Big Data con Apache Spark 3 y Python: Zero to Expert

Data Bootcamp
BEST DATA TRAINING

1

Introduction to Apache Spark

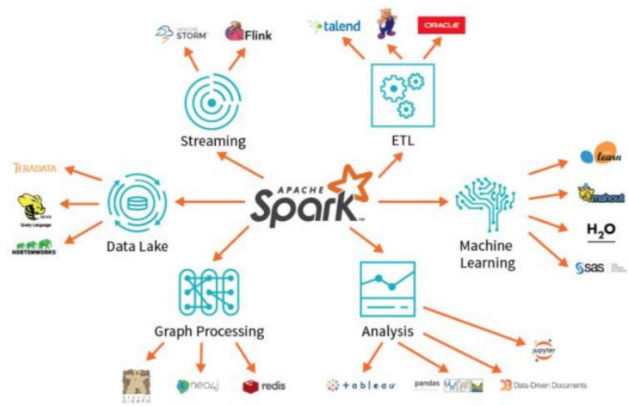
Data Bootcamp
BEST DATA TRAINING

2

Apache Spark

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

It has become a **tool** **of reference** in the field of Big Data.



Data Bootcamp BEST DATA TRAINING

3

Apache Spark vs MapReduce

Easier and faster than Hadoop MapReduce.

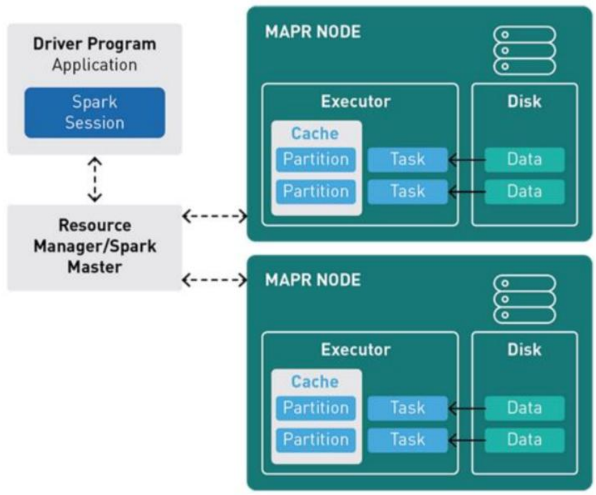
Differences:

- **Spark** much **faster** by caching data in **memory** vs **MapReduce** in the **hard drive** (plus read and write)
- Spark optimized for better **parallelism, CPU** utilization , and faster startup • Spark has a richer **functional programming** model
- Spark is especially useful for **iterative algorithms.**

Data Bootcamp BEST DATA TRAINING

4

How to Spark in a Cluster

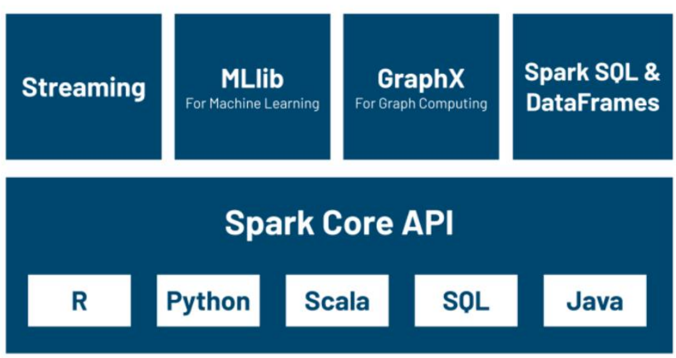


Data Bootcamp
BEST DATA TRAINING

5

Spark Components

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



Data Bootcamp
BEST DATA TRAINING

6

PySpark

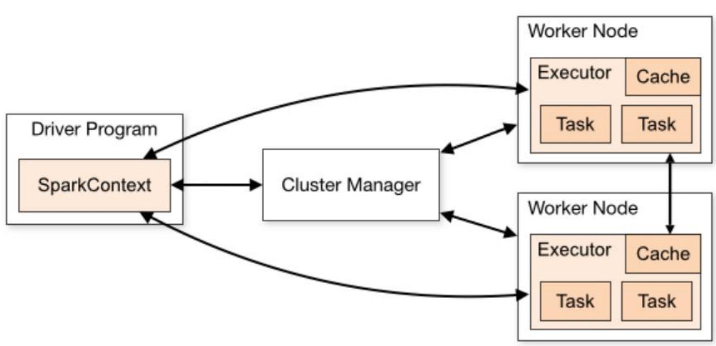
PySpark is a Spark library **written in Python** to run the Python application using the **Apache Spark capabilities.**

