Big Data with Apache Spark and Python: from zero to expert

Data Bootcamp
BEST DATA TRAINING

1

Introduction to Apache Spark

Data Bootcamp
BEST DATA TRAINING

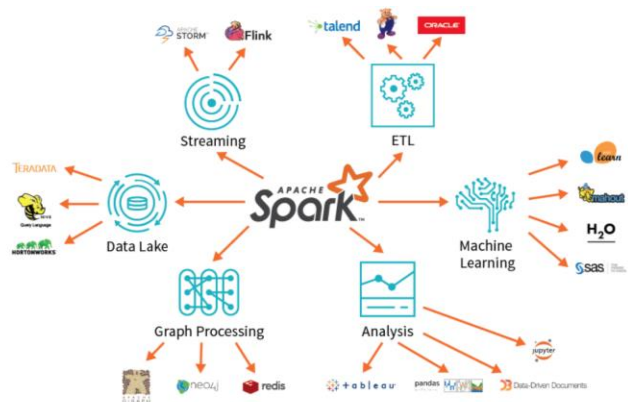
2

Apache Spark

Spark is an **open source Big Data solution**. Developed by the RAD laboratory at UC Berkeley (2009).

It has become the **most used environment**

in Big Data.



Data Bootcamp

3

Apache Spark vs MapReduce

Easier and faster than Hadoop MapReduce.

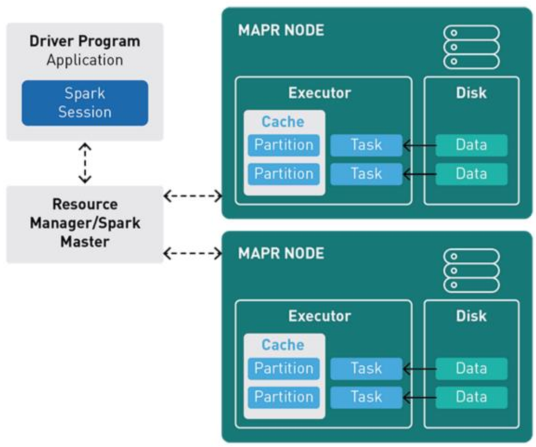
Differences:

- **Spark is faster** as processes data in RAM (memory) while Hadoop reads and writes files to HDFS (on disk)
- **Spark** is optimized for better **parallelism**, **CPU** utilization, and **faster** startup
- Spark has richer **functional programming** model
- Spark is especially useful for **iterative algorithms**

Data Bootcamp

4

How works Spark in a cluster

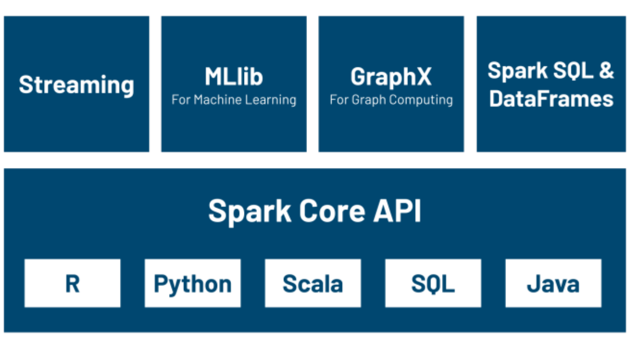


- A Spark application runs as independent processes, coordinated by the **SparkSession** object **in the driver** program.
- The resource or **cluster manager assigns tasks** to workers, one task per partition.
- A **task** applies its **unit of work** to the dataset in its partition and outputs a new partition dataset. Because iterative algorithms apply operations repeatedly to data, they **benefit from caching** datasets across iterations.
- **Results** are **sent back to the driver** application or can be saved to disk.

5

Spark Components

Spark contains a very **complete ecosystem** of tools.



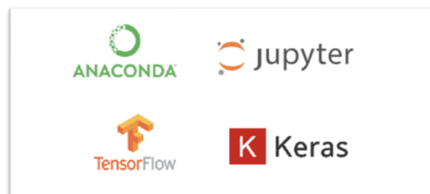
- **Core:** Contains the basic functionality of Spark. Also, home to the API that defines **RDDs**.
- **SQL:** Package for working with **structured data**. It allows querying data via SQL or Hive. It supports various sources.
- **Streaming:** Enables processing of **live streams of data**. Spark Streaming provides an API for manipulating data streams that are similar to Spark Core's RDD API.
- **Mllib:** Provides multiple types of **machine learning algorithms**, like classification, regression, clustering, etc.
- **GraphX:** Library for manipulating **graphs** and performing graph-parallel computations.

6

PySpark

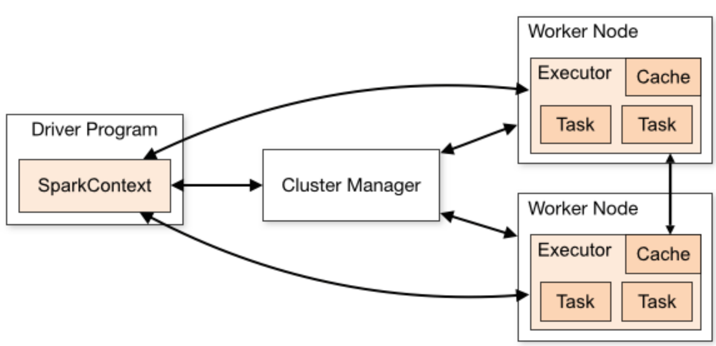
PySpark is the open source, Python API for Apache Spark. It is a distributed computing framework for Big Data processing. Advantages of PySpark:

- **Easy** to learn
- Extensive set of libraries for **Machine Learning and Data Science**
- Great support from the **community**



PySpark Architecture

Apache Spark works on **a master-slave architecture**. **Operations** are executed on **workers**, and the **Cluster Manager** manages resources.



