**A Project Report on**

**Coupon (Buckets) Addition/Replacement on the basis of Gross Revenue (GR %) generated from hotels.**

**For,**

****

(June – July 2019)

**Presented and Developed by:**

**Swapnil Johri**

**Reg. No. : 10617210016**

**B.Tech: Computer Science & Engineering with specialization in Big Data & Analytics.**

**Batch: (2017 – 2021)**



**Certificate**

This is to certify that Swapnil Johri (10617210016) have successfully completed his project titled ‘Coupon (Buckets) Addition/Replacement on the basis of Gross Revenue (GR %) generated from hotels’, during his summer internship/industrial training period at **Make My Trip, Gurgaon** (June – July 2019), after 4th Semester, B.Tech (Computer Science & Engineering with specialization in Big Data & Analytics) of SRM University, Delhi-NCR, Sonepat, Haryana.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Dr. Ajay Sharma**

H.O.D, Computer Science

SRM University, Delhi-NCR, Sonepat, Haryana

**Acknowledgment**

I deem it a pleasure to acknowledge my sense of gratitude to my H.O.D., Dr. Ajay Sharma; who gave his consent, to do internship/industrial training under the esteemed organization, Make My Trip.

I would also like to grab this opportunity to thank my Reporting Manger(during my internship period); Mr. Prakhar Kasar (Director, International Hotels), & Mr. Kewal Krishnan (Data Analyst, Online Revenue, International Hotels); who always helped me & was there for me every time I needed any kind of help.

Finally, I must say that no height is ever achieved without some sacrifices made at some end & it is here, where I owe my special debt to my parents & friends for showing their generous love & care throughout the entire period of time.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Swapnil Johri**

**10617210016**

**About the Organization**

Make My Trip Limited is an Indian online travel company founded in 2000. Headquartered in Gurgaon, Haryana, the company provides online travel services including flight tickets, domestic and international holiday packages, hotel reservations, and rail and bus tickets. As of 31 March 2018, they have 14 company-owned travel stores in 14 cities, over 30 franchisee-owned travel stores in 28 cities, and counters in four major airports in India. Make My Trip has offices in New York, Singapore, Kuala Lumpur, Phuket, Bangkok, and Dubai.

Make My Trip was founded by Deep Kalra, an alumnus of IIM-Ahmedabad. It was launched in the US market in the year 2000 to cater to the overseas Indian community for their US to India travel needs. Make My Trip started Indian operations in September 2005, offering online flight tickets to Indian travelers. The company also started to focus on non-air businesses like holiday packages and hotel bookings. On 17 August 2010, Make My Trip Limited was listed on the NASDAQ after its initial public offering.

In 2016, Make My Trip and Ibibo Group, India's largest travel booking portals, merged through a stock transaction. Through this transaction, Make My Trip acquired its rival, Ibibo Group, in one of the biggest acquisition in India’s online travel space. Post-merger, Make My Trip shareholders own a sixty percent stake and Ibibo Group shareholders get a forty percent stake. Naspers and Tencent, who jointly owned Ibibo Group, became the single largest shareholder in Make My Trip, giving birth to a new group **Go-MMT**.

**** 

**Description of Task**

Coupon or simply ‘Bucket’ is the term in the organization which is used for discounts given to the customer; known as **Sales Promotion (SP)**.

For each hotel booking through Make My Trip (MMT), the hotel owes some commission to the organization; known as **Gross Revenue (GR)** in organization’s terms. On the basis of this GR, SP is decided. Since, if the SP is high; the customers will get more discount & more customers will be attracted. But on the other hand, there will be less profit for the organization. Similarly, if the SP is low; there will be more profit for the organization but less number of customers will be attracted. So, the company has to maintain a balance between this.

Since, GR collected from each hotel is different; so to set SP for each hotel is a very difficult task (There are around 500000 hotels in India only). So, for a range of GR, the organization sets a type of SP. This way, the GR is divided into several ranges, & each range is called a ‘Bucket’; on which a type of coupon is set.

Now for each bucket, there are different type of SP’s based organization policies. Also, the GR keeps changing based on market/booking conditions. For this, SP’s needs to be updated regularly.

The challenges with the task:

* All this addition/replacement of coupons has to be done manually.
* This manually updation of coupons require a human effort of around 4-5 hours/week.
* New hotels keep on adding in the organization’s server every week. So there is no easy way of tracking this. So the bucket updation task becomes even hard for these hotels.
* Since this process is done manually, there is a lot of scope for human error.

Solution for the challenges:

* An overall automated & consistent system, which changes/replaces the coupon bucket by itself, based on set criteria.
* A self-scheduled system which does not require regular maintenance/lookup.
* The hotel list (to be updated) must be updated regularly so that there is no scope of tracking it manually.

