Forecasting Temperature using ARIMA Models

Data is extremely important in life. I believe that within everything, analyzing relevant data creates clear benefits within that particular study. One aspect of life that is influential to everyone is the weather. Weather is important to be able to forecast as it affects many things. When we can predict the weather, it becomes much easier to prepare for disasters or traveling. Weather can be well predicted using data and algorithms. Analyzing past weather patterns, it becomes easier to predict future weather.

Now with AI, machine learning and LLMs, we can much more easily use past data to create predictions and analyze data patterns more efficiently and efficiently. Looking through the many new options the ARIMA model uses past values of the time series to forecast future values. It assumes that the current value is a linear combination of previous values. This methodology is perfect for predicting weather patterns and is being used for the project.

Methodology

The idea is to look at and graph past data and use that past data to train an LLM. Training an LLM with our weather data allows that LLM to make predictions based on the training data. Telling the LLM to predict the weather tomorrow will have the LLM look at past data and make a guess as to what the weather may be. Being able to create these raw predictions can be useful since we can compare those predictions to the predictions of the media and see how they compare. We can compare our findings to the actual weather as well after the set day. It should be possible to analyze how effective predicting the weather with LLMs is using this technique.

The raw data acquired will be daily since that will be enough to capture daily trends and long term trends as well. Using Python libraries we can manipulate the data into a more graphable form for proper analysis.

Data Sets

There are many feasible options for acquiring past weather data. Since we are going to be using python for our overall analysis, finding a good API that allows easy integration of data into a python environment is key. After some research, Tomorrow.io is a good choice for API data. Tomorrow.io allows direct downloads of weather data in many categories like temperature, rain, humidity and sun exposure. Using Tomorrow.io allows easy downloading of weather data in a desired format that will make transformation easy.

After downloading the desired CSV files for our weather data, there are many options for data transformation that we can use to better represent our data. Numpy and Pandas python libraries help our manipulation as we can convert a CSV into a Pandas data frame which is made to be used with python. Using MatPlotLib can finally let us graph our transformed data in a much more readable form. The neat thing about Tomorrow. IO is that you can download CSV weather predictions as well, and should be in a similar format. Using these various libraries along with our data sets and LLMs, showing a perfect relationship between past, future and present data having been calculated in different ways.

Having accurate forecasting data from Tomorrow IO is helpful, but that isn't the only data that will be useful for this study. Having historical data will allow us to attempt to forecast past data and compare it to what that temperature actually was at that point. Getting this data from the National Weather Service will prove worthwhile.

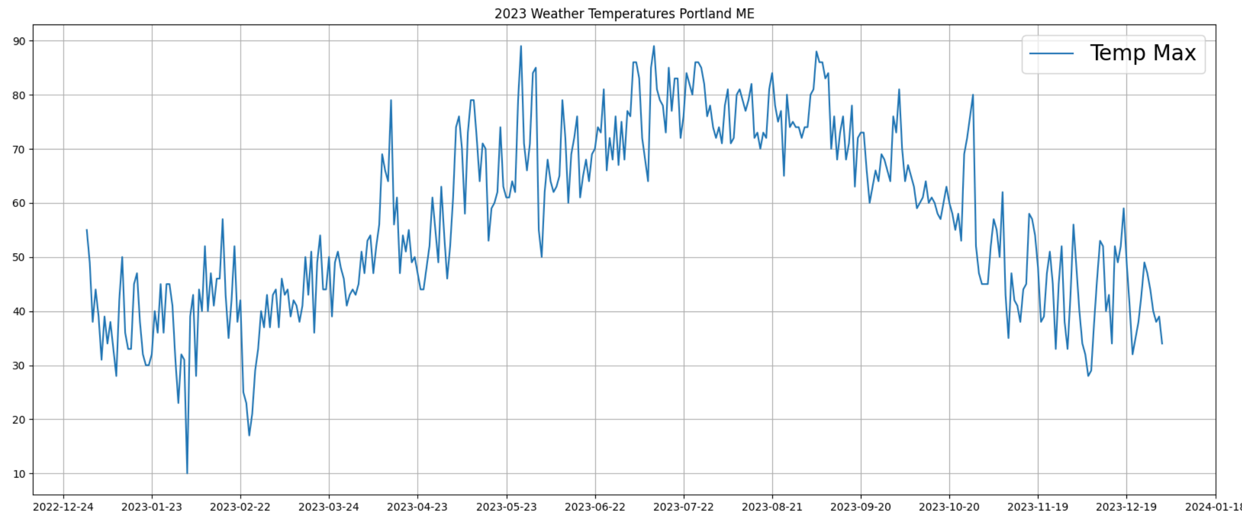
Data gained for this experiment will have many different pieces of information that won't be able to be inserted into an ARIMA model. We would only be able to use one type of training data for an ARIMA model such as temperature. Both historical and forecasted data will be guaranteed to have temperature which makes it most suitable for ARIMA analysis.

