SPARK STREAMING

DR. ALVIN ANG

CONTENTS

I. STRUCTURED STREAMING

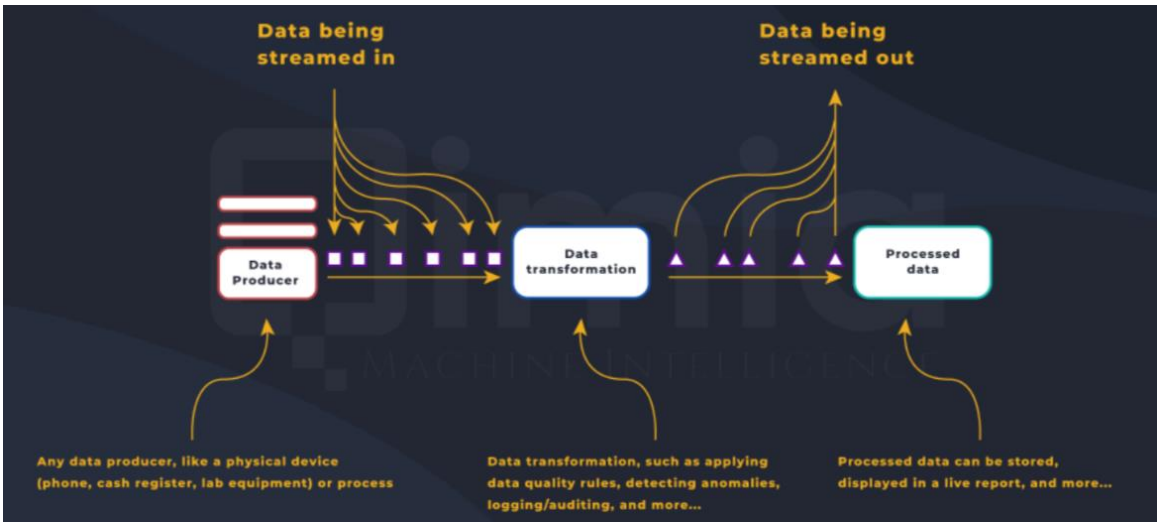
Most of the stuff are referenced from here:

