Pseudo Code

This program will allow the user to check whether the particular bank transaction was fraudulent or not.

1. Import python Libraries for use
2. Read the input file “Dataset\_Datathon\_Fraud\_Detection\_28122022.xlsx”
3. Rename columns to comply with the naming standard in python
4. Impute the missing values of variables in the dataset as below:

Numeric:

* 1. active\_phone\_call: replace missing with 0
  2. remote\_desktop\_protokol: replace missing with 0
  3. emulator: replace missing with 1
  4. screen\_hash: replace missing with 0

Non-numeric: replace missing with blank

1. Create a new variable “pre\_auth\_flag” = 1 if the variable “event\_desc” contains any authorization event, else assign 0
2. Create a new variable “pre\_auth\_amount” which holds the value of the variable “operaton\_amt\_100\_rupees” if the earlier created variable “pre\_auth\_flag” = 1, else assign 0
3. Following steps are followed to slice the dataset on the basis of customer sessions
   1. Sort the dataset by the variables “customer\_id” and “event\_dttm\_Deli”
   2. Temporary Boolean variable “new\_session” is created when the variable “event\_type\_nm” = 'SESSION\_SIGNIN'
   3. A cumulative sum of all the sessions is assigned to new variable “session\_num” after grouping by “customer\_id” and “new\_session”
   4. Temporary variable “new\_session” is dropped
   5. New dataset session\_df is created from the existing one with the following aggregations after grouping by “customer\_id” and “session\_ num” and then index is reset
      * timestamp\_min = min of 'event\_dttm\_Deli'
      * timestamp\_max = max of 'event\_dttm\_Deli’
      * event\_type\_count = count of 'event\_type\_nm'
      * event\_desc = concatenated and comma separated value of 'event\_desc'
      * amount\_sum sum of 'operaton\_amt\_100\_rupees'
      * curr\_device = most frequently used 'device\_model'
      * curr\_location = most frequently occuring 'f'
      * pre\_auth\_count = sum of 'pre\_auth\_flag'
      * pre\_auth\_amt = sum of 'pre\_auth\_amount'
   6. New variable “session duration” = difference of ‘timestamp\_max’ and 'timestamp\_min' typecasted to timedelta64[s]
   7. Convert “event\_desc” values to lowercase
   8. Create a new variable “event\_desc\_flag” =1 if “event\_desc” contains any of the keywords “password“, “payment” or “purchase”, else 0
4. Following steps are followed for customer profiling on the dataset
   1. New dataset customer\_df is created from the input dataset with the following aggregations after replacing null values with ‘ ‘ and grouping by “customer\_id”
      * amount\_stdev = standard deviation of operaton\_amt\_100\_rupees
      * amount\_mean = mean of 'operaton\_amt\_100\_rupees'
      * device = max of 'device\_model'
      * operating\_sys = max of 'os'
      * location = max of 'f'
5. Create a new merged dataset df\_merged by doing a left join on session\_df and customer\_df on ‘customer\_id’
6. Create flags on the merged dataset as below:
   1. 'amount\_flag' = 1 if 'amount\_sum' > 'amount\_stdev', else 0
   2. 'device\_flag' = 1 if ‘'curr\_device' != ‘device', else 0
   3. ‘location\_flag’ = 1 if 'curr\_location' != ‘location’, else 0
7. Create a final dataset from the merged dataset for the rule based prediction of fraudulent transaction with the variables: 'customer\_id','session\_num','event\_type\_count','pre\_auth\_count','pre\_auth\_amt','session\_duration','event\_desc\_flag','amount\_stdev','amount\_mean','amount\_flag','device\_flag' and 'location\_flag'
8. Set index based on 'customer\_id' and 'session\_num'
9. For the rule based classification, the following rules have been followed:
   1. Rules for assigning ‘transaction\_label’:
      * if (‘pre\_auth\_count’ > 1) or (‘pre\_auth amt’ > ‘amount\_stddev’) or (‘event\_desc\_flag’ =1 and ‘amount\_flag’ = 1) then classify as “fraud”
      * if (‘even\_desc\_flag’ =1 and ‘amount\_flag’ = 0) or (‘amount\_flag’ =1 and (‘device\_flag’ =1 or ‘location\_flag’ = 1)) then classify as “possible\_fraud”
      * if (‘device\_flag’ =1 or ‘location\_flag’ =1 or ‘amount\_flag’ =1 or ‘pre\_auth\_count’ > 1) then classify as “to\_review”
      * assign the other records as ‘unknown’
   2. Check the value counts for each category of ‘transaction\_label’