**Formulas & Lookups used in the Project**

* **Gross Market Value (GMV)** = (Total Customer Cost + E-Coupon Amount + Promo Amount) / Conversion Rate
* **Gross Revenue(GR)** = (Total Customer Cost + E-Coupon Amount + Promo Amount – Pay to Hotel Cost – Service Tax) / Conversion Rate
* **GR%** = GR/GMV \* 100
* **List of bucket types used in project :**

1. INTL\_GR
2. IHM
3. INTL\_APPONLY
4. INTL\_INDIAIP
5. INTL\_SDND

* **Sample lookup format used in the project. (Numeric values indicates the GR% value)**

|  |  |  |
| --- | --- | --- |
| Greater\_Than\_ET | Lesser\_Than | Coupon |
| 0.00 | 3.50 | INTL\_GR\_4 |
| 3.50 | 5.70 | INTL\_GR\_6 |
| 5.70 | 6.80 | INTL\_GR\_8 |
| 6.80 | 7.60 | INTL\_GR\_12 |
| 7.60 | 9.10 | INTL\_GR\_15 |
| 9.10 | 10.60 | INTL\_GR\_18 |
| 10.60 | 12.10 | INTL\_GR\_21 |
| 12.10 | 13.60 | INTL\_GR\_high |
| 13.60 | 15.10 | INTL\_GR\_high2 |
| 15.10 | 16.60 | INTL\_GR\_high3 |
| 16.60 |  | INTL\_GR\_high4 |

|  |  |  |
| --- | --- | --- |
| Greater\_Than\_ET | Lesser\_Than | Coupon |
| 0.00 | 3.20 | IHM0 |
| 3.20 | 6.00 | IHM1 |
| 6.00 | 6.10 | IHM2 |
| 6.10 | 9.90 | IHM3 |
| 9.90 | 10.40 | IHM4 |
| 10.40 | 12.50 | IHM5 |
| 12.50 | 14.00 | IHM6 |
| 14.00 | 15.30 | IHM7 |
| 15.30 | 17.40 | IHM8 |
| 17.40 | 19.00 | IHM9 |
| 19.00 | 21.50 | IHM10 |
| 21.50 | 22.40 | IHM12 |
| 22.40 | 24.40 | IHM13 |
| 24.40 | 26.10 | IHM14 |
| 26.10 | 28.40 | IHM15 |
| 28.40 | 30.10 | IHM16 |
| 30.10 |  | IHM17 |

**2.**

|  |  |  |
| --- | --- | --- |
| Greater\_Than\_ET | Lesser\_Than | Coupon |
| 0.00 | 6.10 | INTL\_APPONLY0 |
| 6.10 | 7.60 | INTL\_APPONLY |
| 7.60 | 9.10 | INTL\_APPONLY2 |
| 9.10 | 10.60 | INTL\_APPONLY3 |
| 10.60 | 12.10 | INTL\_APPONLY4 |
| 12.10 | 13.60 | INTL\_APPONLY5 |
| 13.60 | 15.10 | INTL\_APPONLY6 |
| 15.10 | 17.30 | INTL\_APPONLY7 |
| 17.30 |  | INTL\_APPONLY8 |

**3.**

|  |  |  |
| --- | --- | --- |
| Greater\_Than\_ET | Lesser\_Than | Coupon |
| 0.00 | 6.10 | INTL\_INDIAIP0 |
| 6.10 | 7.60 | INTL\_INDIAIP1 |
| 7.60 | 9.10 | INTL\_INDIAIP2 |
| 9.10 | 10.60 | INTL\_INDIAIP3 |
| 10.60 | 12.10 | INTL\_INDIAIP4 |
| 12.10 | 13.60 | INTL\_INDIAIP5 |
| 13.60 | 15.10 | INTL\_INDIAIP6 |
| 15.10 | 17.30 | INTL\_INDIAIP7 |
| 17.30 |  | INTL\_INDIAIP8 |

**4.**

|  |  |  |
| --- | --- | --- |
| Greater\_Than\_ET | Lesser\_Than | Coupon |
| 0.00 | 6.10 | INTL\_SDND0 |
| 6.10 | 7.60 | INTL\_SDND1 |
| 7.60 | 9.10 | INTL\_SDND2 |
| 9.10 | 10.60 | INTL\_SDND3 |
| 10.60 | 12.10 | INTL\_SDND4 |
| 12.10 | 13.60 | INTL\_SDND5 |
| 13.60 | 15.10 | INTL\_SDND6 |
| 15.10 | 17.30 | INTL\_SDND7 |
| 17.30 |  | INTL\_SDND8 |

**5.**

**About the Project**

The project aims to overcome the existing challenges & provide a self-automated, consistent & self-scheduled system.

The overall project runs in two phases:

1. Retrieval of Hotel ID, City Code, Attribute Value (Bucket list for the hotel), GR & GMV for each hotel for the last 30 days (then calculate GR%), from organization’s database.

