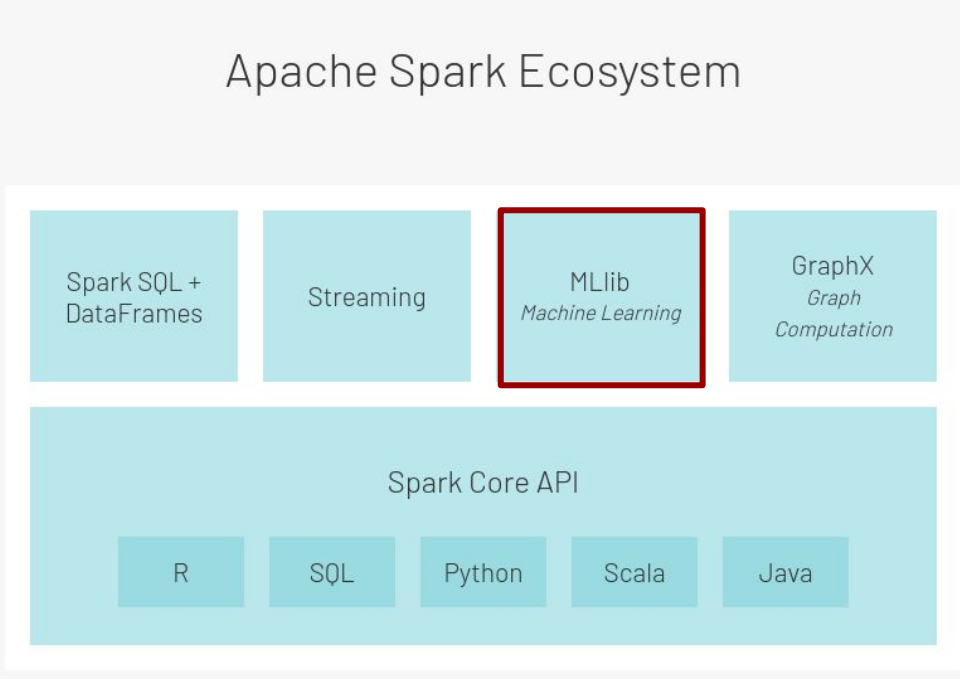
ADALTAS

Data Analysis with

5. Machine Learning With Spark

Spark Ecosystem

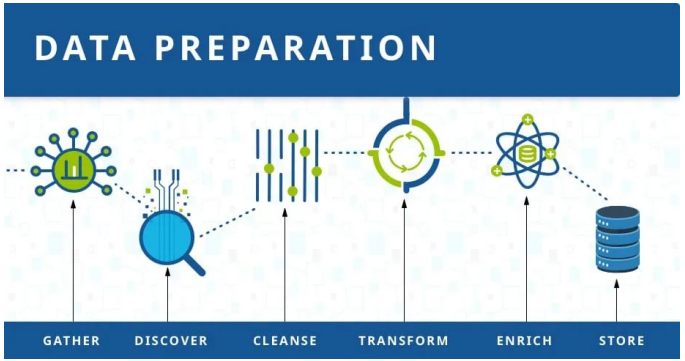


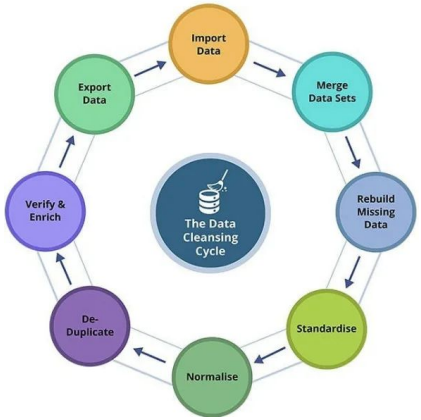
Machine Learning With Spark: packages

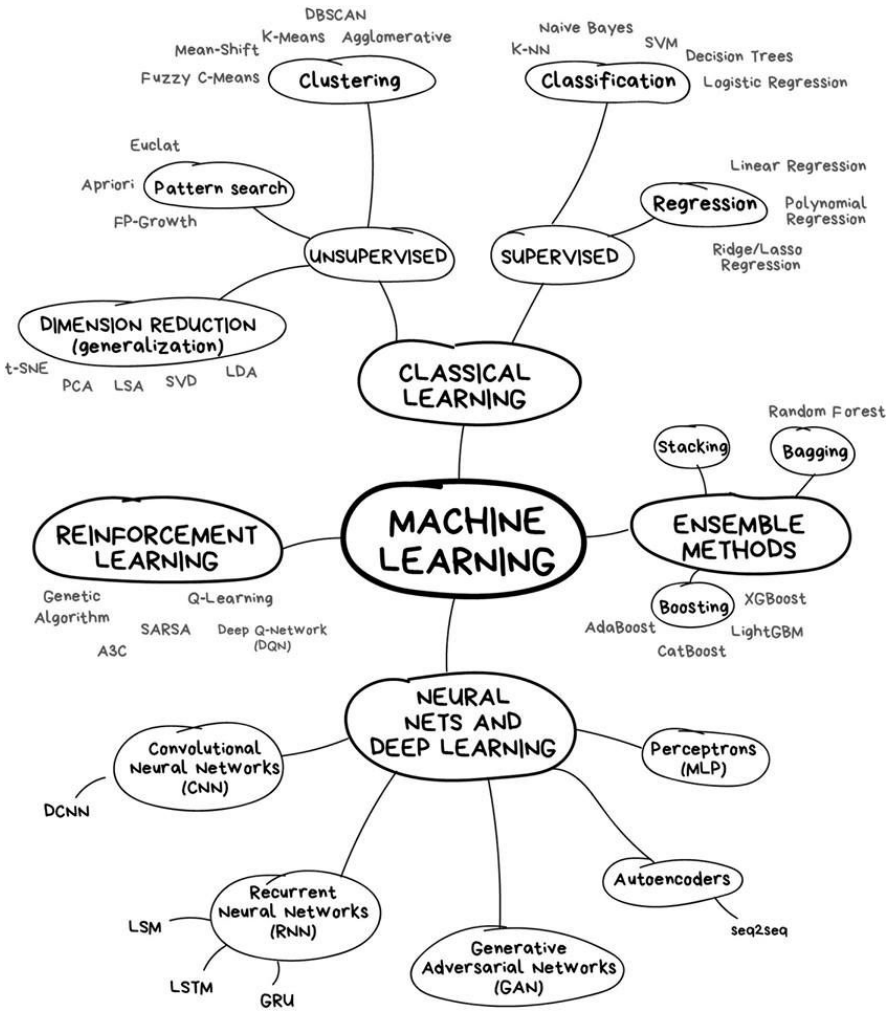
- spark.mllib:
  - RDD-based (DF functionalities being added)
  - In maintenance since Spark 2.0 (now 3.x)
- spark.ml
  - Newer
  - DataFrames-based

The term **MLlib** is used for both.

Machine Learning







MLlib: Functionalities

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,...

