Documento generado de: chapter1f.pdf

# Página 1

Introduction to  
PySpark  
INTRODUCTION TO PYSPARK  
Benjamin Schmidt  
Data Engineer

# Página 2

INTRODUCTION TO PYSPARK  
Meet your instructor  
Almost a Decade of Data Experience with PySpark  
Used PySpark for Machine Learning, ETL tasks, and much more more  
Enthusiastic teacher of new tools for all!  
-



# Página 3

INTRODUCTION TO PYSPARK  
What is PySpark?  
Distributed data processing: Designed to handle large datasets across clusters  
Supports various data formats including CSV, Parquet, and JSON  
SQL integration allows querying of data using both Python and SQL syntax  
Optimized for speed at scale

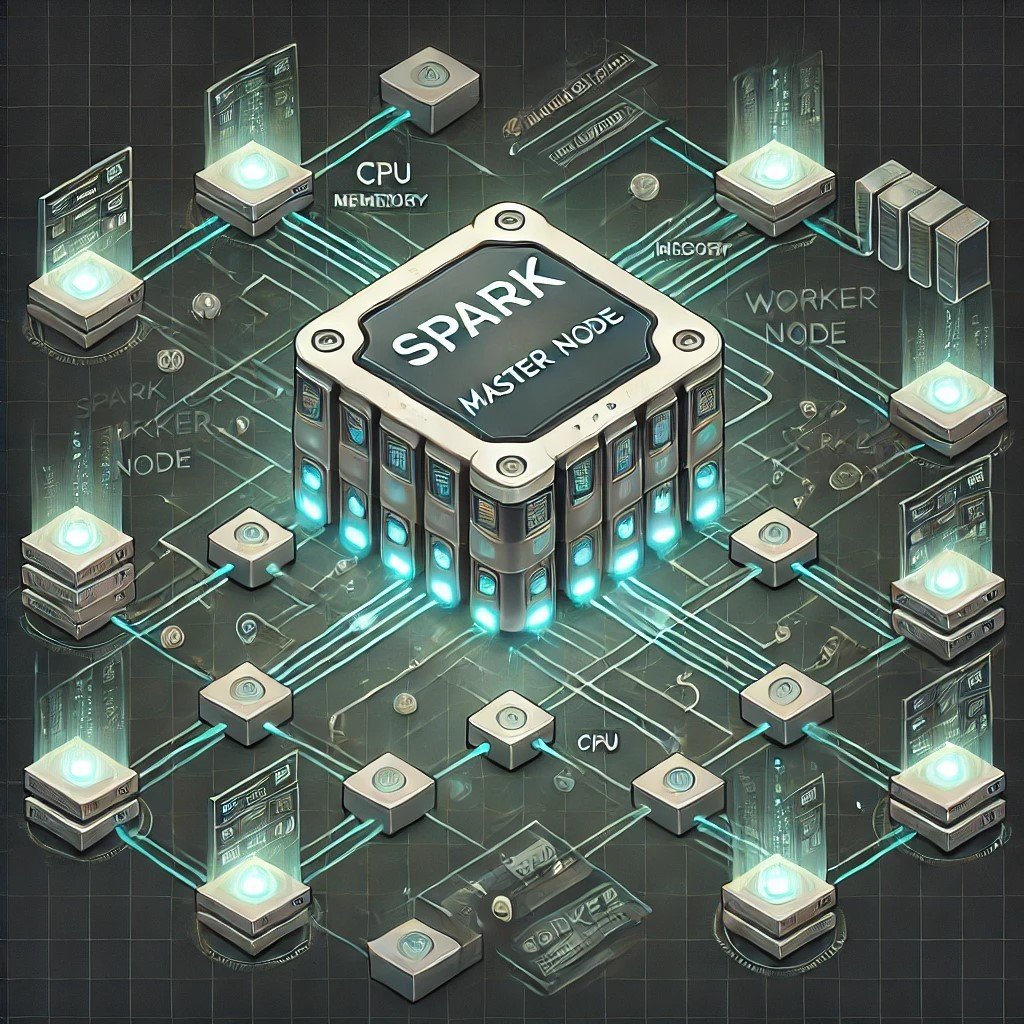


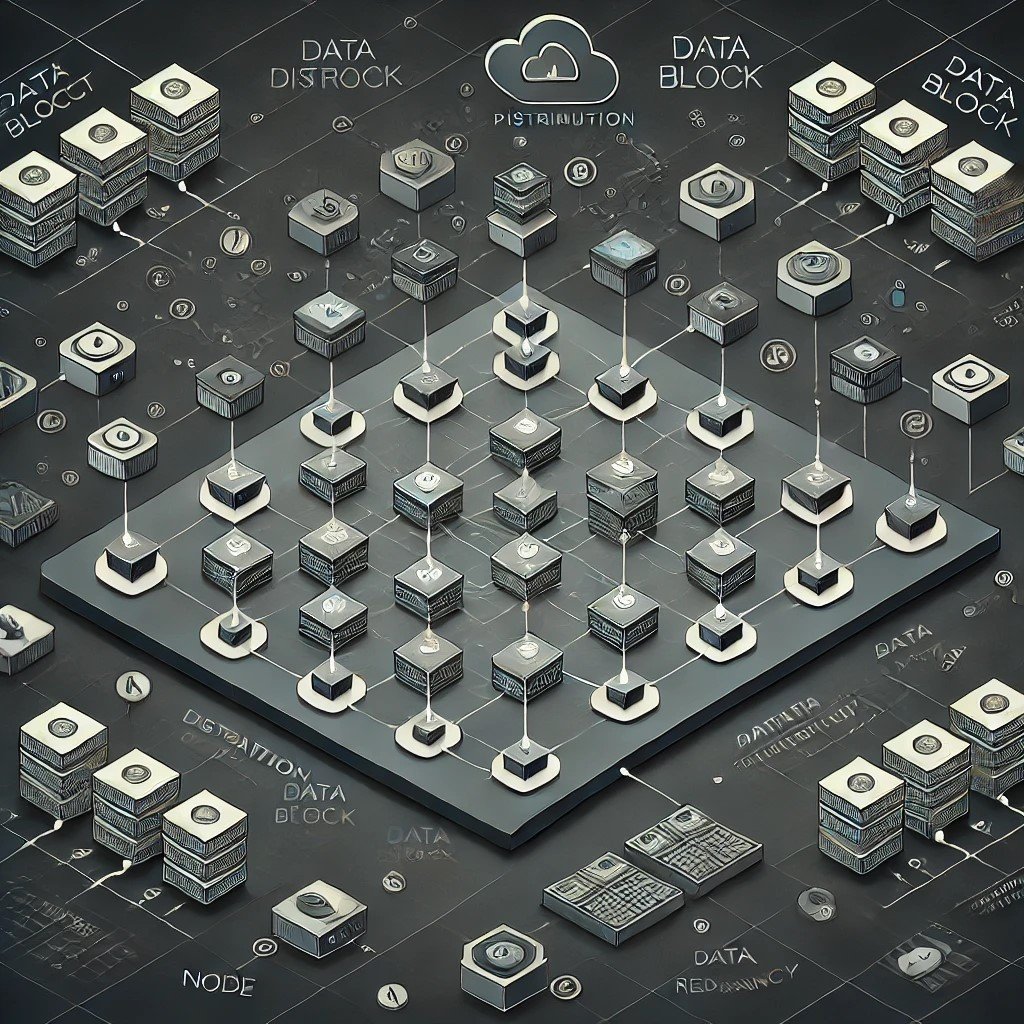
# Página 4

INTRODUCTION TO PYSPARK  
When would we use PySpark?  
Big data analytics  
Distributed data processing  
Real-time data streaming  
Machine learning on large datasets  
ETL and ELT pipelines  
Working with diverse data sources:  
1. CSV  
2. JSON  
3. Parquet  
4. Many Many More

# Página 5

INTRODUCTION TO PYSPARK  
Spark cluster  
Master Node  
Manages the cluster, coordinates tasks,  
and schedules jobs  
Worker Nodes  
Execute the tasks assigned by the master  
Responsible for executing the actual  
computations and storing data in memory  
or disk





i  
  
oe  
4 DISROCK  
  
@>  
  
PISTRIMUTION  
On  
  
© &

# Página 6

INTRODUCTION TO PYSPARK  
SparkSession  
SparkSessions allow you to access your Spark cluster and are critical for using PySpark.  
# Import SparkSession   
from pyspark.sql import SparkSession   
# Initialize a SparkSession   
spark = SparkSession.builder.appName("MySparkApp").getOrCreate()   
.builder() sets up a session  
getOrCreate() creates or retrieves a session  
.appName() helps manage multiple sessions

# Página 7

INTRODUCTION TO PYSPARK  
PySpark DataFrames  
Similar to other DataFrames but  
Optimized for PySpark  
# Import and initialize a Spark session   
from pyspark.sql import SparkSession   
spark = SparkSession.builder.appName("MySparkApp").getOrCreate()   
# Create a DataFrame   
census\_df = spark.read.csv("census.csv",   
 ["gender","age","zipcode","salary\_range\_usd","marriage\_status"])   
# Show the DataFrame   
census\_df.show()

