PROBLEM 2: Drop-off rate analysis & Events Lead to Drop-off

OBJECTIVES:

Calculate the Dropoff Rate:

- Calculate the number of patients who stop taking "Target Drug" each month.
- The drop-off rate is defined as the number of patients who drop off the treatment in a given month.

Analyse Events Leading to Dropoff:

- Identify the events or factors that lead to patients discontinuing the "Target Drug" treatment.
- Generate insights into why patients are stopping the treatment prematurely.

PHASE-1: Drop-off rate analysis

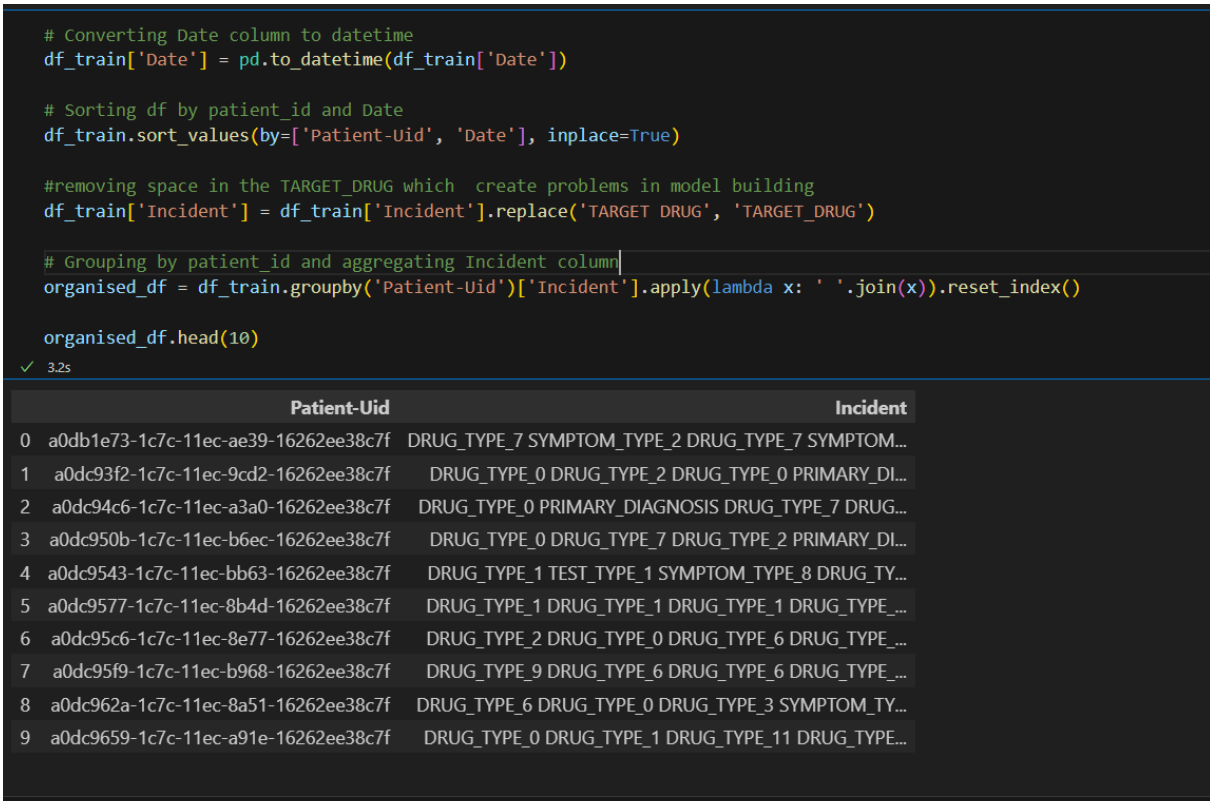
Drugs are typically prescribed or administered by physicians for a specific duration or at regular intervals. However, for various reasons, patients might discontinue the treatment. Consider the following example for a better understanding:

- "Suppose you contract a throat infection, and the physician prescribes you an antibiotic for 10 days. However, you stop taking the treatment after 3 days due to some adverse events."

In the above example, the ideal treatment duration is 10 days, but the patient discontinued the treatment after 3 days due to adverse events. This discontinuation of treatment is referred to as a drop-off.

Part-A: We aim to study the drop-off for the "Target Drug", with the objective of generating insights into what events lead to patients stopping the "Target Drug".

