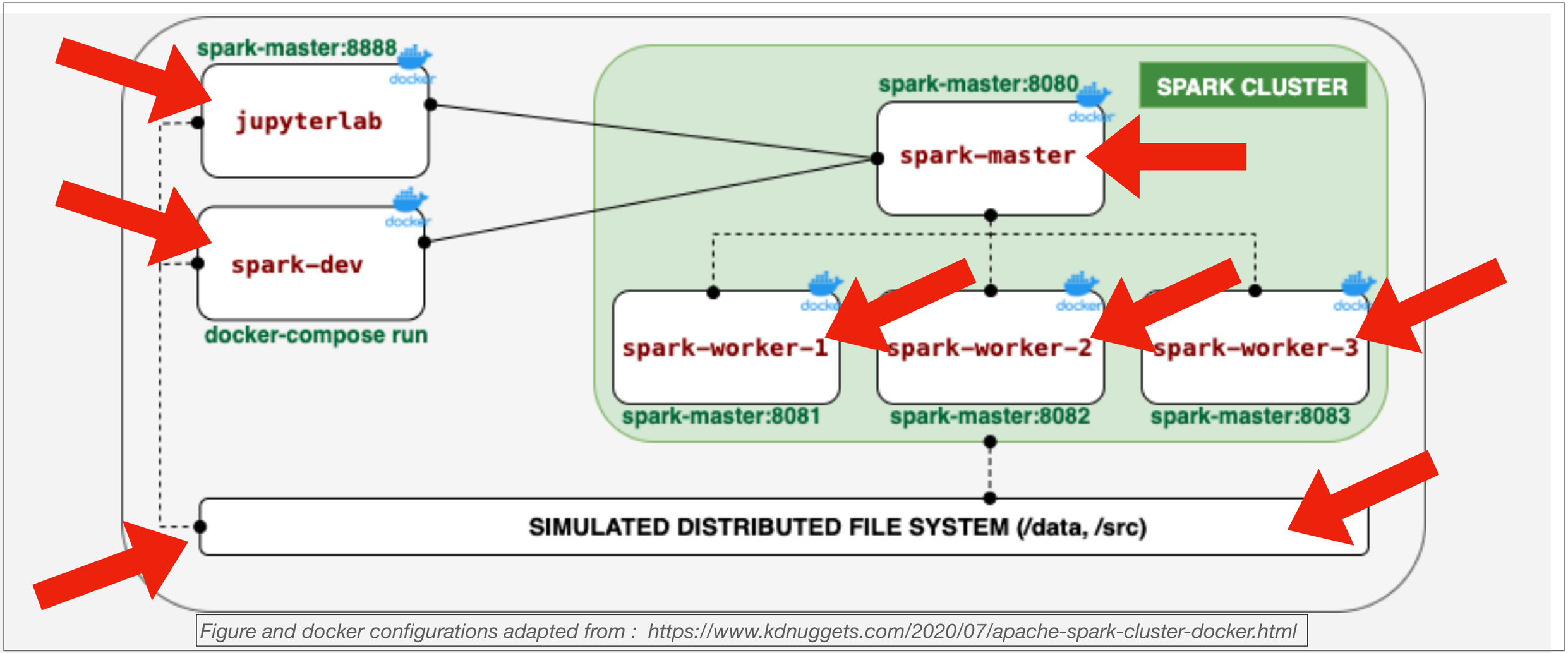


Figure and docker configurations adapted from : https://www.kdnuggets.com/2020/07/apache-spark-cluster-docker.html

Start simulated cluster

Hands-on content

Explore simulated cluster

Hands-on content

04 - SPARK-SUBMIT

HelloWorld batch job examples

- Code API similarity across languages
- Spark-submit utility
  - Client and cluster deploy-modes
  - Resource requests
- A Scala Example

