# Scripts Execution

**Explanation of the solution to the streaming layer problem**

**Task: Create a streaming data processing framework that ingests real-time POS transaction data from Kafka. The transaction data is then validated based on the three rules' parameters (stored in the NoSQL database)**

The steps followed to do this task includes following steps

- Import all necessary libraries and functions.
- Define spark context and add .py files with csv.
- Connect to kafka topic using
  Bootstrap-server: 18.211.252.152
  Port Number: 9092
  Topic: transactions-topic-verified
- Read kafka stream into required schema to map data.
- Look Up Table Name: look_up_table
  Card Transaction table Name: card_transactions
- Defining following UDF to perform required activities and determine whether transaction is fraudulent or genuine.

| FUNCTION | INPUT | OUTPUT |
|---|---|---|
| **ucl_data** | CARD_ID | UCL (look_up_table) |
| **score_data** | CARD_ID | Credit Score (look up tab) |
| **ostcode_data** | CARD_ID | post code (look up table) |
| **distance_calc** | post codes (lookup table & kafka stream) | Distance between 2 locations of current transaction and previous transaction |
| **time_cal** | transaction date (lookup table & kafka stream) | difference between transaction dates in seconds. |
| **TransD_data** | CARD_ID | transaction date (look up table) |
| **speed_calc** | Distance & Time calculated from above distance_calc & time_cal functions | Distance & Time calculated from distance_calc & time_cal functions |
| **status_res** | Amount from current transaction read thru kafka stream, UCL from look up table, Credit_Score (look up table) & Speed calculated (udf) | Status of transaction (genuine or fraud) |

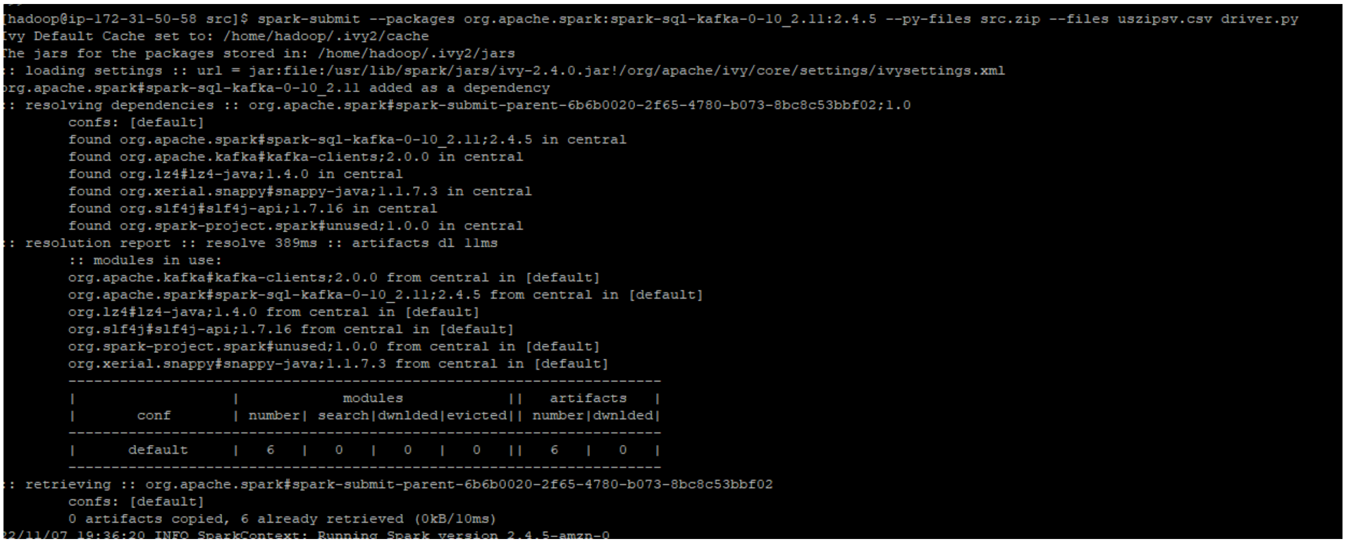
- Executing UDF sequencially. Hence, deriving if transaction is fraud or genuine. These functions work as agents to derive inputs to function status_res (function H).
- The rules performed on inputs supplied to function H.
  If current transaction amount is greater than UCL of look up table for that card_id, mark transaction as Fraud. Else, proceed to check below:
  - If credit score of that card_id under process is less than 250, reject transaction as FRAUD. Else, proceed.

- If speed calculated is greater than 250, recognize the transaction as "FRAUD". If speed is between 0 and 250, mark the transaction as genuine.

