**Data 4 good project**

**Team: Data Dreams**

**Vishwesh Purohit: A20399844**

**Xue Yang: A20224593**

**Sai Chandrika: A20344789**

**Table of Contents**

[Executive Summary 3](#_Toc152752878)

[1. Introduction and Project Objective 4](#_Toc152752879)

[2. Data Introduction/Sourcing 4](#_Toc152752880)

[3. Data Preparation 5](#_Toc152752881)

[4. Data Visualization 5](#_Toc152752882)

[4.1 Distribution of languages 5](#_Toc152752883)

[4.2 Distribution of Nulls after running both the models 6](#_Toc152752884)

[4.3 Distribution of NA, based on the questions 7](#_Toc152752885)

[5. Data Analysis 7](#_Toc152752886)

[6. Generalization/Explanation 9](#_Toc152752887)

[7. Future Scope 9](#_Toc152752888)

[8. Conclusions 9](#_Toc152752889)

# Executive Summary

In today's world, where compassion meets technology, healthcare professionals and volunteers face a challenge: balancing their care for patients with the constraints of time and increasing demands. Particularly in healthcare, these dedicated individuals find themselves overwhelmed by the documentation required for a growing number of patients. However, amidst this pressure, there's hope in the form of advanced technology like Large Language Models (LLMs), such as the new Llama2 model. These innovations offer the promise of streamlining operations, potentially transforming the way healthcare documentation is handled. The emergence of open-access LLMs presents an exciting opportunity to use these powerful tools in a cost-effective manner that respects the privacy needs of the healthcare sector.

This hope and opportunity are encapsulated in the 2023 Purdue "Data for Good" competition. It's a platform brought together by Purdue University, Microsoft, INFORMS, and Prediction Guard, aiming to bridge the gap between healthcare requirements and technological advancements. The focus? Demonstrating how LLMs can revolutionize the painstaking process of medical documentation. The participants must use conversations between the doctor and the patient and prompts to extract information which would be essential for documentation.

The challenge includes automating the process of removing required information from medical conversations. This would be in a question-and-answer format. The participants are given 3 files, including text file with the unique id of the participant and the JSON file with the conversation details. These include 2000 unique entries with 6 questions for each entry. The 6 questions are the same for each participant. The task is cleaning the data and then using the prepared file to call LLMs, the participants use prediction guard to get access to the LLMs, using prompt engineering including one shot and few-shot approach to get responses. It gives participants a good overview of LlMs and in understanding the different models, the constraints with the approaches and the time required for running these heavy models. There is a lot of help provided in terms of sample submissions and sample codes for the participants to get started with. The final output would be a CSV file with 2 columns, the unique identifier for every question per entry and the corresponding response, the uploaded submission is then checked with the correct responses and a corresponding word error rate is determined. The team with the lowest word error rate is considered the winner.

# 1. Introduction and Project Objective

The challenge revolves around automating the extraction of precise information from medical conversations to relieve healthcare workers of documentation burdens. Participants are tasked with creating systems utilizing cutting-edge open Large Language Models (LLMs) like Llama 2, MPT, Falcon, etc., without specific training data provided. The test dataset, structured as a Q&A problem in test.csv, demands zero-shot answers based on corresponding medical conversations in transcripts. Json.

The test.csv comprises questions a medical professional might ask in patient paperwork, while transcripts. Json holds simulated doctor-patient conversations or dictations. Participants must generate natural language responses for each question in the test set using these conversation records. The extraction of pertinent information without direct training data, reflecting real-world scenarios where precise, contextual responses are crucial.

The challenge is structured as a question-answering task where participants must generate natural language responses to various questions posed in the test set based on the content of the provided conversations. The objective isn't solely to provide accurate answers but to recreate preferred natural language responses to validate the information from the given transcripts, mirroring real-world scenarios where precise, contextually appropriate responses are essential for efficient healthcare documentation.

