Oscar Health Case Study

Whenever I get a case study such as this, I first do some tests and analysis on jupyter notebook, before going to actual python files.

The first question took me about 2.5 hours. I threw in some OpenAI in there too, since that’s so relevant these days.

The second one took me just over an hour. Unfortunately, I ran out of time and wasn’t able to train a model which produced good results. None the less, I showed all my steps. Had I had more time, I would have utilized some NLP approaches or even used a custom model using TensorFlow or PyTorch, as I have done in previously in these repos:  
1. <https://github.com/SharhadBashar/Podcast_IQ>

2. <https://github.com/SharhadBashar/ml-algos>

Finally, I spend an extra 45 minutes to set up proper terminal commands to run the different response, and writing this report.

Instructions to run the code is below. Please make sure that all the relevant csv files are in the data folder.

# Question 1

The first question asked to characterize a members health status based on their outpatient data. This is primarily a pandas related question.

Here is what I did:

1. First I took the claim\_lines csv and in the diag column, I removed the “.”
2. Next I did a left join on “diag” column for both claim\_lines and css csv
3. Now I have a detailed patients claims file with member id and details on the diagnosis

For generating the report, all you have to do is run the following code in the python folder:

python main.py q1

This will create the above csv and save it if it isn’t there

After that you can either enter a member id or leave it blank, and it will generate a member report. The image below shows the result:

A computer screen with green text

Description automatically generated

The report is generated based on the users history. If the user had one diagnosis, it outputs that. If the user has between 1 and 5 different diagnoses, it outputs the most frequent diagnosis as well the latest diagnosis. If the user has more diagnosis, then I output the top 2 or 4 most occurring diagnosis, as well as the latest diagnosis. I also output the first and last date of visits of them. I choose ccs\_3\_desc, as it’s the most generic description of the medical issue. But for the latest diagnosis, I use diag\_desc column.

Other things to try: you can look at it from a weekly, monthly, yearly point of view. For example, health status is calculated at the end of the month and takes into account all the information for that month. If the member has no visits, the health status will not be included

Now, since we live in the world of LLMs, I added a little bit of OpenAI.

Normally since we are dealing with sensitive data, we should use a local model like Llama2, but since this is just an exercise, I stripped all the sensitive data away, and used OpenAI

To test it with OpenAI, first add an OpenAI key in the config/openai.json file, as show below:

A screen shot of a computer

Description automatically generated

To run the code, simply enter the following in terminal

python main.py q1 True

This will take all the medical history of the patient and pass it to the LLM, which in turn is instructed to summarize it in 3 sentences. The image below shows the result:

A computer screen with green text

Description automatically generated

Finally, press “x” to exit the report generator.

All the code can be found in python/q1.py or notebook/q1.ipynb files

# Question 2

For this question, first I had to build a dataset to train the model on.

To do so, I utilized the dataset prepared in q1.

It is a reasonable assumption to make that a prescription is filed after a diagnosis.

So to build a dataset, for each prescription filed for a member, I look at the date if was filed, and get the closest date of diagnosis, In the backward direction.

That is the diagnosis. I once again use ccs\_3\_diag

I remove any NaN.

Then I train on 80% data and use the rest for testing.

If I had more time, I would use K cross validation

To run q2, type the following in terminal:

If you want to train a model

python main.py q2 True

or if you only want to make predictions:

python main.py q2

And then follow the instructions