DASCI

Instituto Andaluz Interuniversitario
en Ciencia de Datos e
Inteligencia Computacional

Ciencia de Datos a través del Big Data

Diego García (djgarcia@ugr.es)
Isaac Triguero (isaaktriguero@ugr.es)

Financiado por
la Unión Europea
NextGenerationEU

GOBIERNO
DE ESPAÑA

MINISTERIO
PARA LA TRANSFORMACIÓN DIGITAL
Y DE LA FUNCIÓN PÚBLICA

SECRETARÍA DE ESTADO
DE DIGITALIZACIÓN
E INTELIGENCIA ARTIFICIAL

Plan de
Recuperación,
Transformación
y Resiliencia

UNIVERSIDAD
DE GRANADA

UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

REPSOL

CAMBRIDGE UNIVERSITY PRESS

Chapter 6

Machine Learning with Spark

Learning Outcomes

- Understanding the need for machine learning in Big Data [KU]
- How to use the Spark Machine Learning library as a tool: defining machine learning pipelines [KU, PPS]
- Understand the differences between standard machine learning and machine learning in Big Data [KU]
- Understanding the limitations of distributed machine learning in Big Data [KU]

Recap

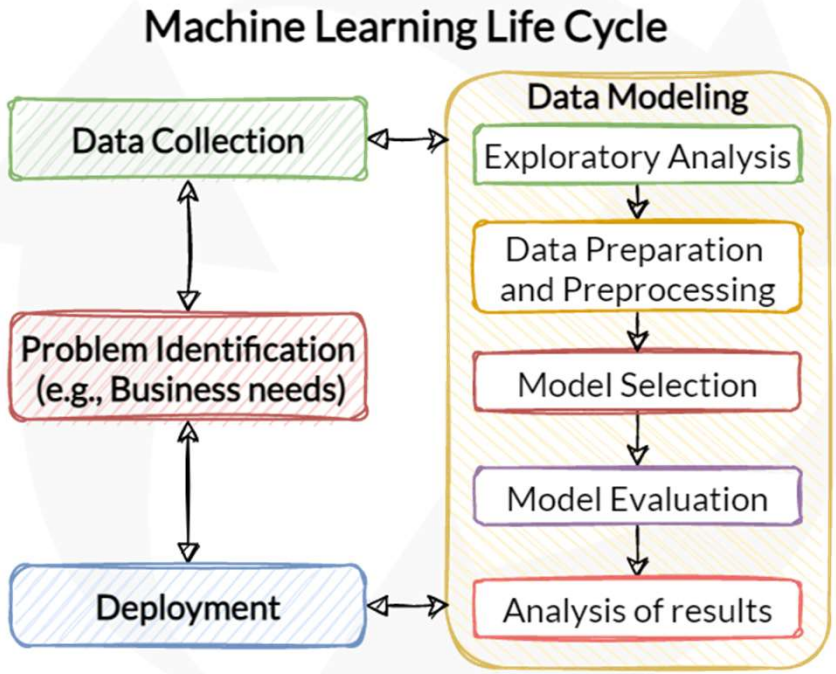
- Big Data Frameworks like Spark provide tools for:
  - Iterative, interactive and streaming processing in Big Data
  - They are based on:
    - The programming paradigm MapReduce (and they extended it)
    - Distributed File Systems (the principle of data locality)
    - Efficient distributed data structures and how to operate them in parallel
    - Fault-tolerance
- They hide the complexity of performing distributed computing, but we still need to be aware of it
- We now move from Big Data Frameworks to Big Data Analytics

Outline

- Introduction and motivation
- ML Libraries: Spark MLlib
- Machine Learning with Big Data

