Sean Toon

IT-460-18500-M01 Machine Learning 2024 C-5 (Sep - Oct)

10/20/2024

**Final Project – Predicting Sunspots with ARIMA**

The case and data I’ve chosen to analyze is the Sunspot data to help predict future sunspots. The CSV file contains records of observed sunspot data, including variables like year, month, date in decimal form, number of sunspots observed, and a mark that might represent significant sunspot activity. My goal is to create a model that helps identify any patterns or connections between earth dates, sunspots, and solar cycles.

The model I’m using to predict future sunspots is ARIMA (AutoRegressive Integrated Moving Average). ARIMA is a great fit for time series data like this because it looks at past values to make predictions about future occurrences. It captures trends and patterns, including the cycles of solar activity. One challenge is tuning the parameters (p, d, q) to make sure the model fits the data correctly, but with some fine-tuning, I expect to generate reliable forecasts. While ARIMA does have limits, like being sensitive to outliers, it’s a strong tool for predicting sunspot activity when applied carefully.

A model like this could be useful for astrophysicists and anyone concerned with solar flares or other solar activity. It can help study how space weather affects technology, infrastructure, and human health. For example, the solar storm in May 2024 caused auroras, disrupted drones, delayed agricultural activities, and more (SIDAC, 2024).

I’ll be using R to build my model. R is a great tool for data manipulation, modeling, and visualization because it offers a lot of helpful libraries. Here’s the process I followed:

1. Collect the data

2. Explore and prepare the data

3. Look for correlations or patterns using graphs, plots, and matrices

4. Train the model

5. Evaluate its performance

6. Improve its performance

First, I downloaded and loaded the “forecast” and “tseries” packages, then loaded the data into an object named “sunspot\_data.” Next, I edited the column names because they weren’t labeled, which I figured would help later.

A screenshot of a computer program

Description automatically generated

A white background with blue text

Description automatically generated

I had to convert the data frame into a time series object for it to work with ARIMA. I used the sunspot number column, set the start date to the first recorded day, and used a frequency of 365 since sunspots were recorded daily. I then plotted the data to confirm that the time series was created successfully.



Plot generated:

A black and white image of a sound wave

Description automatically generated

Next, I checked if the data was stationary, which is important for ARIMA models to work well. I used three different methods: AutoCorrelation, Partial AutoCorrelation, and the Augmented Dickey-Fuller (ADF) test. The autocorrelation plot showed that the data wasn’t stationary because some values exceeded the blue dotted line. The partial autocorrelation plot gave a slightly better view, but it still suggested the data wasn’t stationary. Finally, the ADF test confirmed that the data became stationary when the p-value dropped below 0.05.

AutoCorrelation:



ACF Plot:

A graph with lines and numbers

Description automatically generated

Partial AutoCorrelation Function:



Partial AutoCorrelation plot:

A graph with lines and numbers

Description automatically generated

Augmented Dickey-Fuller test:

A computer code with blue text

Description automatically generated

Once I confirmed stationarity, I fitted the data to an ARIMA model. The auto.arima() function selected the best model for my data, which was ARIMA (5,1,2) because it had the lowest AIC value (630285.2). I then checked the residuals to confirm that the data stayed within the blue dotted lines, further validating the model.

A screenshot of a computer code

Description automatically generated



A graph of a graph

Description automatically generated with medium confidence



A graph with lines and numbers

Description automatically generated

With the best model selected, I used the forecast function to predict sunspot activity. I forecasted ten years ahead, using a 95% confidence level, to compare the forecast to the validation data provided. Unfortunately, the results weren’t ideal. While the prediction intervals expanded over the ten-year period, the actual sunspot predictions stayed stable and became increasingly unreliable as they extended into the future. This led me to think that there might be a better model for predicting sunspots.



The graph I generated shows the historical sunspot data in black and the forecasted values in blue. The gray-shaded area represents the forecast uncertainty. When I compared my forecast to sunspot validation data from the SIDAC website, I noticed that, from 2015 onward, the observed sunspots aligned well with the upper bound of the prediction interval.

A graph showing a graph of a wave

Description automatically generated with medium confidence

I tried improving my model by tweaking the ARIMA parameters and adding a seasonal component. However, since the auto.arima() function already selects the best model, these changes didn’t improve the forecast. Increasing the p-value led to overfitting, while lowering it didn’t make much difference in forecast accuracy.

In conclusion, while ARIMA is a good start for predicting sunspots, the model I built might not be suitable for organizations needing precise forecasts, given the range of unpredictability in my results. Still, it could be helpful in showing the historical patterns of sunspots and the correlation between past cycles and future predictions.





