Introduccion a PySpark

December 17, 2024

Diego Godinez Bravo

Centro de Investigación en Matemáticas

Maestría en Cómputo Estadístico

1 PySpark

1.1 PySpark Dataframe

```
[141]: import pyspark
       import pandas as pd
       from pyspark.sql import SparkSession
       from pyspark.sql import functions as F
       from pyspark.sql.functions import col, when, count # load libraries
 [5]: spark = SparkSession.builder
               .appName("Intro to Spark Dataframes")\
               .getOrCreate() # create a Spark session
[10]: spark # spark session I've created
[10]: <pyspark.sql.session.SparkSession at 0x75e1f83aa000>
[19]: path = '/home/aspphem/Desktop/Statistics-with-Python/Cartwheeldata.csv' # file_
       df = spark.read.csv(path, header = True) # read a csv file
[23]: type(df) # pyspark dataframe object
[23]: pyspark.sql.dataframe.DataFrame
[20]: df.printSchema() # print out the schema in tree format
      root
       |-- ID: string (nullable = true)
       |-- Age: string (nullable = true)
       |-- Gender: string (nullable = true)
       |-- GenderGroup: string (nullable = true)
```

```
|-- Glasses: string (nullable = true)
      |-- GlassesGroup: string (nullable = true)
      |-- Height: string (nullable = true)
      |-- Wingspan: string (nullable = true)
      |-- CWDistance: string (nullable = true)
      |-- Complete: string (nullable = true)
      |-- CompleteGroup: string (nullable = true)
      |-- Score: string (nullable = true)
[21]: df = spark.read.option('header', 'true').csv(path, inferSchema = True) #__
       →overwrite existing data frame and add inferSchema attribute
[22]: df.printSchema() # print out the schema in tree format
     root
      |-- ID: integer (nullable = true)
      |-- Age: integer (nullable = true)
      |-- Gender: string (nullable = true)
      |-- GenderGroup: integer (nullable = true)
      |-- Glasses: string (nullable = true)
      |-- GlassesGroup: integer (nullable = true)
      |-- Height: double (nullable = true)
      |-- Wingspan: double (nullable = true)
      |-- CWDistance: integer (nullable = true)
      |-- Complete: string (nullable = true)
      |-- CompleteGroup: integer (nullable = true)
      |-- Score: integer (nullable = true)
[64]: print("Dataframe shape: {} rows, {} columns".format(df.count(), len(df.
       ⇔columns))) # dataframe dimension
     Dataframe shape: 25 rows, 12 columns
[13]: df.columns # columns names
[13]: ['ID',
       'Age',
       'Gender',
       'GenderGroup',
       'Glasses',
       'GlassesGroup',
       'Height',
       'Wingspan',
       'CWDistance',
       'Complete',
       'CompleteGroup',
       'Score'l
```

[118]: df.show(5) # preview data

----+

 $\label{lem:comp} $$| ID|Age|Gender|GenderGroup|Glasses|GlassesGroup|Height|Wingspan|CWDistance|Complete|CompleteGroup|Score| $$| CompleteGroup|Score| $| CompleteGroup|Score| $$| CompleteGroup|Score| $| CompleteGroup|Score| $| CompleteGroup|Score| $| CompleteGroup|Score| $| CompleteGroup|Score| $| CompleteGroup|Score| $| CompleteGroup|Score| $$

+	++	+		+	+	+-	+	+	
Y 1 7 2 26 F 1 Y 1 62.0 60.0 70 Y 1 8 3 33 F 1 Y 1 66.0 64.0 85 Y 1 7	+		-+	+					
2 26 F 1 Y 1 62.0 60.0 70 Y 1 8	1 56	Fl		1	Υļ	1	62.0	61.0	79
Y 1 8 3 33 F 1 Y 1 66.0 64.0 85 Y 1 7	Υļ	1	7						
3 33 F 1 Y 1 66.0 64.0 85 Y 1 7	2 26	Fl		1	Υļ	1	62.0	60.0	70
Y 1 7	Υļ	1	8						
	3 33	Fl		1	Υļ	1	66.0	64.0	85
4 20 E 1 N 0 64 0 62 0 97	Υ	1	7						
4 39 F 1 N 0 04.0 03.0 07	4 39	Fl		1	N I	0	64.0	63.0	87
Y 1 10	Υ	1	10						
5 27 M 2 N 0 73.0 75.0 72	5 27	M		2	NI	0	73.0	75.0	72
N 0 4	N I	0	4						
+++	++	+-		+	+	+	+	+	

