**ASPIRING AFFLUENTS**

There are 2 parts to the script running, base and feature extraction is in SAS (codes below) and then data is sent to Hive. Next part is the Python script, retrieving data from Hive.

**BASE EXTRACTION**

/\*\*\* LOAD LIBRARIES \*\*\*/

LIBNAME DAPH '/u06/SAS/prod/MIS/Daphanee';

LIBNAME DAPHAA '/u06/SAS/prod/MIS/Daphanee/Aspiring Affluents';

LIBNAME HIVESAS hadoop SUBPROTOCOL=HIVE2 DBMAX\_TEXT=**100** LOGIN\_TIMEOUT=**0** BL\_PORT=**8020** READ\_METHOD=JDBC schema=sas\_ns server="myucbpaabdapp04.cimbmy.cimbdomain.com" port=**10000**;

LIBNAME HIVEMOD hadoop SUBPROTOCOL=HIVE2 DBMAX\_TEXT=**100** LOGIN\_TIMEOUT=**0** BL\_PORT=**8020** READ\_METHOD=JDBC schema=modelling\_ws server="myucbpaabdapp04.cimbmy.cimbdomain.com" port=**10000**;

LIBNAME INCOME2 "/u08/SAS/prod/MIS/Analytic/CIFBASE";

LIBNAME CASA "/u08/SAS/prod/MIS/Analytic/CASA\_TRXN";

LIBNAME HP "/u06/SAS/prod/MIS/Rev/HiddenPreferred";

LIBNAME INTGRPH '/u11/SAS/prod/Modelling/12\_Cust\_Interest';

LIBNAME my\_mis01 sasspds schema='my\_mis01' serv='5400' host='MYUCBPSMMAAPP01' user='spduser' password="{SAS002}B153B44C1599D99F0EC6D7D00CE06B2D" aclspecial=yes;

/\* DATA TO USE - CHANGE DEPENDING ON THE BASE NEEDED \*/

/\* MAKE SURE TO NAME IC COLUMN AS CUST\_ID \*/

/\* TAKE LATEST CUSTOMER BASE - EXCLUDE PRIME AND PREFERRED \*/

**PROC** **SQL**;

CREATE TABLE AVPH\_BASE AS

SELECT A.CUST\_ID,

B.CUST\_SEGMENT

FROM CARV3.CAR\_AVPH\_20200331 AS A

INNER JOIN CARV3.CAR\_CD\_20200331 AS B

ON A.PARTY\_RK = B.PARTY\_RK

WHERE A.TYPE = "RB" AND

A.TTL\_PROD\_HOLD > **0** AND

B.CUST\_SEGMENT = "MASS"

;**QUIT**;

/\* GETTING INCOME BASE \*/

**PROC** **SQL**;

CREATE TABLE INCOME AS

SELECT \*

FROM INCOME2.INCOME\_BASE

WHERE EFF\_DT >= **'01JAN2019'D**

ORDER BY CUST\_ID, REPORT\_DT DESC, EFF\_DT DESC

;**QUIT**;

**DATA** INCOME1;

SET INCOME;

BY CUST\_ID DESCENDING REPORT\_DT DESCENDING EFF\_DT;

IF FIRST.CUST\_ID;

**RUN**;

**PROC** **SQL**;

CREATE TABLE AVPH\_BASE1 AS

SELECT DISTINCT A.CUST\_ID,

B.GROSS\_INCOME

FROM AVPH\_BASE AS A

LEFT JOIN INCOME1 AS B

ON A.CUST\_ID = B.CUST\_ID

;**QUIT**;

/\* Count = 5,318,962 \*/

**PROC** **SQL**;

CREATE TABLE BASE AS

SELECT DISTINCT CUST\_ID

FROM AVPH\_BASE1

WHERE GROSS\_INCOME =**.**

;**QUIT**;

**ATM FEATURES**

/\* MAPPING TABLE - AR\_ID AND CUST\_ID \*/

**PROC** **SQL**;

CREATE TABLE MAPPING AS

SELECT A.AR\_ID,

B.CUST\_ID

FROM CARV3.CAR\_CUST\_ACTBASE\_20200331 AS A

LEFT JOIN CARV3.CAR\_AVPH\_20200331 AS B

ON A.SBJ\_IP\_ID = B.SBJ\_IP\_ID

GROUP BY A.AR\_ID, B.CUST\_ID

;**QUIT**;