Results

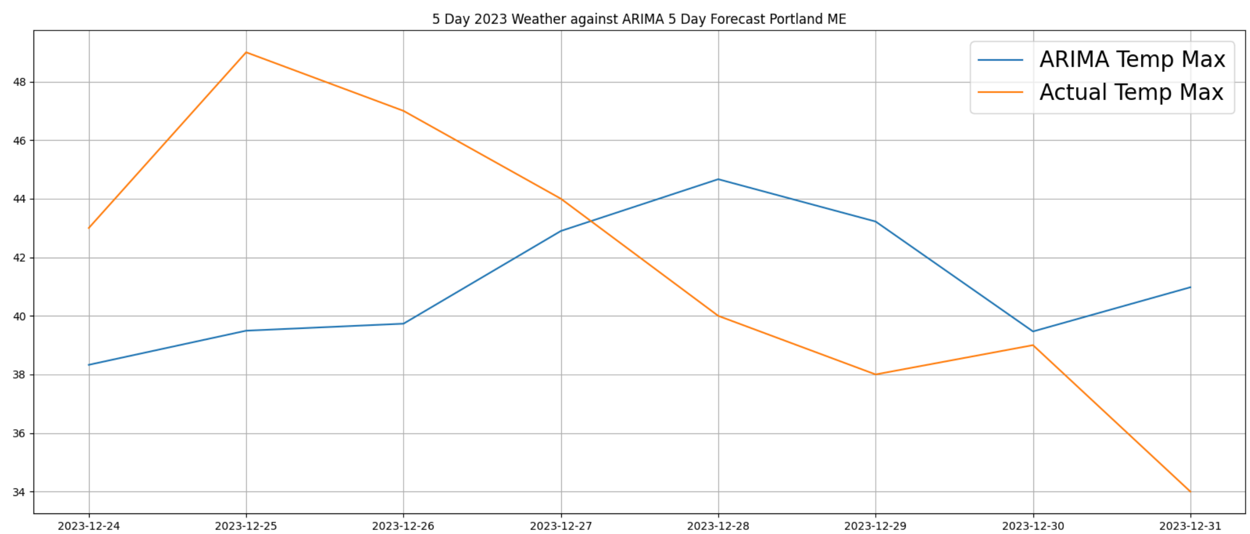
How much data will be necessary to train an LLM?

An LLM needs historical data to be trained to make predictions. Typically these models will need plenty of data in order to make calculations. There is an abundance of weather data but just how much will be necessary to train an LLM?

Beginning with one year of daily temperature data from National Weather Service and training an LLM:



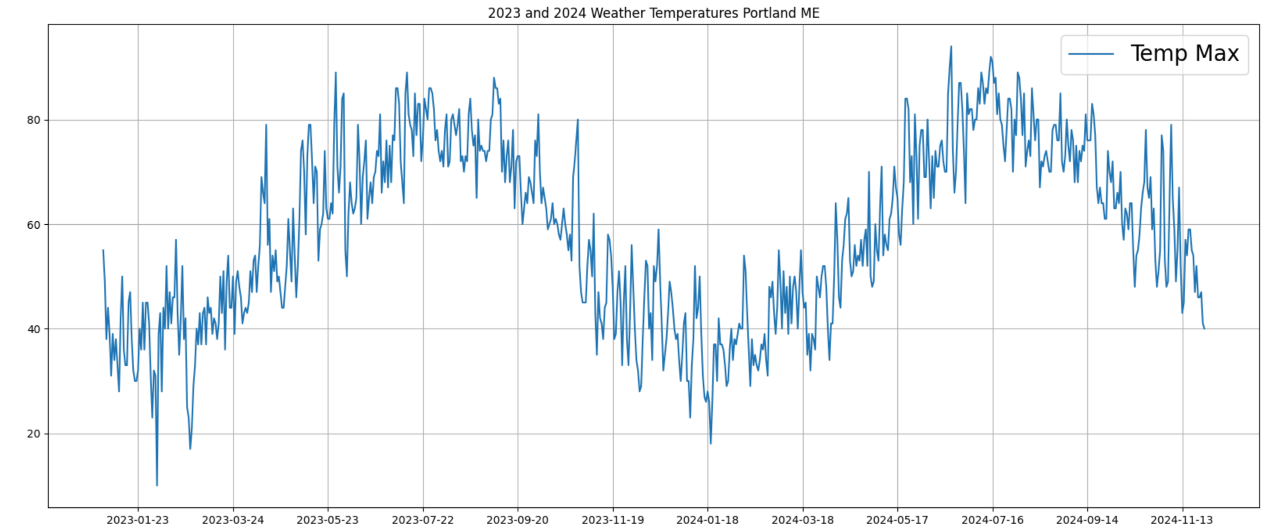
The data looks to have a general seasonal curve with daily dips and peaks relative to the curve position. This data will be good to see how an LLM can spot long term and short term trends. The first experiment will be attempting to forecast the last 5 days of the above data and see how it matches up:



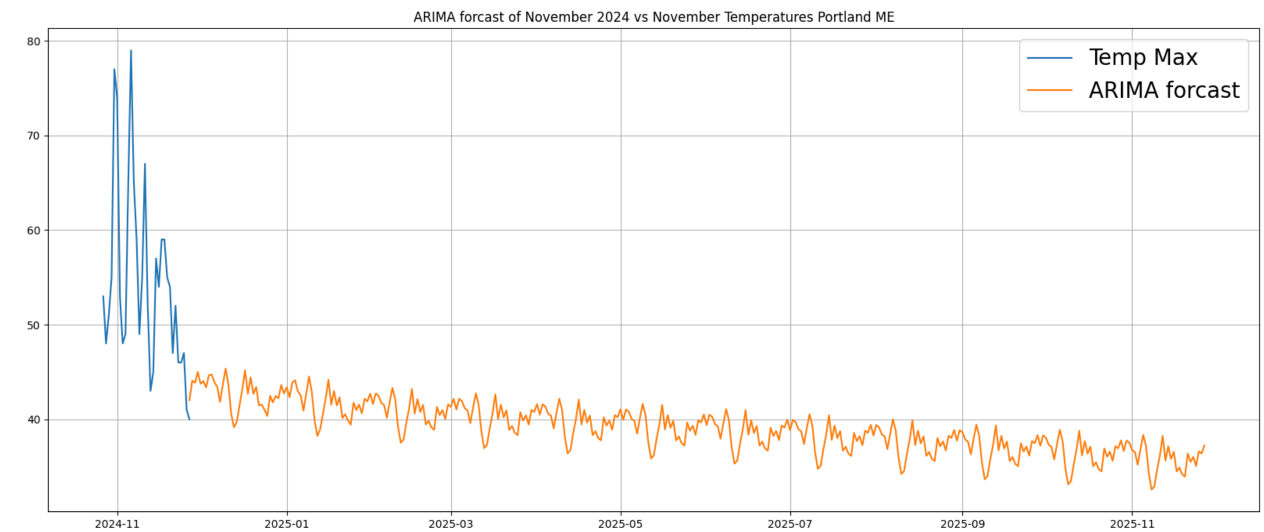
It looks like the LLM does a solid job of having its predicted temperatures within a reasonable range of the real weather. As suspected, the LLM might have a hard time predicting exact numbers without other types of data such as storm forecasts. But clearly here, a year of weather data gives the LLM enough of an idea to make predictions.

How far into the future can these forecasts be accurate?