The retrieval of data first retrieves the data for last 7 days, then looks up for the hotel for last 14 days, which are not booked in last 7 days. And similarly, it retrieves data for last 30 days, which are not booked in last 14 days.

This gives a list of unique hotels for last 30 days which are recently updated.

**This phase of the project runs on SQL.**

1. Replacement of existing bucket with new one, or addition of new bucket if it doesn’t exists.

This replacement is done on the basis of lookup for GR%, provided by the organization.

**This phase of the project runs on Python(Pandas Module).**

After the retrieval of required data from the database, it is imported & converted into a data-frame using Requests & Pandas library in Python. This is data-frame is then traversed through each row and coupon value for each hotel is updated.

A lookup is provided prior to program run as a criteria for the coupon replacement program.

In the end, the final output is then stored in the computer memory as a .csv file; where it can handled/looked up for personal reference.

* **All the codes below uses path for the system that was used during the development for the project.**

**SQL – Data Retrieval Code**

**select \* from**

**((select BKG\_HTLSEQ, BKG\_CITYCD, round((sum(BKG\_TOTMMTCUSTCOST + ifnull(BKG\_ECOUPONAMT, 0) + ifnull(BKP\_PROMOAMT,0))/BKG\_CONVRATE), 2) as GMV,**

**round((sum(BKG\_TOTMMTCUSTCOST + ifnull(BKG\_ECOUPONAMT, 0) + ifnull(BKP\_PROMOAMT,0) - BKG\_MMTPAYHTLCOST - BKG\_SERVICETAX)/BKG\_CONVRATE), 2) as GR,**

**round(((sum(BKG\_TOTMMTCUSTCOST + ifnull(BKG\_ECOUPONAMT, 0) + ifnull(BKP\_PROMOAMT,0) - BKG\_MMTPAYHTLCOST - BKG\_SERVICETAX)/BKG\_CONVRATE)/(sum(BKG\_TOTMMTCUSTCOST + ifnull(BKG\_ECOUPONAMT, 0) + ifnull(BKP\_PROMOAMT,0))/BKG\_CONVRATE)) \* 100, 2) as GRASP, ATTRIBUTE\_VALUE**

**from booking b**

**inner join htl\_extended\_info hei**

**on hei.HTLSEQ = b.BKG\_HTLSEQ**

**inner join booking\_promotion bp**

**on bp.BKP\_BKINGSEQ = b.BKG\_BKINGSEQ**

**where (BKG\_MMTCNFMID = 'NH2%' or BKG\_MMTCNFMID = 'NH3%' or BKG\_MMTCNFMID = 'NH7%' or BKG\_MMTCNFMID != 'NH27%') and**

**(BKG\_FNAME not like "test%" or BKG\_HTLNAME not like "test%") and BKG\_BKINGDT >= subdate(curdate(), INTERVAL 7 DAY) and (ATTRIBUTE\_TYPE = 'HOTEL\_ECOUPON\_CATEGORY')**

**and BKG\_CTYCD not in ('IN','GL') and bkg\_orgbkingstat = 'confirmed' and BKG\_CITYCD != 'HC20100719133546966' and bkg\_bkingsrc in ('b2ctvg','Hotel','b2ctab','mob') and bkg\_cityname != 'testingcityc'**

**and bkg\_mmtpayhtlcost!='0' and (BKG\_EMAIL not like '%-QA@makemytrip.com' and BKG\_EMAIL not like 'computgreen1111222@gmail.com'**

**and BKG\_EMAIL not like 'reachemailherenow0201@gmail.com' and BKG\_EMAIL not like 'emailgreen2020@gmail.com'**

**and BKG\_EMAIL not like 'comput.emailgreen221122@gmail.com' and bkg\_email not like 'emailgreen.idfornet212@gmail.com'**

**and bkg\_email not like 'computemailaddress.here212@gmail.com' and bkg\_email not like 'Rentaavis@gmail.com'**

**and bkg\_Email not like 'Rentavis@prodigy.net.mx') and (BKG\_MMTCNFMID not like 'NH27%')**

**and (BKG\_ORGBKINGSTAT not in ('FAILED'))**

**and bkg\_htlname not like '%HKTTET%'**

**group by(BKG\_HTLSEQ)**

**UNION**

**select BKG\_HTLSEQ, BKG\_CITYCD, round((sum(BKG\_TOTMMTCUSTCOST + ifnull(BKG\_ECOUPONAMT, 0) + ifnull(BKP\_PROMOAMT,0))/BKG\_CONVRATE), 2) as GMV,**

**round((sum(BKG\_TOTMMTCUSTCOST + ifnull(BKG\_ECOUPONAMT, 0) + ifnull(BKP\_PROMOAMT,0) - BKG\_MMTPAYHTLCOST - BKG\_SERVICETAX)/BKG\_CONVRATE), 2) as GR,**