/\* MONTHLY ATM TRANSACTIONS (LAST 3 MONTHS) \*/

/\* L1M \*/

**PROC** **SQL**;

CREATE TABLE CASA\_ATMBASE\_202003 AS

SELECT A.\*,

B.CUST\_ID

FROM CASA.CASA\_TRXN\_202003 AS A

LEFT JOIN MAPPING AS B

ON A.AR\_ID = B.AR\_ID

WHERE A.UNIV\_DESC CONTAINS "ATM" AND

A.DR\_CR\_TP\_ID = "685002" AND /\* DEBIT TRANSACTIONS ONLY \*/

((INDEX(A.UNIV\_DESC, "CHARG") EQ **0** AND INDEX(A.UNIV\_DESC, "CHG") EQ **0** AND INDEX(A.UNIV\_DESC, "FEE") EQ **0** AND

INDEX(A.UNIV\_DESC, "COMMISSION") EQ **0** AND INDEX(A.UNIV\_DESC, "COMMISION") EQ **0** AND INDEX(A.UNIV\_DESC, "INCENTIVE") EQ **0** AND

INDEX(A.UNIV\_DESC, "ADJ") EQ **0** AND INDEX(A.UNIV\_DESC, "PENAL") EQ **0** AND INDEX(A.UNIV\_DESC, "RT") EQ **0** AND

INDEX(A.UNIV\_DESC, "RETURN") EQ **0** AND INDEX(A.UNIV\_DESC, "REFUND") EQ **0** AND INDEX(A.UNIV\_DESC, "REV") EQ **0** AND

INDEX(A.UNIV\_DESC, "INT") EQ **0** AND INDEX(A.UNIV\_DESC, "STAMP DUTY") EQ **0** AND INDEX(A.UNIV\_DESC, "AUTO") EQ **0** AND

INDEX(A.UNIV\_DESC, "GST") EQ **0** AND INDEX(A.UNIV\_DESC, "VOID") EQ **0**) OR

(INDEX(A.UNIV\_DESC, "AUTOPAY") GT **0** AND INDEX(A.UNIV\_DESC, "CHARG") EQ **0**)) /\* REMOVING FEES AND CHARGES \*/

;**QUIT**;

/\* L2M \*/

**PROC** **SQL**;

CREATE TABLE CASA\_ATMBASE\_202002 AS

SELECT A.\*,

B.CUST\_ID

FROM CASA.CASA\_TRXN\_202002 AS A

LEFT JOIN MAPPING AS B

ON A.AR\_ID = B.AR\_ID

WHERE A.UNIV\_DESC CONTAINS "ATM" AND

A.DR\_CR\_TP\_ID = "685002" AND /\* DEBIT TRANSACTIONS ONLY \*/

((INDEX(A.UNIV\_DESC, "CHARG") EQ **0** AND INDEX(A.UNIV\_DESC, "CHG") EQ **0** AND INDEX(A.UNIV\_DESC, "FEE") EQ **0** AND

INDEX(A.UNIV\_DESC, "COMMISSION") EQ **0** AND INDEX(A.UNIV\_DESC, "COMMISION") EQ **0** AND INDEX(A.UNIV\_DESC, "INCENTIVE") EQ **0** AND

INDEX(A.UNIV\_DESC, "ADJ") EQ **0** AND INDEX(A.UNIV\_DESC, "PENAL") EQ **0** AND INDEX(A.UNIV\_DESC, "RT") EQ **0** AND

INDEX(A.UNIV\_DESC, "RETURN") EQ **0** AND INDEX(A.UNIV\_DESC, "REFUND") EQ **0** AND INDEX(A.UNIV\_DESC, "REV") EQ **0** AND

INDEX(A.UNIV\_DESC, "INT") EQ **0** AND INDEX(A.UNIV\_DESC, "STAMP DUTY") EQ **0** AND INDEX(A.UNIV\_DESC, "AUTO") EQ **0** AND

INDEX(A.UNIV\_DESC, "GST") EQ **0** AND INDEX(A.UNIV\_DESC, "VOID") EQ **0**) OR

(INDEX(A.UNIV\_DESC, "AUTOPAY") GT **0** AND INDEX(A.UNIV\_DESC, "CHARG") EQ **0**)) /\* REMOVING FEES AND CHARGES \*/

;**QUIT**;

/\* L3M \*/

**PROC** **SQL**;

CREATE TABLE CASA\_ATMBASE\_202001 AS

SELECT A.\*,

B.CUST\_ID

FROM CASA.CASA\_TRXN\_202001 AS A

LEFT JOIN MAPPING AS B

ON A.AR\_ID = B.AR\_ID

WHERE A.UNIV\_DESC CONTAINS "ATM" AND

A.DR\_CR\_TP\_ID = "685002" AND /\* DEBIT TRANSACTIONS ONLY \*/

((INDEX(A.UNIV\_DESC, "CHARG") EQ **0** AND INDEX(A.UNIV\_DESC, "CHG") EQ **0** AND INDEX(A.UNIV\_DESC, "FEE") EQ **0** AND