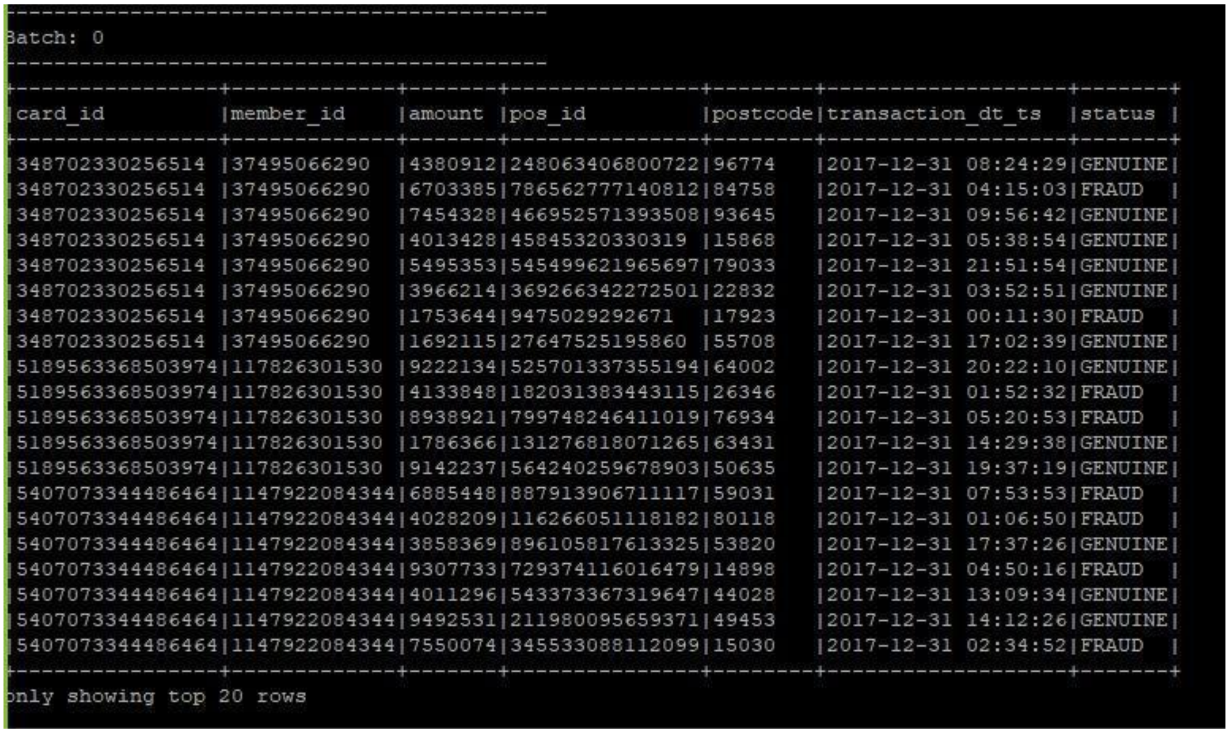
- To summarize, a transaction is qualified to be genuine only when:
  - Credit score of member is greater than 200,
  - Speed is between 0 & 250
  - Amount on current transaction is less than UCL calculated.
- Functions "A", "B", "C", "F" & "H" contact dao.py to call the look up table (given above) for designated purposes.
  - In process of calling dao.py from this driver.py file, I fo called "Import" which loads other .py files in same directory.
  - Establishing spark context to add python files and csv files before command import.
- Function "D" uses geomap.py to calculate distance between last transaction & current transaction locations that is used in calculating speed which is one of factors for determining status of transaction.
- Function "H" status_res also calls look_up_table using write_data function when transaction is genuine.
  - It also updates card_transactions table with latest information of posid, amount, transaction date and member ID.

Command to run:

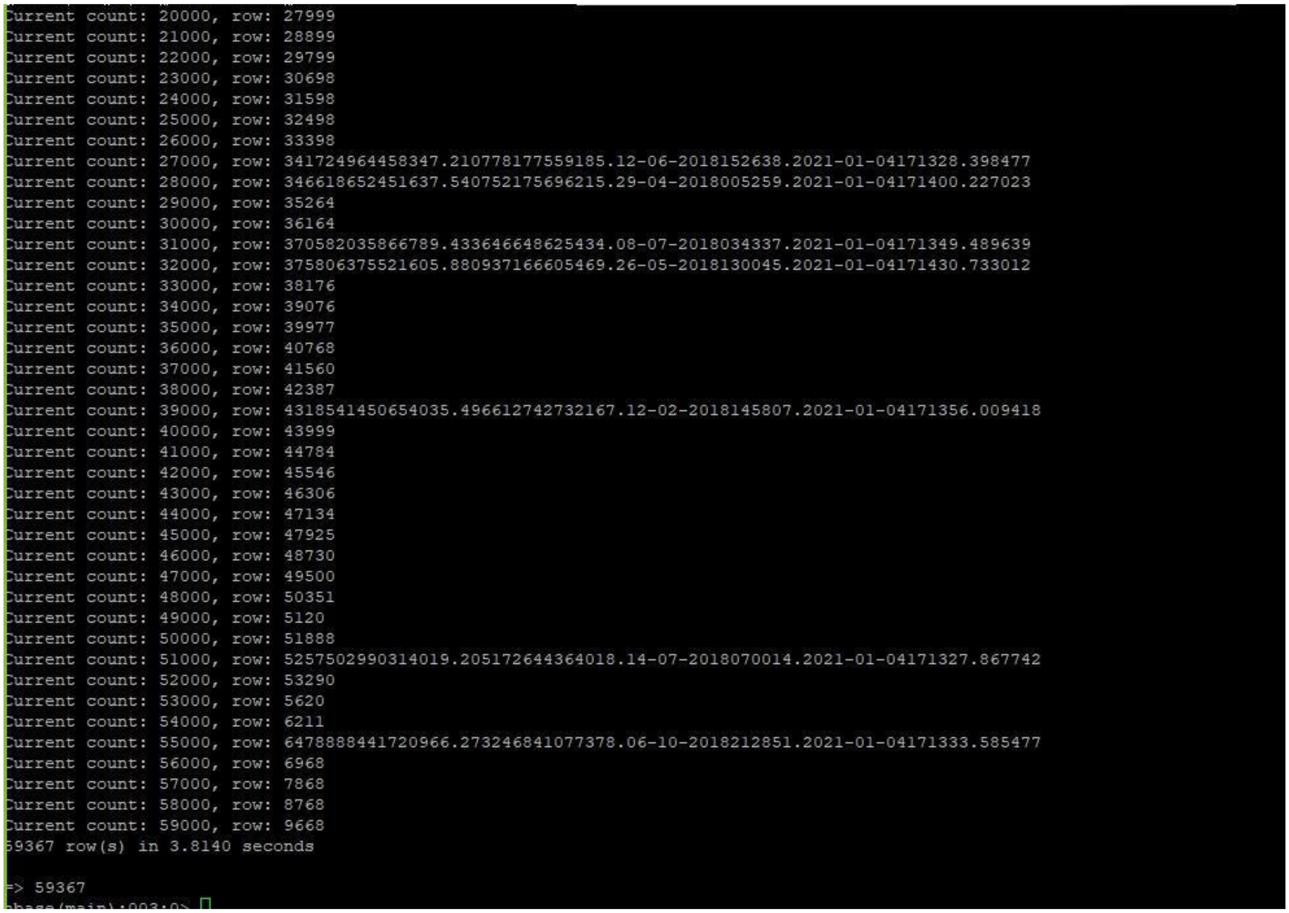
**spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 --py-files src.zip --files uszipsv.csv driver.py**



```
[hadoop@ip-172-31-50-58 src]$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 --py-files src.zip --files uszipsv.csv driver.py
Ivy Default Cache set to: /home/hadoop/.ivy2/cache
The jars for the packages stored in: /home/hadoop/.ivy2/jars
:: loading settings :: url = jar:file:/usr/lib/spark/jars/ivy-2.4.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
org.apache.spark#spark-sql-kafka-0-10_2.11 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-6b6b0020-2f65-4780-b073-8bc8c53bbf02;1.0
        confs: [default]
        found org.apache.spark#spark-sql-kafka-0-10_2.11;2.4.5 in central
        found org.apache.kafka#kafka-clients;2.0.0 in central
        found org.lz4#lz4-java;1.4.0 in central
        found org.xerial.snappy#snappy-java;1.1.7.3 in central
        found org.slf4j#slf4j-api;1.7.16 in central
        found org.spark-project.spark#unused;1.0.0 in central
:: resolution report :: resolve 389ms :: artifacts dl 11ms
        :: modules in use:
        org.apache.kafka#kafka-clients;2.0.0 from central in [default]
        org.apache.spark#spark-sql-kafka-0-10_2.11;2.4.5 from central in [default]
        org.lz4#lz4-java;1.4.0 from central in [default]
        org.slf4j#slf4j-api;1.7.16 from central in [default]
        org.spark-project.spark#unused;1.0.0 from central in [default]
        org.xerial.snappy#snappy-java;1.1.7.3 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   6   |   0   |   0   |   0   ||   6   |   0   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-6b6b0020-2f65-4780-b073-8bc8c53bbf02
        confs: [default]
        0 artifacts copied, 6 already retrieved (0kB/10ms)
22/11/07 19:36:20 INFO SparkContext: Running Spark version 2.4.5-amzn-0
```



```
-------------------------------------------
Batch: 0
-------------------------------------------
+----------------+-------------+-------+---------------+--------+-------------------+-------+
|card_id         |member_id    |amount |pos_id         |postcode|transaction_dt_ts  |status |
+----------------+-------------+-------+---------------+--------+-------------------+-------+
|348702330256514 |37495066290  |4380912|248063406800722|96774   |2017-12-31 08:24:29|GENUINE|
|348702330256514 |37495066290  |6703385|786562777140812|84758   |2017-12-31 04:15:03|FRAUD  |
|348702330256514 |37495066290  |7454328|466952571393508|93645   |2017-12-31 09:56:42|GENUINE|
|348702330256514 |37495066290  |4013428|45845320330319 |15868   |2017-12-31 05:38:54|GENUINE|
|348702330256514 |37495066290  |5495353|545499621965697|79033   |2017-12-31 21:51:54|GENUINE|
|348702330256514 |37495066290  |3966214|369266342272501|22832   |2017-12-31 03:52:51|GENUINE|
|348702330256514 |37495066290  |1753644|9475029292671  |17923   |2017-12-31 00:11:30|FRAUD  |
|348702330256514 |37495066290  |1692115|27647525195860 |55708   |2017-12-31 17:02:39|GENUINE|
|5189563368503974|117826301530 |9222134|525701337355194|64002   |2017-12-31 20:22:10|GENUINE|