Types of cluster administrators

Spark supports the following cluster administrators:

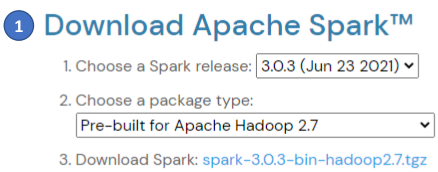
- **Standalone** : Simple cluster administrator
- **Apache Mesos** : is a cluster administrator who can also run Hadoop MapReduce and PySpark.
- **Hadoop YARN** : the resource manager in Hadoop 2
- **Kubernetes**: to automate the deployment and management of containerized applications.

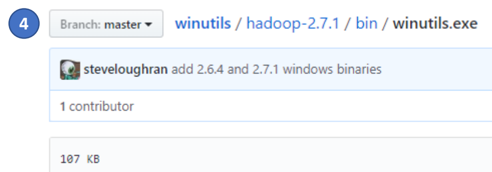
```
# Create SparkSession from builder
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[1]") \
                    .appName('SparkByExamples.com') \
                    .getOrCreate()
print(spark.sparkContext)
print("Spark App Name : "+ spark.sparkContext.appName)

# Outputs
#<SparkContext master=local[1] appName=SparkByExamples.com>
#Spark App Name : SparkByExamples.com
```


Installing Apache Spark

Steps to install Spark (1)

1. Download **Spark** from https://spark.apache.org/downloads.html
2. Modify the **log4j.properties.template,** put log4j.rootCategory=**ERROR** instead of INFO.
3. Install **Anaconda** from https://www.anaconda.com/
4. Download **winutils.exe.** It's a Hadoop binary for Windows. Go to this GitHub repository: https://github.com/steveloughran/winutils/ Select the corresponding Hadoop version with the Spark distribution and look for winutils.exe in /bin.



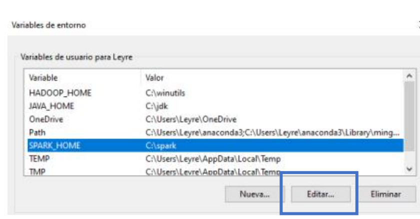


Data Bootcamp
BEST DATA TRAINING

11

Steps to install Spark (2)

1. If you do not have Java or the **Java** version is 7.x or less, download and install Java from Oracle https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html
2. Unzip Spark in **C:\spark**
3. Add the downloaded winutils.exe to a winutils folder in C:. It should look like this: **C:\winutils\bin\winutils.exe**.
4. From **cmd** run: "cd C:\winutils\bin" and then: winutils.exe chmod 777 \tmp\hive
5. Add the environment variables:
   - HADOOP_HOME -> C:\winutils
   - SPARK_HOME -> C:\spark
   - JAVA_HOME -> C:\jdk
   - Path -> %SPARK_HOME%\bin
   - Path -> %JAVA_HOME%\bin



Data Bootcamp
BEST DATA TRAINING

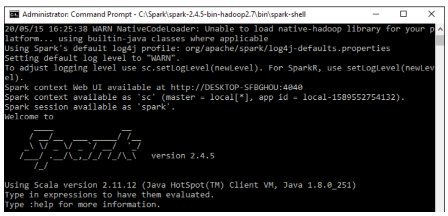
12

Validating the Spark Installation

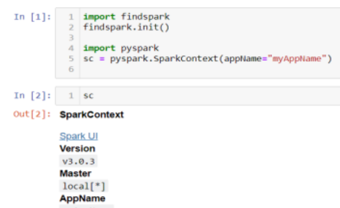
1. From the Anaconda **prompt** run: "cd C:\spark" and then "pyspark". You should see something like picture 1.
2. From **jupyter notebook** install findspark with "pip install findspark" and run the following code

```
import findspark
findspark.init()
import pyspark
sc = pyspark.SparkContext(appName="myAppName")
sc
```

1



2



Data Bootcamp
BEST DATA TRAINING

13

Resilient Distributed Datasets (RDDs)

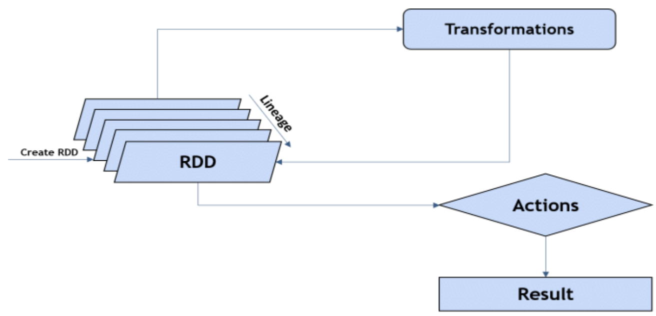
Data Bootcamp
BEST DATA TRAINING

14

Apache Spark RDDs

RDDs are the building blocks of any Spark application. RDD stands for:

- **Resilient:** It is fault tolerant and they can be rebuilt in case of failure
- **Distributed:** Data is distributed across multiple nodes in a cluster
- **Dataset:** Collection of partitioned data



Data Bootcamp
BEST DATA TRAINING

15

Operations in RDDs

With RDDs, you can perform two types of operations:

- **Transformation**: Transformation refers to the operation applied on a RDD to create new RDD. Filter, groupBy and map are the examples of transformations.
- **Actions**: Actions refer to an operation which also applies on RDD, that instructs Spark to perform computation and send the result back to driver. Collect is an example of action.

```
num = [1,2,3,4,5]

num_rdd = sc.parallelize(num)
num_rdd.collect()
```

```
[1, 2, 3, 4, 5]
```

Data Bootcamp
BEST DATA TRAINING

16

DataFrames on Apache Spark

17

Introduction to DataFrames

Dataframes are **tabular** structures. They allow several data types within the same table (**heterogeneous**), while each variable usually has the same data type (**homogeneous**). Dataframes are similar to SQL tables or Excel spreadsheets.

