

# Pipelines and MLFlow

Aula06

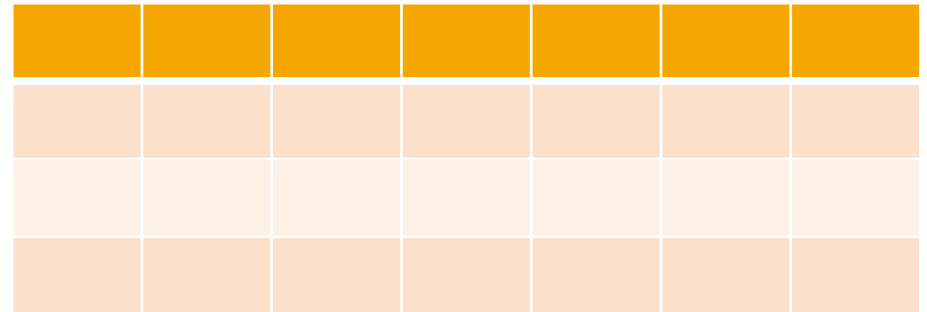


# Pipelines or Workflow Definition

- MLlib standardizes APIs for machine learning algorithms to make it easier to combine multiple algorithms into a single pipeline, or workflow.

# DataFrame

- Machine learning can be applied to a wide variety of data types, such as vectors, text, images, and structured data. This API adopts the DataFrame from Spark SQL in order to support a variety of data types.




# Transformer

- A Transformer is an abstraction that includes feature transformers and learned models. Technically, a Transformer implements a method `transform()`, which converts one DataFrame into another, generally by appending one or more columns.



# Estimator (ML Learning Algorithm)

- An Estimator abstracts the concept of a learning algorithm or any algorithm that fits or trains on data.
- Technically, an Estimator implements a method `fit()`, which accepts a `DataFrame` and produces a `Model`, which is a `Transformer`.



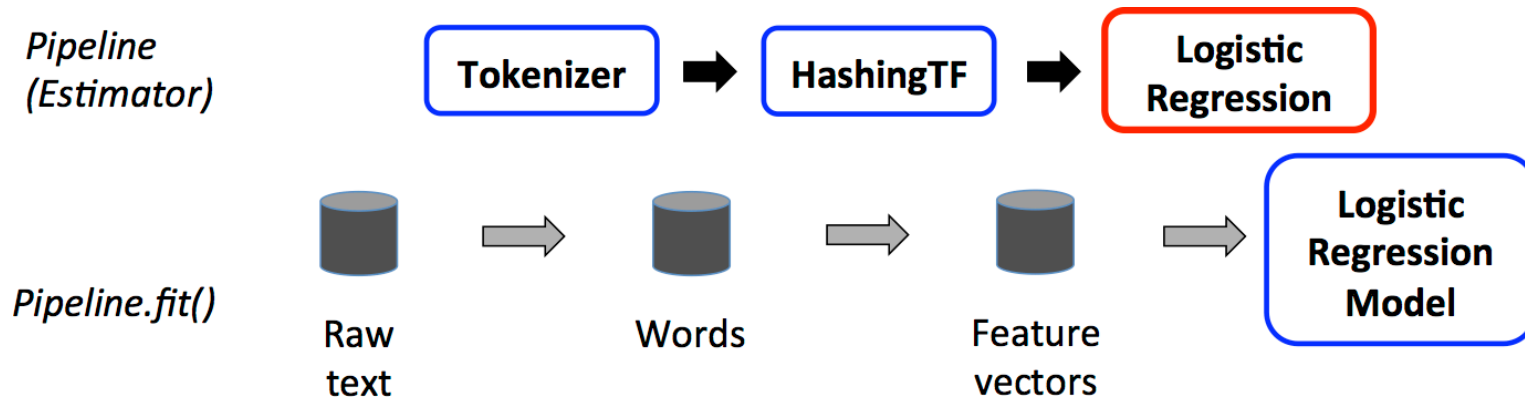
# Pipeline

In machine learning, it is common to run a sequence of algorithms to process and learn from data. E.g., a simple text document processing workflow might include several stages:

- Split each document's text into words.
- Convert each document's words into a numerical feature vector.
- Learn a prediction model using the feature vectors and labels.
- MLlib represents such a workflow as a Pipeline, which consists of a sequence of PipelineStages (Transformers and Estimators).

# Pipeline Example

Above, the top row represents a Pipeline with three stages. The first two (Tokenizer and HashingTF) are Transformers (blue), and the third (LogisticRegression) is an Estimator (red). The bottom row represents data flowing through the pipeline, where cylinders indicate DataFrames.



# Details

- DAG Pipelines: A Pipeline's stages are specified as an ordered array. The examples given here are all for linear Pipelines, i.e., Pipelines in which each stage uses data produced by the previous stage.
- It is possible to create non-linear Pipelines as long as the data flow graph forms a Directed Acyclic Graph (DAG).



# Parameters

- MLlib Estimators and Transformers use a uniform API for specifying parameters.
- A Param is a named parameter with self-contained documentation. A ParamMap is a set of (parameter, value) pairs.
- There are two main ways to pass parameters to an algorithm:
  - Set parameters for an instance. E.g., if lr is an instance of LogisticRegression, one could call lr.setMaxIter(10) to make lr.fit() use at most 10 iterations. This API resembles the API used in spark.mllib package.
  - Pass a ParamMap to fit() or transform(). Any parameters in the ParamMap will override parameters previously specified via setter methods.

# Typical Use Cases

Use Case	Pipeline Stages
Classification	Indexers → Assembler → Classifier
Regression	Imputer → Assembler → Regressor
Clustering	Assembler → Scaler → KMeans
Text Mining	Tokenizer → TF-IDF → Classifier
Hyperparameter Tuning	Full Pipeline + CrossValidator



# MLFlow

- You need MLflow when you're doing machine learning experiments and want to track, reproduce, and deploy them more easily.

MLflow is an open-source platform to manage the ML lifecycle, including:

- Experiment tracking
- Model versioning
- Model packaging
- Deployment



# References

- Read more here:
- [ML Pipelines - Spark 3.5.5 Documentation](#)