https://www.amazon.com/Learn-PySpark-Python-based-Machine-Learning/dp/1484249607

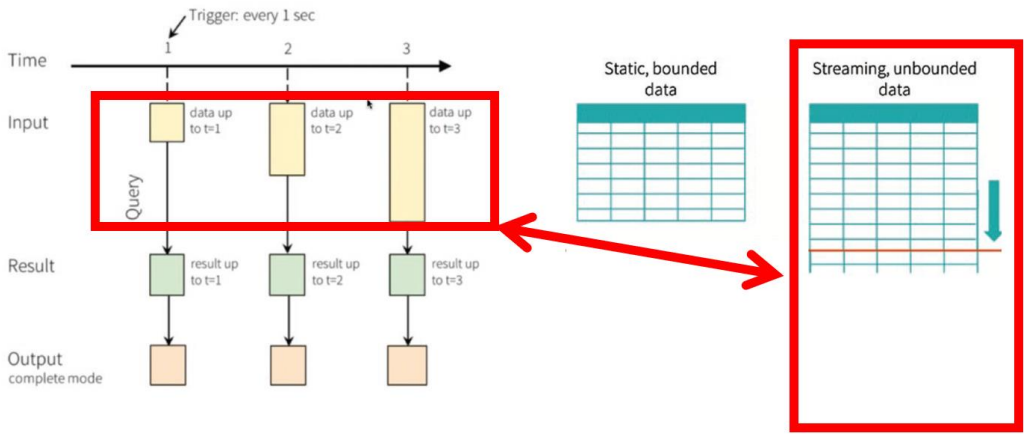
IPYNB

https://www.alvinang.sg/s/Structured_Streaming_with_PySpark.ipynb

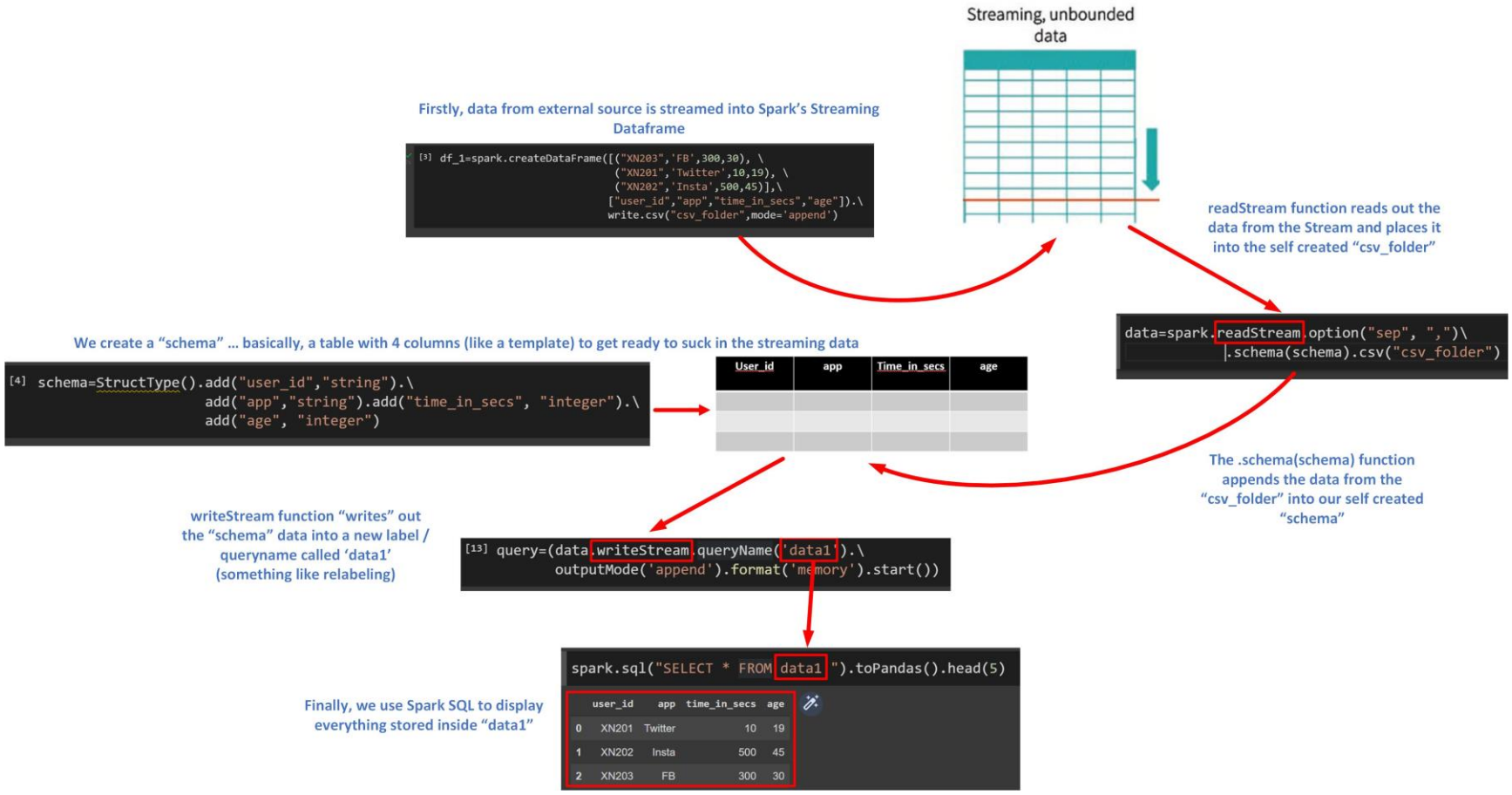
A. IN A NUTSHELL, HOW IT WORKS…



https://qimia.io/en/blog/Developing-Streaming-Applications-Spark-Structured-Streaming