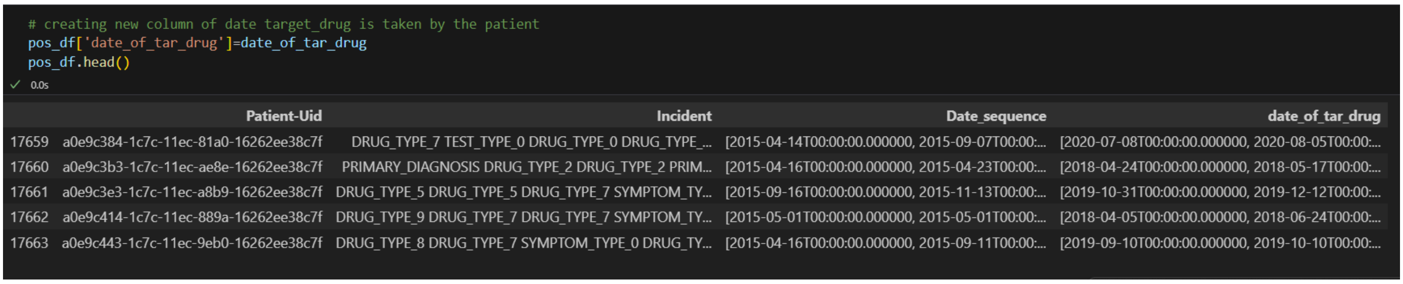
It is given that the ideal treatment duration for the "Target Drug" is 1 year. Therefore, we will conduct an analysis showing how the drop-off rate varies with the number of patients discontinuing each month.



```python
# Converting Date column to datetime
df_train['Date'] = pd.to_datetime(df_train['Date'])

# Sorting df by patient_id and Date
df_train.sort_values(by=['Patient-Uid', 'Date'], inplace=True)

#removing space in the TARGET_DRUG which  create problems in model building
df_train['Incident'] = df_train['Incident'].replace('TARGET DRUG', 'TARGET_DRUG')

# Grouping by patient_id and aggregating Incident column
organised_df = df_train.groupby('Patient-Uid')['Incident'].apply(lambda x: ' '.join(x)).reset_index()

organised_df.head(10)
```

✓ 3.2s

	Patient-Uid	Incident
0	a0db1e73-1c7c-11ec-ae39-16262ee38c7f	DRUG_TYPE_7 SYMPTOM_TYPE_2 DRUG_TYPE_7 SYMPTOM...
1	a0dc93f2-1c7c-11ec-9cd2-16262ee38c7f	DRUG_TYPE_0 DRUG_TYPE_2 DRUG_TYPE_0 PRIMARY_DI...
2	a0dc94c6-1c7c-11ec-a3a0-16262ee38c7f	DRUG_TYPE_0 PRIMARY_DIAGNOSIS DRUG_TYPE_7 DRUG...
3	a0dc950b-1c7c-11ec-b6ec-16262ee38c7f	DRUG_TYPE_0 DRUG_TYPE_7 DRUG_TYPE_2 PRIMARY_DI...
4	a0dc9543-1c7c-11ec-bb63-16262ee38c7f	DRUG_TYPE_1 TEST_TYPE_1 SYMPTOM_TYPE_8 DRUG_TY...
5	a0dc9577-1c7c-11ec-8b4d-16262ee38c7f	DRUG_TYPE_1 DRUG_TYPE_1 DRUG_TYPE_1 DRUG_TYPE_...
6	a0dc95c6-1c7c-11ec-8e77-16262ee38c7f	DRUG_TYPE_2 DRUG_TYPE_0 DRUG_TYPE_6 DRUG_TYPE_...
7	a0dc95f9-1c7c-11ec-b968-16262ee38c7f	DRUG_TYPE_9 DRUG_TYPE_6 DRUG_TYPE_6 DRUG_TYPE_...
8	a0dc962a-1c7c-11ec-8a51-16262ee38c7f	DRUG_TYPE_6 DRUG_TYPE_0 DRUG_TYPE_3 SYMPTOM_TY...
9	a0dc9659-1c7c-11ec-a91e-16262ee38c7f	DRUG_TYPE_0 DRUG_TYPE_1 DRUG_TYPE_11 DRUG_TYPE...



```
# creating new column of date target_drug is taken by the patient
pos_df['date_of_tar_drug']=date_of_tar_drug
pos_df.head()
```

