PRACTICAL 3.4 Spark Machine Learning

[Reference](https://spark.apache.org/docs/latest/ml-pipeline.html)

# Introduction

**spark.ml** is Spark’s machine learning (ML) library inspired by scikit-learn. It provides a uniform set of high-level API built on top of DataFrames for constructing and tuning machine learning pipelines. Some related terminology:

* **Transformer**: an algorithm which can transform one DataFrame into another DataFrame. E.g., an ML model is a Transformer which transforms a DataFrame with features into a DataFrame with predictions.
* **Estimator**: an algorithm which can be fit on a DataFrame to produce a Transformer. E.g., a learning algorithm is an Estimator which trains on a DataFrame and produces a model.
* **Pipeline**: A Pipeline chains multiple Transformers and Estimators together to specify an ML workflow.

Refer to [NOTEBOOK 3.4 SparkML Estimator and Transformer.ipynb](https://drive.google.com/file/d/11p_IosexAG9NjJL9Q0ERDtW5H9cAEakc/view?usp=drive_link) for an example on the use of SparkML’s estimator and transformer.

# Pipelines

A Pipeline is specified as a sequence of stages, and each stage is either a Transformer or an Estimator. These stages are run in order, and the input DataFrame is transformed as it passes through each stage.

* For Transformer stages, the **transform()** method is called on the DataFrame.
* For Estimator stages, the **fit()** method is called to produce a Transformer (which becomes part of the PipelineModel, or fitted Pipeline), and that Transformer’s **transform()** method is called on the DataFrame.

## Training a Pipeline Model

Assume we have a Pipeline with the following three stages:

* Tokenizer (a transformer),
* HashingTF (a transformer) and
* LogisticRegression (an estimator).

The workflow for the above pipeline is as shown in Figure 5.3a.

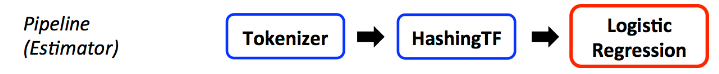


Figure 5.3a Example of Pipeline Stages for an NLP Classification Task

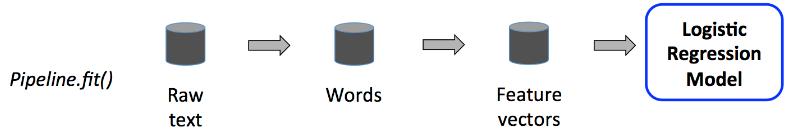


Figure 5.3b Stages that are enacted by the Pipeline.fit() method

Figure 5.3b shows the details of what happens as the different stages of the pipeline are enacted:

* The **Pipeline.fit()** method is called on the original DataFrame, which has raw text documents and labels.
  + The **Tokenizer.transform()** method splits the raw text documents into words, adding a new column with words to the DataFrame.
  + The **HashingTF.transform()** method converts the words column into feature vectors, adding a new column with those vectors to the DataFrame.
  + The **LogisticRegression.fit()** method is called to produce a LogisticRegressionModel.
* After the Pipeline’s **fit()** method runs, it produces a **PipelineModel**, which is a transformer.

## Prediction using a Pipeline Model

The PipelineModel has the same number of stages as the original Pipeline, but all Estimators in the original Pipeline have become Transformers (Figure 5.3c).

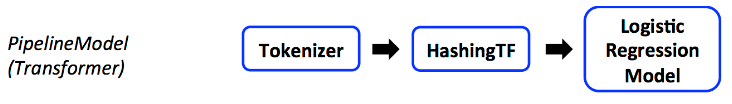


Figure 5.3c The PipelineModel stages.

In Figure 5.3d, When the PipelineModel’s **transform()** method is called on the test dataset, the data are passed through the fitted pipeline in order. Each stage’s **transform()** method updates the dataset and passes it to the next stage.

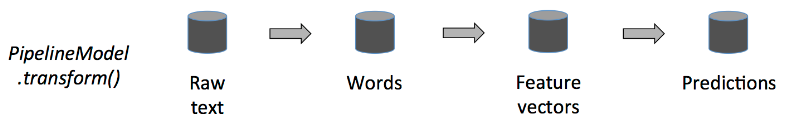


Figure 5.3d The PipelineModel.transform() method

Pipelines and PipelineModels help to ensure that training and test data go through identical feature processing steps.

Refer to [NOTEBOOK 3.5 SparkML Pipeline.ipynb](https://drive.google.com/file/d/1OaS1SwthqF5kTT5L3AWl0zIXen0zcEoW/view?usp=drive_link) for an example on the use of SparkML’s estimator and transformer.

Explore other SparkML modules:

* [Extracting, transforming and selecting features](https://spark.apache.org/docs/latest/ml-features.html)
* [Classification and regression](https://spark.apache.org/docs/latest/ml-classification-regression.html)
* [Clustering](https://spark.apache.org/docs/latest/ml-clustering.html)
* [Collaborative filtering](https://spark.apache.org/docs/latest/ml-collaborative-filtering.html)
* [Frequent pattern mining](https://spark.apache.org/docs/latest/ml-frequent-pattern-mining.html)
* [Model selection and hyperparameter tuning](https://spark.apache.org/docs/latest/ml-tuning.html)
* Etc.