Advantages of PySpark:

- **Easy** to learn
- Extensive set of libraries for **ML and DS**
- Great community **support**

PySpark architecture

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



Types of cluster administrators

Spark supports the following cluster managers:

- **Standalone :** simple cluster manager
- **Apache Mesos** – is a cluster manager that can also run Hadoop MapReduce y PySpark.
- **Hadoop YARN :** the resource manager in Hadoop 2
- **Kubernetes:** to automate the deployment and management of applications in containers.

Data Bootcamp
BEST DATA TRAINING

9

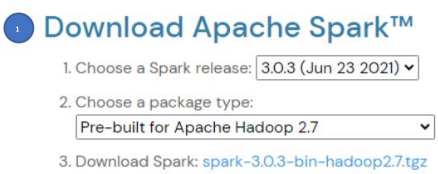
Installing Apache Spark

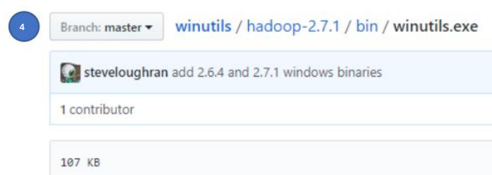
Data Bootcamp
BEST DATA TRAINING

10

Steps to install Spark (1)

1. Download **Spark** from https://spark.apache.org/downloads.html
2. Modify the **log4j.properties.template** to 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 - from the GitHub repository of https://github.com/steveloughran/winutils/ . Navigate to the corresponding Hadoop version with the Spark distribution and look for winutils.exe in **/bin.**



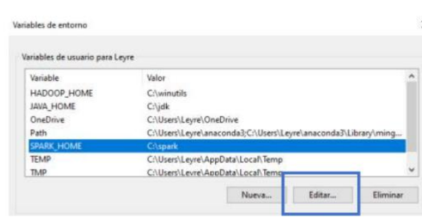


Data Bootcamp
BEST DATA TRAINING

Steps to install Spark (2)

1. If you don't have **Java** or the Java version is 7.x or less, please download and install Java from Oracle https://www.oracle.com/java/technologies/downloads/
2. Unzip Spark to **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



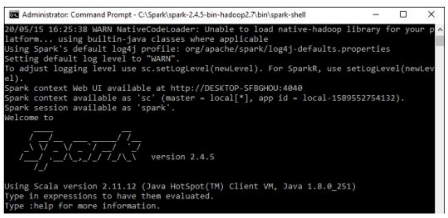
Data Bootcamp
BEST DATA TRAINING

Spark Installation Validation

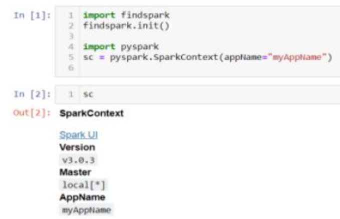
1. From the Anaconda **prompt** run: “cd C:\spark” and then “pyspark”. You should see something like that of image 1.
2. From **jupyter notebook** install findspark with “pip install findspark” and run the following code.

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

1



2

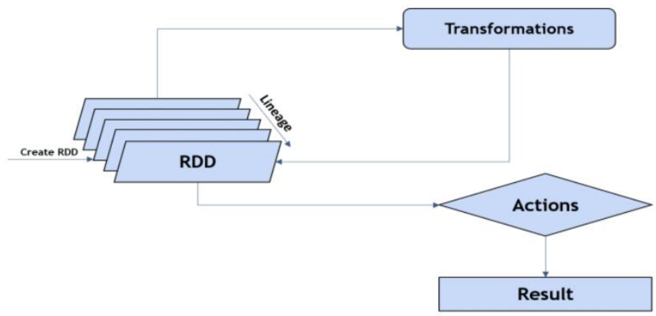


Apache Spark's RDDs

Apache Spark RDDs

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

- **Resilient:** It is fault tolerant and is capable of rebuilding data in case of failure.
- **Distributed –** Data is distributed among multiple nodes in a cluster.
- **Dataset:** A collection of value-partitioned data.



Operations in RDDs

With RDDs, you can perform two types of operations:

- **Transformations:** These operations are applied to create a new RDD.
- **Actions –** These operations are applied in an RDD to tell Apache Spark to apply the calculation and return the result to the controller.