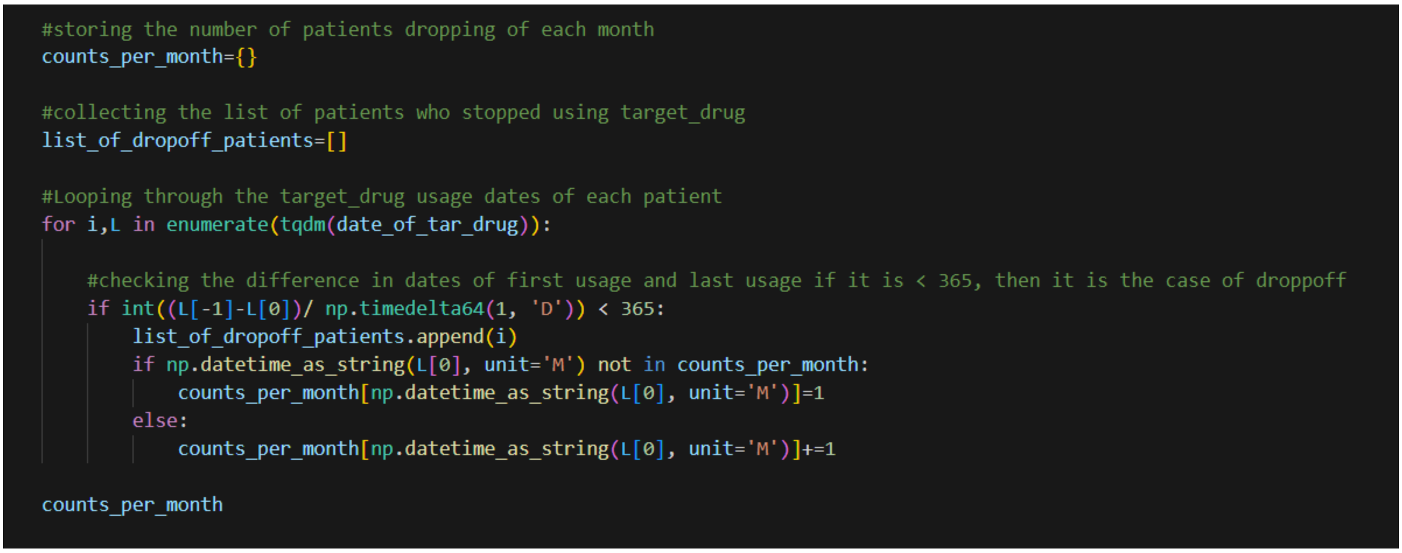
✓ 0.0s

	Patient-Uid	Incident	Date_sequence	date_of_tar_drug
17659	a0e9c384-1c7c-11ec-81a0-16262ee38c7f	DRUG_TYPE_7 TEST_TYPE_0 DRUG_TYPE_0 DRUG_TYPE_...	[2015-04-14T00:00:00.000000, 2015-09-07T00:00:...	[2020-07-08T00:00:00.000000, 2020-08-05T00:00:...
17660	a0e9c3b3-1c7c-11ec-ae8e-16262ee38c7f	PRIMARY_DIAGNOSIS DRUG_TYPE_2 DRUG_TYPE_2 PRIM...	[2015-04-16T00:00:00.000000, 2015-04-23T00:00:...	[2018-04-24T00:00:00.000000, 2018-05-17T00:00:...
17661	a0e9c3e3-1c7c-11ec-a8b9-16262ee38c7f	DRUG_TYPE_5 DRUG_TYPE_5 DRUG_TYPE_7 SYMPTOM_TY...	[2015-09-16T00:00:00.000000, 2015-11-13T00:00:...	[2019-10-31T00:00:00.000000, 2019-12-12T00:00:...
17662	a0e9c414-1c7c-11ec-889a-16262ee38c7f	DRUG_TYPE_9 DRUG_TYPE_7 DRUG_TYPE_7 SYMPTOM_TY...	[2015-05-01T00:00:00.000000, 2015-05-01T00:00:...	[2018-04-05T00:00:00.000000, 2018-06-24T00:00:...
17663	a0e9c443-1c7c-11ec-9eb0-16262ee38c7f	DRUG_TYPE_8 DRUG_TYPE_7 SYMPTOM_TYPE_0 DRUG_TY...	[2015-04-16T00:00:00.000000, 2015-09-11T00:00:...	[2019-09-10T00:00:00.000000, 2019-10-10T00:00:...