Column Index

Row Index

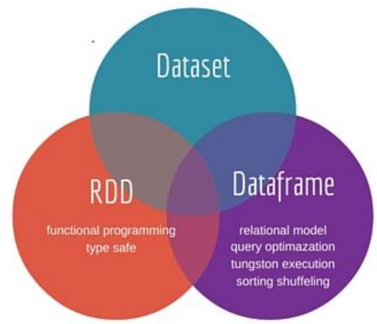
	2018	2019	2020
English	85	60	90
Math	73	80	64
Science	98	58	74
French	88	96	87

18

Advantages of DataFrames

Some of the advantages of working with Dataframes in Spark are:

- Process large amounts of structured or semi-structured data
- **Easy data handling** and imputation of missing values
- Multiple formats as **data sources**
- **Multi-language** support

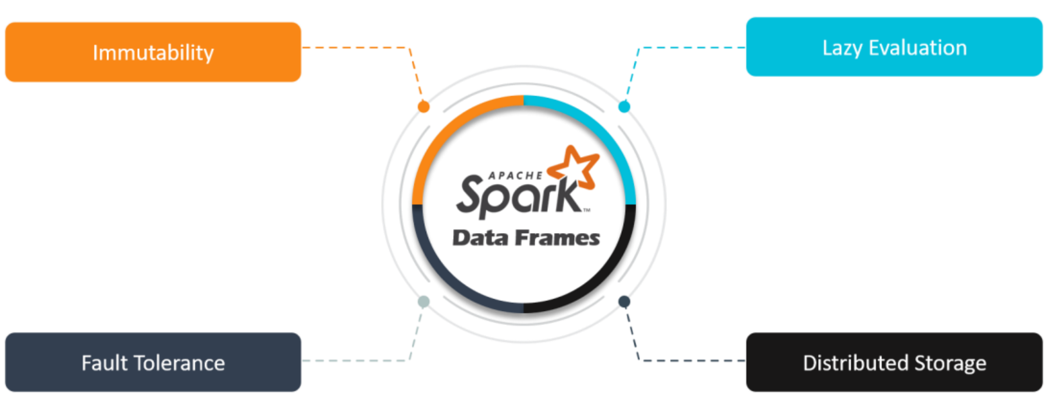


Data Bootcamp
BEST DATA TRAINING

19

Features of DataFrames

Spark **DataFrames are characterized by**: being distributed, have lazy evaluation, immutability and fault tolerance

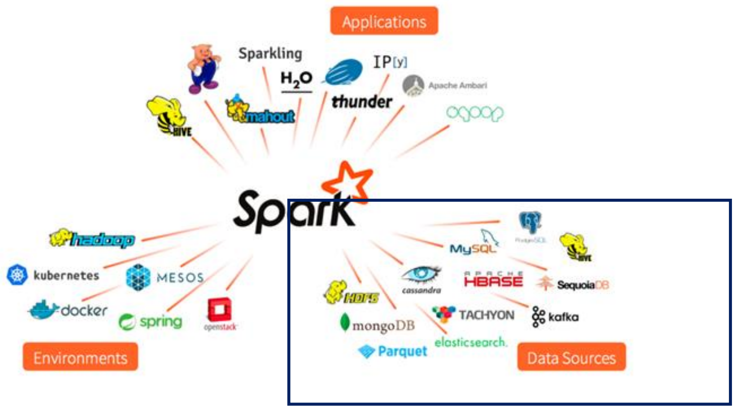


Data Bootcamp
BEST DATA TRAINING

20

DataFrames Data Sources

Data frames in Pyspark can be created in several ways: with files, using RDDs, or with databases.



Data Bootcamp
BEST DATA TRAINING

21

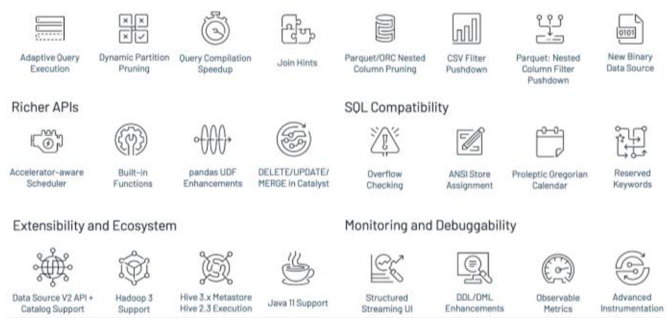
Advanced Spark Features

Data Bootcamp
BEST DATA TRAINING

22

Advanced features

Spark contains **numerous advanced features** to optimize its performance and perform complex transformations on data. Some of them are: UDF, cache ( ), etc.