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

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

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

DataFrames en Apache Spark

17

Introduction to Data Frames

DataFrames **are** tabular in nature . They allow several formats within the same table **(heterogeneous),** while each variable usually has values with a single format **(homogeneous).**

Similar to SQL tables or spreadsheets.

Column Index

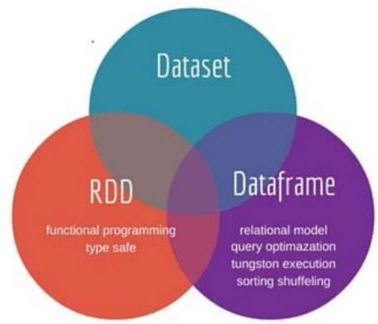
Row Index	2018	2019	2020
English	85	60	90
Math	73	80	64
Science	98	58	74
French	88	96	87

18

Advantages of Data Frames

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

- Ability to process a large amount of structured or semi-structured **data**
- Easy **data handling** and imputation of missing values
- Multiple formats as **data sources**
- Support for **multiple languages**

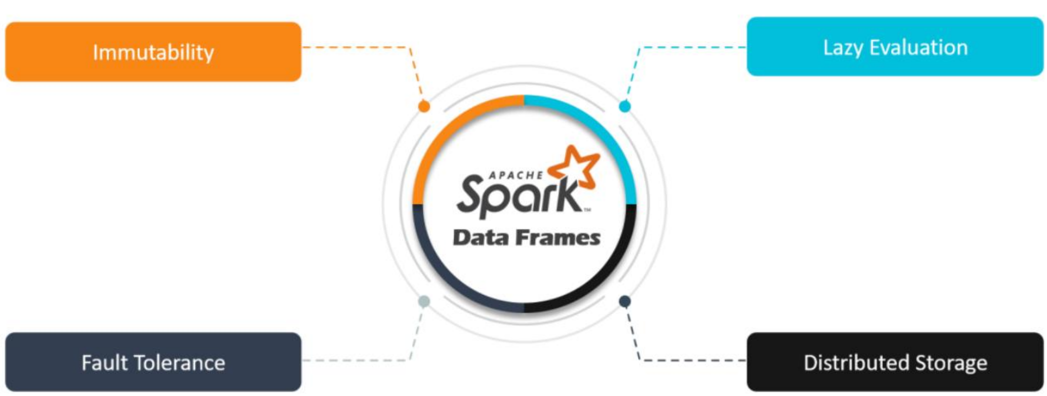


Data Bootcamp
BEST DATA TRAINING

19

DataFrames Characteristics

Spark DataFrames **are characterized** by: **being** distributed, lazy evaluation, immutability and Fault tolerance.



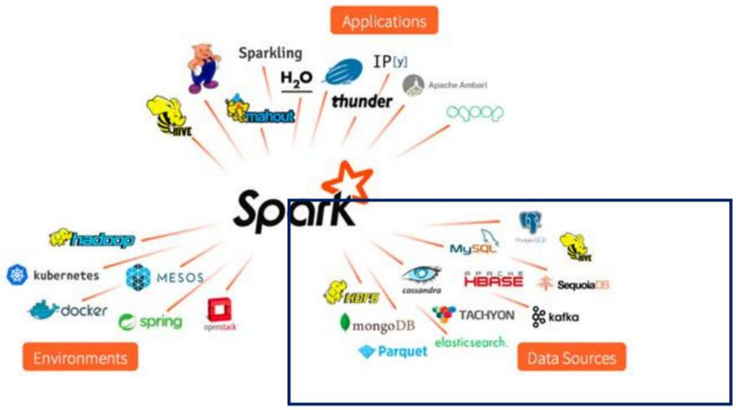
Data Bootcamp
BEST DATA TRAINING

20

DataFrames data sources

Data frames in Pyspark can be created in several ways: via files, using RDDs

or through databases.



