

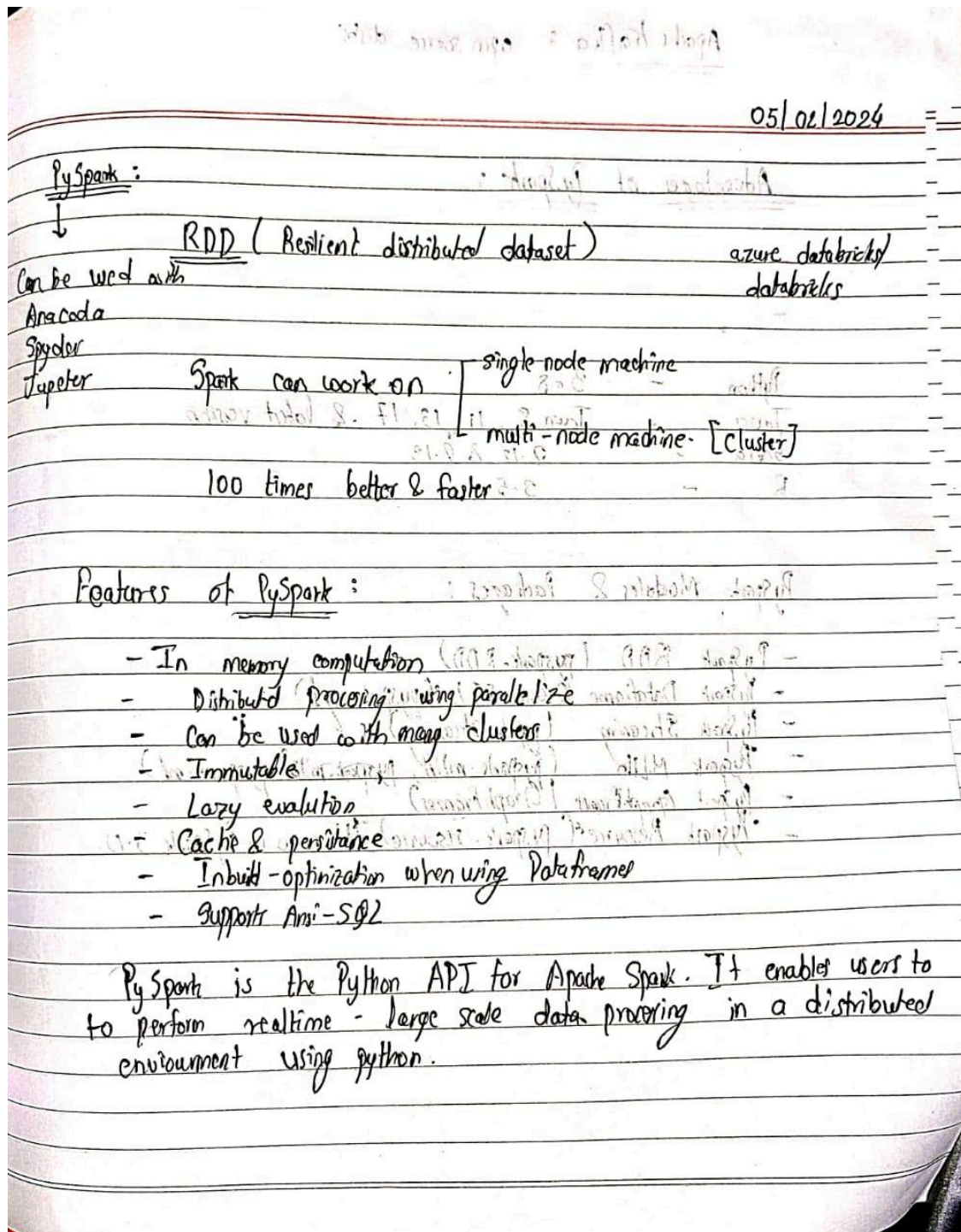
Name : Parth Nandedkar

Date : 05 Feb 2024

Topics : PySpark

Batch : Data Engineering Batch-1

Hand written notes during the session :



Apache Kafka : open source distri

distributed

Advantages of PySpark :

Python	-	3.8	no from con distri
Java	-	Java 8, 11, 13, 17 & latest version	
Scala	-	2.12 & 2.13	
R	-	3.5 latest & latest	with 001

PySpark Modules & Packages :

