Transforming Tabular Data: Leveraging Prompt Engineering with LLMs for Scalable Feature Engineering

Author: Aniket B. Barphe

LinkedIn Profile: https://www.linkedin.com/in/aniiketbarphe/

Competition Category: Role of Prompt Engineering for Tabular Data Tasks

Challenge: Develop solutions that leverage the potential of large language models (LLMs) to transform how we interact with, analyse, and derive insights from tabular data.

Ideathon Platform: MachineHack (Nov 29 - Dec 22, 2024)

Project Name: Runnable Application for Feature Engineering Using Prompt Engineering

Here's a step-by-step guide to running the provided Python script:

Step 1) Prerequisites: Install required libraries using pip in the Anaconda Prompt

```
pip install openai pandas streamlit
```

Step 2) Save the Script:

2.1) Open a text editor (e.g., Notepad).

2.2) Copy the python code provided in the file and paste it into the editor.

2.3) Save the file with the name "feature_engineering_app.py".

Step 3) Set Up OpenAI API Key: Replace "your_openai_api_key" in the script with your actual OpenAI API key

```
openai.api_key = "your_openai_api_key"
```

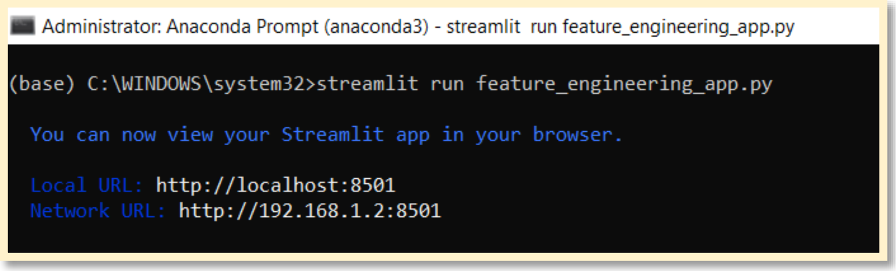
Step 4) Run the Application:

4.1) Open an Anaconda Prompt & Navigate to the directory where you saved the feature_engineering_app.py file.

4.2) Start the Streamlit App: Run the following command to start the app

```
streamlit run feature_engineering_app.py
```

4.3) Access the Application in Your Browser: After running the above command, Streamlit will display a local URL in your terminal, something like this:


Administrator: Anaconda Prompt (anaconda3) - streamlit run feature_engineering_app.py

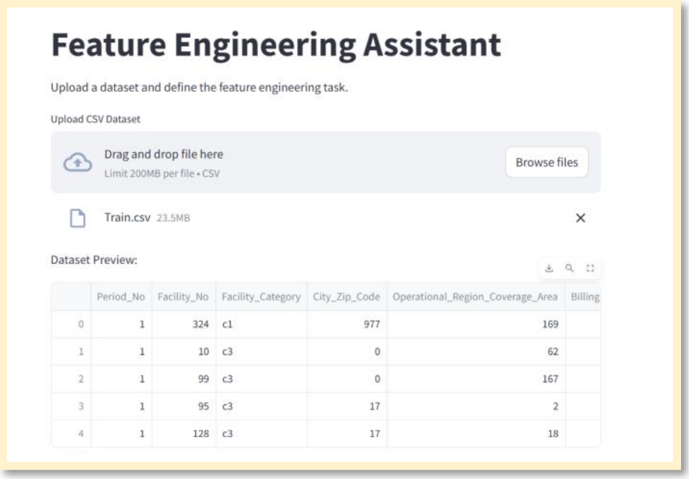
```
(base) C:\WINDOWS\system32>streamlit run feature_engineering_app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.1.2:8501
```

Open the Local URL in your browser to interact with the application.

Step 5) Using the App:

a) **Upload Dataset:** Click on the "Browse files" button in the app to upload a CSV file.

Example CSV:



Feature Engineering Assistant

Upload a dataset and define the feature engineering task.