Data Bootcamp
BEST DATA TRAINING

21

Spark Advanced Features

Data Bootcamp
BEST DATA TRAINING

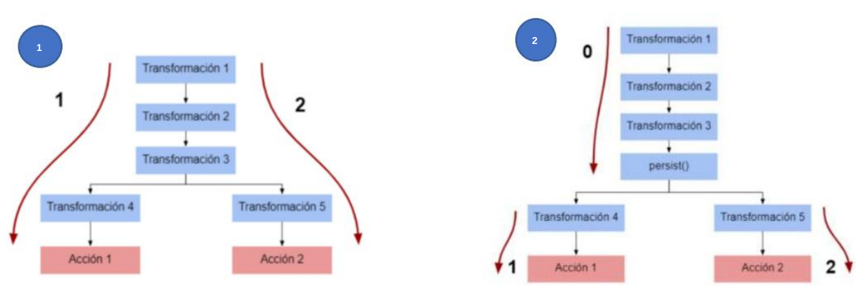
22

advanced features

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

Performance optimization

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

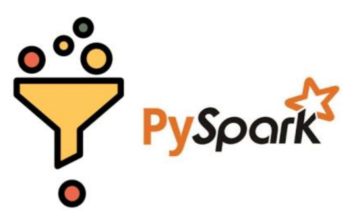


Advanced Analytics with Spark

Functions for data analytics

In order to **train a model** or carry out **statistical analyzes** with our data, it is necessary to following functions and tasks:

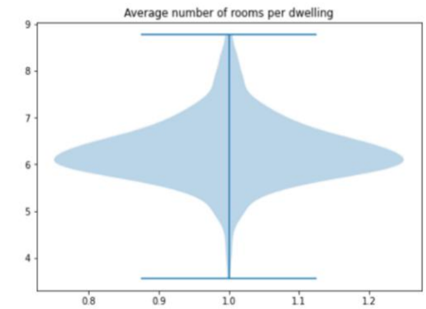
- Generate a Spark session
- Import the data and generate a correct **schema**
- Methods to **inspect** data
- Data and column **transformation**
- Deal with **missing values**
- Run **queries**
- Data **visualization**



data visualization

PySpark is compatible with numerous libraries of **python** data visualization as seaborn, matplotlib, bokehn, etc

```
#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

Machine Learning con Spark

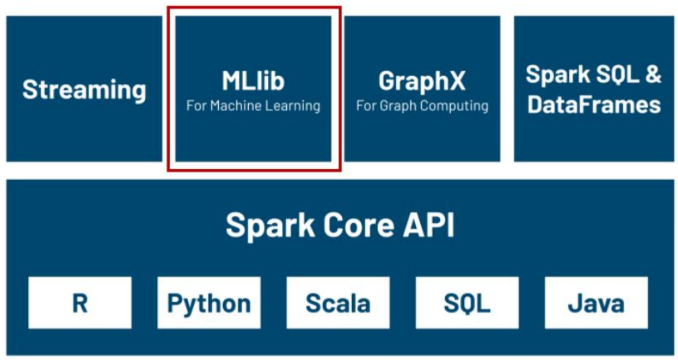
Data Bootcamp
BEST DATA TRAINING

28

Spark Machine Learning

Machine Learning: is the construction of **algorithms** that can learn from the data and make predictions about them.

Spark MLlib is used to perform machine learning on Apache Spark. MLlib consists of algorithms and usual functions.



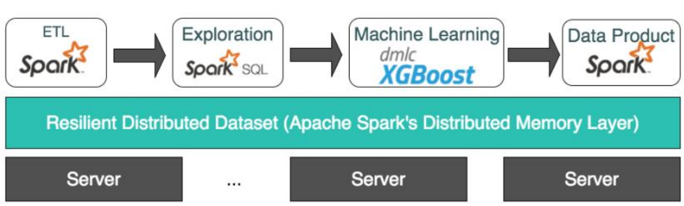
Data Bootcamp
BEST DATA TRAINING

29

Herramientas Spark Machine Learning

MLlib tools:

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



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

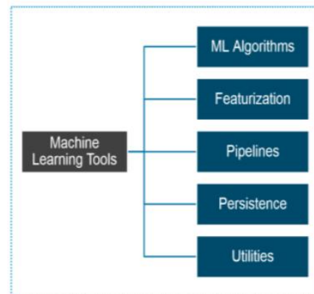
Data Bootcamp
BEST DATA TRAINING

30

Componentes Spark Machine Learning

Spark MLlib provides the following tools:

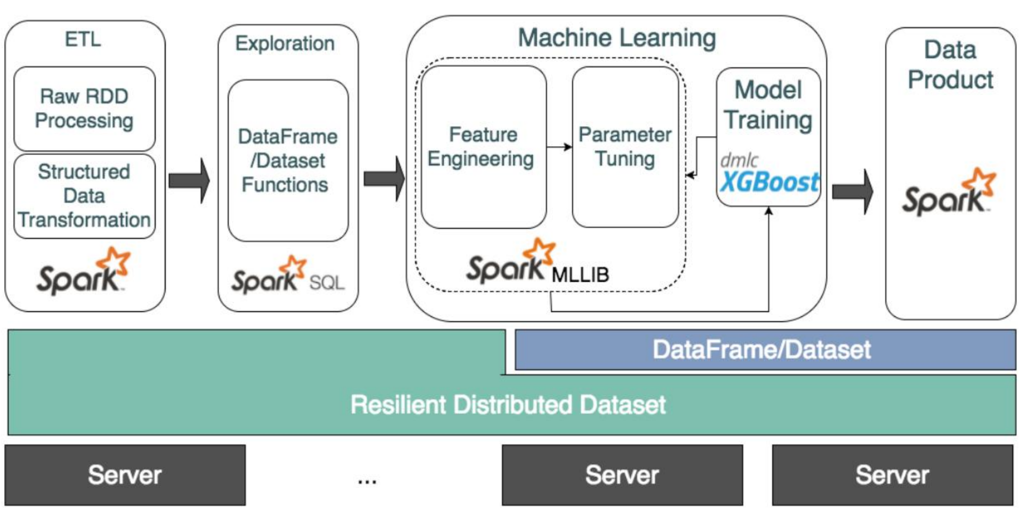
- **ML Algorithms:** These include common learning algorithms such as classification, regression, grouping and collaborative filtering.
- **Characterization:** Includes: extraction, transformation, dimensionality reduction and feature selection.
- **Pipelines:** are tools to build ML models in stages.
- **Persistence:** allows saving and loading algorithms, models y pipelines.
- **Utilities:** for linear algebra, statistics and data management.



Data Bootcamp
BEST DATA TRAINING

31

Proceso de Machine Learning



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

Data Bootcamp
BEST DATA TRAINING

32

Feature Engineering with Spark

The most commonly used data preprocessing techniques in Spark approaches are as follows

- VectorAssembler
- Grouping
- Scaling and normalization
- Work with categorical features
- Text data transformers
- Manipulation of functions
- PCA

Data Bootcamp
BEST DATA TRAINING

33

Feature Engineering with Spark

- **Vector Assembler:** It is basically used to concatenate all the features in a single vector that is can pass to estimator or ML algorithm
- **Grouping:** it is the simplest method to convert continuous variables into categorical variables. HE can be done with the Bucketizer class.
- **Scaling and normalization:** it is another common task in continuous variables. Allows data to have a normal distribution.
- **MinMaxScaler and StandardScaler—** Standardize features with zero mean and standard deviation of 1.
- **StringIndexer :** to convert categorical features to numeric.