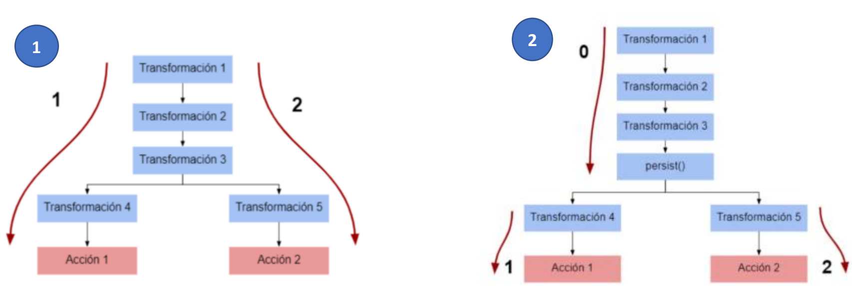


Data Bootcamp
BEST DATA TRAINING

23

Performance optimization

One of the optimization techniques are **cache() and persist()** methods. These methods are used to store an intermediate calculation of an RDD, DataFrame, and Dataset so that they can be **reused** in subsequent actions.



Data Bootcamp
BEST DATA TRAINING

24

Advanced Analytics with Spark

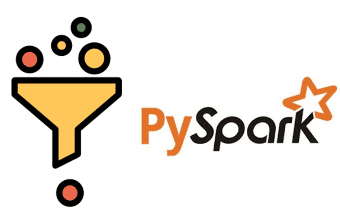
Data Bootcamp
BEST DATA TRAINING

25

Functions for data analytics

In order to **train a model** or perform **statistical analysis** in our data, the following functions and tasks are necessary:

- Generate a Spark session
- Import the data and generate the correct **schema**
- Methods for **inspecting** data
- Data and column **transformation**
- Dealing with **missing values**
- Execute **queries** (SQL, Python, PySpark…)
- Data **visualization**



Data Bootcamp
BEST DATA TRAINING

26

Data visualization

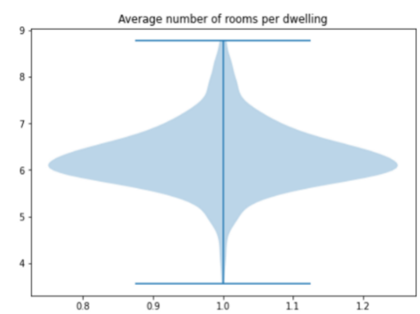
PySpark supports numerous **Python** data visualization libraries such as seaborn, matplotlib, bokehn, ...

```
#Violoin Plot
df5 = sqlContext.sql("SELECT RM from BostonTable").toPandas()
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
bp = ax.violinplot(df5['RM'])
plt.title('Average number of rooms per dwelling')
plt.show()
```

bokeh

matplotlib

seaborn



Data Bootcamp
BEST DATA TRAINING

27

Apache Spark Koalas

Data Bootcamp
BEST DATA TRAINING

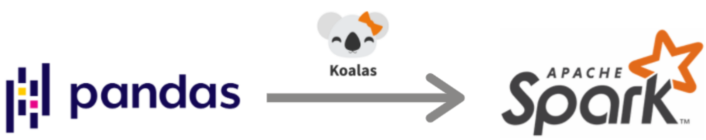
28

Introduction to Koalas

Koalas provides a **direct replacement** for Pandas, allowing efficient scaling to hundreds of nodes for data science and machine learning.

Pandas doesn't scale to Big Data.

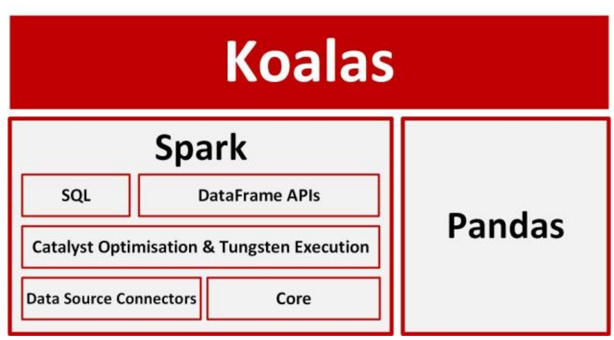
PySpark DataFrame is more compatible with **SQL and Koalas DataFrame** is closer to Python



29

Koalas and PySpark DataFrames

Koalas and PySpark DataFrames are different. **Koalas** DataFrames follows the **structure of Pandas** and implements an **index**. The **PySpark DataFrame** is more compatible with tables in relational databases and has no indexes. Koalas translates pandas APIs to **Spark SQL** logic plan.



30

Example: Feature Engineering with Koalas

In data science, the **get_dummies( )** function of pandas is often needed to encode categorical variables as **dummy (numerical)** variables.

Thanks to Koalas you can do this in Spark with just a few settings.

Pandas

```
import pandas as pd
data = pd.read_csv("fire_department_calls_sf_clean.csv", header=0)
display(pd.get_dummies(data))
```

Koalas

```
import databricks.koalas as ks
data = ks.read_csv("fire_department_calls_sf_clean.csv", header=0)
display(ks.get_dummies(data))
```

	Call_Me?	Money	Target
0	Yes	5	10
1	No	3	4
2	Maybe	5	5
3	Yes	10	7
4	Yes	9	9

	Money	Call_Me?_Maybe	Call_Me?_No	Call_Me?_Yes
0	5	0	0	1
1	3	0	1	0
2	5	1	0	0
3	10	0	0	1
4	9	0	0	1

Example: Feature Engineering with Koalas

In data science you often need to work with **time data**. Pandas allows you to work with this type of data easily.