**round(((sum(BKG\_TOTMMTCUSTCOST + ifnull(BKG\_ECOUPONAMT, 0) + ifnull(BKP\_PROMOAMT,0) - BKG\_MMTPAYHTLCOST - BKG\_SERVICETAX)/BKG\_CONVRATE)/(sum(BKG\_TOTMMTCUSTCOST + ifnull(BKG\_ECOUPONAMT, 0) + ifnull(BKP\_PROMOAMT,0))/BKG\_CONVRATE)) \* 100, 2) as GRASP, ATTRIBUTE\_VALUE**

**from booking b**

**inner join htl\_extended\_info hei**

**on hei.HTLSEQ = b.BKG\_HTLSEQ**

**inner join booking\_promotion bp**

**on bp.BKP\_BKINGSEQ = b.BKG\_BKINGSEQ**

**where (BKG\_MMTCNFMID = 'NH2%' or BKG\_MMTCNFMID = 'NH3%' or BKG\_MMTCNFMID = 'NH7%' or BKG\_MMTCNFMID != 'NH27%') and**

**(BKG\_FNAME not like "test%" or BKG\_HTLNAME not like "test%") and BKG\_BKINGDT >= subdate(curdate(), INTERVAL 14 DAY) and BKG\_BKINGDT < subdate(curdate(), INTERVAL 7 DAY)**

**and (ATTRIBUTE\_TYPE = 'HOTEL\_ECOUPON\_CATEGORY')**

**and BKG\_CTYCD not in ('IN','GL') and bkg\_orgbkingstat = 'confirmed' and BKG\_CITYCD != 'HC20100719133546966' and bkg\_bkingsrc in ('b2ctvg','Hotel','b2ctab','mob') and bkg\_cityname != 'testingcityc'**

**and bkg\_mmtpayhtlcost!='0' and (BKG\_EMAIL not like '%-QA@makemytrip.com' and BKG\_EMAIL not like 'computgreen1111222@gmail.com'**

**and BKG\_EMAIL not like 'reachemailherenow0201@gmail.com' and BKG\_EMAIL not like 'emailgreen2020@gmail.com'**

**and BKG\_EMAIL not like 'comput.emailgreen221122@gmail.com' and bkg\_email not like 'emailgreen.idfornet212@gmail.com'**

**and bkg\_email not like 'computemailaddress.here212@gmail.com' and bkg\_email not like 'Rentaavis@gmail.com'**

**and bkg\_Email not like 'Rentavis@prodigy.net.mx') and (BKG\_MMTCNFMID not like 'NH27%')**

**and (BKG\_ORGBKINGSTAT not in ('FAILED'))**

**and bkg\_htlname not like '%HKTTET%'**

**group by(BKG\_HTLSEQ))**

**union**

**select BKG\_HTLSEQ, BKG\_CITYCD, round((sum(BKG\_TOTMMTCUSTCOST + ifnull(BKG\_ECOUPONAMT, 0) + ifnull(BKP\_PROMOAMT,0))/BKG\_CONVRATE), 2) as GMV,**

**round((sum(BKG\_TOTMMTCUSTCOST + ifnull(BKG\_ECOUPONAMT, 0) + ifnull(BKP\_PROMOAMT,0) - BKG\_MMTPAYHTLCOST - BKG\_SERVICETAX)/BKG\_CONVRATE), 2) as GR,**

**round(((sum(BKG\_TOTMMTCUSTCOST + ifnull(BKG\_ECOUPONAMT, 0) + ifnull(BKP\_PROMOAMT,0) - BKG\_MMTPAYHTLCOST - BKG\_SERVICETAX)/BKG\_CONVRATE)/(sum(BKG\_TOTMMTCUSTCOST + ifnull(BKG\_ECOUPONAMT, 0) + ifnull(BKP\_PROMOAMT,0))/BKG\_CONVRATE)) \* 100, 2) as GRASP, ATTRIBUTE\_VALUE**

**from booking b**

**inner join htl\_extended\_info hei**

**on hei.HTLSEQ = b.BKG\_HTLSEQ**

**inner join booking\_promotion bp**

**on bp.BKP\_BKINGSEQ = b.BKG\_BKINGSEQ**

**where (BKG\_MMTCNFMID = 'NH2%' or BKG\_MMTCNFMID = 'NH3%' or BKG\_MMTCNFMID = 'NH7%' or BKG\_MMTCNFMID != 'NH27%') and**

**(BKG\_FNAME not like "test%" or BKG\_HTLNAME not like "test%") and BKG\_BKINGDT >= subdate(curdate(), INTERVAL 29 DAY) and BKG\_BKINGDT < subdate(curdate(), INTERVAL 14 DAY) and (ATTRIBUTE\_TYPE = 'HOTEL\_ECOUPON\_CATEGORY')**