# Página 8

Let's practice!  
INTRODUCTION TO PYSPARK

# Página 9

Introduction to  
PySpark  
DataFrames  
INTRODUCTION TO PYSPARK  
Benjamin Schmidt  
Data Engineer

# Página 10

INTRODUCTION TO PYSPARK  
About DataFrames  
DataFrames: Tabular format (rows/columns)  
Supports SQL-like operations  
Comparable to a Pandas Dataframe or a SQL TABLE  
Structured Data



Qawts  
  
oe

# Página 11

INTRODUCTION TO PYSPARK  
Creating DataFrames from filestores  
# Create a DataFrame from CSV   
census\_df = spark.read.csv('path/to/census.csv', header=True, inferSchema=True)

# Página 12

INTRODUCTION TO PYSPARK  
Printing the DataFrame  
# Show the first 5 rows of the DataFrame   
census\_df.show()   
 age education.num marital.status occupation income   
0 90 9 Widowed ? <=50K   
1 82 9 Widowed Exec-managerial <=50K   
2 66 10 Widowed ? <=50K   
3 54 4 Divorced Machine-op-inspct <=50K   
4 41 10 Separated Prof-specialty <=50K

# Página 13

INTRODUCTION TO PYSPARK  
Printing DataFrame Schema  
# Show the schema   
census\_df.printSchema()   
Output:   
root   
 |-- age: integer (nullable = true)   
 |-- education.num: integer (nullable = true)   
 |-- marital.status: string (nullable = true)   
 |-- occupation: string (nullable = true)   
 |-- income: string (nullable = true)

# Página 14

INTRODUCTION TO PYSPARK  
Basic analytics on PySpark DataFrames  
# .count() will return the total row numbers in the DataFrame   
row\_count = census\_df.count()   
print(f'Number of rows: {row\_count}')   
# groupby() allows the use of sql-like aggregations   
census\_df.groupBy('gender').agg({'salary\_usd': 'avg'}).show()   
Other aggregate functions are:  
sum()  
min()  
max()

# Página 15

INTRODUCTION TO PYSPARK  
Key functions for PySpark analytics  
.select() : Selects specific columns from the DataFrame  
.filter() : Filters rows based on specific conditions  
.groupBy() : Groups rows based on one or more columns  
.agg() : Applies aggregate functions to grouped data

# Página 16

INTRODUCTION TO PYSPARK  
Key Functions For Example  
# Using filter and select, we can narrow down our DataFrame   
filtered\_census\_df = census\_df.filter(df['age'] > 50).select('age', 'occupation')   
filtered\_census\_df.show()   
   
Output   
+---+------------------+   
|age| occupation |   
+---+------------------+   
| 90| ?|   
| 82| Exec-managerial|   
| 66| ?|   
| 54| Machine-op-inspct|   
+---+------------------+

# Página 17

Let's practice!  
INTRODUCTION TO PYSPARK

# Página 18

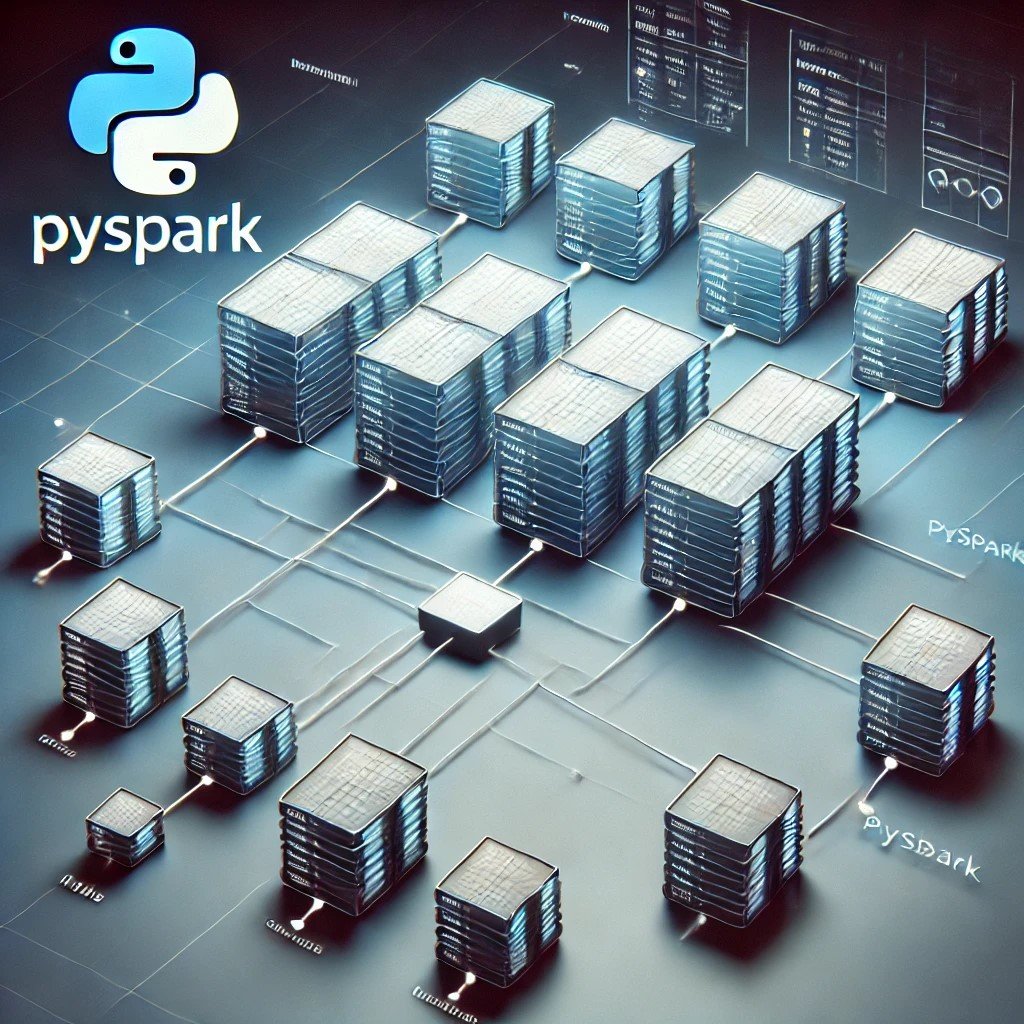
More on Spark  
DataFrames  
INTRODUCTION TO PYSPARK  
Benjamin Schmidt  
Data Engineer