With PySpark it is more complicated.

```
  End_date            Start_date
0 2013-03-17 21:45:00 2012-01-31 12:00:00
1 2013-03-24 21:45:00 2012-02-29 12:00:00
2 2013-03-31 21:45:00 2012-03-31 12:00:00
3 2013-04-07 21:45:00 2012-04-30 12:00:00
4 2013-04-14 21:45:00 2012-05-31 12:00:00
5 2013-04-21 21:45:00 2012-06-30 12:00:00
6 2013-04-28 21:45:00 2012-07-31 12:00:00
```

```
  End_date            Start_date          diff_seconds
0 2013-03-17 21:45:00 2012-01-31 12:00:00 35545500.0
1 2013-03-24 21:45:00 2012-02-29 12:00:00 33644700.0
2 2013-03-31 21:45:00 2012-03-31 12:00:00 31571100.0
3 2013-04-07 21:45:00 2012-04-30 12:00:00 29583900.0
4 2013-04-14 21:45:00 2012-05-31 12:00:00 27510300.0
5 2013-04-21 21:45:00 2012-06-30 12:00:00 25523100.0
6 2013-04-28 21:45:00 2012-07-31 12:00:00 23449500.0
```

Pandas

```
df['diff_seconds'] = df['End_date'] - df['Start_date']
df['diff_seconds'] = df['diff_seconds']/np.timedelta64(1,'s')
print(df)
```

Koalas

```
import databricks.koalas as ks
df = ks.from_pandas(pandas_df)
df['diff_seconds'] = df['End_date'] - df['Start_date']
df['diff_seconds'] = df['diff_seconds'] / np.timedelta64(1,'s')
print(df)
```

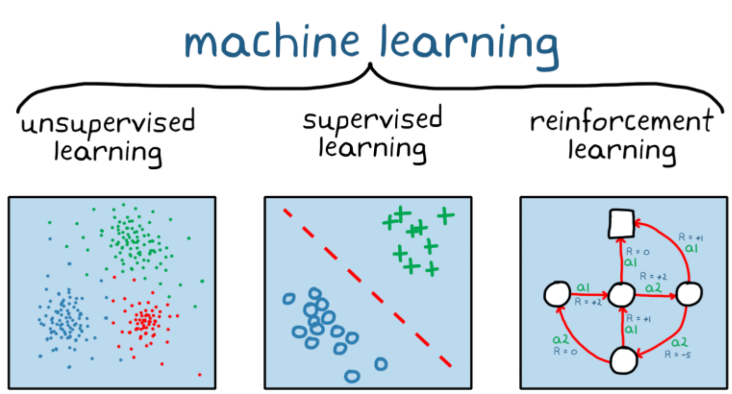
Machine Learning with Spark

Data Bootcamp
BEST DATA TRAINING

33

Spark Machine Learning

Machine Learning: is the construction of **algorithms** that can learn from data and make predictions about it. **Spark ML** has machine learning algorithms and functions.



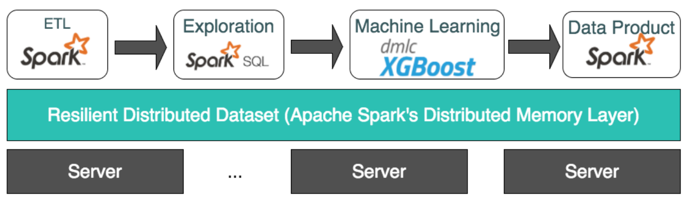
Data Bootcamp
BEST DATA TRAINING

34

Spark Machine Learning Tools

Spark ML libraries:

- **spark.mllib** contains the original API built on top of RDD
- **spark.ml** provides a top-level API built on top of DataFrames for building ML pipelines. The main ML API.



Resource: https://www.r-bloggers.com/

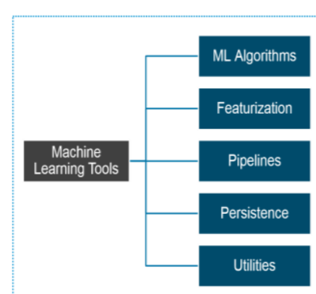
Data Bootcamp
BEST DATA TRAINING

35

Spark Machine Learning Components

Spark ML provides the following tools:

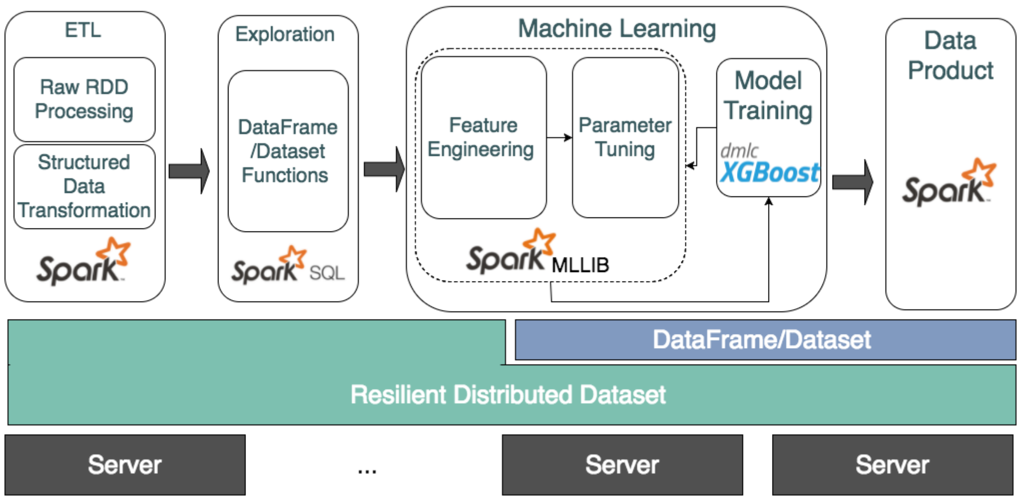
- **ML algorithms**: Include common Machine Learning algorithms such as classification, regression, clustering, and collaborative filtering.
- **Preprocessing functions:** Includes: extraction, transformation, dimensionality reduction and feature selection.
- **Pipelines:** are tools for building ML models in stages.
- **Persistence:** To save and load algorithms, models and pipelines.
- **Utilities:** for linear algebra, statistics and data management.