----+

only showing top 5 rows

[172]: updated_df = df.select('Gender', 'Height', 'CWDistance', 'Score')
 updated_df.show(5) # selecting columns

+----+ |Gender|Height|CWDistance|Score| +----+ F| 62.0| 79| 7| FΙ 62.0| 70| 81 F| 66.0| 85| 7| F| 64.0| 87| 101 M| 73.0| 72| 4| +----+

only showing top 5 rows

[173]: updated_df.describe().show() # dataframe summary

++		+		+
summary		_	CWDistance	Score
count			25	 25
mean	NULL	67.65	82.48	6.4
stddev	NULL	4.431186823715139	15.058552387264852	2.5331140255951103
min	F	61.5	631	2
max	M	75.0	115	10
++		+		+

```
+----+
    |Gender|Height|CWDistance|Score|Height > 67|
    +----+
         FI 62.01
                     79|
                           7|
                                 falsel
         F| 62.0|
                     70|
                          8|
                                 false
         F| 66.0|
                     85 l
                          7|
                                 falsel
         F| 64.0|
                     87 |
                         10|
                                 falsel
                     72|
         M| 73.0|
                           4|
                                  true
    +----+
    only showing top 5 rows
[175]: updated_df = updated_df.drop('Score')
     updated_df.show(5) # drop the columns
    +----+
     |Gender|Height|CWDistance|Height > 67|
         F| 62.0|
                     79|
                            falsel
         F| 62.0|
                     70|
                            false
         F| 66.0|
                     85 l
                            false
         F| 64.0|
                     87 |
                            false
         M| 73.0|
                     72|
                            true
    +----+
    only showing top 5 rows
[176]: updated_df = updated_df.withColumnRenamed("CWDistance", "CartwheelDistance")
     updated_df.show(5) # rename columns
    +----+
    |Gender|Height|CartwheelDistance|Height > 67|
    +----+
         F| 62.0|
                           79|
                                  false
         FI 62.01
                           70 l
                                 falsel
         FI 66.01
                           85 l
                                  false
         F| 64.0|
                           87|
                                  false
         M| 73.0|
                           72|
                                  true
    only showing top 5 rows
```

[174]: | updated_df = updated_df.withColumn('Height > 67', updated_df['Height'] >= 67)