**and BKG\_CTYCD not in ('IN','GL') and bkg\_orgbkingstat = 'confirmed' and BKG\_CITYCD != 'HC20100719133546966' and bkg\_bkingsrc in ('b2ctvg','Hotel','b2ctab','mob') and bkg\_cityname != 'testingcityc'**

**and bkg\_mmtpayhtlcost!='0' and (BKG\_EMAIL not like '%-QA@makemytrip.com' and BKG\_EMAIL not like 'computgreen1111222@gmail.com'**

**and BKG\_EMAIL not like 'reachemailherenow0201@gmail.com' and BKG\_EMAIL not like 'emailgreen2020@gmail.com'**

**and BKG\_EMAIL not like 'comput.emailgreen221122@gmail.com' and bkg\_email not like 'emailgreen.idfornet212@gmail.com'**

**and bkg\_email not like 'computemailaddress.here212@gmail.com' and bkg\_email not like 'Rentaavis@gmail.com'**

**and bkg\_Email not like 'Rentavis@prodigy.net.mx') and (BKG\_MMTCNFMID not like 'NH27%')**

**and (BKG\_ORGBKINGSTAT not in ('FAILED'))**

**and bkg\_htlname not like '%HKTTET%'**

**group by(BKG\_HTLSEQ)) a group by(BKG\_HTLSEQ);**

**The image displays the results generated by the SQL Query.**

**Python – Data Manipulation Code**

**import pandas as pd**

**import os**

**import requests**

**import time**

**import urllib.parse**

**def poll\_job(s, redash\_url, job):**

**# TODO: add timeout**

**while job['status'] not in (3,4):**

**response = s.get('{}/api/jobs/{}'.format(redash\_url, job['id']))**

**job = response.json()['job']**

**time.sleep(1)**

**if job['status'] == 3:**

**return job['query\_result\_id']**

**return None**

**def get\_fresh\_query\_result(redash\_url, query\_id, api\_key):**

**s = requests.Session()**

**s.headers.update({'Authorization': 'Key {}'.format(api\_key)})**

**response = s.post('{}/api/queries/{}/refresh'.format(redash\_url, query\_id))**

**if response.status\_code != 200:**

**raise Exception('Refresh failed.')**

**result\_id = poll\_job(s, redash\_url, response.json()['job'])**

**if result\_id:**

**response = s.get('{}/api/queries/{}/results/{}.json'.format(redash\_url, query\_id, result\_id))**

**if response.status\_code != 200:**

**raise Exception('Failed getting results.')**

**else:**

**raise Exception('Query execution failed.')**

**return response.json()['query\_result']['data']['rows']**

**api\_key = '8IrjI06J618heUl05YG1uKlhCDy0MFTuXC5Y3Y73'**

**df = pd.DataFrame.from\_dict(get\_fresh\_query\_result('http://redash.mmt.com/', 4389, api\_key))**

**df = df[['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GMV', 'GR', 'GRASP', 'ATTRIBUTE\_VALUE']]**

**df1 = df['ATTRIBUTE\_VALUE'].str.split(',', expand = True)**

**df1 = df1.fillna('N/A')**

**df['GR\_Bucket'] = ' '**

**df['IHM\_Bucket'] = ' '**

**df['DND\_Bucket'] = ' '**

**df['APPONLY\_Bucket'] = ' '**

**df['INDIAIP\_Bucket'] = ' '**

**df['SDND\_Bucket'] = ' '**

**df['New\_GR'] = ' '**

**a = []**

**b = []**

**dfgr = pd.read\_excel('C:/Users/int802/Desktop/GR% wise Buckets.xlsx', sheet\_name = 'GR')**

**dfihm = pd.read\_excel('C:/Users/int802/Desktop/GR% wise Buckets.xlsx', sheet\_name = 'IHM')**

**dfao = pd.read\_excel('C:/Users/int802/Desktop/GR% wise Buckets.xlsx', sheet\_name = 'APPONLY')**

**dfii = pd.read\_excel('C:/Users/int802/Desktop/GR% wise Buckets.xlsx', sheet\_name = 'INDIAIP')**

**dfs = pd.read\_excel('C:/Users/int802/Desktop/GR% wise Buckets.xlsx', sheet\_name = 'SDND')**

**dfgr = dfgr.fillna(100)**

**dfihm = dfihm.fillna(100)**

**dfao = dfao.fillna(100)**

**dfii = dfii.fillna(100)**

**dfs = dfs.fillna(100)**

**#TO CREATE A SEPARATE COLUMN FOR GR, IHM, DND, APPONLY, INDIAIP, SDND**

**def GR():**

**lgr = []**

**lgr1 = []**

**for index, row in df1.iterrows():**

**for i in df1.columns:**

**if('INTL\_GR' in