Data Bootcamp
BEST DATA TRAINING

36

Machine Learning Process



Resource: https://www.r-bloggers.com/

Data Bootcamp BEST DATA TRAINING

37

Feature Engineering with Spark

The most commonly used data preprocessing techniques in **Spark** are:

- **VectorAssembler**
- **Grouping**
- **Scaling and normalization**
- **Working with categorical features**
- **Text Data Transformers**
- **Function manipulation**
- **PCA**

Data Bootcamp BEST DATA TRAINING

38

Feature Engineering with Spark

- **Vector Asembler:** It is used to concatenate features into a single vector that can be passed to the estimator or the ML algorithm.
- **Grouping:** is the simplest method for converting continuous variables into categorical variables. It can be done with the Bucketizer class.
- **Scaling and standardization:** is another common task for numerical variables. It transform data to obtain a normal distribution.
- **MinMaxScaler and StandardScaler:** standardize variables with a mean of zero and a standard deviation of 1.
- **StringIndexer :** to convert categorical variables to numerical.

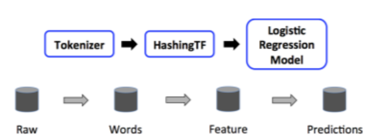
```
+ ---- + ---- + ---- + ------------- +
| int1 | int2 | int3 | características |
+ ---- + ---- + ---- + ------------- +
| 7 | 8 | 9 | [7.0,8.0,9.0] |
| 1 | 2 | 3 | [1.0,2.0,3.0] |
| 4 | 5 | 6 | [4.0,5.0,6.0] |
+ ---- + ---- + ---- + ------------- +
```

Data Bootcamp
BEST DATA TRAINING

39

Pipelines in PySpark

In **pipelines**, the **different stages** of machine learning work can be grouped together as a single entity and can be used as an uninterrupted **workflow**. Each **stage is a Transformer**. They run in sequence and the input data is transformed as they go through each **stage**.




```
tokenizer = Tokenizer(inputCol="SystemInfo", outputCol="words")
hashingTF = HashingTF(inputCol=tokenizer.getOutputCol(), outputCol="features")
lr = LogisticRegression(maxIter=10, regParam=0.01)

# Build the pipeline with our tokenizer, hashingTF, and logistic regression stag
pipeline = Pipeline(stages=[tokenizer, hashingTF, lr])

model = pipeline.fit(training)
```

Data Bootcamp
BEST DATA TRAINING

40

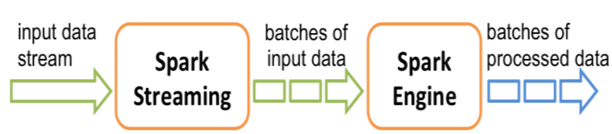
Spark Streaming

41

Spark Streaming Fundamentals

PySpark Streaming is a scalable and fault-tolerant system that follows the RDD batch paradigm. It operates in batch intervals, receiving a **stream of continuous input data** from sources such as Apache Flume, Kinesis, Kafka, TCP sockets, etc.

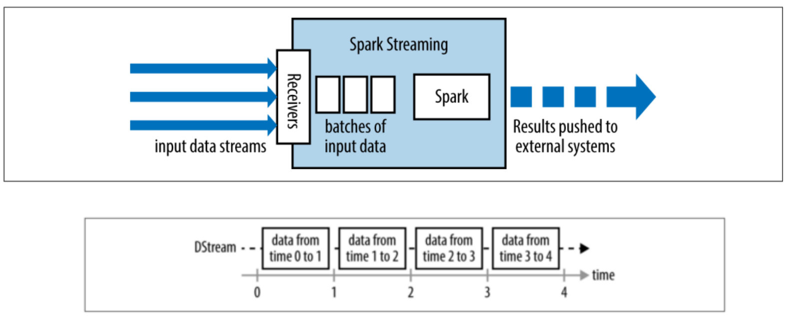
Spark Engine processes them.



42

How Spark Streaming Works

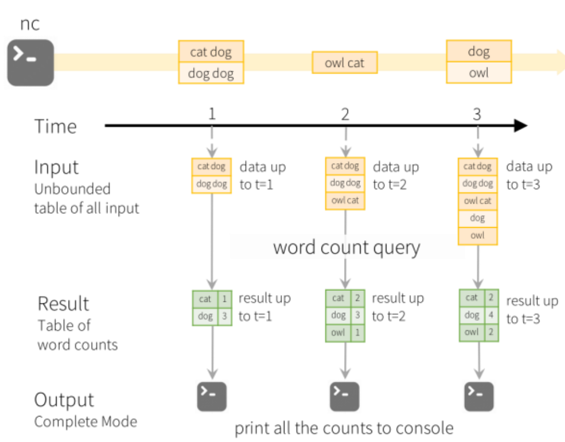
Spark Streaming receives data from multiple sources and groups it into small batches (Dstreams) over a time **interval**. The user can define the range. Each input batch forms an RDD and is processed using Spark jobs to create other RDDs.