INDEX(A.UNIV\_DESC, "COMMISSION") EQ **0** AND INDEX(A.UNIV\_DESC, "COMMISION") EQ **0** AND INDEX(A.UNIV\_DESC, "INCENTIVE") EQ **0** AND

INDEX(A.UNIV\_DESC, "ADJ") EQ **0** AND INDEX(A.UNIV\_DESC, "PENAL") EQ **0** AND INDEX(A.UNIV\_DESC, "RT") EQ **0** AND

INDEX(A.UNIV\_DESC, "RETURN") EQ **0** AND INDEX(A.UNIV\_DESC, "REFUND") EQ **0** AND INDEX(A.UNIV\_DESC, "REV") EQ **0** AND

INDEX(A.UNIV\_DESC, "INT") EQ **0** AND INDEX(A.UNIV\_DESC, "STAMP DUTY") EQ **0** AND INDEX(A.UNIV\_DESC, "AUTO") EQ **0** AND

INDEX(A.UNIV\_DESC, "GST") EQ **0** AND INDEX(A.UNIV\_DESC, "VOID") EQ **0**) OR

(INDEX(A.UNIV\_DESC, "AUTOPAY") GT **0** AND INDEX(A.UNIV\_DESC, "CHARG") EQ **0**)) /\* REMOVING FEES AND CHARGES \*/

;**QUIT**;

/\* COMBINING TOGETHER \*/

**DATA** CASA\_ATMBASE\_COMBINE;

SET CASA\_ATMBASE\_202001

CASA\_ATMBASE\_202002

CASA\_ATMBASE\_202003;

**RUN**;

**PROC** **SQL**;

CREATE TABLE CASA\_ATMBASE\_COMBINE1 AS

SELECT A.\*,

CASE WHEN A.TXN\_TP\_ID = "801369" THEN "OVERSEAS"

WHEN A.TXN\_TP\_ID = "801358" THEN "CIMB REGIONAL"

WHEN A.TXN\_TP\_ID = "801357" THEN "NON CIMB REGIONAL"

WHEN A.TXN\_TP\_ID = "801861" THEN "CIMB MALAYSIA"

WHEN A.TXN\_TP\_ID = "800712" THEN "NON CIMB MALAYSIA" ELSE "NIL" END AS DESCRIPTION,

PUT(A.TXN\_DT, yymmn6.) as TXN\_DATE

FROM CASA\_ATMBASE\_COMBINE AS A

WHERE A.TXN\_AMT > **0** AND

A.TXN\_TP\_ID IN ("801369","801358","801357","801861","800712")

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE ATM\_WITHDRAWALS AS

SELECT CUST\_ID,

TXN\_DATE,

DESCRIPTION,

COUNT(\*) AS TOTAL\_TXN\_CNT,

SUM(TXN\_AMT) AS TOTAL\_TXN\_AMT,

MEAN(TXN\_AMT) AS MEAN\_TXN\_AMT

FROM CASA\_ATMBASE\_COMBINE1

WHERE CUST\_ID IS NOT NULL

GROUP BY CUST\_ID, TXN\_DATE, DESCRIPTION

;**QUIT**;

/\*\*\*\*\* RULE 1: # AND $ OF ATM WITHDRAWALS \*\*\*\*\*/

**PROC** **SQL**;

CREATE TABLE ATM\_1\_1 AS

SELECT CUST\_ID,

TXN\_DATE,

SUM(TOTAL\_TXN\_CNT) AS TTL\_TXN\_CNT,

SUM(TOTAL\_TXN\_AMT) AS TTL\_TXN\_AMT

FROM ATM\_WITHDRAWALS

GROUP BY CUST\_ID, TXN\_DATE

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE ATM\_1\_2 AS

SELECT CUST\_ID,

(SUM(TTL\_TXN\_CNT)/COUNT(\*)) AS L3M\_TXN\_CNT,

(SUM(TTL\_TXN\_AMT)/COUNT(\*)) AS L3M\_TXN\_AMT

FROM ATM\_1\_1

GROUP BY CUST\_ID

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE ATM\_1\_FINAL AS

SELECT \*,

CASE WHEN L3M\_TXN\_CNT < **3** THEN "01. < 3 TIMES"

WHEN L3M\_TXN\_CNT >= **3** AND L3M\_TXN\_CNT < **5** THEN "02. 3-5 TIMES"

WHEN L3M\_TXN\_CNT >= **5** AND L3M\_TXN\_CNT <= **10** THEN "03. 5-10 TIMES"

