The Implementation of Time Series Analysis for Forecasting Temperature using SARIMA

Ömer Tüzün
Department of Computer Science
Florida State University
Tallahassee, U.S.A.
GitHub: Silk-OT

Maddy Burns
Department of Computer Science
Florida State University
Tallahassee, U.S.A.
GitHub: 427am

Alexander Kajda
Department of Computer Science
Florida State University
Tallahassee, U.S.A.
GitHub: Akajda1010

Kaitlyn Franklin
Department of Computer Science
Florida State University
Tallahassee, U.S.A.
GitHub: Kaitlynfranklin

I. Introduction

Everyone interacts with the weather on a daily basis. Whether it is to know whether or not to bring an umbrella with you that day or to plan out events ahead of time, knowledge of the current weather and predicting future weather is a vital aspect of modern life. However, although knowing the current weather and gauging the weather in the short term is easily done, predicting what the weather might be in the long term or even future climate trends can be an arduous task. Meteorologists tackle this issue by using information gathered from radar, satellite, and other ground-based and airborne tools on computer programs to produce an analysis, or graphical representation of the weather based on current data. The meteorologists then analyze this output, using it alongside various other models and their personal experience in the field to present their weather forecast [1]. The goal of this project is to develop a computer program that can process and output a prediction of future weather and climate trends with corresponding visuals.

II. Background

Machine learning algorithms consist of two key steps: data collection and processing, and model selection and training. Datasets can be acquired from multiple sources such as NASA or AerisWeather, however, the data from the National Oceanic and Atmospheric Administration–or NOAA–was chosen for two reasons. For our implementation, the dataset needed to be in a .CSV file format and span many years with monthly data. NOAA's dataset fulfilled both of those requirements while also being easier to access data from. Although some of the others also fulfilled one or two of these as well, the dataset from NOAA was more accessible and thus preferred. The dataset used is from the Automated Surface Observing System–ASOS–at the Annapolis U.S. Naval Academy in Maryland and has data from November 2001 to January 2025. It includes the maximum, minimum, and average temperatures of each month, the average precipitation, and other data gathered that month.

III. Methods

A. Data Processing and Cleanup

The dataset is processed in data_processor.py–where the CSV file is opened–and all the necessary data (the temperature minimum and maximum, alongside the date) is extracted, with the extraneous data excluded. The average temperature is recalculated as the data provided by NOAA is truncated to the nearest tenth. Finally, the date is converted into DateTime format and the extracted and processed data is placed in a pandas DataFrame.

B. Algorithms–Choosing the Ideal Model

There were various machine learning models to choose from that could've been used, such as linear regression and polynomial regression. Through online research, another alternative model was found and deemed preferable: SARIMA or Seasonal Autoregressive Integrated Moving Average. In an article written by Can Özdoğar, Özdoğar shows his approach, implementing time-series temperature forecasting using SARIMA and even compares it to a linear regression approach, claiming that using SARIMA led to a noticeably significant increase in accuracy with its predictions [2]. Although this article was not a scholarly or peer-reviewed source, the anecdote brought SARIMA to our attention.

C. Types of Models to Consider

Linear regression is a machine-learning algorithm that tries to create a line of best fit to model the relationship between a dependent and at least one independent variable. This line can then be used to try and predict future values, making it ideal for variables with linear relationships. Alternatively, the Autoregressive Integrated Moving Average–or ARIMA–is a forecasting model for time series data with only one variable quantity, taking past values (autoregressive or moving average)

into account to estimate future values. Although ARIMA is a versatile model, it's best to use it only for data without any seasonal patterns, making it not ideal for predicting seasonal weather [3]. However, there is an extension of ARIMA called SARIMA which incorporates seasonal data, allowing for cyclical data to be better analyzed [4]. For this project, both linear and SARIMA models were used and compared against each other. Two values were observed and compared to evaluate the models' efficacy: the mean squared error and the mean absolute error. The mean squared error (MSE) is the average of all the squared differences between the predicted and actual data points, with a lower MSE implying a better prediction and vice versa [5]. Similarly, the mean absolute error (MAE) is the absolute difference between the predicted and actual data points, with the value quantifying how inaccurate the predicted values are [5]. Additionally, graphical visualizers were written using the matplotlib package to graph the resulting data.

D. Implementation

Both models run using the same class algorithms.py, with some of the functions shared between the two whilst others are specialized to each model. After the data has been cleaned and processed, the linear and SARIMA models are both fitted to the weather data and trained, with the linear model training its weights and bias for the gradient of the data set and the SARIMA model using the built-in SARIMAX function to fit the data to account for the date-series and seasonal order data. After both models are trained, the dataset is clustered to half the number of data points (N/2), with roughly 11 clusters being made from our dataset and anomaly detection is performed using a threshold of 2, iterating through each value's z-score to determine whether it logically fits within the dataset.

IV. Results

After both models were trained, their predictions were analyzed, and revealed that the anecdote from earlier was correct. The linear model resulted in an MAE of 3.22 and an MSE of 13.54 while the SARIMA model resulted in an MAE of 0.41 and an MSE of 0.21. It is reasonable to assume that SARIMA is more accurate than linear–by a factor of 7.8-64x depending on which score it's compared by–with the consideration that a combination of factors, such as the incompatibility of a linear model with a seasonal date-series dataset and the favorable compatibility of the dataset with SARIMA, affected the final result. In short, through the understanding of both models and the results shown, it can be reasonably assumed that SARIMA is a better fit for this project and more accurate.

V. Discussion

Although our results seem to conclude that SARIMA is a better fit for forecasting weather, future attempts may want to reevaluate this using different models. For example, instead of using a linear model, one may instead try using a polynomial model to account for seasonality. Something else worth considering is the performance impact, although, within the scope of a singular location, the processing wasn't too intensive, with larger datasets–such as multiple locations at a time–this may not be the case as performance efficiency wasn't measured during this project.

References

[1] N. US Department of Commerce, "Forecast process," National Weather Service, https://www.weather.gov/about/forecast-process (accessed Apr. 11, 2025).

[2] C. Özdoğar, "Time Series Forecasting using Sarima (python)," Medium, https://medium.com/@ozdogar/time-series-forecasting-using-sarima-python-8db28f1d8cfc (accessed Apr. 11, 2025).

[3] J. Brownlee, "A gentle introduction to sarima for time series forecasting in Python," MachineLearningMastery.com, https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/ (accessed Apr. 11, 2025).

[4] A. Bajaj, "Arima & Sarima: Real-world time series forecasting," neptune.ai, https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide (accessed Apr. 11, 2025).

[5] S. Mondal, "Understanding Mae, MSE, and RMSE: Key Metrics in machine learning," DEV Community, https://dev.to/mondal_sabbha/understanding-mae-mse-and-rmse-key-metrics-in-machine-learning-4la2 (accessed Apr. 11, 2025).