```
+ ---- + ---- + ---- + ------------- +
| 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] |
+ ---- + ---- + ---- + ------------- +
```

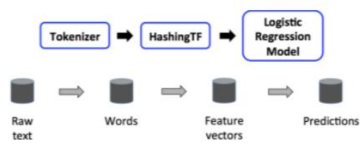
Data Bootcamp
BEST DATA TRAINING

34

Pipelines a PySpark

In the **Pipelines** (pipes) the different **stages of the** machine learning work can be grouped together as a single entity and can be thought of as a seamless workflow.

Each stage is a **Transformer .** They are executed 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

35

Apache Spark Koalas

Data Bootcamp
BEST DATA TRAINING

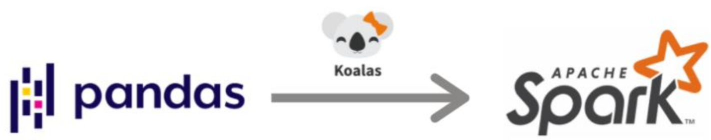
36

Introduction to Koalas

Koalas provide a **drop-in replacement for Pandas,** allowing for efficient scaling to hundreds of nodes for data science and machine learning.

Pandas does not scale to big data.

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



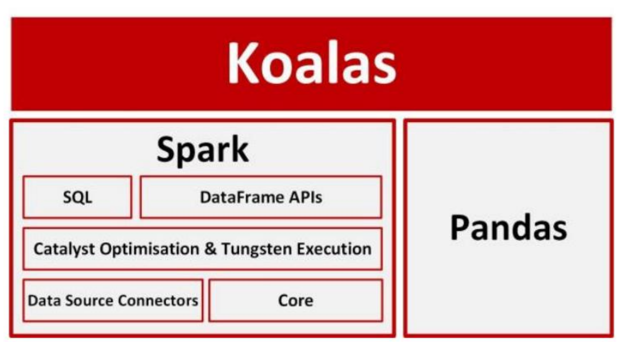
Data Bootcamp
BEST DATA TRAINING

37

Koalas y 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 **databases relational** and has no indexes.

Koalas translates pandas APIs to plan Spark SQL logic .



Data Bootcamp
BEST DATA TRAINING

38

Example: Feature Engineering with Koalas

In data science you often need the **pandas get_dummies( )** function to code categorical variables as (numeric) dummy variables.

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

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, in 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)
```