ELSE "04. > 10 TIMES" END AS ATM\_WITHDRAWAL\_CNT,

CASE WHEN L3M\_TXN\_AMT < **200** THEN "01. < RM200"

WHEN L3M\_TXN\_AMT >= **200** AND L3M\_TXN\_AMT < **500** THEN "02. RM200 - RM500"

WHEN L3M\_TXN\_AMT >= **500** AND L3M\_TXN\_AMT < **1000** THEN "03. RM500 - RM1000"

WHEN L3M\_TXN\_AMT >= **1000** AND L3M\_TXN\_AMT < **3000** THEN "04. RM1000 - RM3000"

WHEN L3M\_TXN\_AMT >= **3000** AND L3M\_TXN\_AMT <= **5000** THEN "05. RM3000 - RM5000"

ELSE "06. > RM5000" END AS ATM\_WITHDRAWAL\_AMT

FROM ATM\_1\_2

;**QUIT**;

/\*RULE 2:# AND $ OF ATM WITHDRAWALS FOR REGIONAL AND OVERSEAS ATMS\*/

/\*\*\* ATM REGIONAL \*\*\*/

**PROC** **SQL**;

CREATE TABLE ATM\_2\_1 AS

SELECT CUST\_ID,

TXN\_DATE,

SUM(TOTAL\_TXN\_CNT) AS TTL\_TXN\_CNT,

SUM(TOTAL\_TXN\_AMT) AS TTL\_TXN\_AMT

FROM ATM\_WITHDRAWALS

WHERE DESCRIPTION IN ("CIMB REGIONAL","NON CIMB REGIONAL")

GROUP BY CUST\_ID, TXN\_DATE

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE ATM\_2\_2 AS

SELECT CUST\_ID,

(SUM(TTL\_TXN\_CNT)/COUNT(\*)) AS L3M\_TXN\_CNT,

(SUM(TTL\_TXN\_AMT)/COUNT(\*)) AS L3M\_TXN\_AMT

FROM ATM\_2\_1

GROUP BY CUST\_ID

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE ATM\_2\_FINAL AS

SELECT \*,

CASE WHEN L3M\_TXN\_CNT < **3** THEN "01. < 3 TIMES"

WHEN L3M\_TXN\_CNT >= **3** AND L3M\_TXN\_CNT < **5** THEN "02. 3-5 TIMES"

WHEN L3M\_TXN\_CNT >= **5** AND L3M\_TXN\_CNT <= **10** THEN "03. 5-10 TIMES"

ELSE "04. > 10 TIMES" END AS ATM\_REG\_WITHDRAWAL\_CNT,

CASE WHEN L3M\_TXN\_AMT < **200** THEN "01. < RM200"

WHEN L3M\_TXN\_AMT >= **200** AND L3M\_TXN\_AMT < **500** THEN "02. RM200 - RM500"

WHEN L3M\_TXN\_AMT >= **500** AND L3M\_TXN\_AMT < **1000** THEN "03. RM500 - RM1000"

WHEN L3M\_TXN\_AMT >= **1000** AND L3M\_TXN\_AMT < **3000** THEN "04. RM1000 - RM3000"

WHEN L3M\_TXN\_AMT >= **3000** AND L3M\_TXN\_AMT <= **5000** THEN "05. RM3000 - RM5000"

ELSE "06. > RM5000" END AS ATM\_REG\_WITHDRAWAL\_AMT

FROM ATM\_2\_2

;**QUIT**;

/\*\*\* ATM OVERSEAS \*\*\*/

**PROC** **SQL**;

CREATE TABLE ATM\_3\_1 AS

SELECT CUST\_ID,

TXN\_DATE,

SUM(TOTAL\_TXN\_CNT) AS TTL\_TXN\_CNT,

SUM(TOTAL\_TXN\_AMT) AS TTL\_TXN\_AMT

FROM ATM\_WITHDRAWALS

WHERE DESCRIPTION IN ("OVERSEAS")

GROUP BY CUST\_ID, TXN\_DATE

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE ATM\_3\_2 AS

SELECT CUST\_ID,

(SUM(TTL\_TXN\_CNT)/COUNT(\*)) AS L3M\_TXN\_CNT,

(SUM(TTL\_TXN\_AMT)/COUNT(\*)) AS L3M\_TXN\_AMT

FROM ATM\_3\_1

GROUP BY CUST\_ID

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE ATM\_3\_FINAL AS

SELECT \*,

CASE WHEN L3M\_TXN\_CNT < **3** THEN "01. < 3 TIMES"