Having the project open for 2 months gives the participants and the team enough time to update the APIs and check with different LLm models, the challenge is part of a larger competition and thus the participants are concurrently learning about Azure. The objective of the challenge is to help participants better understand prompt engineering, LLMs, and how they could use companies like prediction guard to complete their tasks cheaper and faster.

# 2. Data Introduction/Sourcing

There are 3 files given to us, 2 input files and 1 file as a sample submission. From the data input files, one is a csv file called the test file and the other is a JSON file.

* The test.csv file has 12006 rows and 3 columns, containing Id, Transcript and Question, the Id is a alpha numeric combination unique for every question to every patient asked, corresponding to it are the transcript which gives the id of the patient and the question which has been asked, the transcript is a numeric field with 2001 unique values ( for the 2001 patients ) and the question column has 6 unique values, for the six questions asked for each patient.
* The JSON file has the transcript column and the conversation for each transcript\_id thus giving us the conversation text for each patient.
* The sample submission shows us a file with the ID and Text columns, which has the ID from the test.csv column, the text column includes the answers to the question ID associated.

# 3. Data Preparation

To prepare the data we start by combining the datasets to get the final file to work with, so we convert the JSON file into a data frame using python. We then join JSON conversations data frame and the test.csv file on the transcript column in the test.csv file and the ID column in the conversations data frame. We then get the 12006 rows file with the conversation id, patient id, the question, and the conversation text.

Before continuing with the prompt engineering aspect, we look for data discrepancies, we saw that for every 6th patient in the dataset, the conversation was in another language, so we removed them and made a list of all the rows that were not in English, we then translate them to English using the translate library and add them back to the main dataset.

Once we have all the rows in English, we go ahead and start creating the columns for prompt engineering.

Since we are following 2 approaches (one shot and few shot methods), for this we created three separate columns, for one shot we created an instruction column, which has a standard instructions like “please answer the following question based on the text provided “, we then concatenate the question column and the text column. We then can hit the LLM on the prompt row.

We then create an example response column in which we manually write the answers for the first 50 rows. We compare the answers produced by the first LLM and see how its response compares to that of a human, we search for patterns in errors and try to correct them. For the second model we see how many blanks were noticed and for what questions the model failed. We create a separate CSV with the

# 4. Data Visualization

Below we have seen the visualizations on the finally prepared dataset.

## 4.1 Distribution of languages

We see that almost 1/6th of the data is not in English, this has to be treated or discarded, since we would lose a good percentage of records, we call the google translator API to translate all the non-English texts into English for processing.

Figure Languages for English and Non-English

## 4.2 Distribution of Nulls after running both the models

We see that after using Dolly-7B there is a significant decrease in the NA column, the NA columns refer to those columns for which we do not get any response and includes those records where the input text does not have information to provide the response.

Figure 2 Distribution of NA and non-NA with Llama2

Figure 3 Distribution of NA and non-NA with Dolly-7B

## 4.3 Distribution of NA, based on the questions

We see that with both the models they struggle on the same top 2 questions, but the third question has changed from question on condition to question on prescription. We see how Dolly-7B responds better on the top two questions as well.

Figure 4 Distribution of NA Response for Llama2

Figure 3 Distribution of NA Response for Dolly-7B

# 5. Data Analysis

The initial data analysis unveiled crucial insights into the dataset's composition and the challenges posed by incomplete information within medical conversations. A notable observation was the occurrence of missing or insufficient details for certain questions across various patient transcripts. Specifically, numerous instances were identified where essential information required for specific questions was absent. This pattern necessitated the generation of standardized responses for these scenarios. For instance, queries pertaining to patient names, age, diagnosis, symptoms, and prescriptions often lacked explicit mentions in the conversation transcripts. As a mitigation strategy, a consistent set of generalized responses was formulated to address these absent details, ensuring uniformity in data handling and subsequent analyses. Upon comparing responses generated by LLMs against manually written answers, intriguing differences emerged, shedding light on the capabilities and limitations of these language models.