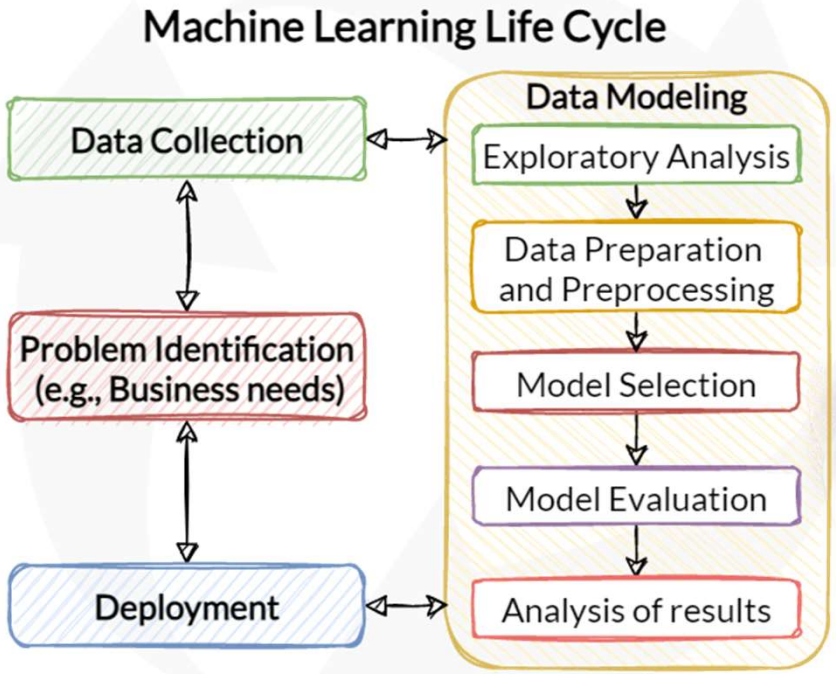
Introduction

- For an ML application to be successful, we will go through various stages
- The Machine Learning Life Cycle
- Iterative process
  - We may go back and forth through the steps



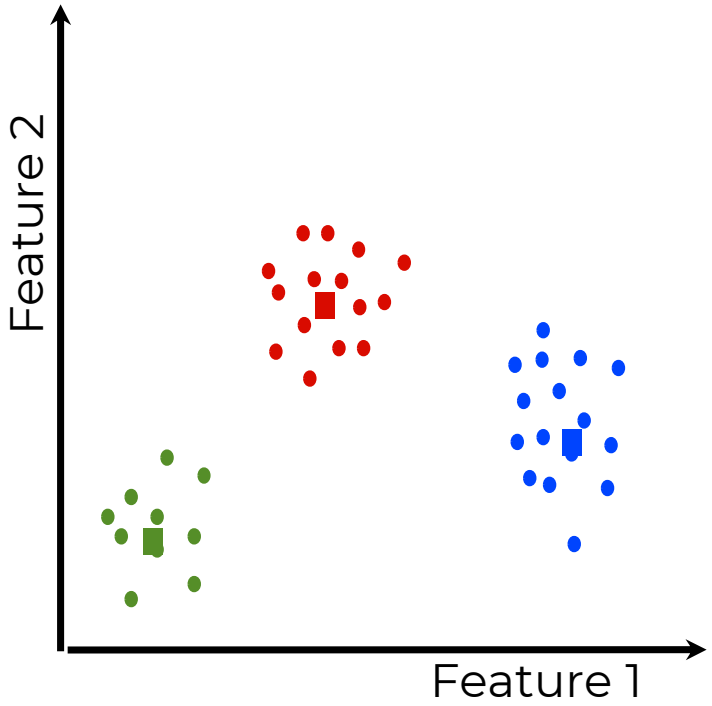
Introduction

- All of the stages **may** (**or may not**) be affected by Big Data
- Shouldn't preprocessing help? (Feature Selection in a 5TB dataset?)
- You may not encounter any problem to train a model (e.g. you do it with GPUs), but the problem lies in deploying it



An Example: k-means in Big Data

- Imagine we run k-means on a dataset with millions of examples
- Big datasets won't fit in the main memory of a single computer
- Finding cluster centers requires lots of distance computations (iterative!)
- Will probably require to keep the dataset in main memory



Source: amplab/databricks

Machine Learning with Big Data

CAMBRIDGE UNIVERSITY PRESS

- Machine Learning techniques have demonstrated to be very useful tools to extract new valuable knowledge from data
- The knowledge extraction process from big data is a very difficult task for most of the **classical** and **advanced** Machine Learning tools
- The **main challenges** are to deal with:
  - The increasing scale of data
    - at the level of **instances**
    - at the level of **features**
  - The velocity in which the data arrives
  - The variety and **complexity** of the problem
- **The idea is to use distributed computing to deal with those challenges!**

The Science of ML for Big Data

CAMBRIDGE UNIVERSITY PRESS

- **Is there any 'science'** in doing ML on Big Datasets? Does it simply consist of re-implementing ML algorithms with distributed data structures like RDDs or DataFrames?
  - The science of it lies in **how to distribute the computation** (**reducing data movement**), maximising what can be learnt in a reasonable amount of time
  - This means:
    - Adapting existing methods
    - Redesigning specific steps that may be intractable with large datasets
  - The assumptions about data availability have changed, and so should the algorithms