row[i] or 'INTL\_Gr' in row[i]):**

**lgr1.append(row[i])**

**lgr.append(lgr1)**

**lgr1 = []**

**df['GR\_Bucket'] = lgr**

**def IHM():**

**lihm = []**

**lihm1 = []**

**for index, row in df1.iterrows():**

**for i in df1.columns:**

**if('IHM' in row[i]):**

**lihm1.append(row[i])**

**lihm.append(lihm1)**

**lihm1 = []**

**df['IHM\_Bucket'] = lihm**

**def DND():**

**ldnd = []**

**ldnd1 = []**

**for index, row in df1.iterrows():**

**for i in df1.columns:**

**if('INTL\_DND' in row[i]):**

**ldnd1.append(row[i])**

**ldnd.append(ldnd1)**

**ldnd1 = []**

**df['DND\_Bucket'] = ldnd**

**def APPONLY():**

**lao = []**

**lao1 = []**

**for index, row in df1.iterrows():**

**for i in df1.columns:**

**if('INTL\_APPONLY' in row[i]):**

**lao1.append(row[i])**

**lao.append(lao1)**

**lao1 = []**

**df['APPONLY\_Bucket'] = lao**

**def INDIAIP():**

**liip = []**

**liip1 = []**

**for index, row in df1.iterrows():**

**for i in df1.columns:**

**if('INTL\_INDIAIP' in row[i]):**

**liip1.append(row[i])**

**liip.append(liip1)**

**liip1 = []**

**df['INDIAIP\_Bucket'] = liip**

**def SDND():**

**lsdnd = []**

**lsdnd1 = []**

**for index, row in df1.iterrows():**

**for i in df1.columns:**

**if('INTL\_SDND' in row[i]):**

**lsdnd1.append(row[i])**

**lsdnd.append(lsdnd1)**

**lsdnd1 = []**

**df['SDND\_Bucket'] = lsdnd**

**GR()**

**IHM()**

**DND()**

**APPONLY()**

**INDIAIP()**

**SDND()**

**# #TO CREATE A NEW COLUMN FOR EACH(MENTIONED ABOVE) NEW BUCKET**

**for index, row in df.iterrows():**

**for index2, row2 in dfgr.iterrows():**

**if(row2.Greater\_Than\_ET <= row.GRASP < row2.Lesser\_Than):**

**row.New\_GR = row.New\_GR.replace(' ', row2.Coupon)**

**elif(row.GRASP < 0):**

**row.New\_GR = row.New\_GR.replace(' ', 'INTL\_GR\_4')**

**a = list(row)**

**b.append(a)**

**df2 = pd.DataFrame(b, columns = ['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GMV', 'GR', 'GRASP', 'ATTRIBUTE\_VALUE', 'GR\_Bucket' ,'IHM\_Bucket', 'DND\_Bucket', 'APPONLY\_Bucket', 'INDIAIP\_Bucket', 'SDND\_Bucket', 'New\_GR'])**

**df = df2**

**del df2**

**a = []**

**b = []**

**df['New\_IHM'] = ' '**

**for index, row in df.iterrows():**

**for index2, row2 in dfihm.iterrows():**

**if(row2.Greater\_Than\_ET <= row.GRASP < row2.Lesser\_Than):**

**row.New\_IHM = row.New\_IHM.replace(' ', row2.Coupon)**

**elif(row.GRASP < 0):**

**row.New\_IHM = row.New\_IHM.replace(' ', 'IHM0')**

**a = list(row)**

**b.append(a)**

**df2 = pd.DataFrame(b, columns = ['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GMV', 'GR', 'GRASP', 'ATTRIBUTE\_VALUE', 'GR\_Bucket' ,'IHM\_Bucket', 'DND\_Bucket', 'APPONLY\_Bucket', 'INDIAIP\_Bucket', 'SDND\_Bucket', 'New\_GR', 'New\_IHM'])**

**df = df2**

**del df2**

**a = []**

**b = []**

**df['New\_APPONLY'] = ' '**

**for index, row in df.iterrows():**

**if(len(row.DND\_Bucket) != 0):**

**for index2, row2 in dfao.iterrows():**

**if(row2.Greater\_Than\_ET <= row.GRASP < row2.Lesser\_Than):**

**row.New\_APPONLY = row.New\_APPONLY.replace(' ', row2.Coupon)**

**elif(row.GRASP < 0):**

**row.New\_APPONLY = row.New\_APPONLY.replace(' ', 'INTL\_APPONLY0')**

**elif(len(row.DND\_Bucket) == 0 and len(row.APPONLY\_Bucket) !=0):**

**row.New\_APPONLY = row.New\_APPONLY.replace(' ', row.APPONLY\_Bucket[0])**

**a = list(row)**

**b.append(a)**

**df2 = pd.DataFrame(b, columns = ['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GMV', 'GR', 'GRASP', 'ATTRIBUTE\_VALUE', 'GR\_Bucket' ,'IHM\_Bucket', 'DND\_Bucket', 'APPONLY\_Bucket', 'INDIAIP\_Bucket', 'SDND\_Bucket', 'New\_GR', 'New\_IHM', 'New\_APPONLY'])**