Data Bootcamp BEST DATA TRAINING

43

Example: Counting Words



Data Bootcamp BEST DATA TRAINING

44

Output modes

Spark uses several output modes to store the data:

- **Complete:** the entire table will be stored
- **Append:** only the new rows of the last process will be added. Only for queries in which existing rows are not expected to change.
- **Update**: only rows that were updated will be stored. This mode only generates the rows that have changed in the last process. If the query does not contain aggregations, it will be equivalent to append mode.

Complete, Append, Update

```
query = wordCounts \
    .writeStream \
    .outputMode("complete") \
    .format("console") \
    .start()
```

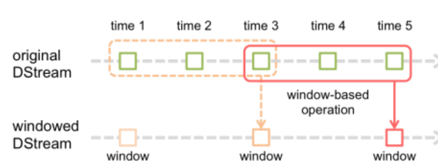
Data Bootcamp
BEST DATA TRAINING

45

Types of transformations

For allow **fault tolerance** the data is copied into two nodes and there is also a mechanism called **checkpointing**. Transformations can be grouped into :

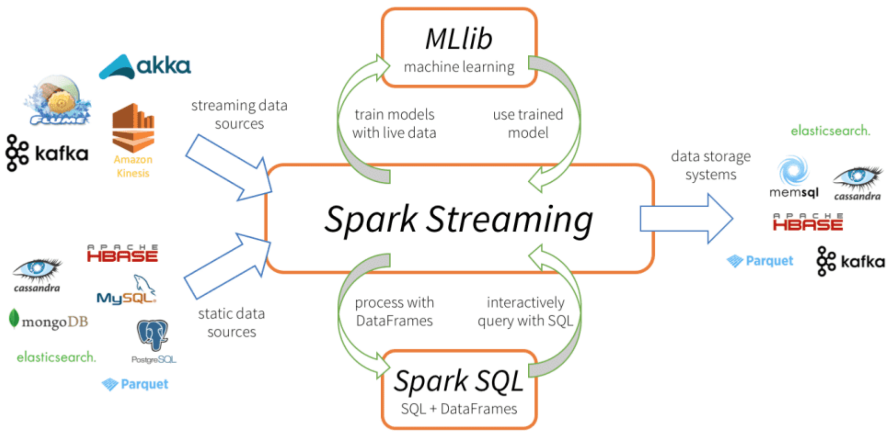
- **Stateless transformation**: each microbatch of data does not depend on the previous data batches, so each batch is fully independent of whatever batches of data preceded it.
- **Stateful transformations**: each microbatch of data depends partially or wholly on the previous batches of data, so each batch considers what happened prior to it and uses that information while being processed.



Data Bootcamp
BEST DATA TRAINING

46

Spark Streaming Capabilities



Data Bootcamp
BEST DATA TRAINING

47

Introduction to Databricks

Data Bootcamp
BEST DATA TRAINING

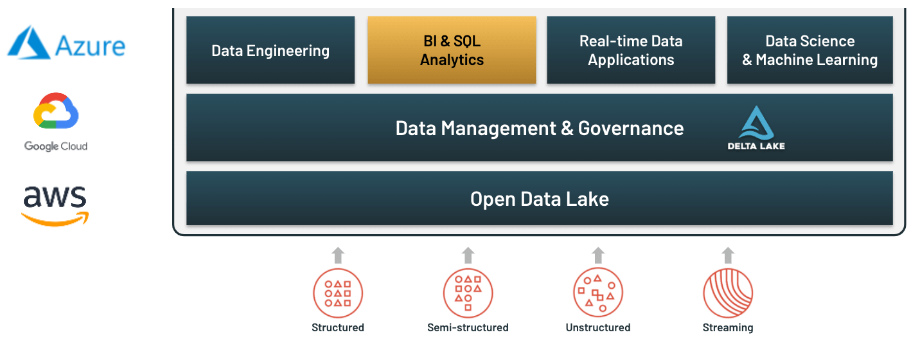
48

Introduction to Databricks

Databricks is the Apache Spark-based **data analytics platform** developed by the creators of **Spark**.

Databricks enables advanced analytics, Big Data and ML in a **simple and collaborative** way.