Spark Streaming

41

Spark Streaming Fundamentals

PySpark Streaming is a scalable, fault-tolerant system that follows the **RDD batch paradigm.**

It operates in batch intervals, receiving a **continuous stream of input data** from sources such as

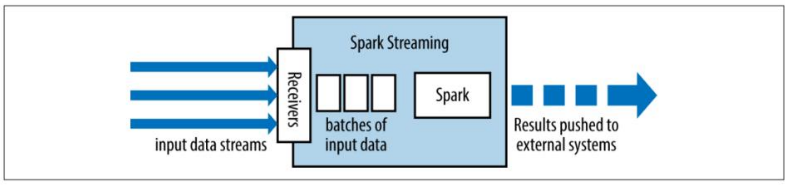
Apache Flume ,Kinesis, Kafka, sockets TCP, etc.

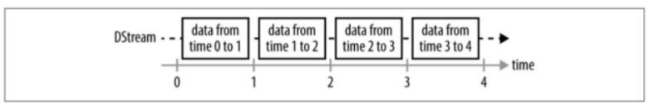
Spark Engine takes care of processing them.

42

How Spark Streaming works

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

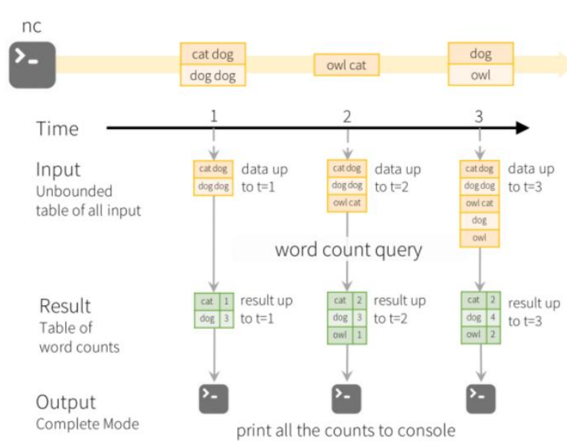




Data Bootcamp
BEST DATA TRAINING

43

Example: count words



Data Bootcamp
BEST DATA TRAINING

44

output modes

Spark uses several output modes to store the data:

- **Complete mode** *(Complete):* the entire table will be stored
- **Addition mode** *(Append):* only the new rows of the last process will be stored. Only for queries in which existing rows are not expected to change.
- **Update mode** *(Update):* Only the rows that have been updated since the last process will be stored. This mode only outputs the rows that have changed since the last process. If the query contains no aggregations, it is equivalent to append mode.

Complete, Append, Update

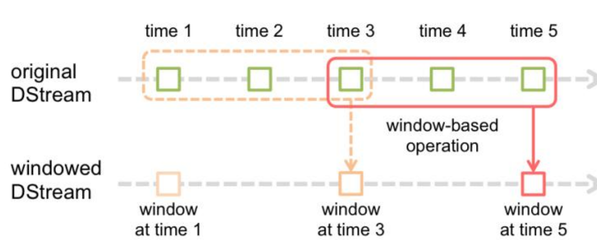
```
query = wordCounts \
    .writeStream \
    .outputMode("complete") \
    .format("console") \
    .start()
```


