Forecasting Precipitation with SARIMA

Nicole Athanasiou
nathanasiou@wisc.edu

Catherine Seymour
caseymour@wisc.edu

Nabhan Kamarudzman
kamarudzman@wisc.edu

December 6, 2021

Abstract

In this project, we predict the amount of precipitation in Madison, Wisconsin by utilizing machine learning techniques, specifically SARIMA with Grid Search. We use the dataset Climate Data Search Tool which contains the monthly precipitation and temperatures from 1900 to 2020 in Madison, Wisconsin. We use Grid Search to find the best combination of hyperparameters and then applied them to the SARIMA Model. Both Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) are used to evaluate the model. The SARIMA model shows evidence of an overall increasing amount of precipitation, likely due to the effects of climate change. The MAPE and RMSE of the SARIMA Model yield 75% and 2.02, respectively. These accuracy scores provide a decent prediction given that weather forecasts are typically inconsistent and that they are solely based off monthly precipitation and inconsistent

1 Introduction

Climate change is leading to an increase in global mean temperatures, as well as the frequency, intensity, and amount of heavy precipitation [4]. Because of these impacts, predicting heavy precipitation events is becoming progressively vital for all people and policymakers. The purpose of our project is to predict the amount of precipitation in Madison, Wisconsin based off preceding data.

Numerical patterns that emerge from weather data sets provide an opportunity to not only project future weather conditions, but to also comprehend drastic changes in the classification of numerical patterns within weather forecasting. This project provides a predictive model to help measure the amount of precipitation in an event by evaluating various weather parameters from the years before. Because weather impacts every individual on Earth, there are many stakeholders that would benefit from such predictive analysis.

Farmers and agricultural developers rely on weather forecasting to proceed with their expectations on times of plantations and harvest. For example, early predictions of cold weather in Wisconsin would help local farmers determine when the best time is to plant crops to harvest right before the cold season begins. An accurate prediction of the weather conditions will improve the production of crops and be more successful for consumers. The next motivating factor would be to provide policymakers with better mediums to evaluate the consequences of past policies, or plan for future policies that mitigate the effects of climate change. Predictions can be based on policymakers "pursuing their current energy policies versus their stated energy policies" [1]. Such analysis provides a visualization for policymakers to evaluate what are the best policies to be implemented to mitigate the current ongoing effects of extreme weather changes.

Building accurate statistical models that predict the amount of precipitation given the date in Madison Wisconsin could have many benefits. It will help farmers make informed decisions about crop production, help policy makers make better decisions for the growth of the community, and help the general population by having an accurate weather statement each day. In this project, we predict the amount of precipitation in Madison, Wisconsin by utilizing machine learning techniques which include SARIMA with Grid Search.

2 Related Work

The first related work comes from an article written by Khoa Lai on implementing a Time Series Analysis and Weather Forecast on Python. This study focuses on implementing the SARIMA Model into forecasting weather[2]. In this study, he used a set of data collection where he "collected the average daily temperatures in Helsinki from September 2015 to May 2019". Methods were run to ensure that the data was suited for the model, in which is known as decomposition. Here we learned to visualize our data using a method called Time-Series Decomposition which decomposed data into three major components: trend, seasonality, and noise.

Here, with the right fitting, it is shown that the data would fit a trend of seasonality, which allowed us to ensure the optimum performance of our predictive model. One of the methods used to validate the model within this study was to check the value of the Root Mean Squared Error (RMSE). Since it was found to be very small, it meant that the data is suitable for the model and is estimating quite accurately. Next, the implementation of Grid Search in here was used to find the best hyperparameters for the performance of the model. The final outcome was graphs which showed that the performance of the SARIMA Time Series model is performing well. This means that the model does manage to estimate weather, assuming there are no exogenous variables.

The next piece of related work was written by Holmstrom, Liu and Vo in which they studied the application of Machine Learning to Weather Forecasting. In this scenario, they first looked a few models, but ended up doing a 4-fold forward chaining time-series cross validation which they also managed to generate a learning curve for the accuracy and performance of the model. A slightly different approach from us, we decided to run a SARIMA model instead, which already considered the concept of seasonality.