updated_df.show(5) # adding columns in data frame

1.1.1 Filter Operations

[177]: df.show(5) # preview data

+ ID Age Ge				اعمودا 12ءمم	esGroupli	leight Wi	nggnan l Cuin-	istanco
lete Comple			_	ιρρερ Ι αταρρ	cear out 11	10181101M1	.rrgshamlown.	recance
++		-		+	+-		+	
+		-+						
1 56	Fl		1	Υļ	1	62.0	61.0	79
Υļ	1	7						
2 26	F		1	Υļ	1	62.0	60.0	70
Υ	1	8						
3 33	Fl		1	Υļ	1	66.0	64.0	85
Υ	1	7		1				
4 39	F	401	1	N	0	64.0	63.0	87
Y	1	10	0.1	27.1	0.1	70.01	75 01	70
5 27	M	4.1	2	N	01	73.0	75.0	72
N ++	0	4						
only showin	g cop	5 TOWS						
df.agg(F.m	ean('CV	/Distano	ce')).c	ollect()[0]][0] # me	an value	of cartwhe	eel dis
df.agg(F.ma	ean('CV	/Distano	ce')).c	ollect()[0][0] # me	an value	of cartwhe	eel dis
82.48								
82.48								
82.48 df.filter('CWDist	ance<82	2').sho	w(5) # car	twheel di	stance l	ess than th	ie mean _l
82.48 df.filter(ovalue	'CWDist	sance<82	2').sho	w(5) # car	twheel di	stance l	ess than th	ie mean _l
82.48 df.filter(>value ++	'CWDist	ance<82	2').sho +	w(5) # car	twheel di	stance l	ess than th	ie mean _l
82.48 df.filter(ovalue ++ ID Age Ge lete Comple	'CWDist	cance<82	2').sho + + oup Gla	w(5) # car +	twheel di	stance l + Height Wi	ess than th	ne mean
82.48 df.filter('CWDist	cance<8: -+ enderGr p Score	2').sho + + oup Gla	w(5) # car +	twheel di	stance l + Height Wi	ess than th	ne mean
82.48 df.filter('CWDist	cance<8: -+ enderGr p Score	2').sho + + oup Gla +	w(5) # car + asses Glass	twheel di +- esGroup H	stance l	ess than th	ne mean
82.48 df.filter('CWDist	cance<82	2').sho + + oup Gla +	w(5) # car +	twheel di +- esGroup H	stance l	ess than th	ne mean
82.48 df.filter('CWDist	cance<8: -+ enderGr p Score	2').sho + + oup Gla + + 1	w(5) # car	twheel di +- esGroup F +-	stance l+ Height Wi	ess than th	istance
82.48 df.filter(value	'CWDist	cance<82 enderGr p Score 7	2').sho + + oup Gla +	w(5) # car + asses Glass	twheel di +- esGroup F +-	stance l+ Height Wi	ess than th	ne mean
82.48 df.filter(CWDist	cance<82	2').sho+ + oup Gla + + 1	w(5) # car+ asses Glass+ Y Y	twheel di+ esGroup H+	stance l+ Height Wi+ 62.0 62.0	ess than the	ne meanuistance
82.48 df.filter('CWDist	cance<82	2').sho + + oup Gla + + 1	w(5) # car	twheel di+ esGroup H+	stance l+ Height Wi	ess than the	istance
82.48 df.filter('CWDist	cance<82 enderGr p Score 7	2').sho+ + oup Gla + + 1 1 2	w(5) # car + asses Glass+ Y Y N	twheel di+ esGroup H+ 1 1 0	stance l+- Height Wi+- 62.0 62.0 73.0	ess than the	ne mean _t istance 79 70 72
82.48 df.filter(CWDist	7 8 4	2').sho+ + oup Gla + + 1	w(5) # car+ asses Glass+ Y Y	twheel di+ esGroup H+ 1 1 0	stance l+- Height Wi+- 62.0 62.0 73.0	ess than the	ne mean _t istance 79 70 72
82.48 df.filter('CWDist	cance<82	2').sho+ + oup Gla + + 1 1 2 2	w(5) # car + usses Glass+ Y Y N N	twheel di+ esGroup F+ 1 0 0	stance l+- Meight Wi+- 62.0 62.0 73.0 75.0	ess than the	ne meanuistance 79 70 72 81
82.48 df.filter('CWDist	7 8 4	2').sho+ + oup Gla + + 1 1 2 2	w(5) # car + asses Glass+ Y Y N	twheel di+ esGroup F+ 1 0 0	stance l+- Meight Wi+- 62.0 62.0 73.0 75.0	ess than the	ne meanuistance 79 70 72 81

```
only showing top 5 rows
[188]: df.filter(df['CWDistance']<82).select(['Age', 'Gender', 'CWDistance', 'Score']).
       ⇒show(5) # select specific columns
     +---+
     |Age|Gender|CWDistance|Score|
     | 56|
              FΙ
                       79|
                             71
     | 26|
              FΙ
                       701
                             81
     | 27|
                       72|
                             41
              Μl
     | 24|
              Μl
                       81|
                             31
     1 331
              FΙ
                       65|
                             81
     +---+
     only showing top 5 rows
[187]: df.filter((df['CWDistance']<82) &
               (df['Age'] <= 30)).select(['Age', 'Gender', 'CWDistance', 'Score']).</pre>
       →show(5) # combine two specific conditions
     +---+
     |Age|Gender|CWDistance|Score|
     +---+
     | 26|
              FΙ
                       701
                             81
     | 27|
              Μl
                       72|
                             41
     | 24|
              Μl
                       81|
                             31
     | 28|
              FΙ
                       79|
                            10|
     | 23|
              FΙ
                       661
                             41
     +---+
     only showing top 5 rows
[190]: df.filter(~(df['CWDistance']<82)).select(['Age', 'Gender', 'CWDistance', u
       →'Score']).show(5) # ~ not operator; anything that is greater than the mean_
       →value will be given
     +---+
     |Age|Gender|CWDistance|Score|
     +---+
     | 33|
              FΙ
                       85|
                             7|
     | 39|
              FΙ
                       87|
                            10|
     | 28|
              Μl
                      107|
                            10|
              FΙ
     | 22|
                       98|
                             91
     | 29|
              Μl
                             51
                      106
     +---+
     only showing top 5 rows
```