Submit HelloWorld.scala

Hands-on content

05 - DATA TRANSFORMATIONS

Motivation

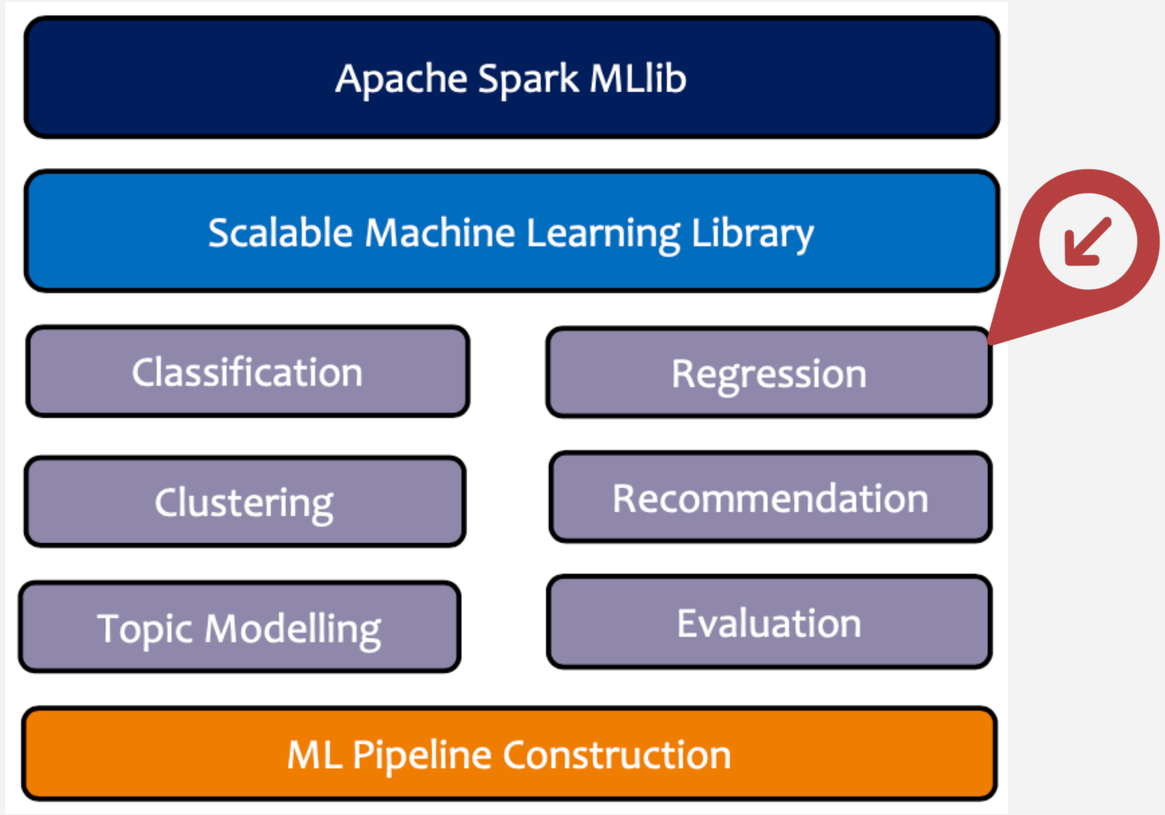
- Using Jupyter notebooks
- Exploratory analytics on structured data can :
  - Inform the feature extraction approach
  - Influence the design of models
  - Lead to valuable insights even before we train a model
- Transform data where structure is preserved
- ML models still make predictions while reducing computation costs

Car data preparation

Hands-on content

06 - SPARK MLLIB

Available algorithms



Source : https://intellipaat.com/mediaFiles/2019/02/MLib-cheat-sheet-design.pdf

HOW MUCH ?

Source : https://bit.ly/3I9OIIP

06 - SPARK MLLIB

Random forest I/O columns

Input Columns

Param name	Type(s)	Default	Description
labelCol	Double	"label"	Label to predict
featuresCol	Vector	"features"	Feature vector

Output Columns (Predictions)

Param name	Type(s)	Default	Description	Notes
predictionCol	Double	"prediction"	Predicted label	
rawPredictionCol	Vector	"rawPrediction"	Vector of length # classes, with the counts of training instance labels at the tree node which makes the prediction	Classification only
probabilityCol	Vector	"probability"	Vector of length # classes equal to rawPrediction normalized to a multinomial distribution	Classification only

Source : https://spark.apache.org/docs/latest/ml-classification-regression.html#random-forests

featuresCol **predictionCol** **labelCol**

```
+-------------------------------------+------------------+-------------+
|algorithmic_input                    |prediction        |Selling_Price|
+-------------------------------------+------------------+-------------+
|[1.6,1200.0,0.0,5.0,1.0,0.0,0.0,1.0] |1.5092715887040848|1.45         |
|[0.75,26000.0,1.0,14.0,1.0,0.0,0.0,1.0]|0.2966107827960572|0.25         |
|[0.99,14500.0,0.0,10.0,1.0,0.0,0.0,1.0]|0.5599952838539594|0.45         |
|[3.46,45280.0,0.0,8.0,1.0,0.0,1.0,1.0]|2.712267776467363 |2.5          |
|[3.95,25000.0,0.0,6.0,1.0,0.0,1.0,1.0]|3.3483788668185857|2.85         |
+-------------------------------------+------------------+-------------+
```

Car data modelling

07 - WRAP UP

Thank you and some resources

- GitHub Link for this workshop:
  - https://github.com/redsofa/dt_seminar_spark_intro
- Spark on HPC infrastructure:
  - https://bit.ly/3Ng9LXB
- Databricks free trial:
  - https://bit.ly/3qsDenn
- Pandas API on Spark :
  - https://bit.ly/3tFACox