Loading the Lookup Table

Commands to load the relevant data in the Lookup Table

Calculate the moving average and standard deviation of the last 10 transactions for each card_id for the data present in hadoop and NoSQL database:

```
window = Window.partitionBy(history['card_id']).orderBy(history['transaction_date'].desc())

history_df = history.select('*', f.rank().over(window).alias('rank')).filter(f.col('rank') <= 10)
```

```
+---------------+-------+--------+---------------+-------+-----+-------------------+----+
|        card_id| amount|postcode|         pos_id| status|score|   transaction_date|rank|
+---------------+-------+--------+---------------+-------+-----+-------------------+----+
|340379737226464|1784098|   26656|000383013889790|GENUINE|  229|2018-01-27 00:19:47|   1|
|340379737226464|3759577|   61334|016312401940277|GENUINE|  229|2018-01-18 14:26:09|   2|
|340379737226464|4080612|   51338|562082278231631|GENUINE|  229|2018-01-14 20:54:02|   3|
|340379737226464|4242710|   96105|285501971776349|GENUINE|  229|2018-01-11 19:09:55|   4|
|340379737226464|9061517|   40932|232455833079472|GENUINE|  229|2018-01-10 20:20:33|   5|
|340379737226464| 102248|   40932|232455833079472|GENUINE|  229|2018-01-10 15:04:33|   6|
|340379737226464|7445128|   50455|915439934619047|GENUINE|  229|2018-01-07 23:52:27|   7|
|340379737226464|5706163|   50455|915439934619047|GENUINE|  229|2018-01-07 22:07:07|   8|
|340379737226464|8090127|   18626|359283931604637|GENUINE|  229|2017-12-29 13:24:07|   9|
|340379737226464|9282351|   41859|808326141065551|GENUINE|  229|2017-12-28 19:50:46|  10|
|345406224887566|1135534|   53034|146838238062262|GENUINE|  349|2017-12-25 04:03:58|   1|
|345406224887566|5190295|   88036|821406924682103|GENUINE|  349|2017-12-20 04:41:07|   2|
|345406224887566|5970187|   28334|024341862357645|GENUINE|  349|2017-11-30 05:24:25|   3|
|345406224887566|3854486|   48880|172521878612232|GENUINE|  349|2017-09-21 00:01:58|   4|
|345406224887566|1242240|   14510|536497882467098|GENUINE|  349|2017-06-11 16:31:45|   5|
```

- Created a window over existing dataframe, grouping using card_id and ordered using transaction date.
- window dataframe above will give transactions of each card_id in chronological order.
- To give rank to each of those identified rows and selected only last 10 transactions

Moving average and standard deviation is calculated using below code
Created separate column with alias as moving_avg and Std_Dev inside the dataframe

```
history_df = history_df.groupBy("card_id").agg(f.round(f.avg('amount'),2).alias('moving_avg'), \
                                               f.round(f.stddev('amount'),2).alias('Std_Dev'))
history_df.show()
```

```
+----------------+----------+----------+
|         card_id|moving_avg|   Std_Dev|
+----------------+----------+----------+
| 340379737226464| 5355453.1|3107063.55|
| 345406224887566| 5488456.5|3252527.52|
| 348962542187595| 5735629.0|3089916.54|
| 377201318164757| 5742377.7|2768545.84|
| 379321864695232| 4713319.1|3203114.94|
|4389973676463558| 4923904.7| 2306771.9|
|4407230633003235| 4348891.3|3274883.95|
|5403923427969691| 5375495.6|2913510.72|
|5508842242491554| 4570725.9|3229905.04|
|6562510549485881| 5551056.9|2501552.48|
| 340028465709212| 6863758.9|3326644.65|
```

Command to see the table created and it's content

Calculating UCL from moving average & standard deviation:

UCL = moving average + 3 *(standard deviation)

```
history_df = history_df.withColumn('UCL',history_df.moving_avg+3*(history_df.Std_Dev))
history_df.show()
```

```
+----------------+----------+----------+--------------------+
|         card_id|moving_avg|   Std_Dev|                 UCL|
+----------------+----------+----------+--------------------+
| 340379737226464| 5355453.1|3107063.55|1.4676643749999998E7|
| 345406224887566| 5488456.5|3252527.52|       1.524603906E7|
| 348962542187595| 5735629.0|3089916.54|1.5005378620000001E7|
| 377201318164757| 5742377.7|2768545.84|1.4048015219999999E7|
| 379321864695232| 4713319.1|3203114.94|       1.432266392E7|
|4389973676463558| 4923904.7| 2306771.9|1.1844220399999999E7|
|4407230633003235| 4348891.3|3274883.95|1.4173543150000002E7|
|5403923427969691| 5375495.6|2913510.72|       1.411602776E7|
|5508842242491554| 4570725.9|3229905.04|1.4260441020000001E7|
|6562510549485881| 5551056.9|2501552.48|       1.305571434E7|
| 340028465709212| 6863758.9|3326644.65|       1.684369285E7|
| 349143706735646| 5453372.9|3424332.26|       1.572636968E7|
|4126356979547079| 4286400.2|2909676.26|       1.301542898E7|
|4484950467600170| 4550480.5|3171538.48|       1.406509594E7|
|4818950814628962| 2210428.9| 958307.87|          5085352.51|
|5464688416792307| 4985938.2|2379084.95|       1.212319305E7|
```

Joining the previous dataframe to this dataframe which has UCL calculated to reproduce a new dataframe with all data required to have for look up table.

```
history_df = history_df.select('card_id','UCL')
```

```
lookup_table = lookup_table.join(history_df,on=['card_id'])
```

```
lookup_table.show()  #Final data set look as below
```

```
+----------------+-------------------+-------+--------+---------------+-------+-----+--------------------+
|         card_id|   transaction_date| amount|postcode|         pos_id| status|score|                 UCL|
+----------------+-------------------+-------+--------+---------------+-------+-----+--------------------+
| 340379737226464|2018-01-27 00:19:47|1784098|   26656|000383013889790|GENUINE|  229|1.4676643749999998E7|
| 345406224887566|2017-12-25 04:03:58|1135534|   53034|146838238062262|GENUINE|  349|       1.524603906E7|
| 348962542187595|2018-01-29 17:17:14|7408949|   27830|453850044027107|GENUINE|  522|1.5005378620000001E7|
| 377201318164757|2017-11-28 16:32:22|4799826|   84302|287431794718846|GENUINE|  432|1.4048015219999999E7|
| 379321864695232|2018-01-03 00:29:37|5702120|   98837|638380208258390|GENUINE|  297|       1.432266392E7|
|4389973676463558|2018-01-26 13:47:46|7196505|   10985|588476547410852|GENUINE|  400|1.1844220399999999E7|
|4407230633003235|2018-01-27 07:21:08|  38579|   50167|697070998627535|GENUINE|  567|1.4173543150000002E7|
|5403923427969691|2018-01-22 23:46:19|1576154|   17350|734614251977032|GENUINE|  324|       1.411602776E7|
|5508842242491554|2018-01-31 14:55:58|2710473|   12986|990193545769550|GENUINE|  585|1.4260441020000001E7|
|6562510549485881|2018-01-17 08:35:27|5939348|   35440|901627725704672|GENUINE|  518|       1.305571434E7|
| 340028465709212|2018-01-02 03:25:35|8696557|   24658|246987608008994|GENUINE|  233|       1.684369285E7|
| 349143706735646|2018-01-29 22:33:14|9246599|   99101|743905143665678|GENUINE|  298|       1.572636968E7|
|4126356979547079|2018-01-24 16:09:03|1770784|   14475|698032801419746|GENUINE|  345|       1.301542898E7|
|4484950467600170|2018-01-10 08:03:13|2284955|   13324|653851258729390|GENUINE|  462|       1.406509594E7|
|4818950814628962|2018-01-31 00:53:15|2316346|   88081|127695801600255|GENUINE|  660|          5085352.51|
|5464688416792307|2018-01-26 19:03:47|4067979|   71670|111365575664933|GENUINE|  469|       1.212319305E7|
|5543219113990484|2018-01-13 18:34:00| 549641|   62273|039213658608911|GENUINE|  494|       1.294090916E7|
|5573293264792992|2018-01-31 14:55:57|4827477|   27012|805073498705051|GENUINE|  284|1.1698505790000001E7|
|6011273561157733|2018-02-01 01:27:58|5272574|   45305|063916192266113|GENUINE|  411|1.3040283309999999E7|
|6011985140563103|2018-01-30 02:03:54|1725430|   36587|914045782120401|GENUINE|  350|1.4569845000000002E7|
+----------------+-------------------+-------+--------+---------------+-------+-----+--------------------+
```