WHEN L3M\_TXN\_CNT >= **3** AND L3M\_TXN\_CNT < **5** THEN "02. 3-5 TIMES"

WHEN L3M\_TXN\_CNT >= **5** AND L3M\_TXN\_CNT <= **10** THEN "03. 5-10 TIMES"

ELSE "04. > 10 TIMES" END AS ATM\_OVER\_WITHDRAWAL\_CNT,

CASE WHEN L3M\_TXN\_AMT < **200** THEN "01. < RM200"

WHEN L3M\_TXN\_AMT >= **200** AND L3M\_TXN\_AMT < **500** THEN "02. RM200 - RM500"

WHEN L3M\_TXN\_AMT >= **500** AND L3M\_TXN\_AMT < **1000** THEN "03. RM500 - RM1000"

WHEN L3M\_TXN\_AMT >= **1000** AND L3M\_TXN\_AMT < **3000** THEN "04. RM1000 - RM3000"

WHEN L3M\_TXN\_AMT >= **3000** AND L3M\_TXN\_AMT <= **5000** THEN "05. RM3000 - RM5000"

ELSE "06. > RM5000" END AS ATM\_OVER\_WITHDRAWAL\_AMT

FROM ATM\_3\_2

;**QUIT**;

**CLICKS FEATURES**

/\*\*\* GETTING BASE FOR BILL PAYMENTS \*\*\*/

/\* 1 . Pull Data Query \*/

**PROC** **SQL**;

CREATE TABLE EMEI\_ICAMS\_SELECTED\_BILLERS AS

SELECT \*

FROM HIVEMOD.EMEI\_ICAMS\_SELECTED\_BILLERS

WHERE CATEGORY IN ("WATERBILL","ELECTRICITY");

**QUIT**;

/\* 2. Find relevant txn from mm\_paymenthistory \*/

**PROC** **SQL**;

CREATE TABLE BILL\_TXN AS

SELECT T1.USERID,

T1.CARDNO,

T1.AMOUNT,

PUT(DATEPART(T1.TRXTIMESTAMP),YYMMN6.) AS DATE,

T2.BILLERCODE,

T2.BILLERNAME,

T2.CATEGORY

FROM MMDM.MM\_PAYMENTHISTORY AS T1

LEFT JOIN EMEI\_ICAMS\_SELECTED\_BILLERS AS T2

ON TRIM(T1.PAYEEID) = TRIM(T2.BILLERCODE)

WHERE DATEPART(T1.TRXTIMESTAMP) >= **'01OCT2019'd**

AND DATEPART(T1.TRXTIMESTAMP) < **'01APR2020'd**

AND T1.CURRENTSTATUS = 'S'

AND TRIM(T1.RECURRINGPAYMENTID) = '0' /\*to remove recurring payments\*/

AND T2.BILLERCODE NE ''

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE MAPPING AS

SELECT DISTINCT USERID,

NEWICNO

FROM MMDM.MM\_ECUSER

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE BILL\_PAYMENTS AS

SELECT A.\*,

B.NEWICNO

FROM BILL\_TXN AS A

LEFT JOIN MAPPING AS B

ON A.USERID = B.USERID

;**QUIT**;

/\*\*\*\*\* RULE 3: UTILITY BILL PAYMENTS \*\*\*\*\*/

/\*\*\* ELECTRICITY \*\*\*/

**PROC** **SQL**;

CREATE TABLE CLICKS\_1\_1 AS

SELECT NEWICNO,

DATE,

COUNT(\*) AS TTL\_CNT,

SUM(AMOUNT) AS TTL\_AMT

FROM BILL\_PAYMENTS

WHERE CATEGORY = "ELECTRICITY"

GROUP BY NEWICNO, DATE

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE CLICKS\_1\_2 AS

SELECT NEWICNO,

(SUM(TTL\_CNT)/COUNT(\*)) AS L6M\_TXN\_CNT,

(SUM(TTL\_AMT)/COUNT(\*)) AS L6M\_TXN\_AMT

FROM CLICKS\_1\_1

GROUP BY NEWICNO

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE CLICKS\_1\_FINAL AS

SELECT \*,

CASE WHEN L6M\_TXN\_AMT < **100** THEN "01. < RM100"

WHEN L6M\_TXN\_AMT >= **100** AND L6M\_TXN\_AMT < **200** THEN "02. RM100 - RM200"

WHEN L6M\_TXN\_AMT >= **200** AND L6M\_TXN\_AMT <= **300** THEN "03. RM200 - RM300"

ELSE "04. > RM300" END AS ELECTRICITY\_PYMT

FROM CLICKS\_1\_2

;**QUIT**;