The results found in the paper were that the forecast models were really good at predicting, but the long run outcome would deviate from the true value. That is to say that the model is short term. Because the model has high bias, it was stated that this bias could be due to the structural decision to forecast weather based upon the weather conditions of the past few days, which is an arbitrary number, which may be "too short to capture trends in weather that functional regression requires"[3]. It is then concluded that more research must be made to lower its bias.

3 Proposed Method

3.1 Problem Description

The goal of this project is to predict precipitation in Madison, Wisconsin from preceding data. To address the goal, we considered regular machine learning such as Random Forests. The initial proposed method was a variation of Random Forests built for Time Series data, but it was required that the dataset be transformed into a supervised learning problem first, which created complications[1]. It also required us to use a walk-forward validation to evaluate the model.

Instead, it was decided that we use a univariate time series as the problem framework. The final proposed method we are using to construct a time series forecast is the Seasonal Auto-Regressive Integrated Moving Average (SARIMA), which is an extension of the ARIMA class of models. We used Grid Search to explore combinations of hyperparameters. The best combination was selected based on Akaike Information Criteria (AIC) as the loss function.

3.2 SARIMA

The SARIMA Time Series model is an extension of the ARIMA model, which stands for autoregressive integrated moving average. ARIMA models serve as an approach to time series forecasting. They focus on autocorrelation, rather than trend and seasonality. The autoregressive term (AR) and the moving-average term (MA) are the basis of ARIMA models. Given a time in an AR model, a value is seen as a weighted sum of prior values. On the other hand, a MA model sees the value as a weight sum of past residuals. Moreover, ARIMA includes an integrated term (I) to difference the time series. The mathematical notation can be represented as such[7]:

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

The c is the intercept of the ARIMA model, Δ is the first-difference operator, and we assume $\epsilon_t \sim N(0, \sigma^2)$.

To emphasize lag polynomials, we can rewrite the expression as such:

$$(1-\phi_1 L)\Delta y_t = c + (1+\theta_1 L)\epsilon_t$$

where L is the lag operator.

ARIMA models are usually written with the given notation:

$$\text{ARIMA}(p, d, q)$$

where

$$p = \text{lags in the autoregressive model}$$

$$d = \text{differencing / integration order}$$

$$q = \text{moving average lags}$$

The limitation to ARIMA Time Series modeling is that it is non-seasonal. This means that the data points must have a consistent linearity in a single direction. In terms of weather in Madison, Wisconsin, it is seasonal, that is to say that Madison has four seasons all year round, which fluctuates the precipitation of Madison's climate per month. Instead of manually separating the data points into seasonal months, it is possible to run SARIMA Time Series Model instead, which is for seasonal data.

Because of ARIMA's inability to control seasonality, SARIMA was developed to address the weakness. SARIMA follows similar traits to the ARIMA model, which is the essential reason behind understanding the application of ARIMA before proceeding with seasonality. This provided the ability to run a Time Series model that controlled seasonality within the data. Before we proceeded, we recompiled the data to ease input within the models and we ran a decomposition analysis to observe its compatibility with SARIMA. In this case, we checked if precipitation within the data has seasonality, which it did.

SARIMA stands for Seasonal Autoregressive Integrated Moving Average, which is an extension of ARIMA which considers seasonality. The mathematical calculations involved a number of differential equations in order to account for the time difference within the data. The main model can be expressed as such[6]:

$$\phi_p(L)\tilde{\phi}_P(L^s)\Delta^d\Delta_s^D y_t = A(t) + \theta_q(L)\tilde{\theta}_Q(L^s)\zeta_t$$

As a univariate structural model, it is represented as:

$$y_t = u_t + \eta_t$$
$$\phi_p(L)\tilde{\phi}_P(L^s)\Delta^d\Delta_s^D u_t = A(t) + \theta_q(L)\tilde{\theta}_Q(L^s)\zeta_t$$

where $n_t$is only applicable in the case of measurement error (although it is also used in the case of a pure regression model, i.e. if p=q=0).

Regression with SARIMA errors can be represented as:

$$y_t = \beta_t x_t + u_t$$
$$\phi_p(L)\tilde{\phi}_P(L^s)\Delta^d\Delta_s^D u_t = A(t) + \theta_q(L)\tilde{\theta}_Q(L^s)\zeta_t$$

This mathematical model is the one used when we consider exogenous regressors within the predictive model.

