Basics of Pyspark Programming for RDD on Jupyter notebook

ISHMEET KAUR · Follow
3 min read · May 18, 2020

SparkContext– represents the connection to a Spark cluster, and can be used to create RDDs, accumulators and broadcast variables on that cluster

```
jupyter  Basic examples of Pyspark Last Checkpoint: 2 hours ago (unsaved changes)
File  Edit  View  Insert  Cell  Kernel  Widgets  Help

In [1]: from pyspark.sql import SparkSession
        spark = SparkSession
                .builder.appName("Python Spark create RDD example")
                .config("spark.some.config.option", "some-value")
                .getOrCreate()
```

Usually, there are two popular ways to create the RDDs: loading an external dataset, or distributing a set of collection of objects. The following examples show some simplest ways to create RDDs by using parallelize() function which takes an already existing collection in your program and pass the same to the Spark Context. Then you will get RDD data.

```
myData = spark.sparkContext.parallelize([(1,2), (3,4), (5,6), (7,8), (9,10)])

myData.collect()
[(1, 2), (3, 4), (5, 6), (7, 8), (9, 10)]
```

a. To Create Dataframe of RDD dataset:

1. With the help of toDF() function in parallelize function.

show() will display in the form of dataframe

collect() will display RDD in the list form for each row

```
In [15]: df = spark.sparkContext.parallelize([(1, 2, 3, 'a b c'),(4, 5, 6, 'd e f'),(7, 8, 9, 'g h i')])
         .toDF(['col1', 'col2', 'col3','col4'])
         df.show()
         df.collect()

+----+----+----+-----+
|col1|col2|col3| col4|
+----+----+----+-----+
|   1|   2|   3|a b c|
|   4|   5|   6|d e f|
|   7|   8|   9|g h i|
+----+----+----+-----+

Out[15]: [Row(col1=1, col2=2, col3=3, col4='a b c'),
          Row(col1=4, col2=5, col3=6, col4='d e f'),
          Row(col1=7, col2=8, col3=9, col4='g h i')]
```

2. With createDataFrame() implicit call both arguments: RDD dataset can be represented in structured dataset with proper schema declared in the second argument of createDataFrame() of spark session.

```
Employee = spark.createDataFrame([
('1', 'Joe', '70000', '1'),
('2', 'Henry', '80000', '2'),
('3', 'Sam', '60000', '2'),
('4', 'Max', '90000', '1')],
['Id', 'Name', 'Sallary','DepartmentId']
)

Employee.show()

+---+-----+-------+------------+
| Id| Name|Sallary|DepartmentId|
+---+-----+-------+------------+
|  1|  Joe|  70000|           1|
|  2|Henry|  80000|           2|
|  3|  Sam|  60000|           2|
|  4|  Max|  90000|           1|
+---+-----+-------+------------+
```

Collect() will show RDD Row format.

```
Employee.collect()

[Row(Id='1', Name='Joe', Sallary='70000', DepartmentId='1'),
 Row(Id='2', Name='Henry', Sallary='80000', DepartmentId='2'),
 Row(Id='3', Name='Sam', Sallary='60000', DepartmentId='2'),
 Row(Id='4', Name='Max', Sallary='90000', DepartmentId='1')]
```

3. With createDataFrame() explicitly call both the arguments: my_list is having the list of values which will be set as rows and col_name is carrying the list of column names values.

```
my_list = [['a', 1, 2], ['b', 2, 3],['c', 3, 4]]
col_name = ['A', 'B', 'C']

spark.createDataFrame(my_list, col_name).show()

+---+---+---+
|  A|  B|  C|
+---+---+---+
|  a|  1|  2|
|  b|  2|  3|
|  c|  3|  4|
+---+---+---+
```

4. With createDataFrame() explicitly calls one of the argument: my_list dataset is called explicitly and column names are declared implicitly.

head()-displays first row of the RDD dataset.

show()-displays the dataframe

```
]: my_list = [['male', 1, None], ['female', 2, 3],['male', 3, 4]]

]: ds = spark.createDataFrame(my_list, ['A', 'B', 'C'])
   ds.head()
]: Row(A='male', B=1, C=None)

]: ds.show()

+------+---+----+
|     A|  B|   C|
+------+---+----+
|  male|  1|null|
|female|  2|   3|
|  male|  3|   4|
+------+---+----+
```

5. createDataFrame() with numpy() array example-When dataset is in keys and values format.

```
d = {'A': [0, 1, 0],
     'B': [1, 0, 1],
     'C': [1, 0, 0]}

import numpy as np

spark.createDataFrame(np.array(list(d.values())).T.tolist(),list(d.keys())).show()

+---+---+---+
|  A|  B|  C|
+---+---+---+
|  0|  1|  1|
|  1|  0|  0|
|  0|  1|  0|
+---+---+---+
```

b. fillna(value)-replace null value to the value provided in fillna() argument.

```
ds.fillna(-99).show()

+------+---+---+
|     A|  B|  C|
+------+---+---+
|  male|  1|-99|
|female|  2|  3|
|  male|  3|  4|
+------+---+---+
```

