'''

To do

timezone date remove timezone info

encode

remove cols after transform

check if data leakage in moving average

df.copy #FIXME

last few transactions vs last few days #TODO

panel data multiple waves

moving average

refer to recession work?

features to be added/considered:

Lag features with various lag periods.

Rolling features with different window types.

Exponential weighted moving average features.

Expanding mean features. expanding().mean()

value minus id based mean

# real data: current + past waves recalculated (day unit???)

IP

- some IP addresses belong to mobile networks, and those will likely reuse one IP for multiple customers

- having two IPs within a short time window (e.g.: VPN, or having a very different country code) can mean stolen credentials, so maybe second transaction could be fraud, or first one too

but you also sort your data by timestamp, and for each user clome the above colums from one row into the next row

- recentIP\_countryCode

- recentIP\_isUsedByMany

- recentIP\_timeAgo DONE

radial basis functions for extracting temporal structure

 sort events chronologically and group them by user, so that events are sorted from each user's perspective

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

so that then you can have even more features from each user's perspective (e.g.: number of purchases over last week, or total spent over last month)

['TX\_AMOUNT','TX\_DURING\_WEEKEND', 'TX\_DURING\_NIGHT', 'CUSTOMER\_ID\_NB\_TX\_1DAY\_WINDOW',

       'CUSTOMER\_ID\_AVG\_AMOUNT\_1DAY\_WINDOW', 'CUSTOMER\_ID\_NB\_TX\_7DAY\_WINDOW',

       'CUSTOMER\_ID\_AVG\_AMOUNT\_7DAY\_WINDOW', 'CUSTOMER\_ID\_NB\_TX\_30DAY\_WINDOW',

       'CUSTOMER\_ID\_AVG\_AMOUNT\_30DAY\_WINDOW', 'TERMINAL\_ID\_NB\_TX\_1DAY\_WINDOW',

       'TERMINAL\_ID\_RISK\_1DAY\_WINDOW', 'TERMINAL\_ID\_NB\_TX\_7DAY\_WINDOW',

       'TERMINAL\_ID\_RISK\_7DAY\_WINDOW', 'TERMINAL\_ID\_NB\_TX\_30DAY\_WINDOW',

       'TERMINAL\_ID\_RISK\_30DAY\_WINDOW']

if you have access to a unique user identifier, then it's almost like you get a bit of "online" behavior at runtime due to some (per-user) features being computed live

to do online learning, updating the weights of the model may be tricky

random forest classifiers work best when your features are clear boolean/scalar features, a concept like "primary IP", "secondary IP", "abnormal IP" does not map directly to boolean/scalar features

group by customer\_id and sort by time, maybe what makes more sense is a "distribution" of IP addresses:

- isNewIP

seasonality you can capture with the radial basis functions

or just have a ton of boolean fields about the current timestamp

- date\_isBankHoliday\_anywhere

- date\_isBankHoliday\_inCountryOfCurrentIP

- date\_isBankHoliday\_inCountryOfPreviousIP

- date\_isBankHoliday\_inCountryOfAnyRecentIP

all generated from the timestamp of each transaction

time\_isMorning\_inCountryOfCurrentIP --> time\_inCountryOfCurrentIP, expressed as seconds since midnight, in that country

tons of bool features about the timestamp of each transaction

'''