The reduced form lag polynomials are written, in mathematical form, as:

$$\Phi(L) \equiv \phi_p(L)\tilde{\phi}_P(L^s)$$
$$\Theta(L) \equiv \theta_q(L)\tilde{\theta}_Q(L^s)$$

In essence, SARIMA models can be written as a shorthand notation:

$$\text{SARIMA}(p, d, q)X(P, D, Q, S)$$

where

$$p = \text{non-seasonal autoregressive (AR) order}$$

$$d = \text{non-seasonal difference}$$

$$q = \text{non-seasonal moving average (MA) order}$$

$$P = \text{seasonal AR order}$$

$$D = \text{seasonal difference}$$

$$Q = \text{seasonal MA order}$$

$$S = \text{length of repeating seasonal pattern}$$

This shorthand notation makes it easier to implement such models in code, after cleaning and readjusting the dataset.

SARIMA confronts seasonality with the addition of seasonal AR and MA terms. SARIMAX further extends ARIMA by including the ability to handle exogenous variables. As of now, in order to maximize the performance of the model, we decided to stick with precipitation as our main variable of focus.

DATE	PRCP
1900-01-01	1.57
1900-02-01	1.23
1900-03-01	1.53

Table 1: The table shows a subset of the preprocessed data.

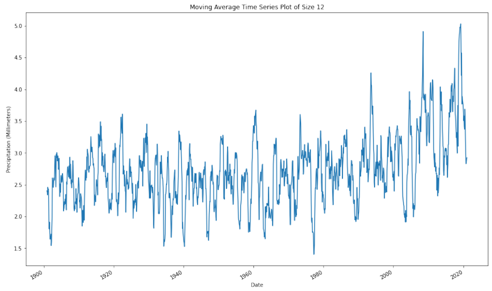

Figure 1: Moving Average Time Series Plot.

4 Experiments

4.1 Dataset

The dataset was obtained through the Climate Data Search Tool from the National Oceanic and Atmospheric Administration [5]. The dataset contains monthly precipitation and temperature measures from 1900 to 2000, recorded by a station in Madison, Wisconsin. The preprocessed time series data can be viewed as a plot in Figure 1, which has been transformed through moving average smoothing to reduce the noise from 1452 training examples. We can see a clear seasonality pattern, along with an increase in precipitation over time.

As we originally wanted to solve a multivariate problem with both precipitation and temperature, we attempted to implement Vector Autoregression (VAR). The model generalizes univariate autoregressive model by allowing for multivariate time series. However, the data are not stationary, meaning one criteria of the model was not satisfied. We did not move forward with the model as a result. A subset of the preprocessed dataset, without temperature, can be viewed in Table 4.1.

4.2 SARIMA

The baseline model was constructed based on the assumption that the precipitation the current month is dependent on the precipitation from the month before. 1-step prediction was applied to model the precipitation as a time series. We evaluated the model based on RMSE and MAPE and went on to implement SARIMA. To select the best combination of hyperparameters, we implemented Grid Search. The algorithm iteratively explored the combinations, allowing us to select the best combination based on the Akaike Information Criteria (AIC) values. We then confirmed that residuals were normally distributed by interpreting diagnostic plots. We set the forecasts to start at 1996 to the end of the data so that 20% of the data were forecasts. We looked at both non-dynamic and dynamic forecasts. Non-dynamic forecasts produce one-step-ahead forecasts so that forecasts at each point are based on the entire history until that point. On the contrary, dynamic forecasts leverage data from the time series until a specified point; after that point, values from previous forecasted time points are used to produce forecasts. To explore further, the dataset was divided into three bins: 1900-1940, 1940-1980, and 1980-2020. The non-dynamic forecasts were set to start 35 years from the start date of each bin so that 20% of the data were forecasts. We then evaluated the non-dynamic and dynamic forecasts of the entire dataset and the non-dynamic binned datasets using RMSE and MAPE.

4.3 Software

For model training, we used Jupyter Notebook.

4.4 Hardware

Each group member utilized his/her own laptop.