The Science of ML for Big Data

CAMBRIDGE UNIVERSITY PRESS

- ML has some interesting properties we need to bear in mind:
  - May require long runs – **fault tolerance** is essential
  - Prior to training a model we may need to **transform** our data
  - Many algorithms may require to **iterate** through the data multiple times
- We have to think which operations may not simply work in this context, or may not be useful at all
- The use of Big Data Frameworks is inevitable, using Spark (over others like Flink or Dask) is simply a matter of choice

Machine Learning Libraries

CAMBRIDGE UNIVERSITY PRESS

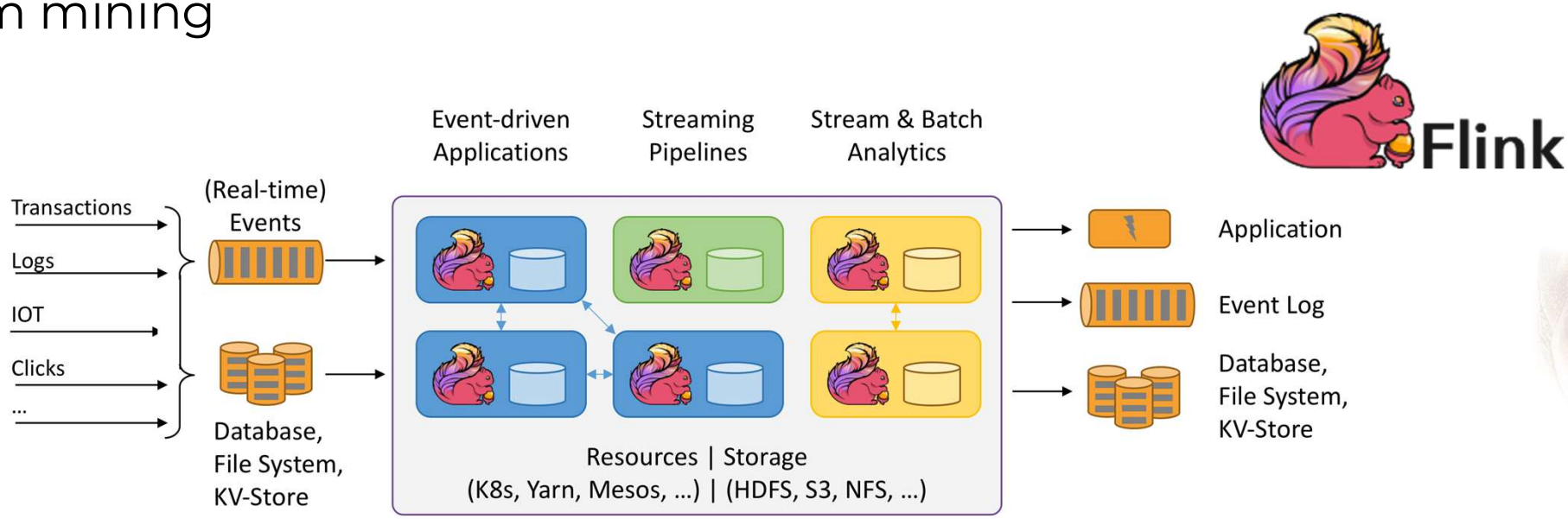
- Machine learning end-users are usually not experts in machine learning
- They usually rely on open-source libraries that allow them to perform data analytics - **Off-the-shelf techniques**
- **Minimum knowledge still required**:
  - Understanding the behavior of the techniques
  - Appropriateness for the problem at hand
  - Hyperparameters' optimization
  - Experiment validation/performance metrics
- In the most extreme scenario, complete newbies could use Auto-ML (not much for Big data though! Have a look at *H2O*)

The ML Libraries' Landscape

- Traditional libraries such as *sklearn* are designed for sequential processing
  - Won't scale out and don't offer fault-tolerance
- With the idea of MapReduce and Hadoop, many libraries appeared, e.g., Mahout, Flink and many more
- Spark offers its own ML library exploiting the DataFrames API
  - Offers **a limited set of algorithms** for classification, regression, clustering and collaborative filtering
  - Also, lots of tools to construct ML **pipelines**

