Supplementary Material

SheetChat: a chatbot for making data science easier

Jorge Guerra Pires[1]*

Abstract: On this document, I am going to present examples of conversations using SheetChat. SheetChat was designed to use the technology I have discussed on the main paper. It is using the openAI API Assistant, the coder interpreter. For the interface, I are using Angular, and Nest.js as backend. For deploy, I am using Heroku. One nice feature of the app created is that extra functions can be added with time for complementing the ones proposed by coder interpreter.
: openAI — data science — statistical inference — python — open source — diabetes

[1] *Founder at IdeaCoding Lab and JovemPesquisador.com*
DOI: http://dx.doi.org/10.22456/2175-2745.XXXX • **Received:** dd/mm/yyyy • **Accepted:** dd/mm/yyyy

1. Introduction

During the main paper, I have discussed the openAI API tool called *code interpreter*. Their playground is very friendly, nonetheless, I have decided to build my own. It may open space for instance to create functions with time, which is not possible on their playground. I also think that it could be easier for beginners to use this interface that I have created.

I have deployed an app on Heroku here. I have also deployed demos here. I am going to discuss one demo more closely. See the other SM with more examples: it is the same tool, except that I have created an interface herein. The reader is invited to explore the demos on their own.

Feel free to leave me a good review on Product Hunt.

For using the tool, it is necessary an openAI API Key, which is released for free at the first usage. They are also very generous on the first key credits.

You may want to read these posts:

- Where do I find my API Key?;
- Create an account on openAI API;
- openAI API keys;

There are in fact several tutorials on YouTube teaching to create your openAI Key, it is generally straightforward.

Tip. Create a key to test our app and when you are done, just delete it. After being deleted, the key validity will also be removed. It will no longer be accepted for API calls.

2. Chess pain as heart attack predictor: Not all pains lead to heart attack, but most of them lead

Full dataset

The full conversation can be found on this link as a demo. I am going to focus on the overall picture.

The idea is to get to know our dataset better, which I have never used before. The most interesting behavior is about the chest pain column (cp). The tool guesses the meaning of the values, even though I have not provided a description. Better, the original description seems to be unclear on the meaning. The tool used a description that is closer to the true description, even without access to the description. It may happen because behind the tool is a LLM, which may have learnt this information from texts in the internet. If we go to the discussions on the dataset, the users had similar doubt, and the solution by the tool was almost what the users proposed. The values are just numbers, but the tool correctly guessed the meaning of each number.

> .....
>
> 3. cp: Chest pain type (0 = typical angina, 1 = atypical angina, 2 = non-anginal pain, 3 = asymptomatic)
>
> .....

See their public discussion here. There are different proposals; some of them is in line with the one by the tool.

This is interesting that the model can access its knowledge to name columns, even when the user does not provide a description.

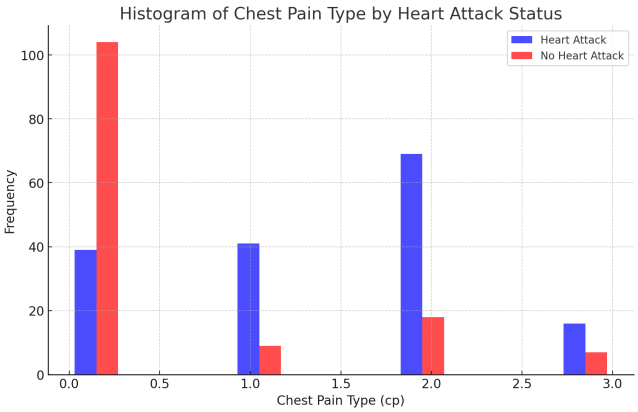

Figure 1. Heart attack by pain type. Source: SheetChat.

One way to doubly check our finding is see what the literature says.

There is some evidence to suggest that **typical angina** is less likely to lead to a heart attack than other types of chest pain. According to a study published in the journal **Heart**, typical angina is a clinical manifestation of ischemic heart disease (IHD) and is characterized by constricting discomfort in the front of the chest or in the neck, shoulders, jaw, or arms. It is precipitated by physical exertion and relieved by rest or sublingual glyceryl trinitrate within about 5 minutes [1]. The study also mentions that around one-half of angina patients have no obstructive coronary disease, and many of these patients have microvascular and/or vasospastic angina [1].

It is important to stress that **typical angina** can also lead to heart attack (Fig. 1). What it is possible to infer is that as we go to other pain types, most of the cases lead to heart attack.

What is interesting to mention is all the work done under the hood. For instance, the tool did without being asked, to join the pains into two groups, for making the test.

One interesting work to do is to compare the results of this tool with manual testing using a tool like Excel. This would be interesting just to make sure there is no mistake. The results seem right. Remember that under the hood the tool is using Python libraries, and you can ask which tool it used. That is a work left for the future. Generally, the tool is very informative.

Important. It is advisable that calculations that are essential for a research be double checked. One could use other tools. It is also possible to ask for the tool for more information of the under the hood tools used, generally, from Python data science tools. It is a new tool, thus, it must be used more before we could trust it completely.

An interesting fact about the code interpreter is that it allows *function calling*. Thus, I may add new tools when the default ones are not enough.