5 Results and Discussion

We used Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) to evaluate the model.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} ||y(i) - \hat{y}(i)||^2}{N}}$$

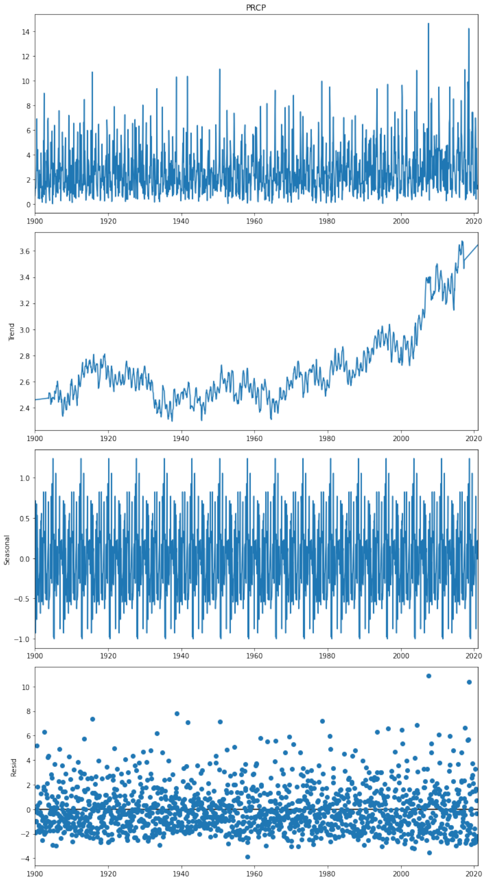

Figure 2: Additive Decomposition Plot.

$$\text{MAPE}= \frac{1}{n} * \sum \frac{|y(i)-y(\hat{i})|}{|y(i)|} * 100$$

where n is the sample size, $y(i)$ is the actual data value, and $\hat{y}$ is the predicted data value. Both RMSE and MAPE can be used to measure the predictive accuracy of a forecast. In particular, MAPE can be easily interpreted [4]. It is best used for high-volume data, making it ideal for our dataset. Any missing data were filled by propagating the last valid observation forward. Since the data are monthly, missing instances were minimal.

5.1 Baseline Model

We constructed the baseline model with 1-step predictions. To validate how well the baseline model performed, we looked at the RMSE and MAPE, which were 2.36 and 111%, respectively. The small RMSE indicates that the 1-step prediction baseline model can predict the precipitation of the upcoming month with an average error of 2.36 millimeters. The MAPE implies that the average difference between the forecasted value and the true value is 1.11%. The errors are acceptable, as the weather forecast field involves other attributes, such as humidity and gust wind speed. Moving forward with the baseline model, we explored a method to reduce the prediction error.

5.2 SARIMA Model

We chose to apply SARIMA, since the method is generally used for time-series forecasting with seasonality. The output of Grid Search suggested that $SARIMA(0, 1, 1)x(0, 1, 1, 12)12$ yields lowest AIC value of 5547. Thus, we selected hyperparameters for p, d, q, P, D, Q, and S correspondingly.

Once we selected the ideal set of hyperparameters, we had to check that residuals were normally distributed by interpreting diagnostic plots. From the top-right plot in Figure 3, we can tell that the red KDE line loosely follows the $N(0, 1)$ line. The overlap provides evidence that the residuals are normally distributed. The residuals do not follow the qq-plot line closely at the head and tail. The graph provides hesitant evidence that the residuals are normally distributed. The top left plot does not hint at seasonality. Moreover, the correlogram plot depicts that there is low correlation in the time series residuals. Based on the plots, we conclude that the model is a satisfactory fit. We can attribute the lack of normality in the QQ-plot due to the changing climate, as when the plot is generated with an earlier end date, the residuals fit the line closely.

Since the one-step time series model was determined to be satisfactory, we moved forward with producing forecasts. We began with comparing the forecasted values to the actual values of the time series. The comparison allows for validation of the forecast accuracy.

The non-dynamic forecast (Figure 4) yielded a

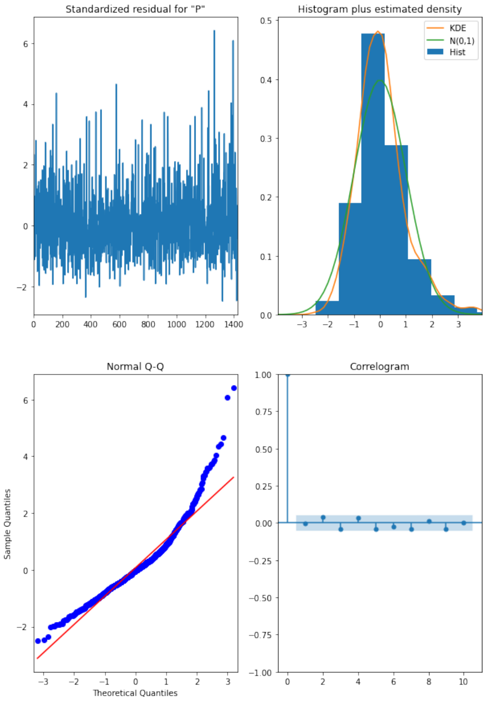

Figure 3: Diagnostic plots for SARIMA model with optimal hyperparameter combination.

