Strive

SCHOOL

/ Date and Time data

Rich information

Date

- Day of week (Monday, ...)
- Day of month (1st, 2nd, ...)
- Month (January, ...)
- Year (2021, ...)
- Is weekend (yes, no)
- Is vacation (yes, no)

Time

- Hour (24-hour format)
- Minute
- Second

Encoding formats: Timestamp

1/17/07 has the format "%m/%d/%y"

17-1-2007 has the format "%d-%m-%Y"

%a	Weekday as locale's abbreviated name.	Mon
%A	Weekday as locale's full name.	Monday
%w	Weekday as a decimal number, where 0 is Sunday and 6 is Saturday.	1
%d	Day of the month as a zero-padded decimal number.	30
%-d	Day of the month as a decimal number. (Platform specific)	30
%b	Month as locale's abbreviated name.	Sep
%B	Month as locale's full name.	September
%m	Month as a zero-padded decimal number.	09
%-m	Month as a decimal number. (Platform specific)	9
%y	Year without century as a zero-padded decimal number.	13
%Y	Year with century as a decimal number.	2013

Rich information: Pandas

● Minute	→ df["min"]	= df.myDateVar.dt.minute.astype(np.int8)
● Hour (24-hour format)	→ df["hour"]	= df.myDateVar.dt.hour.astype(np.int8)
● Day of week (Monday, ...)	→ df[weekDay]	= df.myDateVar.dt.dayofweek.astype(np.int8)
● Day of month (1st, 2nd, ...)	→ df["day"]	= df.myDateVar.dt.day.astype(np.int8)
● Day of year	→ df[yearDay]	= df.myDateVar.dt.dayofyear.astype(np.int16)
● Month (January, ...)	→ df["month"]	= df.myDateVar.dt.month.astype(np.int8)
● Year (2021, ...)	→ df["year"]	= df.myDateVar.dt.year.astype(np.int16)
● Is weekend (yes, no)	→ # To do	
● Is vacation (yes, no)	→ # To do	

2 categories

Current datetime

Useful for capture patterns and repetition. Example: On friday afternoon, shopping increases.

Datetime past since a particular event (LAG)

Very useful to measure. Example: Time since a patient took a pill.

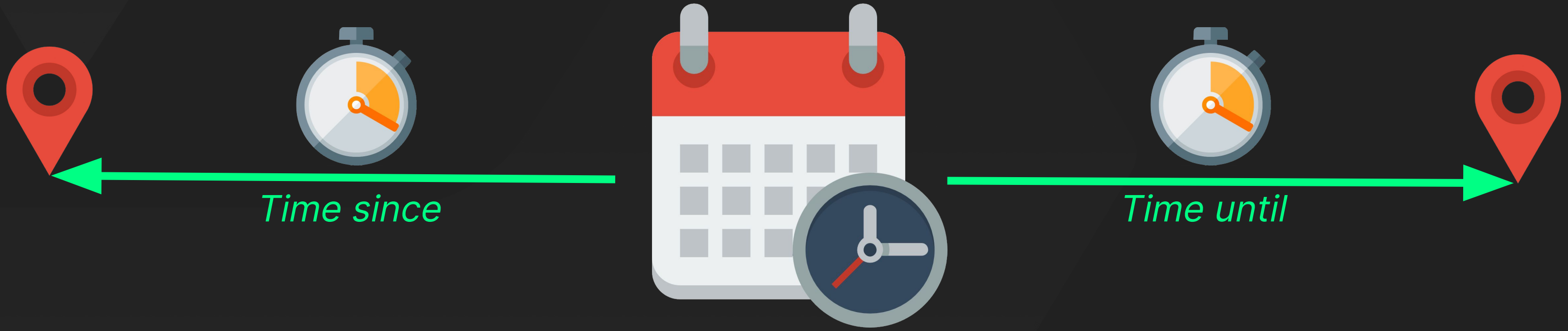
Lag features: Time since, Time until

/ Row-independent moment

For example: since 00:00:00 UTC, 1 January 1970;

/ Row-dependent important moment

Time passed since the last holiday, weekend, sales campaign, ...
Number of days left until next holidays.



Lag features: Example

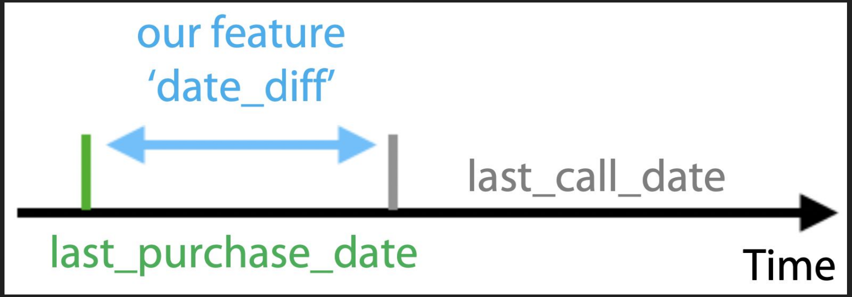
		Time since		*Time until*	
Date	weekday	daynumber_since_year_2014	is_holiday	days_till_holidays	*sales*
01.01.2014	5	0	True	0	*1213*
02.01.2014	6	1	False	3	*938*
03.01.2014	0	2	False	2	*2448*
04.01.2014	1	3	False	1	*1744*
05.01.2014	2	4	True	0	*1732*
06.01.2014	3	5	False	9	*1022*

Lag features with target var

/ it is very easy to make mistakes and insert data leaks or leakages when we extract lags with the target variable.

Several dates: Diff

/ If we have several dates, we can compute the difference between them.



user_id	registration_date	*last_purchase_date*	*last_call_date*	date_diff
14	10.02.2016	21.04.2016	26.04.2016	5
15	10.02.2016	03.06.2016	01.06.2016	-2
16	11.02.2016	11.01.2017	11.01.2017	1
20	12.02.2016	06.11.2016	08.02.2017	94

Conclusion

1. Periodicity

   Day number in week, month, season, year second, minute, hour.

2. Lag Features: Time since/until
   a. Row-independent moment

      For example: since 00:00:00 UTC, 1 January 1970

   b. Row-dependent important moment

      Days passed after last holiday.

      Days left until next holidays.

3. Difference between dates

   datetime_feature_1 - datetime_feature_2

/ Q&A

What are your doubts?

Strive
SCHOOL