The automated responses from LLMs, particularly the Nous-Hermes-Llama2-13B model, displayed a remarkable level of depth and accuracy in their generated information. However, this comprehensive coverage was not consistent across all queries and transcripts, especially in instances where specific questions were directed at absent or ambiguous information. Notably, while the LLMs provided exhaustive answers, the variation in accuracy surfaced when dealing with questions regarding prescriptions or unmentioned details. This variability underscored the need for further exploration and refinement of LLMs to enhance their contextual understanding and response accuracy. Another critical aspect of the analysis involved investigating the performance of different LLMs beyond Llama2. Early evaluations showed that while Llama2 yielded an approximate 40% null response rate, alternative LLMs like Dolly-7B exhibited improvements, particularly in answering specific questions such as patient identification (name and age). This initial comparison between LLMs hinted at the potential variability in their performance concerning different types of queries within the medical conversation context.

Consequently, an expanded exploration into other LLMs provided in the challenge dataset was deemed necessary to comprehensively assess their effectiveness and suitability in addressing specific informational gaps encountered in the medical conversation.

We look at sample responses from individual models and we compare them with the responses we would write looking at the question.

Table Responses Example from the Nous-hermes-Llama2-13B

|  |  |
| --- | --- |
| Questions | Answers |
| What is the patient's name? | The patients name is Mr. Don Hicks |
| What is the patient's age? | The patient is 81 years old. |
| What is the patient's condition? | The patient has a fungal infection. |
| What symptoms is the patient experiencing? | The patient has dichromic patches, nodal skin eruptions, and skin rash as symptoms. |
| What precautions did the doctor advise? | I prescribed the patient to use Dettol and Neem |
| What drug did the doctor prescribe? | Did not mention any prescriptions |

We see that the answers are pretty in-depth, As compared to the responses we manually write, we see that all the questions are accurately answered however, this is not true for many cases, especially when there are questions. I then used Dolly-7B and it showed much more improvement on the questions for what the patient names and patient age is.

To further improve we converted all the rows that have blanks due to not enough sufficient data as NA as mentioned by the competition organizers.

We collect the remaining nulls as 2 separate lists. We perform different tasks on them to see which method performs better.

1. We pass the blanks again to the same model; this would help increase the number of blanks due to API call error.
2. We pass the blanks with the model Dolly-7B and see how many of these the model helps in creating a response.

# 6. Generalization/Explanation

We define how well our result CSV performed based on the Word Error Rate (WER), the WER is a metric used to evaluate the accuracy of speech recognition or machine translation systems. It measures the disparity between the original (reference) transcript and the transcribed or translated text produced by an automated system. WER is calculated as the minimum number of edits required to transform the reference text into the system's output, normalized by the total number of words in the reference text. The edits include:

1. Substitutions: The number of words that differ between the reference and the system output.
2. Insertions: Words present in the system output but not in the reference.
3. Deletions: Words present in the reference but not in the system output

Since we have a WER of 5.2 we have a high scope of improvement. We noticed the models used by us failed in majorly two question sets. We have discussed approaches to deal with them in the future scope.

# 7. Future Scope

The future scope of this competition could be to reduce the word error rate and get results closer to the true responses. We could do so in three major ways

* Using different LLMs and looking at which model performs better for which question, and then we could mix and match the LLMs output into a final submission.
* We could collect and pass the blanks reiteratively for the different language, there could be an API call error which would be mitigated by trying with the same prompt rows again and again.
* We could try and see how to phrase the example prompts, to better get responses. This would help the model in finding patterns to look for, this would require some more research on how to create those examples.

# 8. Conclusions

The project saw a total of 3 submissions but with little to no improvement in the WER rates, we see that the best WER we receive is **5.2.** This is generally a percentage letting us know that approximately 95% of the words in our entry are accurate. We, however, see that the WER for the other participating teams lies on average at 2%, showing us many of the teams must have taken the additional steps we mentioned in our future scope. The competition was helpful for the team as it introduced us to the world of LLMs and prompt engineering. We could see how to create prompts and the limitations of certain models. It also showed the limitations of google collab and exposed us how to deal with increasing processing speed using parallel processing.