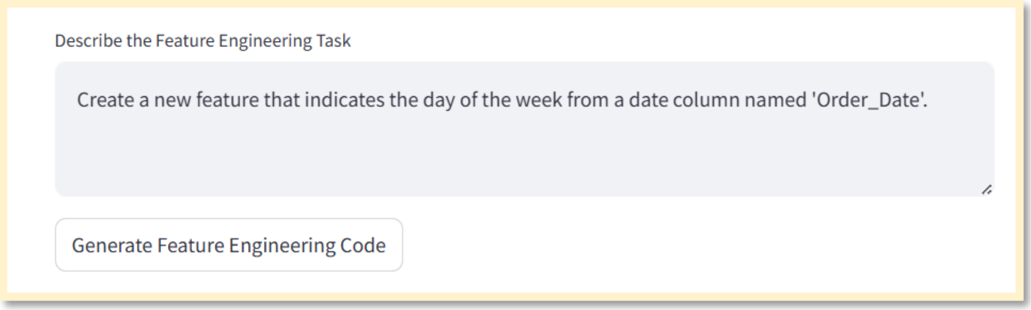
Upload CSV Dataset

Train.csv 23.5MB ✕

Dataset Preview:

	Period_No	Facility_No	Facility_Category	City_Zip_Code	Operational_Region_Coverage_Area	Billing
0	1	324	c1	977	169	
1	1	10	c3	0	62	
2	1	99	c3	0	167	
3	1	95	c3	17	2	
4	1	128	c3	17	18	

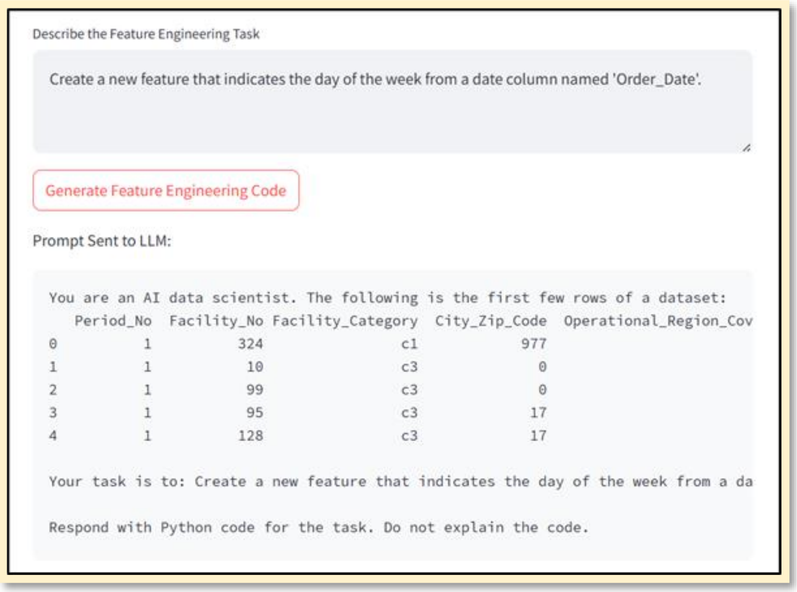
b) **Describe the Feature Engineering Task:** Enter your desired feature engineering instruction in the text area, such as:



Describe the Feature Engineering Task

Create a new feature that indicates the day of the week from a date column named 'Order_Date'.

c) **Generate Code:** Click on the "Generate Feature Engineering Code" button.



Describe the Feature Engineering Task

Create a new feature that indicates the day of the week from a date column named 'Order_Date'.

Prompt Sent to LLM:

```
You are an AI data scientist. The following is the first few rows of a dataset:
   Period_No  Facility_No Facility_Category  City_Zip_Code  Operational_Region_Cov
0          1          324                c1            977
1          1           10                c3              0
2          1           99                c3              0
3          1           95                c3             17
4          1          128                c3             17

Your task is to: Create a new feature that indicates the day of the week from a da

Respond with Python code for the task. Do not explain the code.
```

The app will display the prompt sent to the LLM, the Python code generated, and (optionally) the transformed dataset.

Results: This application demonstrates the integration of LLMs with prompt engineering to automate feature engineering, highlighting its scalability across various tasks. It provides an accessible, user-friendly interface to enhance the productivity of data analysts and non-technical users.