03_Basic_operations

March 28, 2020

0.1 Load and quick look

[1]:
```
from pyspark.sql import SparkSession
```

[2]:
```
spark = SparkSession.builder.appName("ops").getOrCreate()
```

[3]:
```
path = "Python-and-Spark-for-Big-Data-master/Spark_DataFrames/appl_stock.csv"
df = spark.read.csv(path, inferSchema=True, header=True)     # infer schema
```

[4]:
```
df.show(5)
```

```
+-------------------+----------+----------+-----------------+-----------------
+---------+-----------------+
|               Date|      Open|      High|              Low|
Close|   Volume|        Adj Close|
+-------------------+----------+----------+-----------------+-----------------
+---------+-----------------+
|2010-01-04 00:00:00|213.429998|214.499996|212.38000099999996|
214.009998|123432400|         27.727039|
|2010-01-05 00:00:00|214.599998|215.589994|        213.249994|
214.379993|150476200|27.774976000000002|
|2010-01-06 00:00:00|214.379993|    215.23|        210.750004|
210.969995|138040000|27.333178000000004|
|2010-01-07 00:00:00|    211.75|212.000006|        209.050005|
210.58|119282800|          27.28265|
|2010-01-08
00:00:00|210.299994|212.000006|209.06000500000002|211.98000499999998|111902700|
27.464034|
+-------------------+----------+----------+-----------------+-----------------
+---------+-----------------+
only showing top 5 rows
```

[5]:
```
df.printSchema()
```

```
root
 |-- Date: timestamp (nullable = true)
 |-- Open: double (nullable = true)
```

```
 |-- High: double (nullable = true)
 |-- Low: double (nullable = true)
 |-- Close: double (nullable = true)
 |-- Volume: integer (nullable = true)
 |-- Adj Close: double (nullable = true)
```

0.2 Filtering

[6]:
```
# filtering using sql syntax
df.filter("Close < 500").show(5)
```

```
+-------------------+----------+----------+-----------------+-----------------
+---------+-----------------+
|               Date|      Open|      High|              Low|
Close|   Volume|        Adj Close|
+-------------------+----------+----------+-----------------+-----------------
+---------+-----------------+
|2010-01-04 00:00:00|213.429998|214.499996|212.38000099999996|
214.009998|123432400|         27.727039|
|2010-01-05 00:00:00|214.599998|215.589994|        213.249994|
214.379993|150476200|27.774976000000002|
|2010-01-06 00:00:00|214.379993|    215.23|        210.750004|
210.969995|138040000|27.333178000000004|
|2010-01-07 00:00:00|    211.75|212.000006|        209.050005|
210.58|119282800|          27.28265|
|2010-01-08
00:00:00|210.299994|212.000006|209.06000500000002|211.98000499999998|111902700|
27.464034|
+-------------------+----------+----------+-----------------+-----------------
+---------+-----------------+
only showing top 5 rows
```

[7]:
```
# filtering using column object (pyspark style)
df.filter(df["Close"] < 500).show(5)
```

```
+-------------------+----------+----------+-----------------+-----------------
+---------+-----------------+
|               Date|      Open|      High|              Low|
Close|   Volume|        Adj Close|
+-------------------+----------+----------+-----------------+-----------------
+---------+-----------------+
|2010-01-04 00:00:00|213.429998|214.499996|212.38000099999996|
214.009998|123432400|         27.727039|
|2010-01-05 00:00:00|214.599998|215.589994|        213.249994|
214.379993|150476200|27.774976000000002|
```

```
|2010-01-06 00:00:00|214.379993|    215.23|        210.750004|
210.969995|138040000|27.333178000000004|
|2010-01-07 00:00:00|    211.75|212.000006|        209.050005|
210.58|119282800|           27.28265|
|2010-01-08
00:00:00|210.299994|212.000006|209.06000500000002|211.98000499999998|111902700|
27.464034|
+-------------------+----------+----------+------------------+------------------
+---------+------------------+
only showing top 5 rows
```

[8]:
```
# we can select columns after filtering using "select"
```

[9]:
```
df.filter(df["Close"] < 500).select(["Open", "Close"]).show(5)
```

```
+----------+------------------+
|      Open|             Close|
+----------+------------------+
|213.429998|        214.009998|
|214.599998|        214.379993|
|214.379993|        210.969995|
|    211.75|            210.58|
|210.299994|211.98000499999998|
+----------+------------------+
only showing top 5 rows
```

0.3 Filtering using multiple coditions

Use & for AND, | for OR, ~ for NOT. Each condition need to be wrapped inside ()

[10]:
```
df.filter((df["Close"] <200) & (df["Open"]>200)).show()
```

```
+-------------------+------------------+----------+----------+----------+-------
--+------------------+
|               Date|              Open|      High|       Low|     Close|
Volume|         Adj Close|
+-------------------+------------------+----------+----------+----------+-------
--+------------------+
|2010-01-22 00:00:00|206.78000600000001|207.499996|    197.16|
197.75|220441900|         25.620401|
|2010-01-28 00:00:00|
204.930004|205.500004|198.699995|199.289995|293375600|25.819922000000002|
|2010-01-29 00:00:00|
201.079996|202.199995|190.250002|192.060003|311488100|         24.883208|
+-------------------+------------------+----------+----------+----------+-------
--+------------------+
```

[11]:
```
df.filter((df["Close"] <200) & ~(df["Open"]>200)).show(5)
```

```
+-------------------+-----------------+----------+-----------------+----------
+---------+------------------+
|               Date|             Open|      High|              Low|
Close|   Volume|         Adj Close|
+-------------------+-----------------+----------+-----------------+----------
+---------+------------------+
|2010-02-01 00:00:00|192.36999699999998|
196.0|191.29999899999999|194.729998|187469100|         25.229131|
|2010-02-02 00:00:00|        195.909998|196.319994|193.37999299999998|195.859997
|174585600|25.375532999999997|
|2010-02-03 00:00:00|        195.169994|200.200003|
194.420004|199.229994|153832000|25.812148999999998|
|2010-02-04 00:00:00|        196.730003|198.370001|
191.570005|192.050003|189413000|         24.881912|
|2010-02-05 00:00:00|192.63000300000002|     196.0|
190.850002|195.460001|212576700|25.323710000000002|
+-------------------+-----------------+----------+-----------------+----------
+---------+------------------+
only showing top 5 rows
```

[12]:
```
df.filter(df["Low"]==197.16).show()
```

```
+-------------------+-----------------+----------+------+------+---------+-----
----+
|               Date|             Open|      High|   Low| Close|   Volume|Adj
Close|
+-------------------+-----------------+----------+------+------+---------+-----
----+
|2010-01-22
00:00:00|206.78000600000001|207.499996|197.16|197.75|220441900|25.620401|
+-------------------+-----------------+----------+------+------+---------+-----
----+
```

0.4 Using collect() to get the data as an object

[13]:
```
result = df.filter(df["Low"]==197.16).collect()
```

[14]:
```
result
```

[14]:
```
[Row(Date=datetime.datetime(2010, 1, 22, 0, 0), Open=206.78000600000001,
High=207.499996, Low=197.16, Close=197.75, Volume=220441900, Adj
Close=25.620401)]
```

[15]:
```
# convert row to dictionary
mydict = result[0].asDict()
```

[16]:
```
mydict
```

[16]:
```
{'Date': datetime.datetime(2010, 1, 22, 0, 0),
 'Open': 206.78000600000001,
 'High': 207.499996,
 'Low': 197.16,
 'Close': 197.75,
 'Volume': 220441900,
 'Adj Close': 25.620401}
```