- At t = 1, data streams in and is stored in the Spark Dataframe.
- We may query it and extract it out to store it someplace else.
- At t = 2, this process continues. Data is constantly appended into the initial Dataframe.
- We may query it every second if we want, but each time we query, the dataframe is constantly expanding.
- There are 3 main areas (refer to figure above):
    - INPUT → where data is appended to the dataframe
    - QUERY → .readStream; where the streaming data is extracted using SQL
    - OUTPUT → .writeStream; where the data is stored in another dataframe someplace else



B. IMPORT LIBRARIES

Import Libraries

```
from pyspark.sql import SparkSession
spark=SparkSession.builder.appName('structured_streaming').getOrCreate()

import pyspark.sql.functions as F
from pyspark.sql.types import *
```

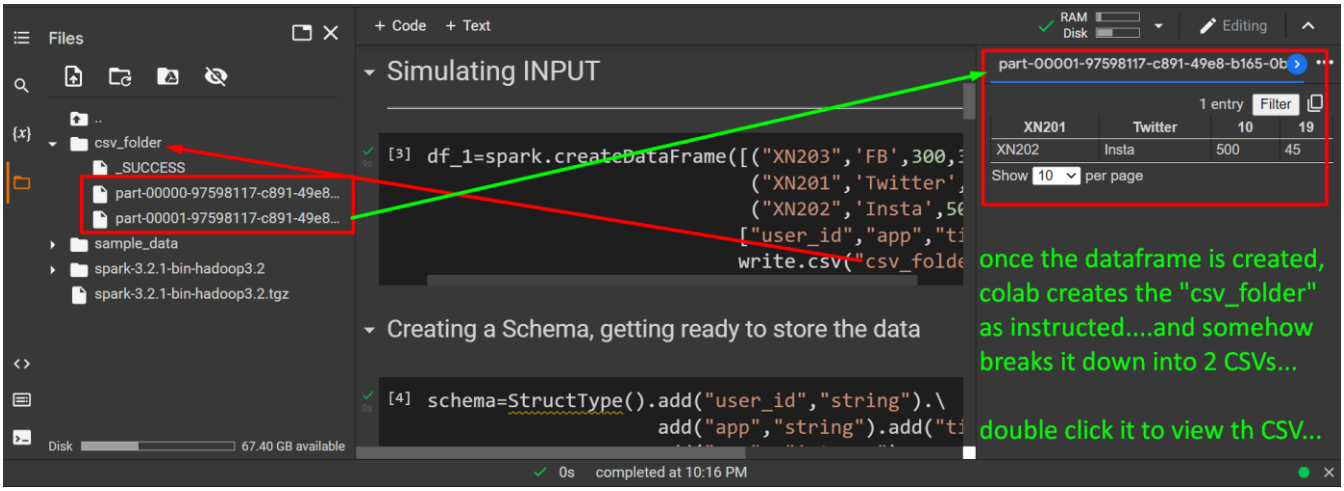
C. START A SPARK SESSION

- Please refer here: https://www.alvinang.sg/s/Installing-Spark-on-Colab-by-Dr-Alvin-Ang.pdf

D. SIMULATING STREAMING DATA

Simulating INPUT

```
df_1=spark.createDataFrame([("XN203",'FB',300,30), \
                            ("XN201",'Twitter',10,19), \
                            ("XN202",'Insta',500,45)],\
                           ["user_id","app","time_in_secs","age"]).\
                           write.csv("csv_folder",mode='append')
```



- Since we are just doing simulation, we pretend that the data is streamed in through "creating a dataframe"

- By right, df_1 should contain the table as below, but note that streaming data is unlike Pandas dataframe (you can't just show the dataframe like that, you need to use readStream and writeStream).

	user_id	app	time_in_secs	age
0	XN201	Twitter	10	19
1	XN202	Insta	500	45
2	XN203	FB	300	30

E. CREATING A SCHEMA

Creating a Schema, getting ready to store the data

```
[4] schema=StructType().add("user_id","string").\
                        add("app","string").add("time_in_secs", "integer").\
                        add("age", "integer")
```

F. READSTREAM

QUERY: readStream to read in data from the Data Frame

```
[5] data=spark.readStream.option("sep", ",")\
                .schema(schema).csv("csv_folder")
```

