Predictive Weather Analysis using an ARIMA-based Machine Learning Model

Bryce Hayes

Computational Physics, Westminster College, Pennsylvania, United States.

Abstract

Using an autoregressive integrated moving average model (ARIMA), we will predict weather patterns within Los Angeles, California by using an open source data set containing Los Angeles's surface temperatures spanning 160 years. In doing so, we will also plot the data in an efficient and intuitive way to differentiate differences and predictions in temperatures.

Keywords: ARIMA, Python, Machine Learning, Climate Change

1 Introduction

Modern weather forecasting is one of the most complex system available, thus making it ideal for today's computer architecture and machine learning techniques. Even with the rise in popularity and growth technologically, modern systems are only able to predict weather patterns accurately within a few days. However, as machine learning algorithms continue to be optimized and technological growth continues, accuracy for weather modeling increases incrementally. Thus, within this paper, we will describe an autoregressive moving average model (ARIMA) that can predict weather patterns in Los Angeles over a period of 62 months with an average error of 1.928 degrees Celsius.

2 Data Set

The data set that will be utilized in this model is "Climate Change: Earth Surface Temperature Data."(1) Within this data set, lies average temperatures using a monthly period time scale ranging from the year 1750 to 2013. This data set also includes a 95% confidence interval around the average temperature, maximum temperature, and minimum temperature beginning in 1850. The CSV file that we will be using is GlobalLandTemperaturesByMajorCity.csv. Since we will be looking at Los Angeles, our first initial average temperatures begin in 1849 and end in September of 2013, thus giving us 1,976 data points. However of those data points, we will only be utilizing year's 1962 to 2013, thus giving us 620 data points, in which 62 will be within our test set and 558 in our training set.

3 Machine Learning Model

Within this section we will be discussing the autoregressive moving average model (ARIMA) that we will be using to determine our weather predictions. Before continuing to the model that was chosen, we must first understand what an ARIMA model is, and what it does. ARIMA is a statistical analysis model that utilizes non-stationary time series data (data that does depend on the time at which the series is observed) to predict future trends. This is modeled by the equation below.

$$\Delta y_t = c + \phi_1 \Delta y_{t-1} + \theta_1 \epsilon_{t-1} + \epsilon_t \tag{1}$$

Where c is an intercept of the autoregressive moving average, Δ is the first difference operator, and y is the time lags (fixed amount of passing time). The remaining terms are determined computationally. The final step before a model can be trained, is to identify the parameters that optimize our model. These parameters are given by (p, d, q), where p is the lags in the autoregressive model, d is the differentiation order, and q is the moving average lags. These values are determined computationally as well.

The specific ARIMA model that we will be using is a seasonal autoregressive integrated moving average with exogenous factors (SARIMAX). SARIMAX is an updated version of ARIMA that includes seasonal effects and exogenous factors with the autoregressive and moving average components. Seasonal effects mean that it can handle data with a repeating cycle, an area in which the ARIMA model suffers. Exogenous factors allow for weights to be implemented, primarily through introducing data that has trends. SARIMAX is modeled by the equation below.

$$\phi_p(L)\widetilde{\phi}_P(L^s)\Delta^d\Delta_s^D y_t = A(t) + \theta_q(L)\widetilde{\theta}_q(L^s)\epsilon_t \tag{2}$$

Where $\phi_p(L)$ is the nonseasonal autoregressive lag potential, $\widetilde{\phi}_P(L^s)$ is the seasonal autoregressive lag polynomial, $\Delta^d\Delta_s^D y_t$ is the time series differentiated d times and seasonally differenced D times, A(t) is the trend polynomial that includes the intercept, $\theta_q(L)$ is the non-seasonal moving average lag polynomial, and $\widetilde{\theta}_q(L^s)\epsilon_t$ is the seasonal moving average lag polynomial. Therefore, the SARIMAX model was decided upon because the average temperature trends are seasonal, which the ARIMA model can not implement successfully.

4 Machine Learning Code

In this section we will step through the code that was written to create a SARIMAX model. The libraries that were implemented include numpy, pandas, matplotlib, statsmodels, itertools, tqdm, and sklearn. The file will be read in using pandas, then any duplicates are dropped and any non-existent values are dropped using back fill (prior element is put in). We can now create a data frame using all the data involving Los Angeles. However, before continuing, a method needs to be implemented to ease getting data within some time interval. This is done by taking in the starting and ending year, and returning the data from the data frame.

Before we can begin building our model, we use the Augmented Dickey–Fuller test (adfuller) to determine whether our data is stationary. This value can be identified by the p-value, which in this case is 2.17×10^{-6}, and since it is much less than 0.5 (standard), it can be considered as stationary. Next, we split our data into a 90-10 training and testing set. We can now begin to optimize our model. Within a method to optimize our model, we pass in a list that will be ordered, and our exogenous factors. We then apply the SARIMAX keyword using the exogenous factors passed in, and make sure the output is ordered. Next we use the Akaike information criterion (metric that is used to compute prediction error) to determine our (p, d, q), and append it to a data frame and return it. This method can then be called. The ideal optimization parameters for this model were (6, 1, 6), which provided an AIC of 1977. The summary and diagnostics can then be printed. Finally, we can find our prediction by using the ideal optimization parameters, and applying them to the training set. The predictions can then be compared to the test set using a graphical output. These outputs included graphing the test set and prediction set with the error range and confidence interval. The resulting mean squared deviation was 1.928 degrees Celsius, which is the average of all of our errors.

5 Conclusion

Within this paper, we explained what an ARIMA and SARIMAX model are and how they work, modeled a complex system, and used error analysis to determine how accurate our model was. As a result, we created a model that span 62 months and averaged an error of 1.928 degrees Celsius. Although this error is high, it allowed us to see how a statistics based machine learning model could predict weather patterns. Further research could be done by implementing a deep learning algorithm such as the convolutional neural network.

References

[1] "Climate Change: Earth Surface Temperature Data." Kaggle, 1 May 2017, https://www.kaggle.com/berkeleyearth/climate-change-earth-surface-temperature-data?select=GlobalLandTemperaturesByMajorCity.csv.

[2] Patel, Harsh. "What Is Arima and Sarima Model?" Medium, Becoming Human: Artificial Intelligence Magazine, 14 Sept. 2020, https://becominghuman.ai/what-is-arima-and-sarima-model-10972b5e13c0.

[3] Verma 30/07/2021, Yugesh, et al. "Complete Guide to SARIMAX in Python for Time Series Modeling." Analytics India Magazine, 29 July 2021, https://analyticsindiamag.com/complete-guide-to-sarimax-in-python-for-time-series-modeling/.