c. replace(listOfRowValues, listOfRowvaluesToReplace)-replacing row listed values with the list of values to replace with.

```
#caution: Mixed type replacements are not supported
ds.replace(['male','female'],['1','0']).show()

+---+---+----+
|  A|  B|   C|
+---+---+----+
|  1|  1|null|
|  0|  2|   3|
|  1|  3|   4|
+---+---+----+
```

d. toDF(listOfColvaluesToReplace)-To replace all the column names of dataframe with the new set of column names in the dataframes

e. To rename a single column withColumnRenamed()

f. More than one column needs to be renamed with mapping.get(col,col) in for loop and dataframe is created with help of toDF(*new_names)

g. drop(*drop_name)-When column needs to be dropped drop from the dataframe

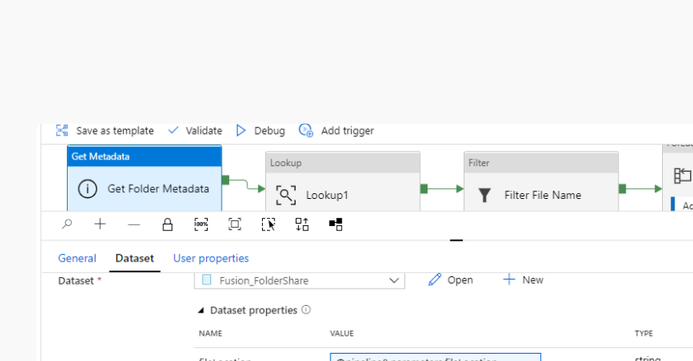
Pyspark · Rdd · Spark Rdd · Lists · Jupyter Notebook

Written by ISHMEET KAUR

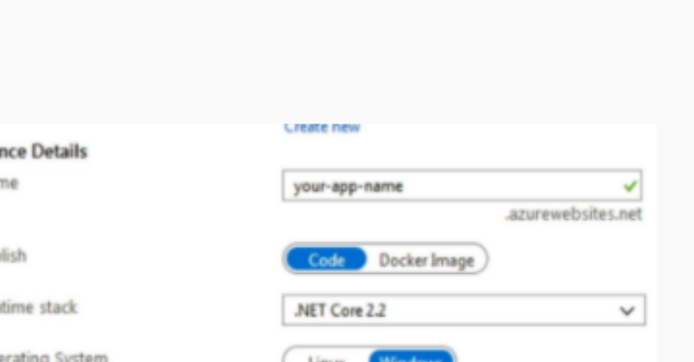
8 Followers

Follow

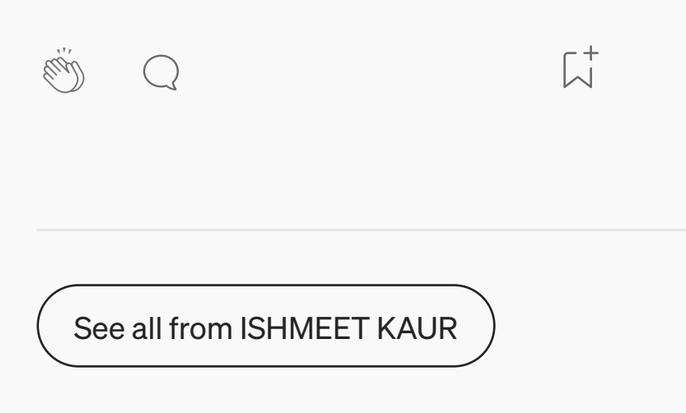
More from ISHMEET KAUR



ISHMEET KAUR
Hadoop ecosystem
Apache Hadoop is an open source framework for storing and processing large scale data,...
10 min read · May 25, 2020



ISHMEET KAUR
Azure DevOps CI/CD for Data Platform-Azure Databricks
For gaining the maximum value of data products, they should be delivered in a timel...
7 min read · Jan 30, 2021
1

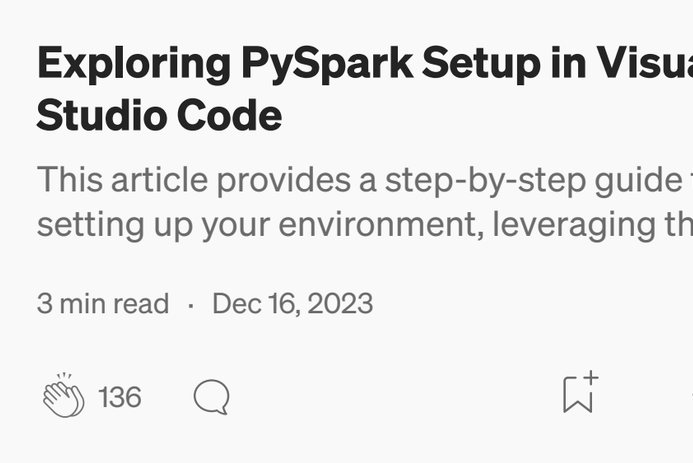


ISHMEET KAUR
Copying unique zip filenames to Azure BLOB through Azure Data...
Copy the zip files from SFTP which are not copied in the past and making sure zip file...
4 min read · May 14, 2020