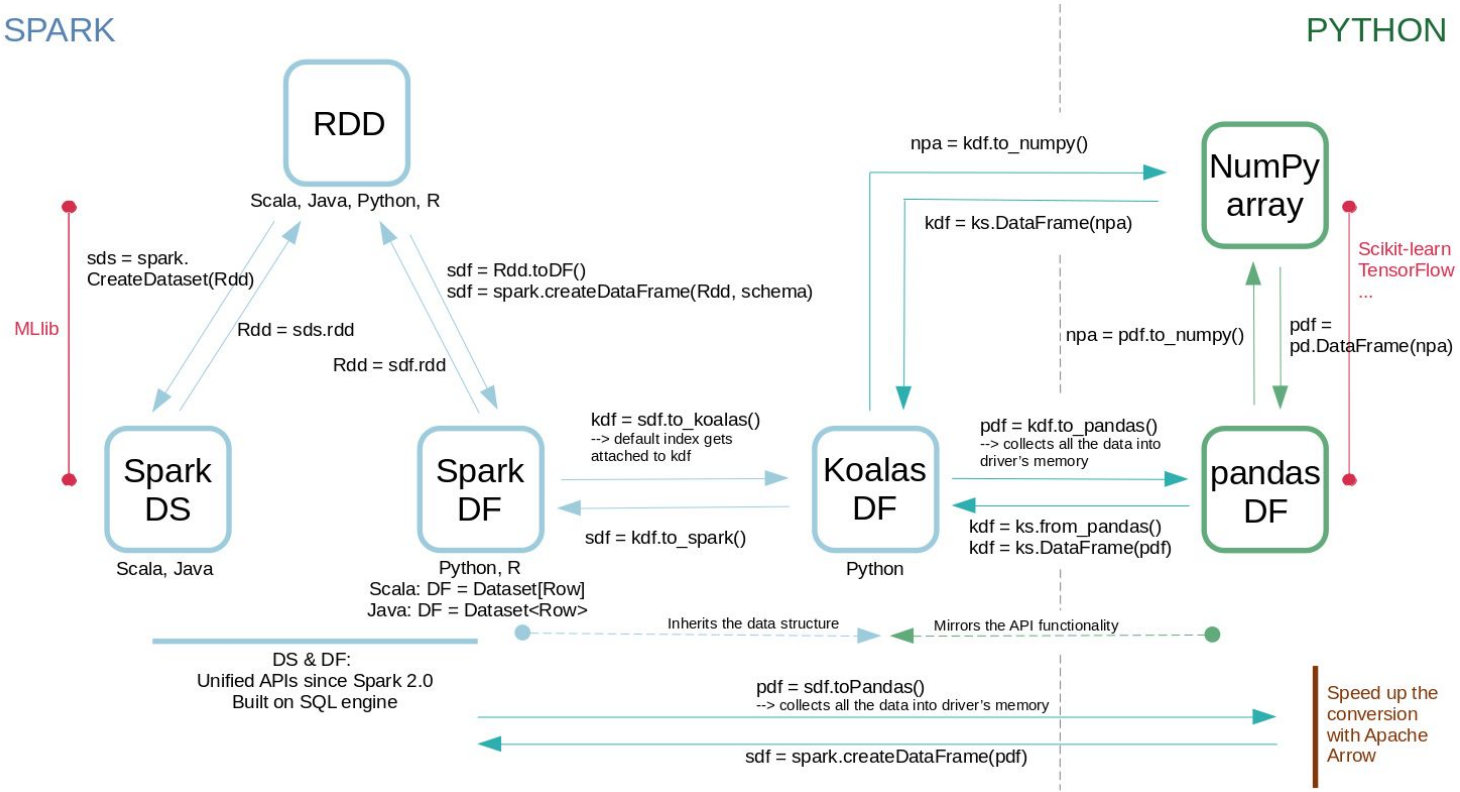
https://spark.apache.org/mllib/
https://spark.apache.org/docs/latest/ml-guide.html

When do we choose ML with Spark?

- Pros:
  - We can process big datasets
  - One framework from data preparation to modeling

- Cons:
  - The algorithms need to be re-written for distributed computing
  - No visualization natively present in Spark
  - Different functionalities with different APIs (e.g. XGBoost only in Scala)
  - Not necessarily faster

Spark Data Structures and connection to Python



Types of Parallelism

Could be set up:

- cv**.setParallelism(n)**.fit()
- Spark-scikit ...

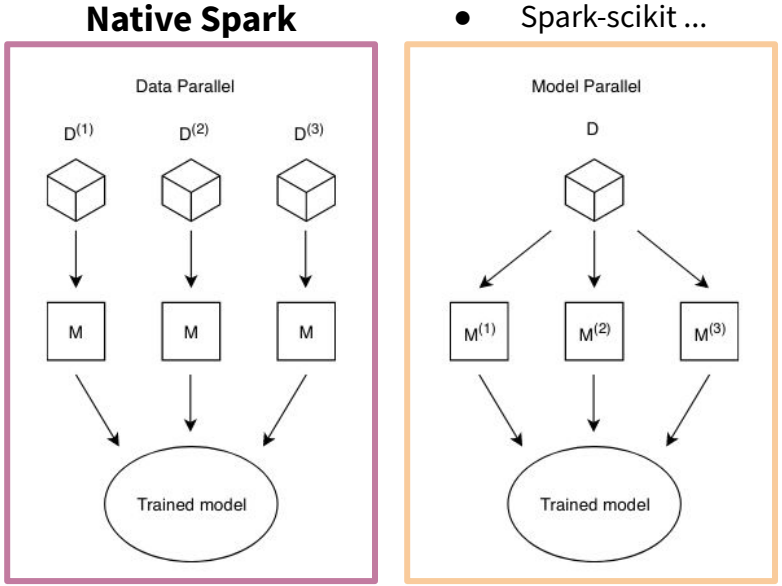


Fig. 2. Parallelism in Distributed Machine Learning. Data parallelism trains multiple instances of the same model on different subsets of the training dataset, while model parallelism distributes parallel paths of a single model to multiple nodes.