- PySpark RDD (pyspark.RDD)
- PySpark DataFrame (and SQL) (pyspark.sql)
- PySpark Streaming (pyspark.streaming)
- PySpark MLlib (pyspark.mllib, pyspark.ml, pyspark.ml)
- PySpark GraphFrames (GraphFrames)
- PySpark Resource (pyspark.resource). It's new in PySpark 3.0.

of new version 3.0 - Spark 3.0 is the first version of Spark that is distributed in a community-driven way. It is the first version of Spark that is distributed in a community-driven way.

mail → althina.21910516@
vif.ac.in
pass → Shaonmish@114

Pyspark (RDDs)

```
val filePath = "Path"
```

i) Create RDD →

```
spark = SparkSession \
    .builder \
    .appName("Python Spark Credit RDD example") \
    .config("spark
```

```
    .master("spark://localhost:7077") \
    .getOrCreate()

rdd = spark.sparkContext.parallelize(dataList)
```

Setting up Databricks :

As databricks is a paid platform we created an account with community version with signing up :

Create your Databricks account

1/2

Sign up with your work email to elevate your trial with expert assistance and more.

First name

Last name

Email

Company

Title

Phone (Optional)

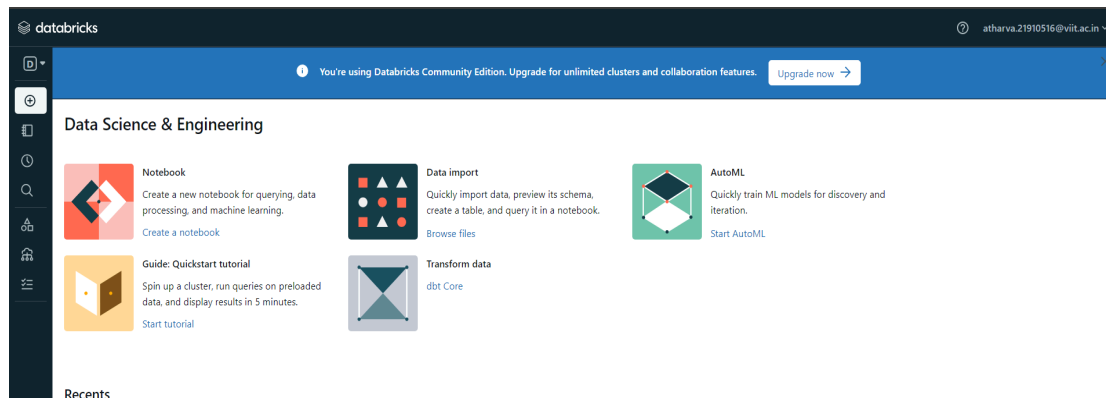
Country

India ▼

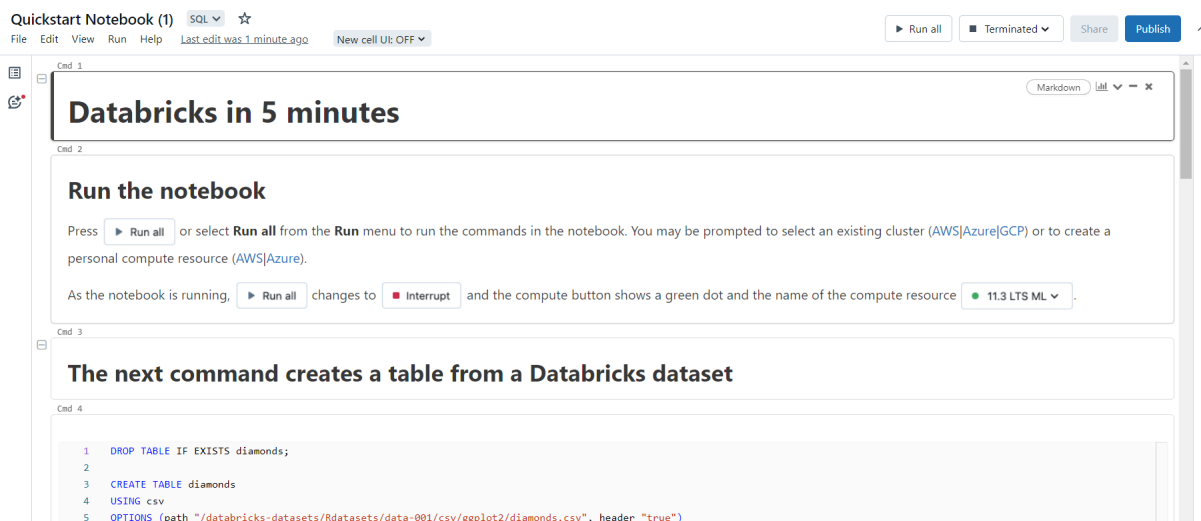
By submitting, I agree to the processing of my personal data by Databricks in accordance with our [Privacy Policy](#). I understand I can [update my preferences](#) at any time.