Types of transformations

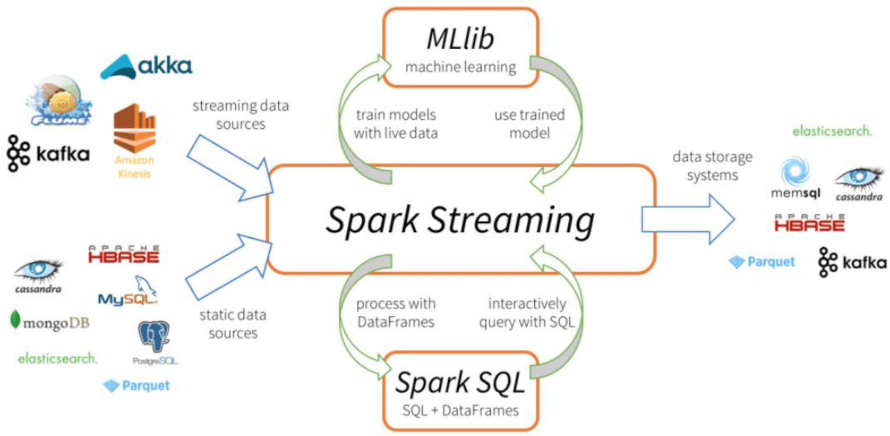
For **fault tolerance** the received data is copied to two nodes and there is also a mechanism called **checkpointing.**

The transformations can be grouped into:

- **stateless:** does not depend on the data from previous batches.
- **stateful:** use data from previous batches



Spark Streaming Capabilities



Data Bootcamp
BEST DATA TRAINING

47

Databricks

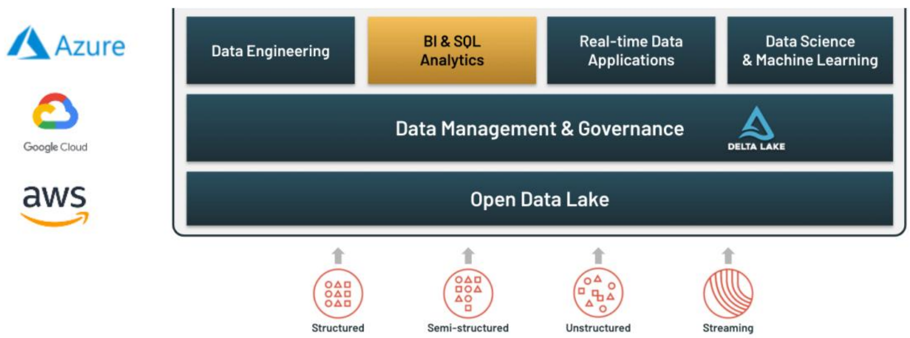
Data Bootcamp
BEST DATA TRAINING

48

Introduction to Databricks

Databricks is the Apache **Spark-** based **data analytics platform** developed by the forerunners of Spark. It allows advanced analytics, Big Data and ML in a **simple** and **collaborative way.**