- readStream grabs the data out of the "csv_folder" and stores it into the "schema"

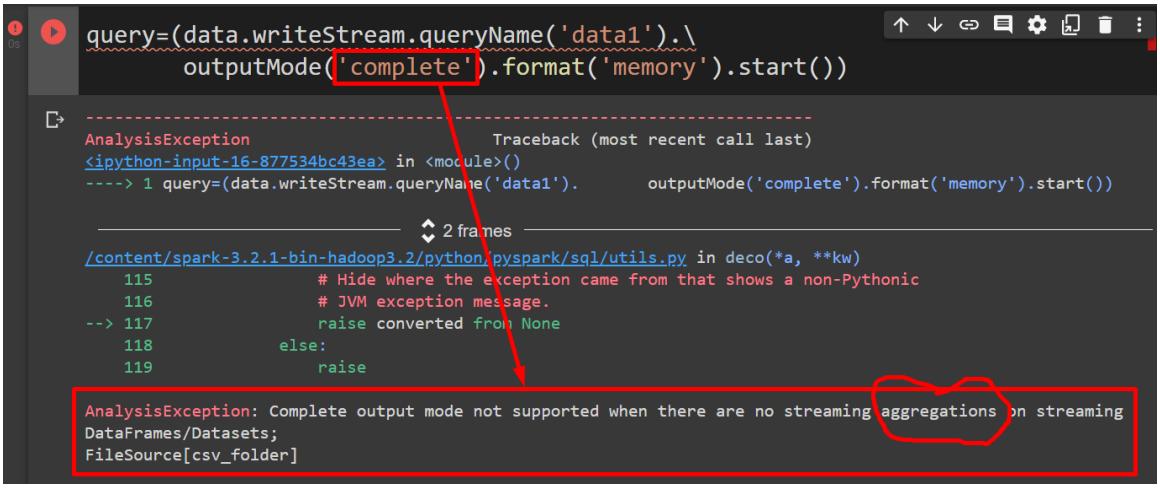
```
[6] data

DataFrame[user_id: string, app: string, time_in_secs: int, age: int]
```

```
data.printSchema()

root
 |-- user_id: string (nullable = true)
 |-- app: string (nullable = true)
 |-- time_in_secs: integer (nullable = true)
 |-- age: integer (nullable = true)
```

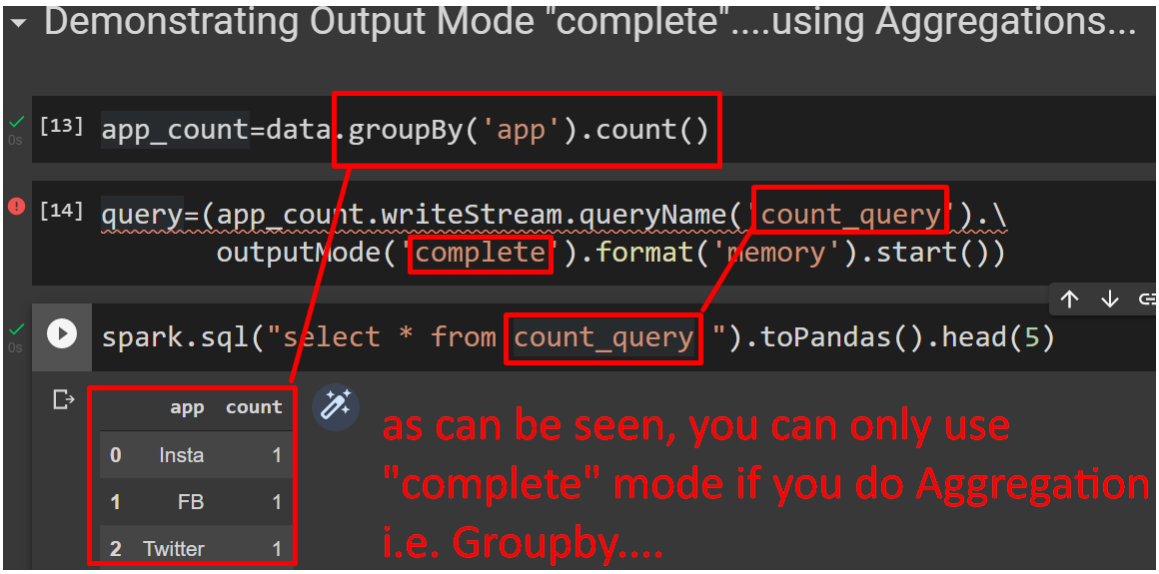
G. WRITESTREAM

```
[13] query=(data.writeStream.queryName('data1').\
          outputMode('append').format('memory').start())

[14] query

<pyspark.sql.streaming.StreamingQuery at 0x7fc160883c90>
```