Continue

Dashboard after logging in :



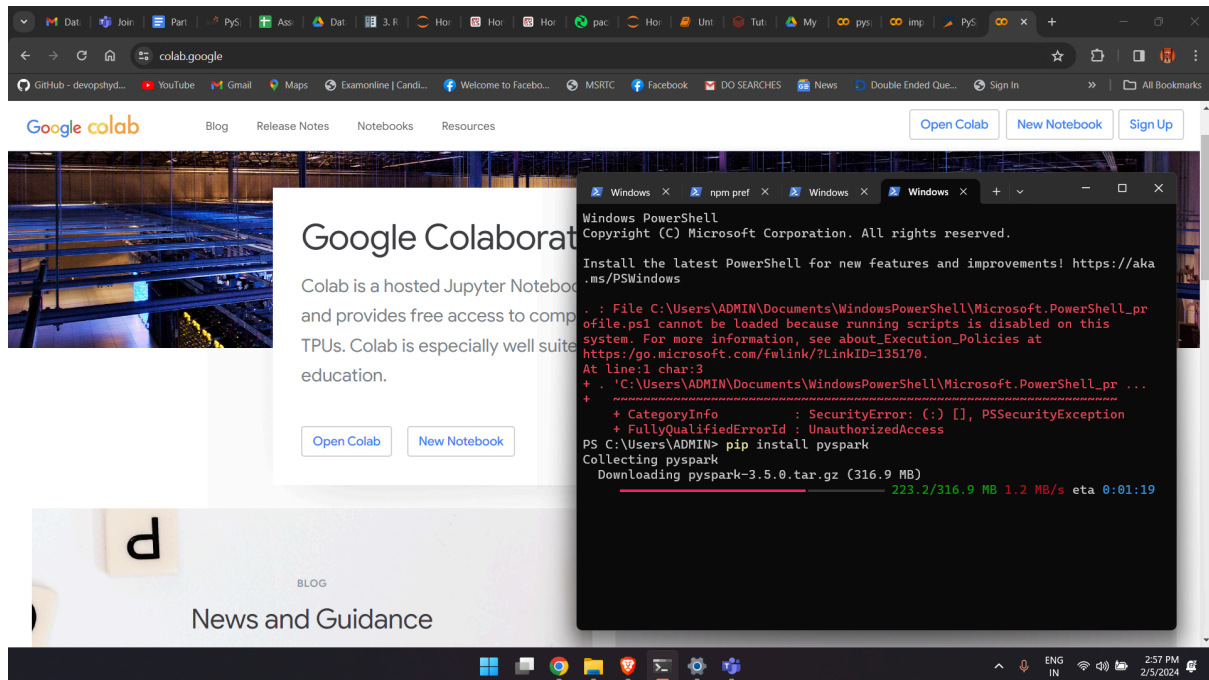
Quickstart program execution :



Installing PySpark in Local :

As there was no PySpark installed in my system using following command I installed PySpark.