Available as a **cloud service on Azure, AWS, and GCP.**

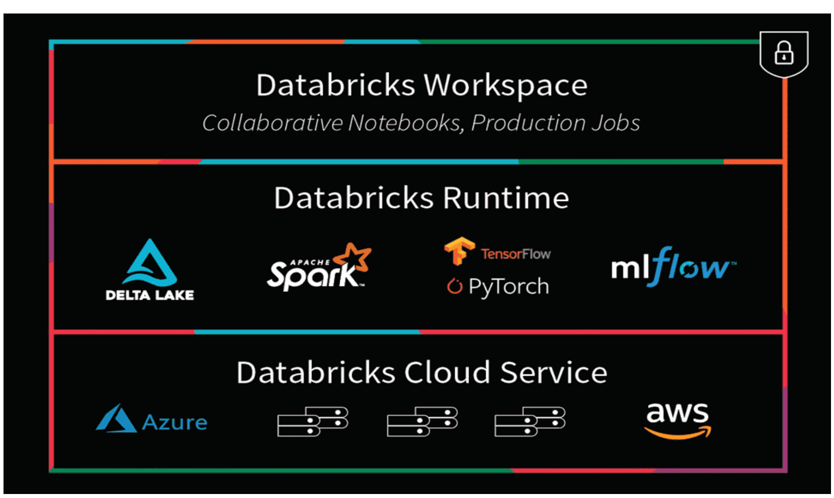


Data Bootcamp
BEST DATA TRAINING

49

Features of Databricks

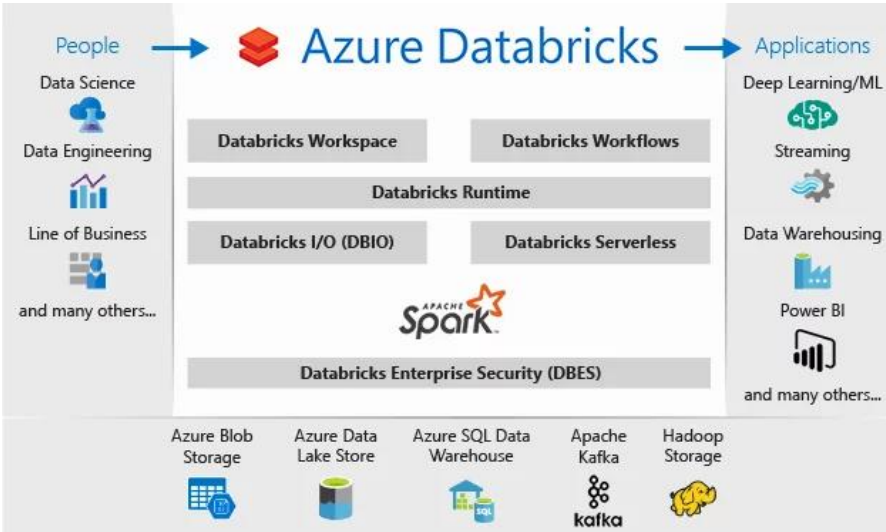
Databricks **auto-scale** and size **Spark environments** in a **simple way**. Facilitates deployments and accelerates the installation and configuration of Big Data environments



Data Bootcamp
BEST DATA TRAINING

50

Databricks Architecture

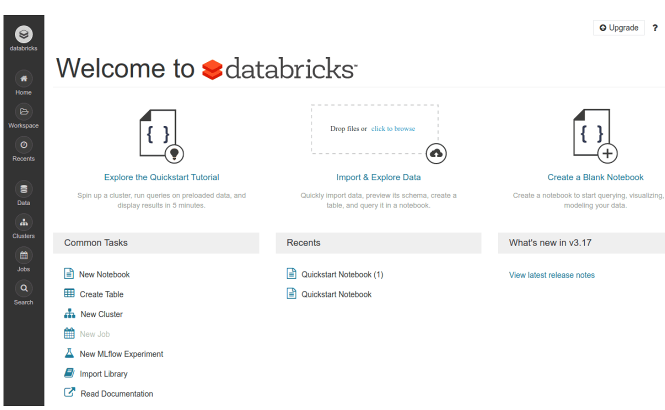


Data Bootcamp
BEST DATA TRAINING

51

Databricks Community

Databricks community is the **free** version. It allows you to use a **small cluster** with limited resources and **non-collaborative notebooks**. Paid version has more capabilities



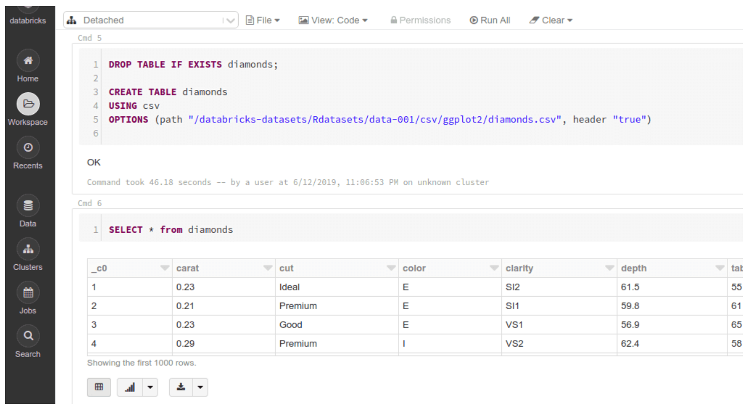
Data Bootcamp
BEST DATA TRAINING

52

Terminology

Important terms to know:

1. Workspaces
2. Notebooks
3. Libraries
4. Tables
5. Clusters
6. Jobs



Data Bootcamp
BEST DATA TRAINING

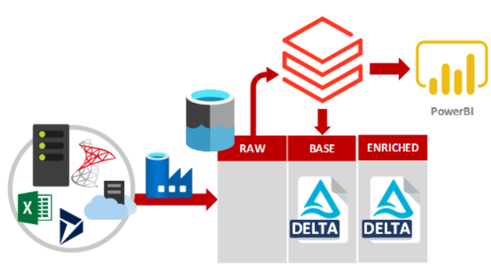
53

Delta Lake

Delta Lake is the open source storage **layer developed** for **Spark and Databricks**. Provides ACID transactions and advanced **metadata** management.

It includes a Spark-compatible query engine that **accelerates operations** and improves performance.

The data stored in **Parquet** format.



Data Bootcamp
BEST DATA TRAINING

54

Resources

55

Resources:

- **https://spark.apache.org/docs/2.2.0/index.html** Official Spark Documentatio
- **https://colab.research.google.com/** Google Colab to be able to have additional computing capacity

56