# Página 19

INTRODUCTION TO PYSPARK  
Creating DataFrames from various data sources  
CSV Files: Common for structured,  
delimited data  
JSON Files: Semi-structured, hierarchical  
data format  
Parquet Files: Optimized for storage and  
querying, often used in data engineering  
Example:  
spark.read.csv("path/to/file.csv")   
Example:  
spark.read.json("path/to/file.json")   
Example:  
 https://spark.apache.org/docs/latest/api/python/reference/pyspark.pandas/api/pyspark.pandas.read\_csv  
spark.read.parquet("path/to/file.parquet")   
1

# Página 20

INTRODUCTION TO PYSPARK  
Schema inference and manual schema definition  
Spark can infer schemas from data with inferSchema=True  
Manually define schema for better control - useful for fixed data structures



sere  
  
SN  
Sos  
  
Me /i/t/.  
a  
  
ANAK’  
Di

# Página 21

INTRODUCTION TO PYSPARK  
DataTypes in PySpark DataFrames  
IntegerType : Whole numbers  
E.g., 1 , 3478 , -1890456  
LongType: Larger whole numbers  
E.g., 8-byte signed numbers, 922334775806  
FloatType and DoubleType: Floating-point numbers for decimal values  
E.g., 3.14159  
StringType: Used for text or string data  
E.g., "This is an example of a string."  
...

# Página 22

INTRODUCTION TO PYSPARK  
DataTypes Syntax for PySpark DataFrames  
# Import the necessary types as classes   
from pyspark.sql.types import (StructType,   
 StructField, IntegerType,   
 StringType, ArrayType)   
   
# Construct the schema   
schema = StructType([   
 StructField("id", IntegerType(), True),  
 StructField("name", StringType(), True),   
 StructField("scores", ArrayType(IntegerType()), True)   
])   
   
# Set the schema   
df = spark.createDataFrame(data, schema=schema)

# Página 23

INTRODUCTION TO PYSPARK  
DataFrame operations - selection and filtering  
Use .select() to choose specific columns  
Use .filter() or .where() to filter rows based on conditions  
Use .sort() to order by a collection of columns  
# Select and show only the name and age columns   
df.select("name", "age").show()   
# Filter on age > 30   
df.filter(df["age"] > 30).show()   
# Use Where to filter match a specific value   
df.where(df["age"] == 30).show()

# Página 24

INTRODUCTION TO PYSPARK  
Sorting and dropping missing values  
Order data using .sort() or .orderBy()  
Use na.drop() to remove rows with null values  
# Sort using the age column   
df.sort("age", ascending=False).show()   
# Drop missing values   
df.na.drop().show()

# Página 25

INTRODUCTION TO PYSPARK  
Cheatsheet  
spark.read\_json() : Load data from JSON  
spark.read.schema() : Define schemas explicitly  
.na.drop() : Drop rows with missing values  
.select() , .filter() , .sort() , .orderBy() : Basic data manipulation functions

# Página 26

Let's practice!  
INTRODUCTION TO PYSPARK