Prompt Engineering Document

Introduction

We used prompt engineering throughout the project to assist with idea generation, problem solving, and program modifying to meet our needs. This document includes examples of prompts used to assist with various steps of the process.

Prompt Examples

1. Problem Domain
    a. ChatGPT was used to assist with problem domain idea generation and selection using scripts such as the following:
    b. Request block 1, requirement formulation:
        i. The culminating event of this course is a team-based final project which will require teams to integrate and utilize at least three Google Cloud Platform (GCP) technologies (e.g., BigQuery, Dataflow, and Cloud pub/sub), in addition to Looker Studio.

           This project must incorporate aspects of prompt engineering, testing the team's ability to optimize AI models using tailored input prompts.

           Requirements for the final project:

           Problem Domain: Select a relevant and engaging problem domain of your choice.

           Data Sources: Utilize one archived and one real-time streaming data source via an open API.

           Analysis Goals: Clearly state and describe your goals for analysis.

    c. Request block 2, idea generation:
        i. Apply the initial requirements to a public domain of real-time traffic analysis and prediction
        ii. Apply the initial requirements to a public domain of weather and space
        iii. Apply the initial requirements to a public domain of mortgage rates and government
        iv. Apply the initial requirements to a public domain of weather and residential energy use

Prompt Engineering Document

2. Data Sources
    a. The previously described prompt fulfilled this requirement, offering open source data sources for the team to use as a starting point for our data review, to see if they would be technically acceptable data sources for our project.
    b. In collection and linking of data, for prompt engineering was used to assist general code development and debugging. Prompts such as the following were used.
        i. API data call: “write a script to leverage the openweather api to gather the date, day (of the week), maximum temperature (temp_max), maximum humidity (hum_max), average wind speed (wind_avg), and precipitation level (precipt)”
        ii. Program troubleshooting and debugging: why does the below code not run?
        iii. For syntax clarification: In BigQuery, how do you compare a timestamp to a date?
3. Analysis Goals
    a. Original prompt provided several ideas for goal selection, as a team we settled on the following:
        i. Correlation Analysis
        ii. Predictive Modeling
4. Metric Selection and Justification
    a. No prompt engineering was necessary at this point as metrics logically followed from previous input.

Performance Improvement Observations:
Prompt Engineering was largely effective in improving productivity during the course of the project. In the beginning, using prompt engineering helped the team get support in developing project concepts. Prompt engineering was also very helpful in general code development. Given most of the team’s unfamiliarity with the API import into BigQuery, retrieving code segments that were generally effective was very fruitful in that we could save time in doing so ourselves.

That said we found that prompt engineering can have flaws specifically, in development of increasingly more complex levels. Our engineering service allowed for editing of the model prompts, so at some points, things were made easier, by further specifying the prompt as necessary for the desired outcome, but many times errors were so close to the finish line that it would be simpler to adjust the code and names.