----+

1.2 PySpark Handling Missing Values

```
[156]: updated_df = df.select('Gender', 'Height', 'CWDistance', 'Score')
[157]: updated_df.select([count(when(col(c).isNull(), c)).alias(c) for c in updated_df.
       ⇔columns]).show() # check for NULL values
     +----+
     |Gender|Height|CWDistance|Score|
     +----+
         01
                0|
                         0|
     +----+
[158]: updated_df = updated_df.replace({'F': None}, subset = ['Gender'])
     updated_df.show(5) # adding NULL values
     +----+
     |Gender|Height|CWDistance|Score|
     +----+
                        79 l
      NULL| 62.0|
                               71
     | NULL| 62.0|
                        70 l
                               81
     | NULL| 66.0|
                        85|
                              7|
      NULL| 64.0|
                        87|
                              10|
          M| 73.0|
                        72|
     +----+
     only showing top 5 rows
[159]: print("Updated dataframe shape: {} rows, {} columns".format(updated_df.count(),__
       →len(updated_df.columns))) # dataframe dimension
     Updated dataframe shape: 25 rows, 4 columns
[160]: updated_df = updated_df.na.drop() # by default 'how = any' so it will drop a_
      →row if it contains any nulls ('how = all' will drop a row only if all itsu
       ⇔values are NULL)
     updated_df.show(5) # drop rows with NULL values
     +----+
     |Gender|Height|CWDistance|Score|
     +----+
          M| 73.0|
                        72|
                               4|
             75.0
                               31
          Μl
                        81|
          M| 75.0|
                        107|
                              10|
          M \mid 74.0 \mid
                        106 l
                               5 l
          M| 69.5|
                        96|
                               6 I
```

```
[161]: print("Updated dataframe shape: {} rows, {} columns".format(updated_df.count(),__
        →len(updated_df.columns))) # dataframe dimension
      Updated dataframe shape: 13 rows, 4 columns
[163]: restored_df = df.select('Gender', 'Height', 'CWDistance', 'Score').replace({85:___
        →None}, subset = ['CWDistance'])
      restored df.show(5) # define a new dataframe with NULL values on it
      +----+
      |Gender|Height|CWDistance|Score|
      +----+
           F| 62.0|
                           79|
                                  71
           FΙ
               62.01
                                  81
                           70 l
           FΙ
               66.0|
                         NULL
                                  71
           F|
               64.0|
                            87|
                                 10|
           M| 73.0|
                           72|
                                  41
      +----+
      only showing top 5 rows
[164]: restored_df.groupBy('CWDistance').count().orderBy(F.col("count").desc()).show()__
        →# get unique values and the no. of times each value appears
      +----+
      |CWDistance|count|
            NULLI
                     21
              721
                     21
              791
                     21
              66|
                     21
              65 l
                     11
             115|
                     1|
             101
                     1|
              81|
                     1|
              961
                     1|
              921
                     1 l
              64|
                     1|
             107|
                     1|
              87|
                     1|
              63 l
                     1 l
              82|
                     1|
              70 l
                     1 l
              981
                     11
              90 l
                     1 l
             106
                     11
```

```
67 1
     only showing top 20 rows
[152]: restored_df.agg(F.mean('CWDistance')).collect()[0][0] # mean value of cartwheel_
       \rightarrow distance
[152]: 82.26086956521739
[165]: from pyspark.ml.feature import Imputer
      imputer = Imputer(
         inputCols = ['CWDistance'],
         outputCols = ['{}_imputed'.format(c) for c in ['CWDistance']]
      ).setStrategy('mean') # create a new column with the NULL values of the
       ⇒specified column replaced by the mean value
[166]: restored_df = imputer.fit(restored_df).transform(restored_df)
      restored_df.show(5) # add imputation cols to df
     +----+
     |Gender|Height|CWDistance|Score|CWDistance_imputed|
     +----+
          F| 62.0|
                         79|
                               7|
                                               791
                                              70 l
          F| 62.0|
                         70|
                            8|
          F| 66.0|
                      NULL | 7 |
                                              821
          F| 64.0|
                         87 |
                            10|
                                              87|
                         72|
          M| 73.0|
                               4|
                                               72|
     +----+
     only showing top 5 rows
[191]: spark.stop() # stop spark session
```