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

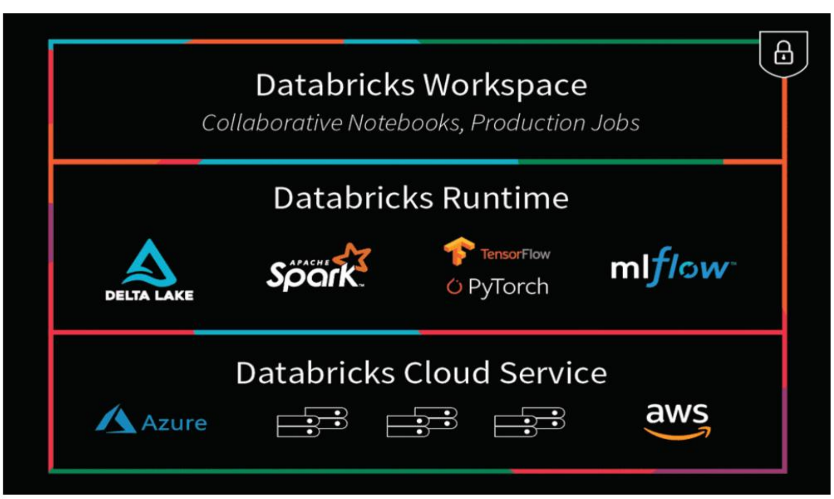


Data Bootcamp
BEST DATA TRAINING

49

Databricks features

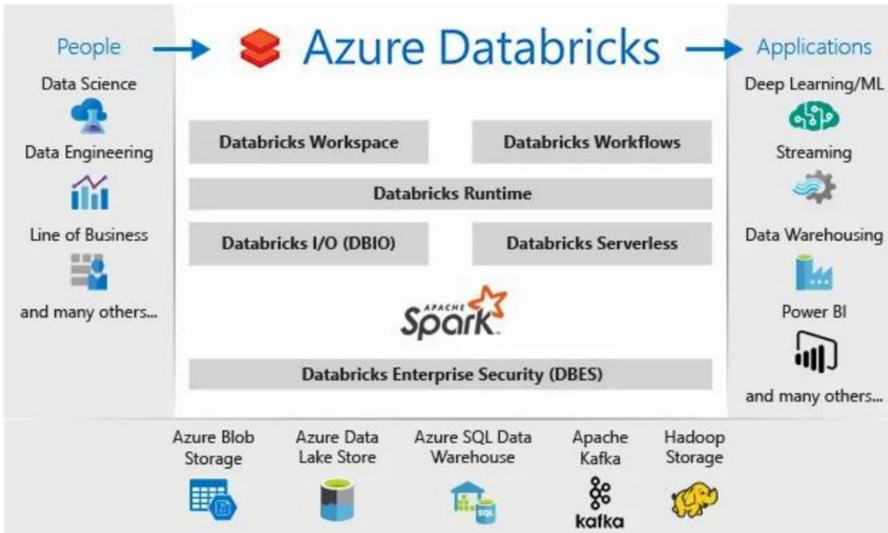
It allows easy **auto-scaling** and sizing **of Spark environments .** Facilitates deployments and speeds up installation and configuration of the environments.



Data Bootcamp
BEST DATA TRAINING

50

Databricks Architecture

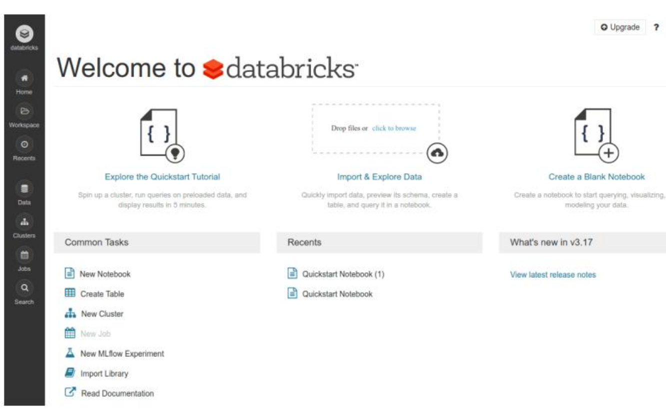


Data Bootcamp
BEST DATA TRAINING

51

Databricks Community

Databricks community is the **free version.** Allows you to use a **small cluster** with limited resources and **non-collaborative notebooks.** The paid version increases the capabilities.



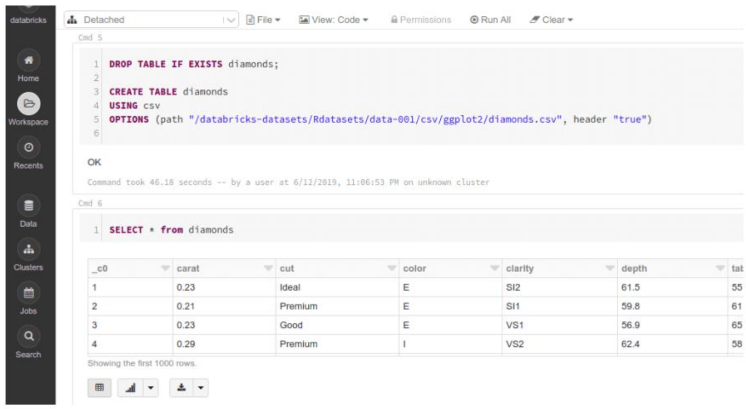
Data Bootcamp
BEST DATA TRAINING

52

Terminology

Important terms to know:

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



Data Bootcamp
BEST DATA TRAINING

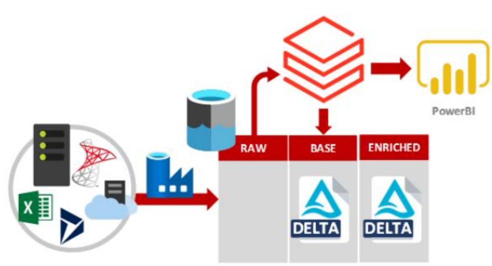
Delta Lake

Delta Lake is the open source **storage layer** developed for **Spark** and **Databricks.**

It provides **ACID transactions** and advanced **metadata management.**

It includes a Spark-compatible query engine that **speeds up operations** and improves the performance. Data stored in **Parquet format.**

DELTA LAKE



Data Bootcamp
BEST DATA TRAINING

Resources

Resources:

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