https://arxiv.org/pdf/1912.09789.pdf

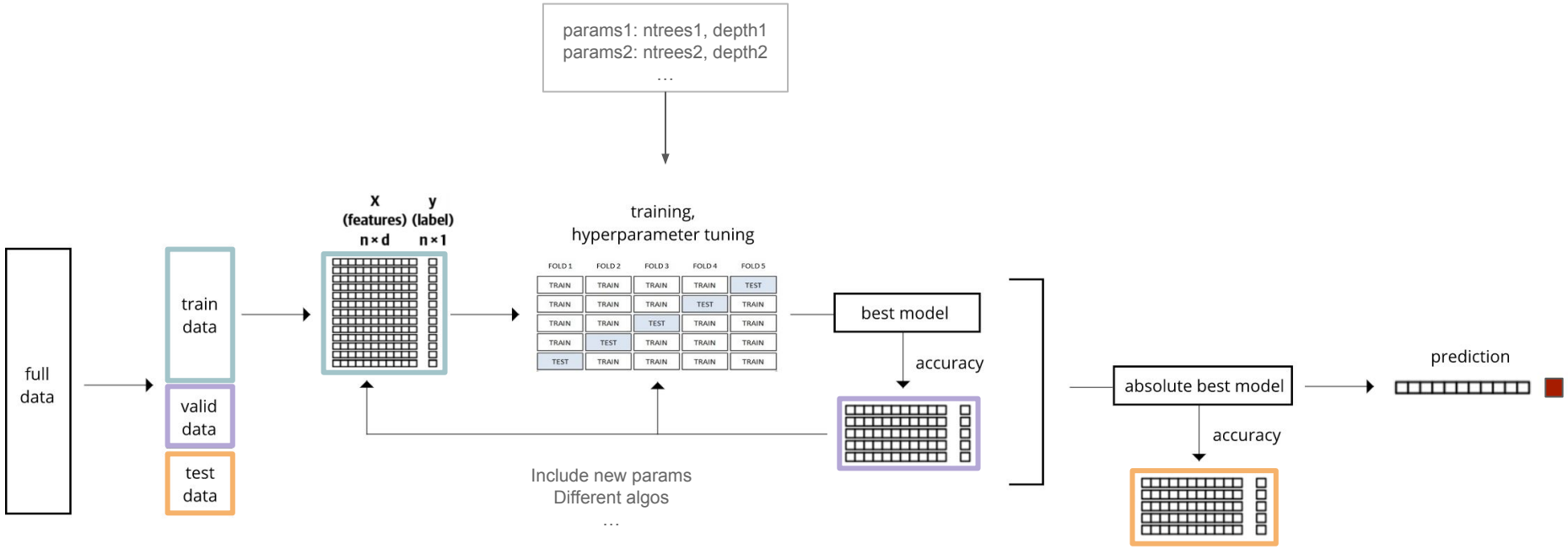
ML Terminology (Spark, sklearn…)

- Transformer
  - Input: DF → output: DF via **.transform()**
  - Apply rule-based transformation, **no learning**
  - Example: One-hot encoding, model
- Estimator
  - **Learns** parameters via **.fit()** method
  - Returns a model
- Pipeline
  - Putting a series of transformers and estimators in a sequence
  - **Automation**

Steps of a Machine Learning project (1/2)

- Dataset preparation (cleaning, feature engineering)
- Train/test (train/validation/test) split: 80/20 (60/20/20)
- Train the model (on the **train** data)
  - Cross-validation, leave-one-out
- Estimate the quality of the model (on the **validation** data)
- Tune the model to improve the performance
  - Grid search
- Test the final performance on the **test** data

Steps of a Machine Learning project (2/2)



Linear Regression and Random Forest regression