Created two columns Date sequence and Date of target drug is taken by the patient.

Part-B: Following this, we will conduct an analysis to generate insights into what events drive a patient to stop taking the "Target Drug".

- Filtering the required data, which includes patients who have taken the 'TARGET_DRUG'.
- Creating a list of all the incident dates for each patient and adding it to the dataset.
- Creating a list of only those dates when the patient has taken the "target_drug" and adding this also to the dataset.

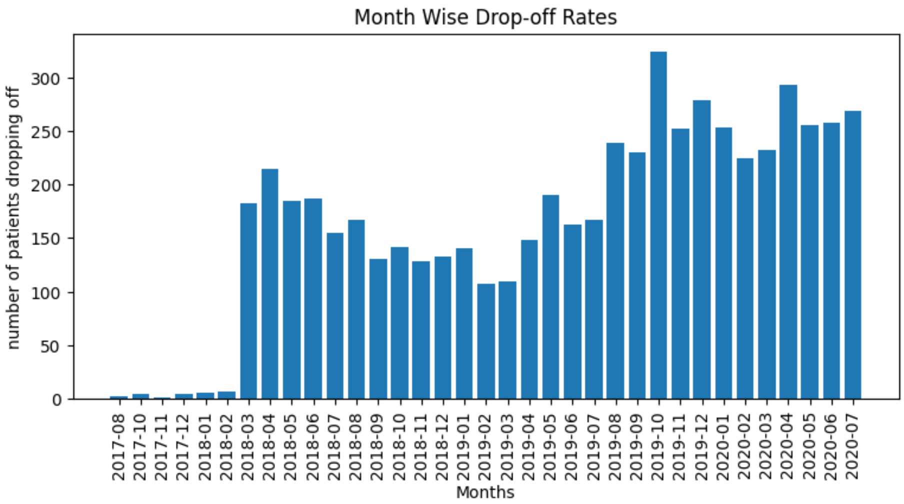


```
#storing the number of patients dropping of each month
counts_per_month={}

#collecting the list of patients who stopped using target_drug
list_of_dropoff_patients=[]

#Looping through the target_drug usage dates of each patient
for i,L in enumerate(tqdm(date_of_tar_drug)):

    #checking the difference in dates of first usage and last usage if it is < 365, then it is the case of droppoff
    if int((L[-1]-L[0])/ np.timedelta64(1, 'D')) < 365:
        list_of_dropoff_patients.append(i)
        if np.datetime_as_string(L[0], unit='M') not in counts_per_month:
            counts_per_month[np.datetime_as_string(L[0], unit='M')]=1
        else:
            counts_per_month[np.datetime_as_string(L[0], unit='M')]+=1

counts_per_month
```