RMSE of 2.02 and a MAPE of 75%, while the dynamic forecasted generated a RMSE of 2.06 and MAPE of 71%. The errors of the forecasts are not significantly different. We can use the non-dynamic model to predict the precipitation of the upcoming month with an average error of 2 millimeters or with an average difference of 75%. A MAPE value of this magnitude is typically interpreted as poor; however, given that the predictions were based solely off monthly precipitation and the typical uncertainty in weather forecasts, the prediction accuracy seems acceptable. The average precipitation in the dataset is 2.71 millimeters. The MAPE indicates predictions would be off by around 1.5 millimeters for an entire month. The error does not appear to be excessive in this context.

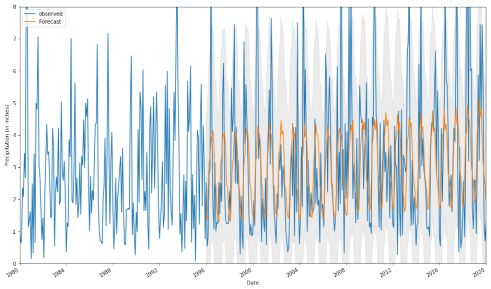

Figure 4: Non-dynamic forecast visualization. 1900-1980 were cut out to reduce noise.

Next, we divided the dataset into three bins to see how prediction error has changed over time using a non-dynamic forecast. The bins had RMSE and MAPE of 1.65 and 105%, 1.63 and 120%, and 2.58 and 94% from earliest to latest bin, respectively. The RMSE increased as time progressed. The relationship noted may provide evidence that precipitation is becoming more difficult to predict as time and climate change progress. Because of the magnitude of the errors and the hesitation regarding the diagnostic plot assumptions, we cannot conclude that there is a clear pattern.

6 Conclusions

We implemented a SARIMA model to forecast precipitation in Madison, Wisconsin. The results may support evidence for overall increasing precipitation due to climate change. Additionally, the results may imply that uncertainty in precipitation forecasts is increasing as time and climate change progress. We accomplished the initial task of predicting precipitation in Madison, although the error measures imply improvements could be made to our forecasts. The related works reached similar conclusions, suggesting that more research was needed to reduce deviations.

Future works should include daily data with variables such as snow depth, wind speed, and temperature. Our work is limited in this area, as the process for obtaining data through the Climate Data Search Tool from NOAA was time consuming. Given the constraints on the amount of data points the tool allows downloaded at a time and the time provided for the project, we were unable to include daily records. As well, future works should include an expanded set of variables. The time period we were interested in, 1900-2020, had large spans of missing data for snow depth, wind speed, and temperature. Considering these weather conditions in modeling and analysis would lead to greater robustness, since there would be enhanced information about how overall trends and uncertainty in forecasting are changing over time.

7 Acknowledgements

We are grateful for the data from NOAA, as well as the tutorial from Khoa Lai. Both resources played a key role in the success of the project.

8 Contributions

Catherine wrote the abstract and introduction. Nabhan wrote the related work and most of the proposed method sections. Nicole preprocessed the dataset, implemented the SARIMA model, and wrote the experiments, results and discussion, and part of the proposed method sections.

References

[1] J. Brownlee. Random forest for time series forecasting, 2020.

[2] K. Lai. Time series analysis and weather forecast in python, 2020.

[3] C. V. Mark Holstrom, Dylan Liu. Machine learning applied to weather forecasting, 2016.

[4] B. Nguyen. End-to-end time series analysis and forecasting: a trio of sarimax, lstm and prophet (part 1), 2021.

[5] N. Oceanic and A. Administration. Climate data online: Dataset discovery, 2021.

[6] statsmodels. Sarima model notes, 2021.

[7] Y. Verma. Complete guide to sarimax in python for time series modeling, 2021.