**LWP-GP-Version**

**Algo Steps:**

Step 0: Take Multiple of 7 days - min 14 days, max 28 days of data (4 Weeks)

 MSISIDN, Timestamp, cellid, Lat, Long and remove all rows with missing lat/long values.

* Select date range (eg 1-sep to 28-sep) in multiple of 7
* Select user which have data in Multiple of 7 days min 7 days max 28 days of data.
* To reduce data size remove data on same cell id & Lat-long per user.(not mandatory just to improve computation )

Additional step: Keep only cell change events in raw data

Step 1 :

Create 30 min Bins ie for each hour divide it into two bins ie 2\_0, 2\_1 where 2\_0 means 2:00 to 2:30 AM and 2\_1  means 2:30 to 3:00 AM

Step 2 : In case the values are missing then create new missing bins and embed the values by last reported Cell/ Lat/Long for time of that particular bin use max time for the imputed bin e.g. if Bin 9\_0 was missing and was inserted put 9:00 as time if Bin 10\_1 was missing and was inserted put 10:30 as time.

3)Once the data is ready then further clean up the data by taking only the first and  last time stamp value with in the bin, Ex- say the bin  under consideration is from 2:30 to 3:00 AM  and the measurements are recorded at 2:35 Am, 2:42:Am, 2:53 AM and 2:56 AM then only the measurement recorded at 2:35 AM and  2:56 AM will be considered and rest (2:42AM and 2:53 AM) can be discarded :Optional

Do not apply for first & last sample Remove the above step

Handle triangulation error on basis of illogical distance travelled in very less time

4)Now your data is ready and we can compute the following : Optional

Distance travelled between consecutive rows on time  in KM/Mts

Time difference in secs/ mins

5)Now aggregate the data for Per Msisdn, Per date, per day of the week, Per hour, per HalfBin and calculate the

* Total distance
* Total time with in the bin (Caution time can be higher than 1800 as this is being calculated wrt previous row)
* Speed (Total distance / Total time in Hours)
* Total distinct Cells per Bin on actual data (Output of step no. 2)

Date\_Hour\_Halfhour(0 or 1) - (28\*48 = 1,344)

Convert the time to 1800s where the value is greater than equals to 1800

**Ensure we do not have extraordinary high values, permitted value with in 30 min bin of distance travelled is 45-50 KM Max (need to discuss)**

6) with in each bin check if the speed is > 1 KM/H & the distance travelled is > 3 KM then tag as Mobile else stationary  (Why 4 Kms  because we are computing distance between multiple points and then aggregating and hence the triangulation error needs to be overcome 2.5 KM travelled can result in false triggers)

**Mobility Analysis:**

1. Use Aggregated data of step 6 (**Suggestion: Use data after Stationary & Mobility correction step)**
2. Map week number with the data using 7-7 days buckets (Wk1, 2, 3, 4)
3. Filter rows where TAG is "Mobility"
4. Groupby on MSISDN, Day of the Week , hour and aggregate:
   1. Count of events – A (Calculated from output of Step2)
   2. Total Distance Travelled - B
   3. Total Time spent - C
   4. Distinct cells reported - D
   5. New - Distinct Weeks
5. Now create a weighted Mobility factor "E" by multiplying A\*B\*C\*D
6. Separate Weekdays and Weekend Data.
7. Select Weekday data
8. Get 10th percentile of all non zero samples for each msisdn i.e. 24hr\*7 days = 168 Samples(only non zero) - Call it as "10th\_perc\_cut\_off"
9. Condtion1: If Mobility Factor "E" > "10th\_perc\_cut\_off" and condition 2: "Distinct Weeks >=2" then mark hourly\_probability as '1'(if data more than 3 week or 21 day)

* Note condition 2 will apply only if data is more than 3 week or 28 days

1. Groupby on MSISDN and hour and aggregate (sum) hourly\_probability
2. Find Probability of Mobility in each hour out of 7 days, if probability is >= "3/7" then mark those hours as Mobile and rest stationary (currently not required)
3. Find Probability of Mobility in each hour out of 2 days(weekday/weekend), if probability is >= "1/2" then mark those hours as Mobile and rest stationary - weekend pattern

**Weekday**

**If user have data more than 21 days:**

* 1. Find Probability of Mobility in each hour out of 7 days, if probability is > "2/5" then mark those hours as Mobile and rest stationary

**If user have data less than 21 days:**

1. Find Probability of Mobility in each hour out of 7 days, if probability is >= "2/5" then mark those hours as Mobile and rest stationary

**Weekend**

* 1. Find Probability of Mobility in each hour out of 7 days, if probability is > "1/2" then mark those hours as Mobile and rest stationary (currently not required)

**New Steps:**

1. Create a 24 hours pattern for each user, fill "S" where in null hours. Include those users with 'S' which are not in Mobility pattern but available in raw data.
2. Correct Mobility Pattern:

Condition 1: If out of 24 hours only 1 hour is 'M' then convert that into 'S'

Condition 2: If Before and after 'S' there is 'M' then convert that into 'M'(Check Distance if not significant change )

1. Home/Work/Mobile Hours Segmentation:
   1. If all 24 hrs are 'S' then change the flag to 'H'
   2. Find 2nd Top 2 seq of 'S's in between 'M's and change their flag to 'W'
   3. Rest all 'S' need to be changed to 'H'
2. Do unpivoting of data and create columns of 'msisdn', 'hour', 'TAG\_HWM'

Cleanup:

1. Join data with step 2 on msisdn & hour.
2. Groupby on ['msisdn',”Cell”,"Lat ",'Lon, 'tag'] in basic LWP output datframe to get: total hour count & unique days
3. Factor calculation:

= pow(unique days, 2) \* hours count

1. Providing the rank to each msisdn for each (home, work, others) category according to the calculated factor. Rank provided in descending manner.

**df\_factor["Rank"] = df\_factor.groupby(['msisdn','tag'])["factor"].rank("dense", ascending=False)**

1. Select the top 6 records from each category.
2. Now calculate the distance of H1 (top rank home category point) from all other location points available in top-6 dataframe.
3. Select the < 3 km distance (it can be tuneable further according to inter-site distance) records & provide those a New\_tag as ‘Home’ category.
4. Now select ‘Home ’ category & keep it separate DataFrame.
5. Now select the remaining records other than ‘Home category’ and again calculate the rank on the basis of factor for these remaining records.
6. Now calculate the distance of W1 (top rank work category point) from all other location points available in df\_remaining\_t6 DataFrame.
7. Select the < 3 km distance (it can be tuneable further according to inter-site distance) records & provide those a New\_tag as ‘Work’ category.
8. Provide New\_tag as ‘Others’ for the remaining records (not tagged as work category)
9. Merge both dataframes – Home & Work-Others into a single dataframe & take output as Master LWP sheet.