/\*\*\* WATER \*\*\*/

**PROC** **SQL**;

CREATE TABLE CLICKS\_2\_1 AS

SELECT NEWICNO,

DATE,

COUNT(\*) AS TTL\_CNT,

SUM(AMOUNT) AS TTL\_AMT

FROM BILL\_PAYMENTS

WHERE CATEGORY = "WATERBILL"

GROUP BY NEWICNO, DATE

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE CLICKS\_2\_2 AS

SELECT NEWICNO,

(SUM(TTL\_CNT)/COUNT(\*)) AS L6M\_TXN\_CNT,

(SUM(TTL\_AMT)/COUNT(\*)) AS L6M\_TXN\_AMT

FROM CLICKS\_2\_1

GROUP BY NEWICNO

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE CLICKS\_2\_FINAL AS

SELECT \*,

CASE WHEN L6M\_TXN\_AMT < **100** THEN "01. < RM100"

WHEN L6M\_TXN\_AMT >= **100** AND L6M\_TXN\_AMT < **200** THEN "02. RM100 - RM200"

WHEN L6M\_TXN\_AMT >= **200** AND L6M\_TXN\_AMT <= **300** THEN "03. RM200 - RM300"

ELSE "04. > RM300" END AS WATER\_PYMT

FROM CLICKS\_2\_2

;**QUIT**;

/\*\*\* COMBINED \*\*\*/

**PROC** **SQL**;

CREATE TABLE CLICKS\_3\_1 AS

SELECT NEWICNO,

DATE,

COUNT(\*) AS TTL\_CNT,

SUM(AMOUNT) AS TTL\_AMT

FROM BILL\_PAYMENTS

GROUP BY NEWICNO, DATE

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE CLICKS\_3\_2 AS

SELECT NEWICNO,

(SUM(TTL\_CNT)/COUNT(\*)) AS L6M\_TXN\_CNT,

(SUM(TTL\_AMT)/COUNT(\*)) AS L6M\_TXN\_AMT

FROM CLICKS\_3\_1

GROUP BY NEWICNO

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE CLICKS\_3\_FINAL AS

SELECT \*,

CASE WHEN L6M\_TXN\_AMT < **100** THEN "01. < RM100"

WHEN L6M\_TXN\_AMT >= **100** AND L6M\_TXN\_AMT < **200** THEN "02. RM100 - RM200"

WHEN L6M\_TXN\_AMT >= **200** AND L6M\_TXN\_AMT <= **300** THEN "03. RM200 - RM300"

ELSE "04. > RM300" END AS UTILITY\_PYMT

FROM CLICKS\_3\_2

;**QUIT**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\* GETTING DATA FOR PAYING OTHER BANKS CC \*\*\*/

**PROC** **SQL**;

CREATE TABLE PAYING AS

SELECT A.\*,

PUT(A.RPT\_DT, YYMMN6.) AS MONTH

FROM MMDM.MM\_FUNDTRANSFERHISTORY AS A

WHERE A.TRANSFERMETHOD IN ("CC","LP") AND

A.TOACCTYPE = "" AND

A.CURRENTSTATUS = "S" AND

A.RPT\_DT >= **'1OCT2019'd**

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE PAYING1 AS

SELECT USERID,

COUNT(\*) AS TTL\_CNT,

SUM(AMOUNT) AS TTL\_AMT,

TRANSFERMETHOD,

MONTH

FROM PAYING

GROUP BY USERID, TRANSFERMETHOD, MONTH

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE MAPPING AS

SELECT DISTINCT USERID,

NEWICNO

FROM MMDM.MM\_ECUSER

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE PAYING\_OTHER\_BANKS AS

SELECT B.NEWICNO,

A.\*

FROM PAYING1 AS A

LEFT JOIN MAPPING AS B

ON A.USERID = B.USERID

;**QUIT**;

/\*\*\*\*\* RULE 5: PAYING FOR OTHER CC/LOANS WITH CLICKS \*\*\*\*\*/

**PROC** **SQL**;

CREATE TABLE CLICKS\_4\_1 AS

SELECT DISTINCT A.NEWICNO,

SUM(A.TTL\_CNT) AS COUNT,

SUM(A.TTL\_AMT) AS AMOUNT,

A.MONTH

FROM PAYING\_OTHER\_BANKS AS A

WHERE MONTH IN ("202003","202002","202001","201912","201911","201910")

GROUP BY A.NEWICNO, A.MONTH

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE CLICKS\_4\_2 AS

SELECT NEWICNO,

(SUM(COUNT)/COUNT(\*)) AS L6M\_TXN\_CNT,

(SUM(AMOUNT)/COUNT(\*)) AS L6M\_TXN\_AMT

FROM CLICKS\_4\_1

GROUP BY NEWICNO

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE CLICKS\_4\_FINAL AS

SELECT \*,

CASE WHEN L6M\_TXN\_CNT < **2** THEN "01. < 2 TIMES"