Screenshot of the created table

```
In [76]: create_table('lookup_table', {'info' : dict(max_versions=5) })

creating table lookup_table
fetching all table
all tables fetched
table created
```

```
In [79]: batch_insert_data(lookup_table,'lookup_table')

starting batch insert of events
batch insert done
```

```
hbase(main):007:0> list
TABLE
card_transactions
lookup_table
2 row(s) in 0.0070 seconds

=> ["card_transactions", "lookup_table"]
hbase(main):008:0>
```

root@ip-10-0-0-149:~

```
 6591175617713393                    column=info:transaction_date, timestamp=1634375007372, value=2018-01-31 13:10:37
 6592184145413632                    column=info:UCL, timestamp=1634375006787, value=13734342.65
 6592184145413632                    column=info:card_id, timestamp=1634375006787, value=6592184145413632
 6592184145413632                    column=info:postcode, timestamp=1634375006787, value=53186
 6592184145413632                    column=info:score, timestamp=1634375006787, value=456
 6592184145413632                    column=info:transaction_date, timestamp=1634375006787, value=2018-01-28 00:54:30
 6594248319343442                    column=info:UCL, timestamp=1634375006872, value=15065362.77
 6594248319343442                    column=info:card_id, timestamp=1634375006872, value=6594248319343442
 6594248319343442                    column=info:postcode, timestamp=1634375006872, value=24927
 6594248319343442                    column=info:score, timestamp=1634375006872, value=350
 6594248319343442                    column=info:transaction_date, timestamp=1634375006872, value=2018-01-31 23:42:38
 6595638658736751                    column=info:UCL, timestamp=1634375007621, value=14005069.97
 6595638658736751                    column=info:card_id, timestamp=1634375007621, value=6595638658736751
 6595638658736751                    column=info:postcode, timestamp=1634375007621, value=68328
 6595638658736751                    column=info:score, timestamp=1634375007621, value=310
 6595638658736751                    column=info:transaction_date, timestamp=1634375007621, value=2018-01-30 10:50:34
 6595814135833988                    column=info:UCL, timestamp=1634375007288, value=14332708.84
 6595814135833988                    column=info:card_id, timestamp=1634375007288, value=6595814135833988
 6595814135833988                    column=info:postcode, timestamp=1634375007288, value=22508
 6595814135833988                    column=info:score, timestamp=1634375007288, value=210
 6595814135833988                    column=info:transaction_date, timestamp=1634375007288, value=2018-01-30 02:03:54
 6595928469079750                    column=info:UCL, timestamp=1634375008323, value=11824730.01
 6595928469079750                    column=info:card_id, timestamp=1634375008323, value=6595928469079750
 6595928469079750                    column=info:postcode, timestamp=1634375008323, value=98349
 6595928469079750                    column=info:score, timestamp=1634375008323, value=412
 6595928469079750                    column=info:transaction_date, timestamp=1634375008323, value=2018-01-24 12:38:22
 6597703848279563                    column=info:UCL, timestamp=1634375007681, value=15250624.49
 6597703848279563                    column=info:card_id, timestamp=1634375007681, value=6597703848279563
 6597703848279563                    column=info:postcode, timestamp=1634375007681, value=95699
 6597703848279563                    column=info:score, timestamp=1634375007681, value=218
 6597703848279563                    column=info:transaction_date, timestamp=1634375007681, value=2018-01-27 10:51:49
 6598830758632447                    column=info:UCL, timestamp=1634375007878, value=12685782.48
 6598830758632447                    column=info:card_id, timestamp=1634375007878, value=6598830758632447
 6598830758632447                    column=info:postcode, timestamp=1634375007878, value=19421
 6598830758632447                    column=info:score, timestamp=1634375007878, value=293
 6598830758632447                    column=info:transaction_date, timestamp=1634375007878, value=2018-01-30 00:18:34
 6599900931314251                    column=info:UCL, timestamp=1634375008288, value=12487392.07
 6599900931314251                    column=info:card_id, timestamp=1634375008288, value=6599900931314251
 6599900931314251                    column=info:postcode, timestamp=1634375008288, value=97423
 6599900931314251                    column=info:score, timestamp=1634375008288, value=297
 6599900931314251                    column=info:transaction_date, timestamp=1634375008288, value=2018-01-31 11:25:16
999 row(s) in 0.4250 seconds

hbase(main):009:0>
```