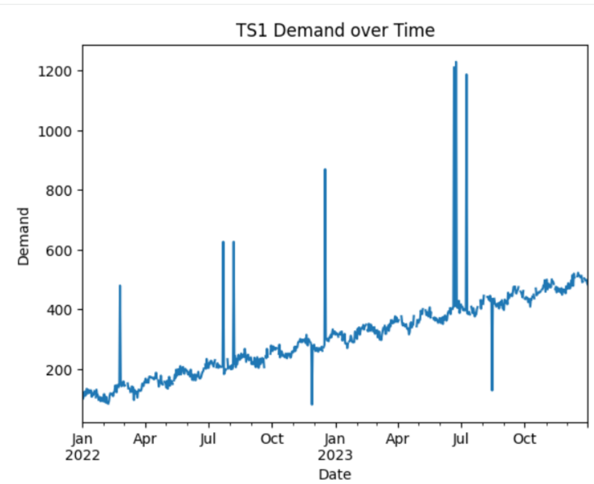
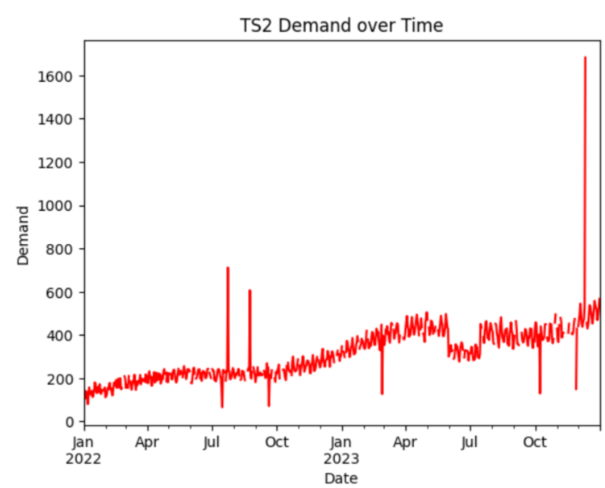
Report Out

1. What did you observe initially with the two time series? Mention missing values and anything you observed in regard to trend, seasonality, and outliers and other noteworthy features. Include screenshots of your initial plots.
    a. I decided that I did not like my dates to be out of order, so after setting the index, I sorted the index from earliest to most recent. When doing that, I immediately noticed that ts1 had a NA value for 2022-01-01.
    b. In regard to trend, the overall appearance of each dataset shows an increase.
    c. In terms of seasonality, ts2 seems like it starts increasing during the winter months and decreases during the summer months. ts1 has an increasing sin pattern going on. I am not sure what to make of it as it is increasing through the duration, but every other month there is an increase/decrease pattern.
    d. There are some very distinct outlier patterns showing in the initial plots that do not account for cleaning the outliers or NA values.





2. Explain what steps you went through to prepare the dataset for AutoML. What steps did you use AI for and what methods did you end up using for filling missing values and handling outliers? Explain the reasoning or rationale behind these choices. (You may have to lookup a particular function or method that was suggested by AI that you used if you do not know what it is or does).
    a. # Fill NaN values using forward fill and backward fill
        i. I used fillna() method declaring either forward fill or backward fill because I wanted the data point to represent a similar value to the days around it, not just using the mean or median values which are affected by many more dates.
    b. # Determine outliers using Z-score
        i. I imported z-score from scipy.stats to determine outliers in each dataset that were 3 standard deviations from the mean or more. I think with this dataset, outliers should be held to the 3 SD rule to qualify it as an outlier.

c. # List each date that was determined to be an outlier
   i. I used the previous code and np.abs(z-score) > 3 to list the outliers in each set.
d. # Smooth the outliers found in the previous step by replacing them with the mean of the surrounding values
   i. With the assistance of copilot, I was able to use a for loop to iterate through the dates so that I could use the demand value on either side of the outlier day in question, and use the mean of those two days to smooth the outliers. The pd.timedelta was new to me so I had to look that up. The best I can tell, it is used when needing to make calculations of time data (ie. minutes, hours, days, months, years).

3. Give a brief summary of the three methods you used within AutoML for making forecasts. You can (but are not required to) use any GenAI (e.g. ChatGPT) of your choice to help with this, but don't copy and paste. I suggest an initial prompt and then following up with a couple questions of your own to clarify things you might initially not understand. Then synthesize it all in your own words. Two to three sentences max per method. Consider also comparing and contrasting the three methods as to their strengths and weaknesses at the end of this section.
   a. Auto Arima
      i. ARIMA stands for Autoregressive Integrated Moving Average. The model repeatedly calculates the difference of the previous value and current value within a dataset. This causes the data to become stationary to generate a stable mean for forecasting.

   b. Exponential Smoothing
      i. The recent observations use a weighted average to forecast, with more recent observations being deemed as more important. Exponential smoothing uses the data to show a general pattern. It is most useful when the data does not seem to have a trend or seasonality that is apparent.

   c. Prophet Model
      i. The Prophet Model takes components trend, seasonality, holidays, and error to combine them to forecast data. The Prophet Model was created for datasets that can be very easily determined to have an apparent trend or seasonality to them.

4. What was the winning forecast for each time series and for each timeframe? Were they the same? After learning about the models are you at all surprised based on your initial views of the time series (before doing any cleaning or preparation) which ones were chosen?
   a. Time Series and their winners:
      i. Time_series_ts1_90: Winning forecast was Prophet Model
      ii. Time_series_ts1_180: Winning forecast was Prophet Model (Although it said something about VotingEnsemble, which was technically better by 0.0023)
      iii. Time_series_ts2_90: Winning forecast was Auto Arima Model
      iv. Time_series_ts2_180: Winning forecast was Auto Arima Model

b. I think that it makes sense, now after researching the prophet model that it won ts1. The seasonality (sin looking pattern) is perfect for forecasting with that model. The same can be said for the Auto Arima model winning ts2, albeit for different reasons. There is a pattern to ts2, but that pattern seems more stable in producing the mean for forecasting. There are not a lot of big peaks and valleys in ts2 and Auto Arima model thrives on that for forecasting.

5. Finally, summarize your overall experience. Possible things to address: Was this assignment more difficult or less difficult than you anticipated? How much time do you think it took overall? What was the most time-consuming part, and what parts went the quickest?

    a. I started the final project this morning at 10:30. As I am writing up this summary document, it is going on 4:15. I still have two more Azure runs to complete, 90 and 180 days on ts2. I have taken several 5 minute breaks every 25-30 minutes or so. I am just finishing this report at 6:20.

    b. The first part was good, but it took me longer than I wanted it to. I was trying to play around with seaborn to make the dates as nice as the og pyplot, but couldn't, and I decided I had to keep moving on to finish the other parts of the project.

    c. The slickest part, after I figured out how to get the copilot working, was utilizing copilot for creating code on Github. That was much smoother than anticipated.

    d. The thing that was taking the longest/most frustrating was figuring out how to make Azure work. There were a lot of struggles. Once I got the forecast finally up and running, I felt good. Then the results came in and for the life of me I could not find/get a csv file. JSON files were to be had, but not csv. As soon as I was about to stop for a meal break, and email you, then you sent that announcement in the class. I am glad I had my email up which notified me. I am not as frustrated at this point. Now I am going to power through to finish.