Computing the Drop-off Rates

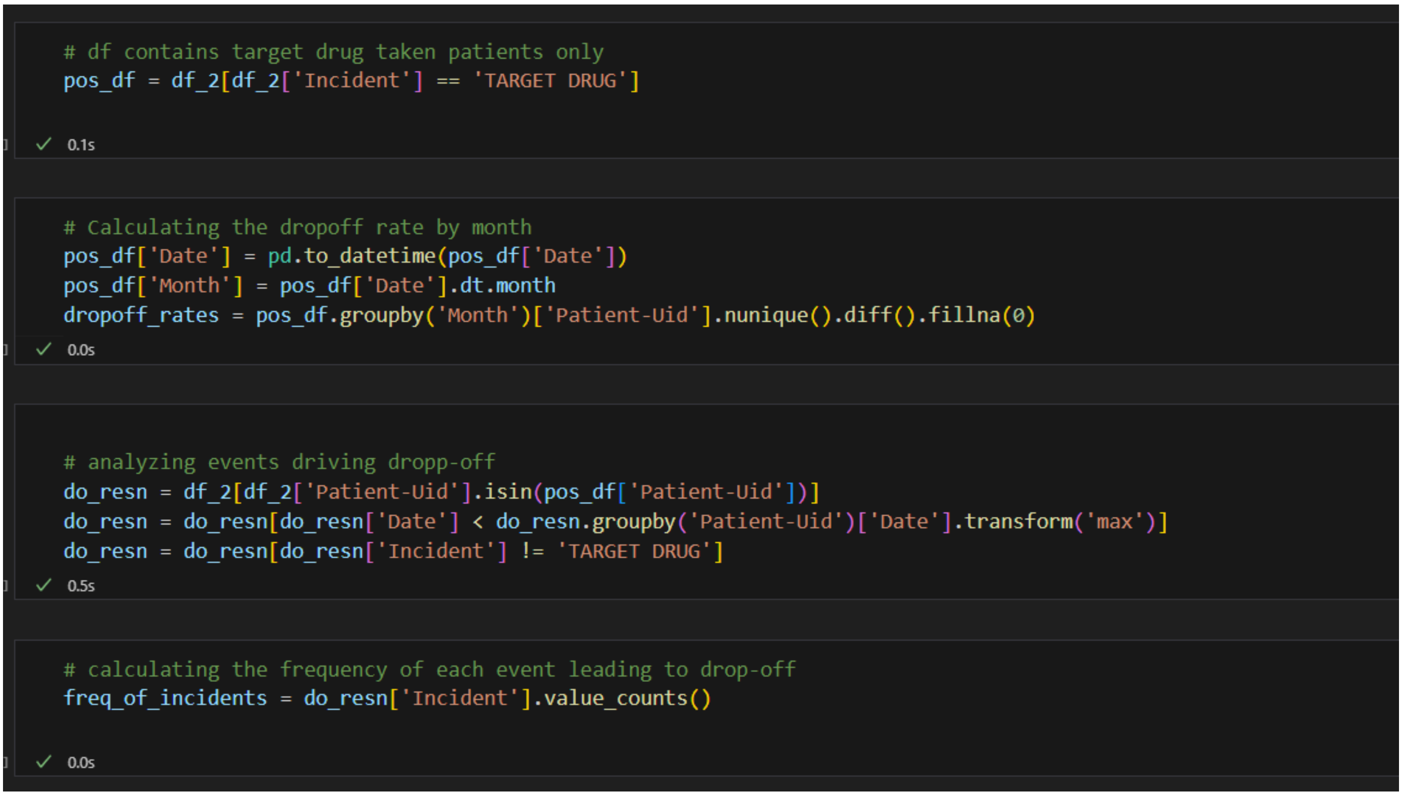
Given that the ideal duration for the "TARGET_DRUG" is 1 year, we can determine when the patient started and stopped using it. Then, we can check if the duration is one year.

If the duration is not 1 year, we can consider it as a drop-off case and record the stoppage date for further analysis.



PHASE 2: Gaining Insight on Which Events Lead to Drop-off

To analyze what events are leading patients to stop the "target_drug", we need to compile a list of symptoms on the dates preceding the stoppage date of the target drug for the patients who are stopping it.



```
# df contains target drug taken patients only
pos_df = df_2[df_2['Incident'] == 'TARGET DRUG']
```

✓ 0.1s

```
# Calculating the dropoff rate by month
pos_df['Date'] = pd.to_datetime(pos_df['Date'])
pos_df['Month'] = pos_df['Date'].dt.month
dropoff_rates = pos_df.groupby('Month')['Patient-Uid'].nunique().diff().fillna(0)
```