Weather forecasts typically get less accurate as the forecast gets farther into the future. Forecasting weather also becomes less helpful as the forecast gets farther into the future. Having a solid weekly forecast is the most beneficial. But, lets see how accurate a longer forecast can be. Using two years of historical weather data for training:



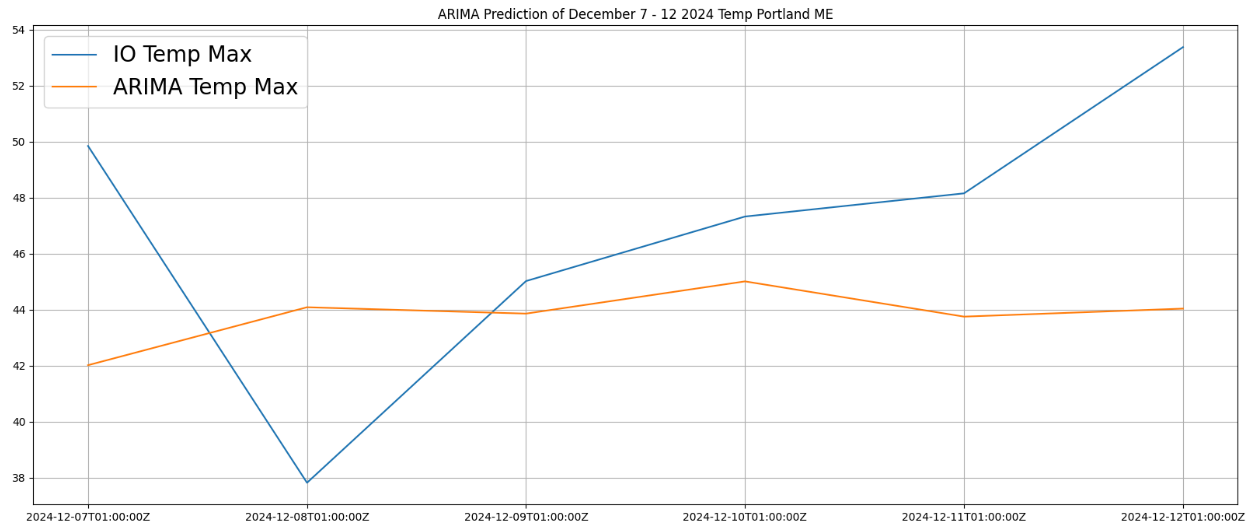
And attempting to forecast an entire year:



As expected, the ARIMA model has a hard time past the first few weeks of the year and becomes completely unusable soon after. This is not a super disappointing outcome since a year forecast would probably be pretty inaccurate even if it was usable. The most disappointing aspect of this experiment is that the training data has a very clear sine-like trend over the two years of time. The ARIMA model should have been able to catch that yearly trend even if the predicted individual numbers are repeating. This proves that the ARIMA model cannot capture long term trends like this. So we shall stick to analyzing short term forecasts.

Can this method still be viable without any future input data?

So far, we have only attempted to forecast weather data for dates that have already passed. Now, using Tomorrow.IOs own forecast data, we can try to forecast dates that have not happened yet. This is where we can prove whether or not these forecasts may be practical for use. Using the same two years of data above, the trained LLM is set to forecast 5 days into the future and graphed over Tomorrow.IOs own forecast:

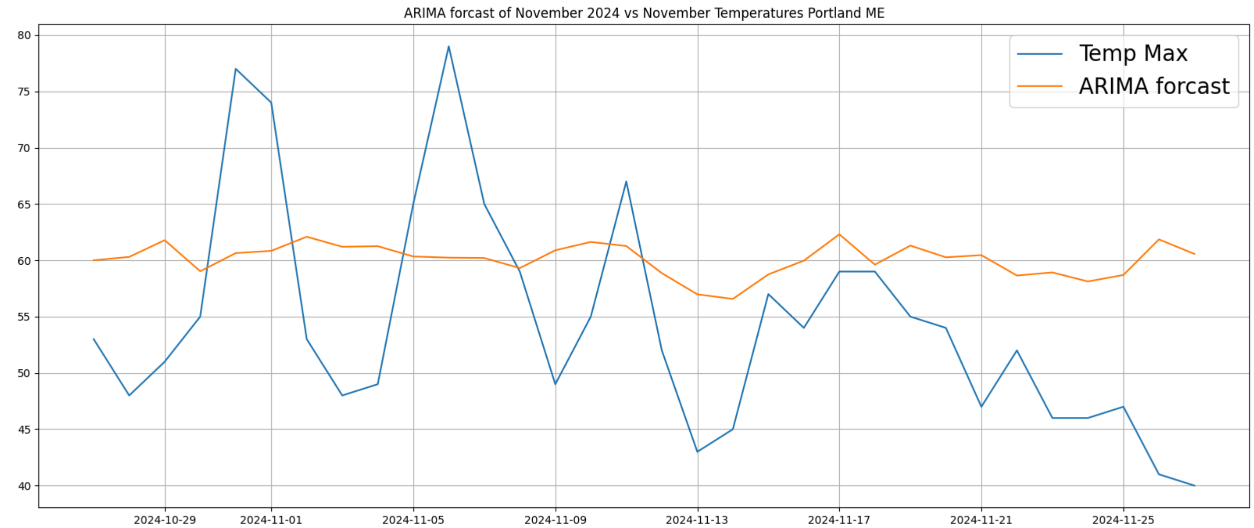


The ARIMA model does get its individual numbers in the right areas, but has a hard time picking up on one off spikes or dips that may occur in the forecast. It is set to snow tomorrow which is why the IO temperature dips on the 8th, the ARIMA model might see this dip given this information

What kinds of data would be good for forecasting and which types of data can best assist in forecasting? And how accurate can these forecasts be?

Based on our discoveries above, it would seem that creating a custom ARIMA LLM model that can have storm information imputed into it would greatly improve the LLMs ability to forecast. With an advanced model, adding weather factors that induce storms, like humidity, air fluctuations or pressure might allow for more complex predictions.

ARIMA models might not be able to predict weather on its own. A more viable use of LLMs for weather forecasting might need to be semi automated with human input data in specific areas, leaving the LLM to the calculations. Asking an LLM to use past weather data to calculate future weather data might be too much of an ask for a computational language model. See a final month long prediction vs existing data:



The ARIMA model simply cannot calculate these highs and lows purely based on numeric calculations.

Discussion

After many experiments with ARIMA models and predicting weather patterns, it has been determined that an ARIMA model is not practical for weather forecasting. ARIMA models fail to capture short and long term trends within historical weather data. The ARIMA model does get its predicted numbers in the right areas with no real outliers which is a positive, but overall yields unusable data past a few days of forecasting. Possibly more practical may be a custom ARIMA model that is used in unison with a meteorologist for pure calculation purposes as weather predictions seem to be too hands on for pure automation.

Per ARIMA, models can be inaccurate if there are certain outliers in reality such as a financial crisis or depression. For weather predictions, storms or other weather anomalies will yield those same inaccuracies for ARIMA models. Based on this, it would be important to focus on analyzing the individual numbers outputted by an ARIMA model in a stable environment.

In the future, it may still be feasible to have an autonomous weather forecasting system using ARIMA LLMs that would have higher levels of complexity and data input than current models. For future studies, it could be possible to have multiple simple ARIMA LLMs alongside one another that focus on different types of data required for proper forecasting. For example, there could be one LLM for temperature, one for humidity, one for air pressure and so on. With all these models together, Python programs could place the different results of each ARIMA LLM in the system together into a single data set that is then analyzed by a final ARIMA model for a single forecast.

There are many undiscovered possibilities of ARIMA models and LLMs in general, but to fully understand their abilities, more research and experiments need to be done.

Reference

Data sources:
https://docs.tomorrow.io/reference/welcome
https://www.weather.gov/

R. H. Shumway and D. S. Stoffer, ARIMA Models, pp. 75–163. Cham: Springer International Publishing, 2017

A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, Cambridge, UK, 2014, pp. 106-112, doi: 10.1109/UKSim.2014.67.

Yi Yan, Linjing Guo, Jiangting Li, Zhouxiang Yu, Shuji Sun, Tong Xu, Haisheng Zhao, Lixin Guo, Hybrid GRU–Random Forest Model for Accurate Atmospheric Duct Detection with Incomplete Sounding Data, Remote Sensing, 10.3390/rs16224308, 16, 22, (4308), (2024).