**df = df2**

**del df2**

**a = []**

**b = []**

**df['New\_INDIAIP'] = ' '**

**for index, row in df.iterrows():**

**if(len(row.DND\_Bucket) != 0):**

**for index2, row2 in dfii.iterrows():**

**if(row2.Greater\_Than\_ET <= row.GRASP < row2.Lesser\_Than):**

**row.New\_INDIAIP = row.New\_INDIAIP.replace(' ', row2.Coupon)**

**elif(row.GRASP < 0):**

**row.New\_INDIAIP = row.New\_INDIAIP.replace(' ', 'INTL\_INDIAIP0')**

**elif(len(row.DND\_Bucket) == 0 and len(row.INDIAIP\_Bucket) !=0):**

**row.New\_INDIAIP = row.New\_INDIAIP.replace(' ', row.INDIAIP\_Bucket[0])**

**a = list(row)**

**b.append(a)**

**df2 = pd.DataFrame(b, columns = ['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GMV', 'GR', 'GRASP', 'ATTRIBUTE\_VALUE', 'GR\_Bucket' ,'IHM\_Bucket', 'DND\_Bucket', 'APPONLY\_Bucket', 'INDIAIP\_Bucket', 'SDND\_Bucket', 'New\_GR', 'New\_IHM', 'New\_APPONLY', 'New\_INDIAIP'])**

**df = df2**

**del df2**

**a = []**

**b = []**

**df['New\_SDND'] = ' '**

**for index, row in df.iterrows():**

**if(len(row.DND\_Bucket) != 0):**

**for index2, row2 in dfs.iterrows():**

**if(row2.Greater\_Than\_ET <= row.GRASP < row2.Lesser\_Than):**

**row.New\_SDND = row.New\_SDND.replace(' ', row2.Coupon)**

**elif(row.GRASP < 0):**

**row.New\_SDND = row.New\_SDND.replace(' ', 'INTL\_SDND0')**

**elif(len(row.DND\_Bucket) == 0 and len(row.SDND\_Bucket) !=0):**

**row.New\_SDND = row.New\_SDND.replace(' ', row.SDND\_Bucket[0])**

**a = list(row)**

**b.append(a)**

**df2 = pd.DataFrame(b, columns = ['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GMV', 'GR', 'GRASP', 'ATTRIBUTE\_VALUE', 'GR\_Bucket' ,'IHM\_Bucket', 'DND\_Bucket', 'APPONLY\_Bucket', 'INDIAIP\_Bucket', 'SDND\_Bucket', 'New\_GR', 'New\_IHM', 'New\_APPONLY', 'New\_INDIAIP', 'New\_SDND'])**

**df = df2**

**del df2**

**a = []**

**b = []**

**#TO REPLACE THE EXISTING BUCKET IN 'ATTRIBUTE\_VALUE' WITH NEW BUCKET**

**for index, row in df.iterrows():**

**if(len(row.GR\_Bucket) != 0):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(row.GR\_Bucket[0], row.New\_GR)**

**if(len(row.GR\_Bucket) == 2):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(row.GR\_Bucket[1], '', 1)**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(',,', ',')**

**elif(row.New\_GR != ' '):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE + ',' + row.New\_GR**

**a = list(row)**

**b.append(a)**

**df2 = pd.DataFrame(b, columns = ['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GMV', 'GR', 'GRASP', 'ATTRIBUTE\_VALUE', 'GR\_Bucket' ,'IHM\_Bucket', 'DND\_Bucket', 'APPONLY\_Bucket', 'INDIAIP\_Bucket', 'SDND\_Bucket', 'New\_GR', 'New\_IHM', 'New\_APPONLY', 'New\_INDIAIP', 'New\_SDND'])**

**df = df2**

**del df2**

**a = []**

**b = []**

**for index, row in df.iterrows():**

**if(len(row.IHM\_Bucket) != 0):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(row.IHM\_Bucket[0], row.New\_IHM)**

**if(len(row.IHM\_Bucket) == 2):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(row.IHM\_Bucket[1], '', 1)**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(',,', ',')**

**elif(row.New\_IHM != ' '):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE + ',' + row.New\_IHM**

**a = list(row)**

**b.append(a)**

**df2 = pd.DataFrame(b, columns = ['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GMV', 'GR', 'GRASP', 'ATTRIBUTE\_VALUE', 'GR\_Bucket' ,'IHM\_Bucket', 'DND\_Bucket', 'APPONLY\_Bucket', 'INDIAIP\_Bucket', 'SDND\_Bucket', 'New\_GR', 'New\_IHM', 'New\_APPONLY', 'New\_INDIAIP', 'New\_SDND'])**