|5189563368503974|117826301530 |4133848|182031383443115|26346   |2017-12-31 01:52:32|FRAUD  |
|5189563368503974|117826301530 |8938921|799748246411019|76934   |2017-12-31 05:20:53|FRAUD  |
|5189563368503974|117826301530 |1786366|131276818071265|63431   |2017-12-31 14:29:38|GENUINE|
|5189563368503974|117826301530 |9142237|564240259678903|50635   |2017-12-31 19:37:19|GENUINE|
|5407073344486464|1147922084344|6885448|887913906711117|59031   |2017-12-31 07:53:53|FRAUD  |
|5407073344486464|1147922084344|4028209|116266051118182|80118   |2017-12-31 01:06:50|FRAUD  |
|5407073344486464|1147922084344|3858369|896105817613325|53820   |2017-12-31 17:37:26|GENUINE|
|5407073344486464|1147922084344|9307733|729374116016479|14898   |2017-12-31 04:50:16|FRAUD  |
|5407073344486464|1147922084344|4011296|543373367319647|44028   |2017-12-31 13:09:34|GENUINE|
|5407073344486464|1147922084344|9492531|211980095659371|49453   |2017-12-31 14:12:26|GENUINE|
|5407073344486464|1147922084344|7550074|345533088112099|15030   |2017-12-31 02:34:52|FRAUD  |
+----------------+-------------+-------+---------------+--------+-------------------+-------+
only showing top 20 rows
```



```
Current count: 20000, row: 27999
Current count: 21000, row: 28899
Current count: 22000, row: 29799
Current count: 23000, row: 30698
Current count: 24000, row: 31598
Current count: 25000, row: 32498
Current count: 26000, row: 33398
Current count: 27000, row: 341724964458347.210778177559185.12-06-2018152638.2021-01-04171328.398477
Current count: 28000, row: 346618652451637.540752175696215.29-04-2018005259.2021-01-04171400.227023
Current count: 29000, row: 35264
Current count: 30000, row: 36164
Current count: 31000, row: 370582035866789.433646648625434.08-07-2018034337.2021-01-04171349.489639
Current count: 32000, row: 375806375521605.880937166605469.26-05-2018130045.2021-01-04171430.733012
Current count: 33000, row: 38176
Current count: 34000, row: 39076
Current count: 35000, row: 39977
Current count: 36000, row: 40768
Current count: 37000, row: 41560
Current count: 38000, row: 42387
Current count: 39000, row: 4318541450654035.496612742732167.12-02-2018145807.2021-01-04171356.009418
Current count: 40000, row: 43999
Current count: 41000, row: 44784
Current count: 42000, row: 45546
Current count: 43000, row: 46306
Current count: 44000, row: 47134
Current count: 45000, row: 47925
Current count: 46000, row: 48730
Current count: 47000, row: 49500
Current count: 48000, row: 50351
Current count: 49000, row: 5120
Current count: 50000, row: 51888
Current count: 51000, row: 5257502990314019.205172644364018.14-07-2018070014.2021-01-04171327.867742
Current count: 52000, row: 53290
Current count: 53000, row: 5620
Current count: 54000, row: 6211
Current count: 55000, row: 6478888441720966.273246841077378.06-10-2018212851.2021-01-04171333.585477
Current count: 56000, row: 6968
Current count: 57000, row: 7868
Current count: 58000, row: 8768
Current count: 59000, row: 9668
59367 row(s) in 3.8140 seconds

=> 59367
hbase(main):003:0>
```