WHEN L6M\_TXN\_CNT >= **2** AND L6M\_TXN\_CNT < **4** THEN "02. 2-3 TIMES"

WHEN L6M\_TXN\_CNT >= **4** AND L6M\_TXN\_CNT < **6** THEN "03. 4-5 TIMES"

ELSE "04. > 5 TIMES" END AS PAYING\_OTHERBANK\_CC\_CNT,

CASE WHEN L6M\_TXN\_AMT < **500** THEN "01. < RM500"

WHEN L6M\_TXN\_AMT >= **500** AND L6M\_TXN\_AMT < **1000** THEN "02. RM500 - R1000"

WHEN L6M\_TXN\_AMT >= **1000** AND L6M\_TXN\_AMT < **3000** THEN "03. RM1000 - RM3000"

WHEN L6M\_TXN\_AMT >= **3000** AND L6M\_TXN\_AMT < **5000** THEN "04. RM3000 - RM5000"

WHEN L6M\_TXN\_AMT >= **5000** AND L6M\_TXN\_AMT <= **10000** THEN "05. RM5000 - RM10000"

ELSE "06. > RM10K" END AS PAYING\_OTHERBANK\_CC\_AMT

FROM CLICKS\_4\_2

;**QUIT**;

**OTHER FEATURES**

/\*\*\* EBCB DIRECTORS \*\*\*/

**DATA** OTHER\_1\_1;

SET MMDM.BDW\_PROSPECT\_LIST (keep=CFSSNO CFCIFN);

RENAME CFSSNO=CUST\_ID

CFCIFN=DIRECTOR\_COMPANY;

**RUN**;

**PROC** **SQL**;

CREATE TABLE OTHER\_1\_FINAL AS

SELECT DISTINCT CUST\_ID,

"YES" AS EBCB\_DIRECTOR

FROM OTHER\_1\_1

WHERE CUST\_ID IS NOT NULL

;**QUIT**;

/\*\*\* FCCA HOLDERS \*\*\*/

**PROC** **SQL**;

CREATE TABLE FCCA AS

SELECT A.CUST\_ID,

CASE WHEN SUM(A.FCCA\_CNT) > **0** OR SUM(B.FCCA\_CNT) > **0** OR SUM(C.FCCA\_CNT) > **0** OR SUM(D.FCCA\_CNT) > **0** OR

SUM(E.FCCA\_CNT) > **0** OR SUM(F.FCCA\_CNT) > **0** THEN **1** ELSE **0** END AS FCCA\_TAG

FROM CARV3.CAR\_AVPH\_20200331 AS A

LEFT JOIN CARV3.CAR\_AVPH\_20200229 AS B

ON A.CUST\_ID = B.CUST\_ID

LEFT JOIN CARV3.CAR\_AVPH\_20200131 AS C

ON A.CUST\_ID = C.CUST\_ID

LEFT JOIN CARV3.CAR\_AVPH\_20191231 AS D

ON A.CUST\_ID = D.CUST\_ID

LEFT JOIN CARV3.CAR\_AVPH\_20191130 AS E

ON A.CUST\_ID = E.CUST\_ID

LEFT JOIN CARV3.CAR\_AVPH\_20191031 AS F

ON A.CUST\_ID = F.CUST\_ID

WHERE A.TYPE = B.TYPE = C.TYPE = D.TYPE = E.TYPE = F.TYPE = "RB"

GROUP BY A.CUST\_ID

;**QUIT**;

**DATA** FCCA\_HOLDERS;

SET FCCA;

IF FCCA\_TAG = **1**;

**RUN**;

**PROC** **SQL**;

CREATE TABLE OTHER\_2\_FINAL AS

SELECT \*,

"YES" AS FCCA\_HOLDER

FROM FCCA\_HOLDERS

;**QUIT**;

/\*\*\* SAFEDEPOSITBOX \*\*\*/

**PROC** **SQL**;

CREATE TABLE SAFEDEPOSIT AS

SELECT \*,

PUT('CIF NO.'n, BEST14.) as CIF\_KEY\_CHAR

FROM DAPH.SAFE\_DEPOSIT\_BASE\_MAY2019

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE MAPPING\_TABLE AS

SELECT A.CIFNO,

A.PARTY\_RK,

B.CUST\_ID,

B.ALT\_ID

FROM CARV3.CAR\_CD\_20200331 AS A

LEFT JOIN CARV3.CAR\_AVPH\_20200331 AS B

ON A.PARTY\_RK = B.PARTY\_RK

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE SAFEDEPOSIT1 AS

SELECT A.\*,

B.CUST\_ID,

B.ALT\_ID

FROM SAFEDEPOSIT AS A

LEFT JOIN MAPPING\_TABLE AS B

ON A.CIF\_KEY\_CHAR = B.CIFNO

;**QUIT**;