**df = df2**

**del df2**

**a = []**

**b = []**

**for index, row in df.iterrows():**

**if(len(row.APPONLY\_Bucket) != 0):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(row.APPONLY\_Bucket[0], row.New\_APPONLY)**

**if(len(row.APPONLY\_Bucket) == 2):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(row.APPONLY\_Bucket[1], '', 1)**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(',,', ',')**

**elif(row.New\_APPONLY != ' '):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE + ',' + row.New\_APPONLY**

**a = list(row)**

**b.append(a)**

**df2 = pd.DataFrame(b, columns = ['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GMV', 'GR', 'GRASP', 'ATTRIBUTE\_VALUE', 'GR\_Bucket' ,'IHM\_Bucket', 'DND\_Bucket', 'APPONLY\_Bucket', 'INDIAIP\_Bucket', 'SDND\_Bucket', 'New\_GR', 'New\_IHM', 'New\_APPONLY', 'New\_INDIAIP', 'New\_SDND'])**

**df = df2**

**del df2**

**a = []**

**b = []**

**for index, row in df.iterrows():**

**if(len(row.INDIAIP\_Bucket) != 0):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(row.INDIAIP\_Bucket[0], row.New\_INDIAIP)**

**if(len(row.INDIAIP\_Bucket) == 2):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(row.INDIAIP\_Bucket[1], '', 1)**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(',,', ',')**

**elif(row.New\_INDIAIP != ' ') :**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE + ',' + row.New\_INDIAIP**

**a = list(row)**

**b.append(a)**

**df2 = pd.DataFrame(b, columns = ['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GMV', 'GR', 'GRASP', 'ATTRIBUTE\_VALUE', 'GR\_Bucket' ,'IHM\_Bucket', 'DND\_Bucket', 'APPONLY\_Bucket', 'INDIAIP\_Bucket', 'SDND\_Bucket', 'New\_GR', 'New\_IHM', 'New\_APPONLY', 'New\_INDIAIP', 'New\_SDND'])**

**df = df2**

**del df2**

**a = []**

**b = []**

**for index, row in df.iterrows():**

**if(len(row.SDND\_Bucket) != 0):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(row.SDND\_Bucket[0], row.New\_SDND)**

**if(len(row.SDND\_Bucket) == 2):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(row.SDND\_Bucket[1], '', 1)**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(',,', ',')**

**elif(row.New\_SDND != ' ') :**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE + ',' + row.New\_SDND**

**a = list(row)**

**b.append(a)**

**df2 = pd.DataFrame(b, columns = ['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GMV', 'GR', 'GRASP', 'ATTRIBUTE\_VALUE', 'GR\_Bucket' ,'IHM\_Bucket', 'DND\_Bucket', 'APPONLY\_Bucket', 'INDIAIP\_Bucket', 'SDND\_Bucket', 'New\_GR', 'New\_IHM', 'New\_APPONLY', 'New\_INDIAIP', 'New\_SDND'])**

**df = df2**

**del df2**

**a = []**

**b = []**

**#To remove extra ,'s from data frame**

**for index, row in df.iterrows():**

**if(',,,' in row.ATTRIBUTE\_VALUE):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(',,,', ',')**

**if(',,' in row.ATTRIBUTE\_VALUE):**

**row.ATTRIBUTE\_VALUE = row.ATTRIBUTE\_VALUE.replace(',,', ',')**

**a = list(row)**

**b.append(a)**

**df2 = pd.DataFrame(b, columns = ['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GMV', 'GR', 'GRASP', 'ATTRIBUTE\_VALUE', 'GR\_Bucket' ,'IHM\_Bucket', 'DND\_Bucket', 'APPONLY\_Bucket', 'INDIAIP\_Bucket', 'SDND\_Bucket', 'New\_GR', 'New\_IHM', 'New\_APPONLY', 'New\_INDIAIP', 'New\_SDND'])**

**df = df2**

**del df2**

**a = []**

**b = []**

**df\_final = df[['BKG\_HTLSEQ', 'BKG\_CITYCD', 'GRASP', 'ATTRIBUTE\_VALUE']]**

**df\_final[['BKG\_HTLSEQ', 'BKG\_CITYCD']] = df\_final[['BKG\_HTLSEQ', 'BKG\_CITYCD']].astype(str)**

**df[['BKG\_HTLSEQ', 'BKG\_CITYCD']] = df[['BKG\_HTLSEQ', 'BKG\_CITYCD']].astype(str)**

**df.to\_csv('C:/Users/int802/Desktop/Final Python df.csv', index = False)**

**df\_final.to\_csv('C:/Users/int802/Desktop/Final Python.csv', index = False)**