Apache FlinkML

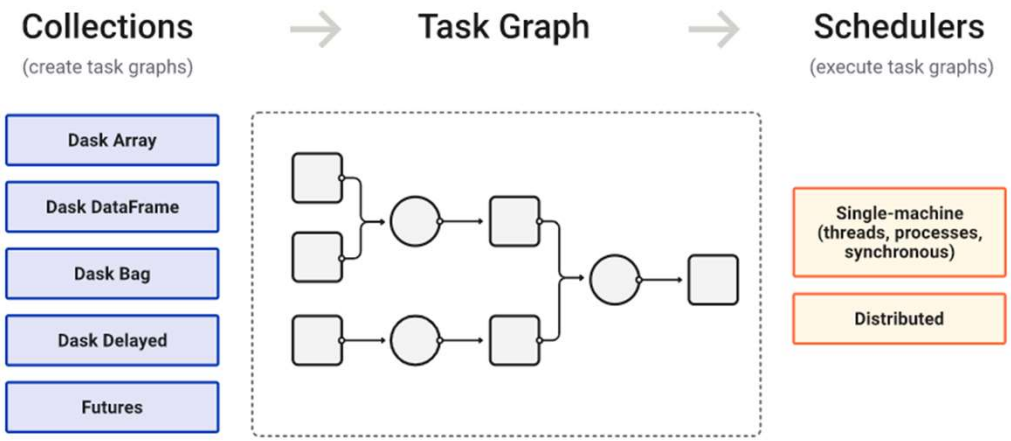
- Spark is not the only Apache library for Machine Learning
- Flink is an alternative which is particularly interesting for Data Stream mining



https://flink.apache.org/

DaskML

- Dask is another interesting framework for parallel computing in **Python**!
  - Offer interfaces like numpy, Pandas or Python iterators for distributed environments
- DaskML allows us to us sklearn and other ML libraries



MLlib: What Does It Offer?

- **Traditional algorithms** for classification, regression, clustering and collaborative filtering
- **Feature transformation** methods such as feature extraction, dimensionality reduction, feature selection, etc.
- **Pipelines** tools to construct and evaluate ML pipelines
- Other utilities such as persistence tools to save and load models, compute statistics or data handling

- Why Spark Mllib does not have all the algorithms provided in the scikit-learn library?

MLlib: Basic Concepts

- A **pipeline** is a programming construct that allows us to automatize part of the ML life cycle
- Two key concepts/operations in Mllib
  - **Transformer**: Operations that preprocess/transform your data for the subsequent ML algorithm (e.g., apply one-hot encoding to categorical variables)
  - **Estimator**: Operations that use the data to learn the parameters of a model that can later be used for prediction
- MLlib **pipeline**: A sequence of multiple transformers and estimators

Transformer

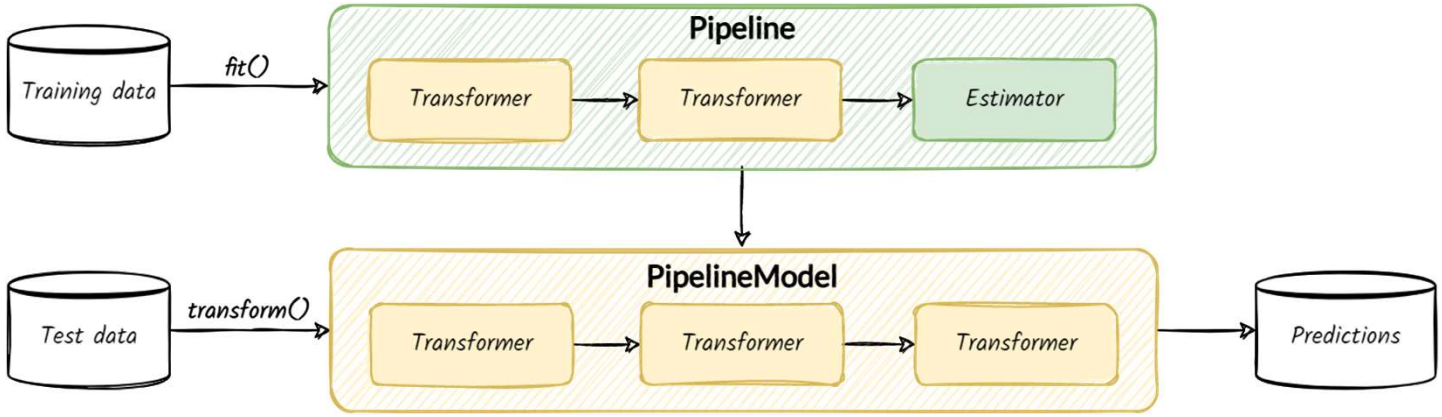
- **An algorithm that transform our DataFrame** into a new one
  - E.g., adding one or more columns
  - It **doesn't** need to **fit any parameters**
- **Examples**
  - Pre-processing method without any parameters (e.g., one hot encoding)
  - Pre-processing with previously fitted parameters (e.g., max-min scaler)
  - An already learned ML model to make predictions adding a a new column
- Transformers implement the `transform()` method