✓ 0.0s

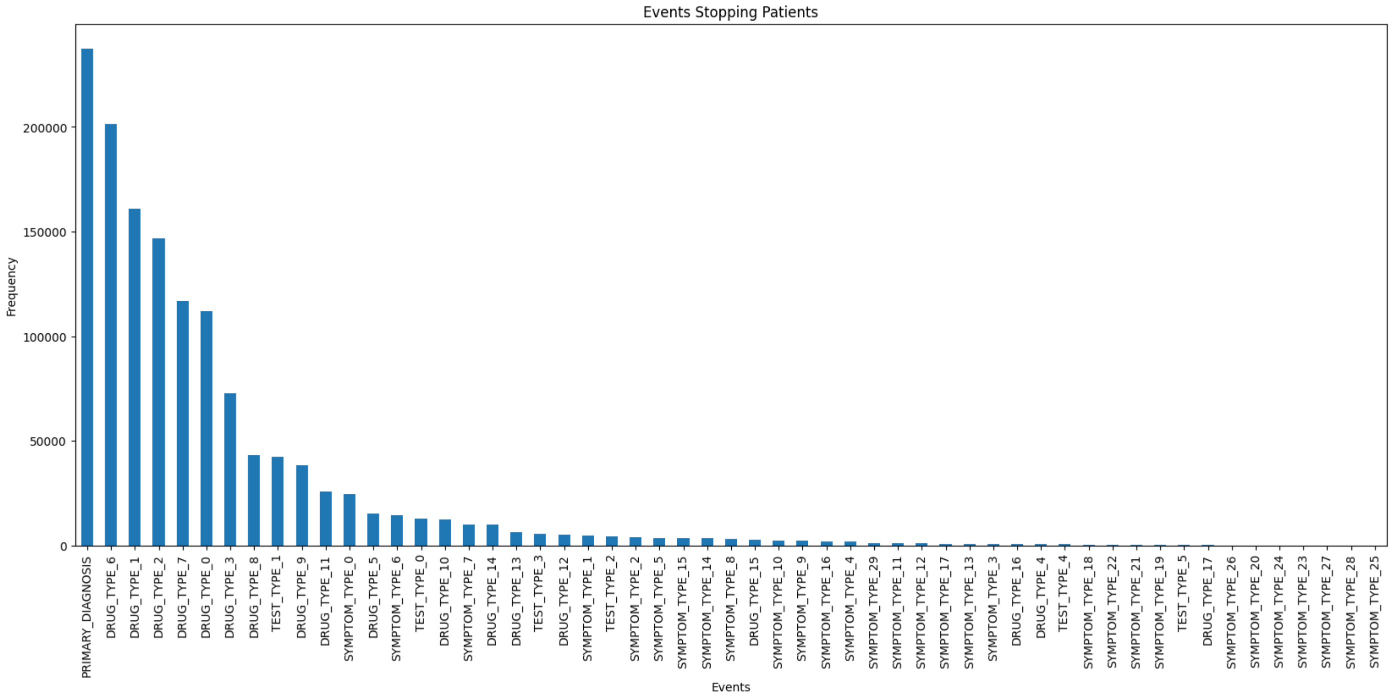
```
# analyzing events driving dropp-off
do_resn = df_2[df_2['Patient-Uid'].isin(pos_df['Patient-Uid'])]
do_resn = do_resn[do_resn['Date'] < do_resn.groupby('Patient-Uid')['Date'].transform('max')]
do_resn = do_resn[do_resn['Incident'] != 'TARGET DRUG']
```

✓ 0.5s

```
# calculating the frequency of each event leading to drop-off
freq_of_incidents = do_resn['Incident'].value_counts()
```

✓ 0.0s

- From the above, we already have a list of drop-off patients. Let's examine the events before the drop-off dates.
- insight which events might have led to the discontinuation of the target drug, plot the events according to their frequency.


Events Stopping Patients
Frequency
Events
PRIMARY_DIAGNOSIS
DRUG_TYPE_6
DRUG_TYPE_1
DRUG_TYPE_2
DRUG_TYPE_7
DRUG_TYPE_0
DRUG_TYPE_3
DRUG_TYPE_8
TEST_TYPE_1
DRUG_TYPE_9
DRUG_TYPE_11
SYMPTOM_TYPE_0
DRUG_TYPE_5
SYMPTOM_TYPE_6
TEST_TYPE_0
DRUG_TYPE_10
SYMPTOM_TYPE_7
DRUG_TYPE_14
DRUG_TYPE_13
TEST_TYPE_3
DRUG_TYPE_12
SYMPTOM_TYPE_1
TEST_TYPE_2
SYMPTOM_TYPE_2
SYMPTOM_TYPE_5
SYMPTOM_TYPE_15
SYMPTOM_TYPE_14
SYMPTOM_TYPE_8
DRUG_TYPE_15
SYMPTOM_TYPE_10
SYMPTOM_TYPE_9
SYMPTOM_TYPE_16
SYMPTOM_TYPE_4
SYMPTOM_TYPE_29
SYMPTOM_TYPE_11
SYMPTOM_TYPE_12
SYMPTOM_TYPE_17
SYMPTOM_TYPE_13
SYMPTOM_TYPE_3
DRUG_TYPE_16
DRUG_TYPE_4
TEST_TYPE_4
SYMPTOM_TYPE_18
SYMPTOM_TYPE_22
SYMPTOM_TYPE_21
SYMPTOM_TYPE_19
TEST_TYPE_5
DRUG_TYPE_17
SYMPTOM_TYPE_26
SYMPTOM_TYPE_20
SYMPTOM_TYPE_24
SYMPTOM_TYPE_23
SYMPTOM_TYPE_27
SYMPTOM_TYPE_28
SYMPTOM_TYPE_25