**PROC** **SQL**;

CREATE TABLE OTHER\_3\_FINAL AS

SELECT DISTINCT CUST\_ID,

"YES" AS SAFEDEPOSITBOX

FROM SAFEDEPOSIT1

WHERE CUST\_ID IS NOT NULL

;**QUIT**;

**COMBINE TOGETHER**

/\* ADDING ATM \*/

**PROC** **SQL**;

CREATE TABLE BASE\_1 AS

SELECT A.\*,

B.ATM\_WITHDRAWAL\_CNT,

B.ATM\_WITHDRAWAL\_AMT,

C.ATM\_REG\_WITHDRAWAL\_CNT,

C.ATM\_REG\_WITHDRAWAL\_AMT,

D.ATM\_OVER\_WITHDRAWAL\_CNT,

D.ATM\_OVER\_WITHDRAWAL\_AMT

FROM BASE AS A

LEFT JOIN ATM\_1\_FINAL AS B

ON A.CUST\_ID = B.CUST\_ID

LEFT JOIN ATM\_2\_FINAL AS C

ON A.CUST\_ID = C.CUST\_ID

LEFT JOIN ATM\_3\_FINAL AS D

ON A.CUST\_ID = D.CUST\_ID

;**QUIT**;

/\* ADDING CLICKS AND CC \*/

**PROC** **SQL**;

CREATE TABLE BASE\_2 AS

SELECT A.\*,

B.ELECTRICITY\_PYMT,

C.WATER\_PYMT,

D.UTILITY\_PYMT,

E.PAYING\_OTHERBANK\_CC\_CNT,

E.PAYING\_OTHERBANK\_CC\_AMT,

F.CROSSBORDER\_CNT,

F.CROSSBORDER\_AMT,

G.PETROL\_TICKET\_SIZE,

G.PETROL\_CNT,

H.FINEDINING\_CNT,

H.FINEDINING\_AMT,

I.LUXURY\_CNT,

I.LUXURY\_AMT

FROM BASE\_1 AS A

LEFT JOIN CLICKS\_1\_FINAL AS B

ON A.CUST\_ID = B.NEWICNO

LEFT JOIN CLICKS\_2\_FINAL AS C

ON A.CUST\_ID = C.NEWICNO

LEFT JOIN CLICKS\_3\_FINAL AS D

ON A.CUST\_ID = D.NEWICNO

LEFT JOIN CLICKS\_4\_FINAL AS E

ON A.CUST\_ID = E.NEWICNO

LEFT JOIN CC\_1\_FINAL AS F

ON A.CUST\_ID = F.S\_IC\_PP

LEFT JOIN CC\_2\_FINAL AS G

ON A.CUST\_ID = G.S\_IC\_PP

LEFT JOIN CC\_3\_FINAL AS H

ON A.CUST\_ID = H.S\_IC\_PP

LEFT JOIN CC\_4\_FINAL AS I

ON A.CUST\_ID = I.S\_IC\_PP

;**QUIT**;

/\* ADDING OTHERS \*/

**PROC** **SQL**;

CREATE TABLE BASE\_3 AS

SELECT A.\*,

B.EBCB\_DIRECTOR,

C.FCCA\_HOLDER,

D.SAFEDEPOSITBOX

FROM BASE\_2 AS A

LEFT JOIN OTHER\_1\_FINAL AS B

ON A.CUST\_ID = B.CUST\_ID

LEFT JOIN OTHER\_2\_FINAL AS C

ON A.CUST\_ID = C.CUST\_ID

LEFT JOIN OTHER\_3\_FINAL AS D

ON A.CUST\_ID = D.CUST\_ID

;**QUIT**;

/\* CODE TO CHANGE NULL TO N/A - RUN ONCE ALL COLUMNS IN\*/

**DATA** BASE\_4;

SET BASE\_3;

ARRAY CHANGE \_CHARACTER\_;

DO OVER CHANGE;

IF CHANGE = "" THEN CHANGE = "N/A";

END;

**RUN**;

/\* SENDING TO HIVE \*/

**DATA** HIVEMOD.DAPH\_AA\_202003;

SET BASE\_4;

**RUN**;

**Once final data is sent to Hive, then just run the Python script to get output.**