FIDDLE

I started off with cloning the repo, reading the source article and the slides, However my basic understanding was just FIDDLE stands for Flexible Data Driven Pipeline, and it takes in EHR data to covert them into eigen vectors and use them to create better ML models.

Since there were many files in the repo I wasn't sure how many and which ones to run and how to run them. I got guided to read the example usage in the readme file.

Next, I tried to run the command given in the example usage directly but ran into a very long error. I tried contacting my mentor and he asked me to run it on google collab. The demo file ran but I ran into similar issues while running the run-docker.ipynb and had to make multiple changes according to the given input format in the readme doc. After this, I was getting an error while giving the path which was not getting resolved even after much effort. The problem persisted with me after making multiple changes, searching the web and using ChatGPT. This problem was not getting solved.

I installed pip and docker once again, cloned the repo, and ran the system requirements commands, and this time even the example usage commands worked almost correctly after some editing(like making python into python3, removing extra spaces, correcting the paths, adding data\_fname and output\_dir and removing data\_path as was required by the run.py).

Yet another problem was occurring; the error shown was in the code of one of the files given. I tried to make changes to the code in steps.py.

Due to this error, the output for small\_test was not complete- the X\_npz file was not created even though the s\_npz file was created.

The following is my understanding of FIDDLE and how it works:

FIDDLE is a flexible data-driven pipeline. It somewhat generalized the method to convert structured Electronic Health Records to feature vectors. These feature vectors are used by ML and AI algorithms to produce useful outcomes. These may include predictions of viruses that may affect the particular patient or developing new drugs based on large amounts of data. The advantages advertised are:

Fewer decisions required.

Generalizable to suit different task(although the repository we were given was not the same one used for genralization.)

Converting EHR to feature vectors consists of 3 steps: prefiltering, transformation, and post-filtering.

Prefiltering is the removal of rows with timestamps outside the observation period and the removal of variables that occur rarely i.e. all rows that occur less than theta1 \* 100 times are removed.

The transformation step converts data into two parts: A matrix S for the Time invariant data and a tensor X for the time dependent data.

Post filtering step removes data that carry little information i.e. features equal to 1 less than theta2 \* 100 times are removed.

The variable used has the following meaning :

N is the number of examples.

L is the time bin calculated using floor(T/dt)

theta1 is the threshold of the prefilter step

theta2 is the threshold of the post-filter step.

thetafreq is the threshold to determine a variable is frequent or not.

d is the final dimensionality of the time-invariant variables after the post-filtering step.

D is the final dimensionality of the time dependent variables after the post-filtering step.

The readme doc guides to keep the variables as theta1=theta1=thetafreq = 0.01 \* "minority class".

The example given was that if predictions need to be made every 4 hours over 48 hour time period, then dt = 4 and T = 48. Hence L = 12 hours