Estimator

- This can be any ML algorithm, or simply **an algorithm fitting some parameters**
  - Takes a DataFrame as input, **fits the parameters**
  - Returns a `Transformer`
- **Examples**
  - An ML method (e.g., Linear Regression fitting its parameters)
  - A preprocessing technique that requires parameters (e.g., max-min scaler)
- Estimators implement the `fit()` method

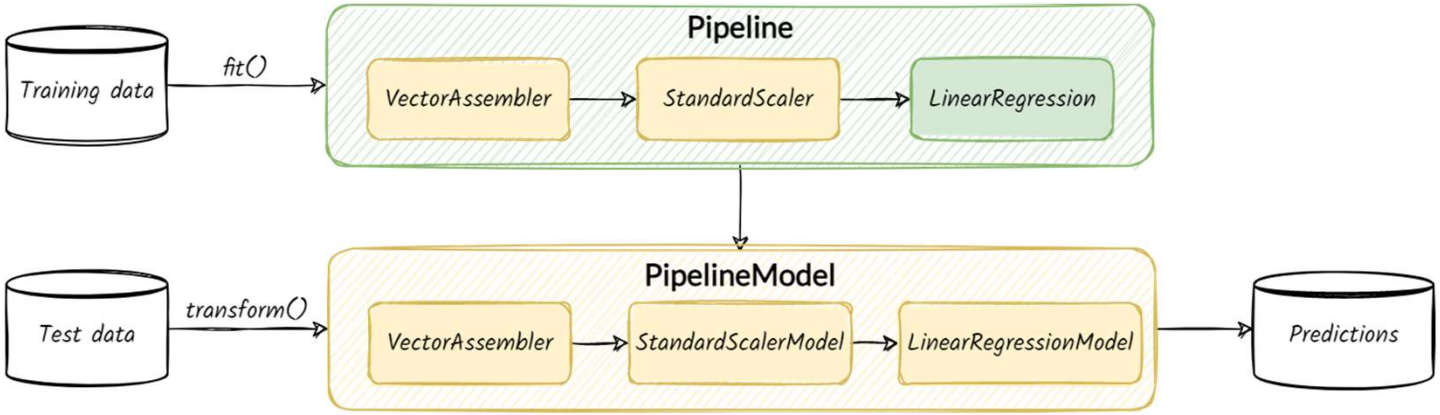
Pipelines

- A Pipeline **concatenates multiple Transformer and Estimator** objects
  - A Pipeline is an Estimator itself. It has a `fit()` method
  - The result of fitting a Pipeline is a Transformer! Which will allow us to make predictions (adding them as an additional column)



Pipelines

- An example



Pipelines

- Details
  - **Runtime checks**: Feeding a wrongly formatted DataFrame will produce a runtime error before running the Pipeline
  - **Unique Pipeline stages**: You can't run the same preprocessing operation more than once in the same pipeline
- There are other details about Pipelines that must be discussed, e.g., use of parameters, storing your models, etc.
  - **Let's do this with a hands-on example**

Hands-on Mllib Outline

- Learning a Linear Regression model
  - Toy Example
- Intro to Pipelines
- Evaluating ML algorithms
- Hyper-parameter tuning
  - CrossValidator vs TrainValidationSplit

Take-Home Message

- Big Data may affect all stages of the ML life cycle
- The science of doing Machine learning with Big Data
- Basic concepts about the MLlib: Pipelines, Transformer, Estimator

To Learn More

CAMBRIDGE UNIVERSITY PRESS

- **If you are not very familiar with Machine Learning:**
  - Classic books: (Bishop, 2006), (Hastie, 2001), (Ng, 2017)
  - Data pre-processing books: (Garcia, 2015), (Luengo, 2020)
  - On-line courses: Machine Learning Specialization by Coursera, Machine Learning in Python with Scikit-learn MOOC
  - Hands on with Python: (Garreta, 2017), (Geron, 2022)
- **To deal with Big Data and Machine Learning**
  - MLlib: (Xiangrui, 2016)
  - Spark Packages
  - Other libraries: FlinkML, *DaskML* (Daniel, 2019).

CAMBRIDGE UNIVERSITY PRESS

Chapter 6

Machine Learning with Spark