**Abstract**

I will attempt to use prompt engineering for ChatGPT to create a Python program to create a neural network sequence model for time series forecasting. In addition, I will ask ChatGPT to create a streamlit app that will load the results of a trained model and output forecasts for a user input datafile.

**Loosely following the COSTAR framework**

*I am doing an assignment on generative AI and would like help from chatGPT to provide sample code for my assignment using pytorch, Pandas, NumPy, and the streamlit package in Python.*

*Your objective is to write clean and understandable code in Python so that the end result is a program that can be run with no or minimal manual edits.*

*Your audience is someone with solid understanding of Python but not the Pytorch package. Be sure to have lots of comments around what each API call to Pytorch package is doing.*

*Your code should be well commented and written in a way that is easy to read. Be sure to consider runtimes and efficiency of the code that is returned.*

*You should also retain memory of previous code that is generated so that you can make edits easily from further prompts.*

A screenshot of a chat

Description automatically generated

**Data**

The data is proprietary data from my work which is sales data from grocery stores in USA. I have included a snippet of fake data as an example.

**Prompt – major subsequent revisions in RED or ~~strikethrough~~**

I have a csv file with the following columns in double quotes, “CATEGORY, DATE, LINE\_GROUP\_NAME, TOTAL\_QUANTITY, ON\_PROMOTION”. The dataset is unique by LINE\_GROUP\_NAME and DATE columns. ON\_PROMOTION is a continuous variable ranging from 0 to 1. There are multiple separate time series in the data. Below is an example of the data in csv format with headers.

``````

CATEGORY,DATE,LINE\_GROUP\_NAME,TOTAL\_QUANTITY,ON\_PROMOTION

175-2,9/6/23,123@abc,1,0.5

175-2,9/7/23,123@abc,2,0

175-2,9/8/23,123@abc,2,1

175-2,9/6/23,999@abc,10,0.999

``````

Create a Python script that will do the following steps.

1. Read in the csv file located at “./data/forecast\_data.csv”

2. Create a time series model using Pytorch that will train each LINE\_GROUP\_NAME and DATE time series. The model should forecast both ON\_PROMOTION as well as TOTAL\_QUANTITY. Train each LINE\_GROUP\_NAME separately on the same model. There is no need to do feature scaling. Do not use the Dataloader or Dataset Pytorch API.

3. The model should be able to forecast off of an input time series of DATE, TOTAL\_QUANTITY, and ON\_PROMOTION

4. After training the script should save off the model in the following directory “./model”

``````````````````````

Create a second Python script that will do the following steps.

1. Read in a model and load the model that was saved off from the previous script. Streamlit should cache the results so that subsequent calls will be faster.

2. Use streamlit for the user to input a time series of DATE, TOTAL\_QUANTITY, and ON\_PROMOTION.

3. There should also be another user input cell for the user to input the number of days to forecast.

4. Upon press of a button, Streamlit code should read in the user inputs above and call the model to generate the forecast values, both TOTAL\_QUANTITY as well as ON\_PROMOTION.

5. Streamlit should be used to then display the forecasted results.

Deleted Instructions

The model should consider the CATEGORY which the time series belongs to. A single time series can only belong to a single CATEGORY.

. ON\_PROMOTION can vary with each DATE.

The model should forecast values assuming the same ON\_PROMOTION value from the user input during the forecasted period.

, a constant value representing ON\_PROMOTION value during the forecasted days. The CATEGORY of the forecast period will be the same as the CATEGORY of the input series in the previous step.

~~3. There should be a second, separate series of cells for the user to input a series of CATEGORY, ON\_PROMOTION values to serve as the covariates for the forecasted series. The number of forecasted time series steps should be equal to the number of covariate CATGORY, ON\_PROMOTION values.~~

**Prompt Engineering summary**

Originally I tried to have the neural network take into account the CATEGORY as an additional feature but it was too difficult. I also wanted to use ON\_PROMOTION as an additional feature instead of another output but that also ran into issues during the forecasting portion of the code. Therefore I had to simplify the code to do a more simplistic forecast instead. The code now takes in a series of TOTAL\_QUANTITY and ON\_PROMOTION and forecasts the same two columns.

**Code Revisions**

You can see my code revisions by going to the Github account and looking at the difference between the first and last commits. The first commit is the raw code output from GPT4 and the last commit is my work that produces the streamlit app.