there are a few output modes.....
'append' _'complete'



- Can you see the difference in the above 2 pictures?
- One is using "append", while the other is using "complete" output mode.
- As can be seen, the "complete" mode gives an error.
- To give a brief explanation, is because "complete" mode only allows the data to be "written out" IF you have performed AGGREGATIONS on it….
- It doesn't allow you to "extract out" the entire "streamed data"…
- Below gives an exmple using "complete" mode….



- We shall only explain "append" in this manuscript (to keep things simple).

H. SQL

```
spark.sql("SELECT * FROM data1 ").toPandas().head(5)
```

	user_id	app	time_in_secs	age
0	XN201	Twitter	10	19
1	XN202	Insta	500	45
2	XN203	FB	300	30

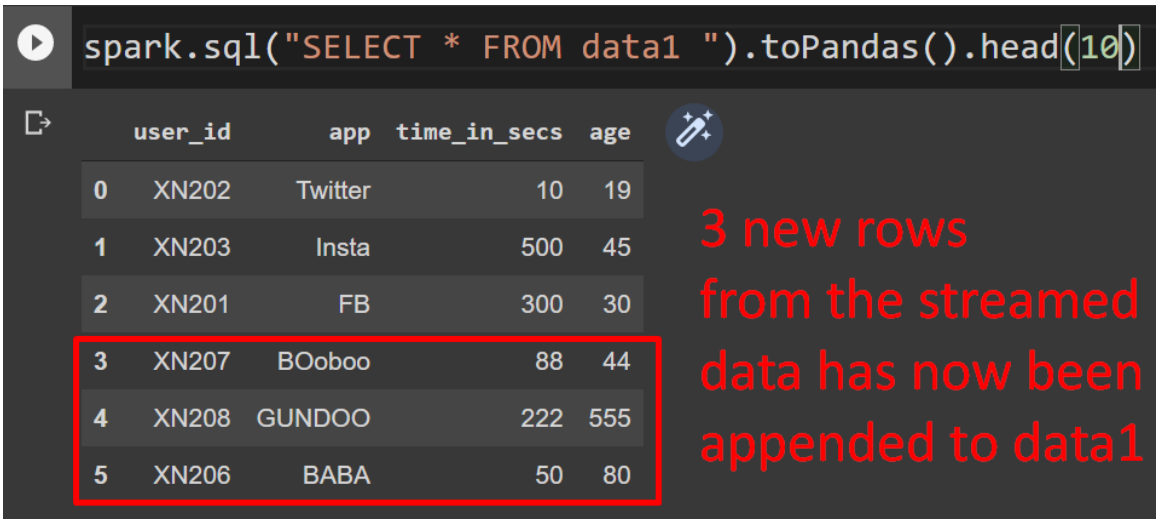
- Finally, we get the very first streamed table!
- We have to use Spark SQL to query and display the data out of data1.

I. CONTINUING…. 2ND DATAFRAME STREAMED IN….

```
df_2=spark.createDataFrame([("XN206",'BABA',50,80),\
                            ("XN207",'BOoboo',88,44),\
                            ("XN208",'GUNDOO',222,555)],\
                           ["user_id","app","time_in_secs","age"]).\
                           write.csv("csv_folder",mode='append')
```

- We simulate another new data streaming in.. this time is df2….

J. SQL AGAIN….



- This time, when you run the exact SQL code again, you realized that df2 has been appended on top of df1…..

ABOUT DR. ALVIN ANG

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.