## 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**