`pip install PySpark`



Accessing CSV file using Pyspark :

```
[2]: import pyspark

[5]: from pyspark.sql import SparkSession

[6]: spark = SparkSession.builder.appName("Jupyter Notebook").getOrCreate()

[7]: spark

[7]: SparkSession - in-memory

SparkContext

Spark UI

Version      v3.5.0
Master       local[1]
AppName      SparkByExamples.com

[12]: df = spark.read.csv("E:\\Hexaware\\PySpark\\Demo2.csv")

[13]: df

[13]: DataFrame[_c0: string, _c1: string, _c2: string, _c3: string, _c4: string, _c5: string, _c6: string, _c7: string, _c8: string, _c9: string]

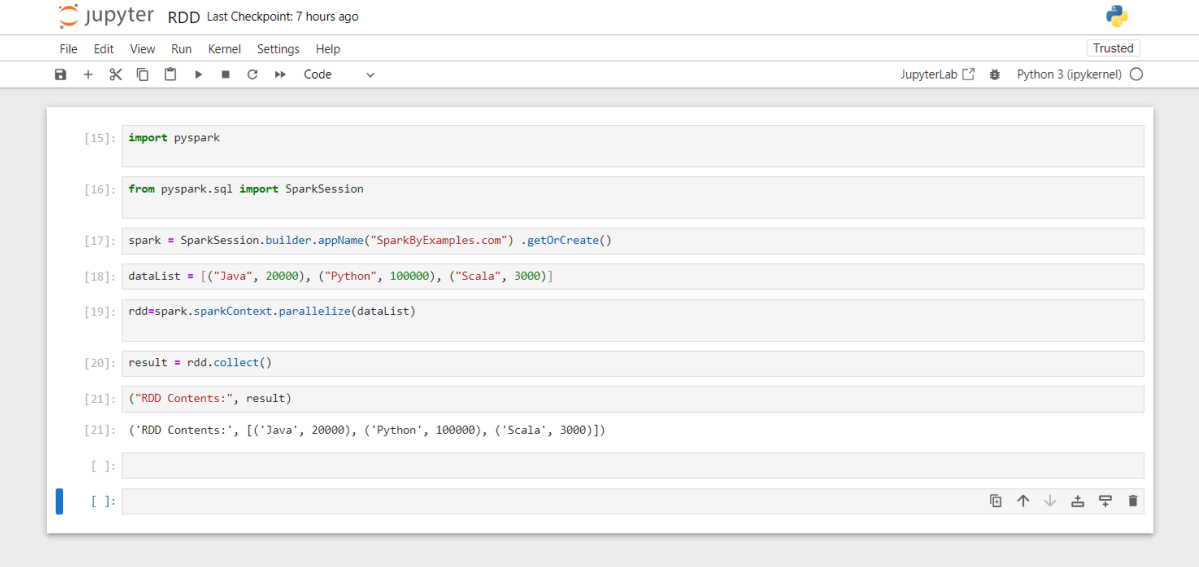
[14]: df.show()

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|_c0|_c1|_c2|_c3|_c4|_c5|_c6|_c7|_c8|_c9|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|Year|Industry_aggregat...|Industry_code_NZSIIOC|Industry_name_NZSIIOC|Units|Variable_code|Variable_name|Variable_category|Value|
|1|Industry code 4M7|I| | | | | | | | |
```

```
[14]: df.show()
```

	_c0	_c1	_c2	_c3	_c4	_c5	_c6	_c7	_c
8	Year Industry_aggregat...	Industry_code_NZSIOC	Industry_name_NZSIOC	Units Variable_code	Variable_name	Variable_category	Valu		
e Industry_code_ANZ...									
4 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H01	Total income Financial perform...	757,50			
0 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H04 Sales, government...	Financial perform...	674,89			
3 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H05 Interest, dividen...	Financial perform...	49,59			
0 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H07 Non-operating income	Financial perform...	33,02			
4 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H08 Total expenditure	Financial perform...	654,40			
8 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H09 Interest and dona...	Financial perform...	26,13			
1 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H10 Indirect taxes	Financial perform...	6,99			
1 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H11 Depreciation	Financial perform...	27,80			
0 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H12 Salaries and wage...	Financial perform...	123,62			
5 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H13 Redundancy and se...	Financial perform...	27			
5 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H14 Salaries and wage...	Financial perform...	2,08			
3 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H19 Purchases and oth...	Financial perform...	452,96			
6 ANZSIC06 division...	2021 Level 1	99999	All industries Dollars (millions)	H20 Non-operating exp...	Financial perform...	14,80			

Creating and Accessing RDD format using PySpark :



The screenshot displays a JupyterLab environment with a notebook titled "RDD". The interface includes a top menu bar with options like File, Edit, View, Run, Kernel, Settings, and Help. Below the menu is a toolbar with icons for file operations and code execution. The notebook area shows a series of code cells being executed. The code imports PySpark, creates a SparkSession, and defines an RDD with data for Java, Python, and Scala. The final output shows the contents of the RDD as a list of tuples.

```
[15]: import pyspark

[16]: from pyspark.sql import SparkSession

[17]: spark = SparkSession.builder.appName("SparkByExamples.com").getOrCreate()

[18]: dataList = [("Java", 20000), ("Python", 100000), ("Scala", 3000)]

[19]: rdd=spark.sparkContext.parallelize(dataList)

[20]: result = rdd.collect()

[21]: ("RDD Contents:", result)

[21]: ('RDD Contents:', [('Java', 20000), ('Python', 100000), ('Scala', 3000)])

[ ]:

[ ]:
```