ISHMEET KAUR
What is Azure App Service Plan?
Azure App service(in short): Azure PaaS service is the platform that handles...
4 min read · Nov 18, 2020

See all from ISHMEET KAUR

Recommended from Medium



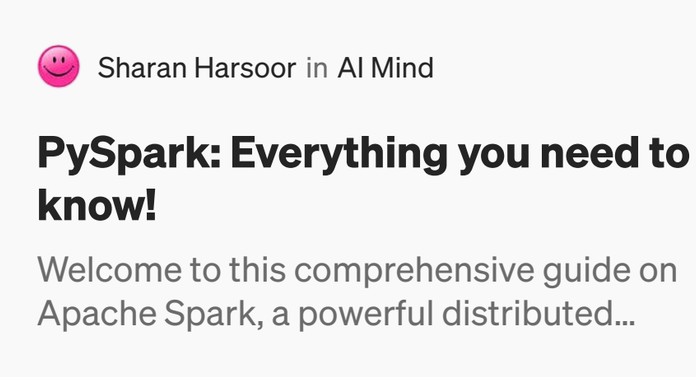
Nidhi Gupta
Exploring PySpark Setup in Visual Studio Code
This article provides a step-by-step guide to setting up your environment, leveraging the...
3 min read · Dec 16, 2023
136

Ritesh Bakliwal
PySpark: A Powerful Tool in the Finance Industry
Introduction to PySpark
3 min read · Nov 25, 2023
8

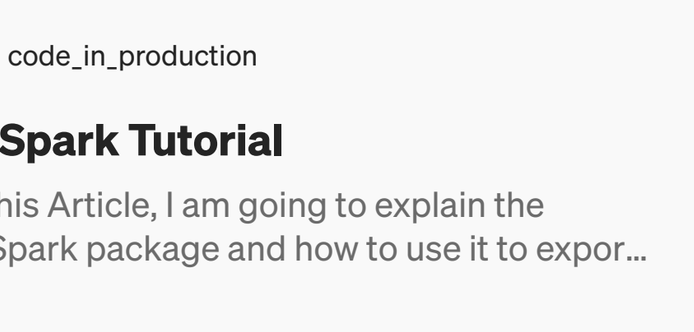
Lists

Staff Picks
606 stories · 859 saves

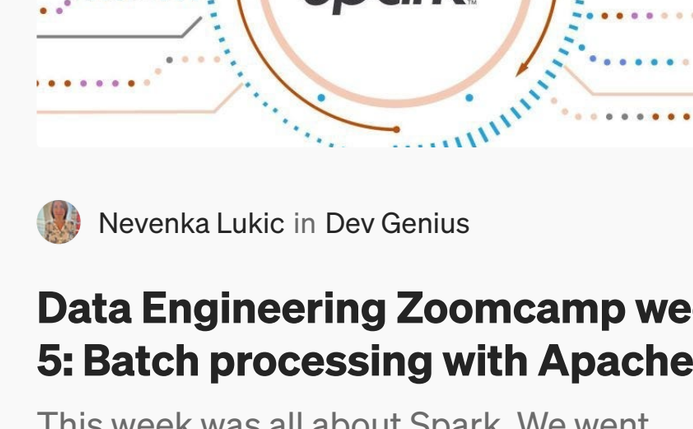
Natural Language Processing
1320 stories · 811 saves



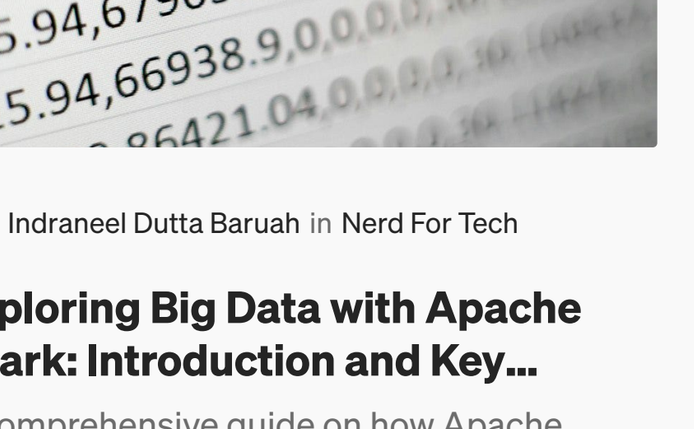
Sharan Harsoor in AI Mind
PySpark: Everything you need to know!
Welcome to this comprehensive guide on Apache Spark, a powerful distributed...
22 min read · Oct 27, 2023
108



code_in_production
PySpark Tutorial
In this Article, I am going to explain the PySpark package and how to use it to expor...
3 min read · Sep 28, 2023



Nevenka Lukic in Dev Genius
Data Engineering Zoomcamp week 5: Batch processing with Apache...
This week was all about Spark. We went through the installation, explored the batch...
7 min read · Feb 27, 2024
11



Indraneel Dutta Baruah in Nerd For Tech
Exploring Big Data with Apache Spark: Introduction and Key...
A comprehensive guide on how Apache Spark works and how to use it efficiently!
12 min read · Dec 16, 2023
10

See more recommendations