Part 4: Time Series Forecasting Using ARIMA Modeling

Alexander Bonnet

Grand Canyon University

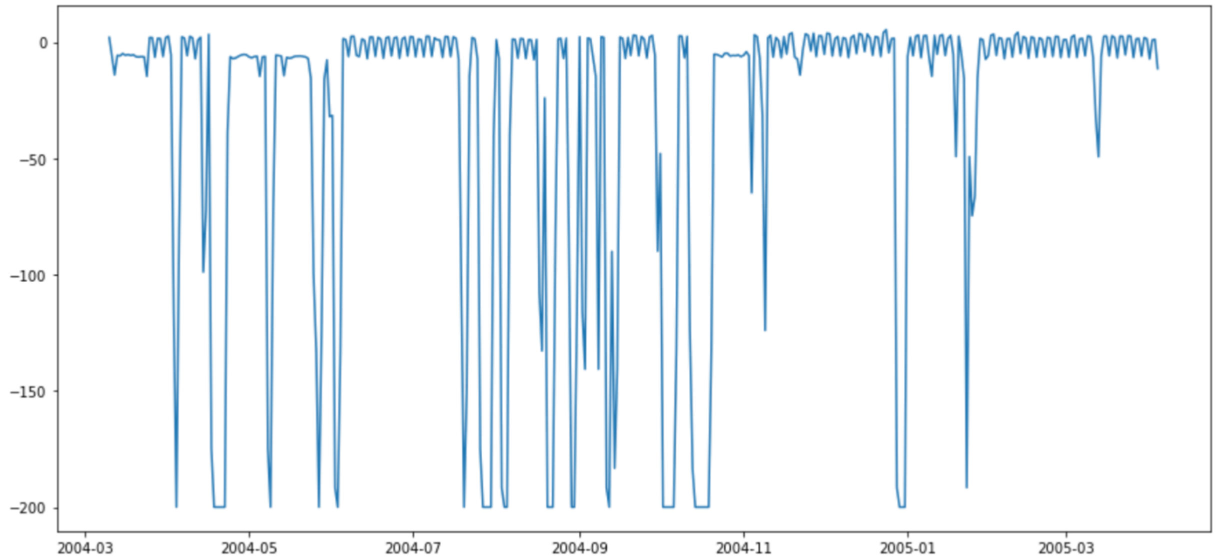
DSC - 530

Brian Stout

7/3/2024

1.

```
plt.figure(figsize = (15,7))
plt.plot(year, label = 'Average CO2 by Day')
plt.show()
```



There does not appear to be a need for differencing as there is no noticeable change in the overall average of CO(GT) over the span of the plot.

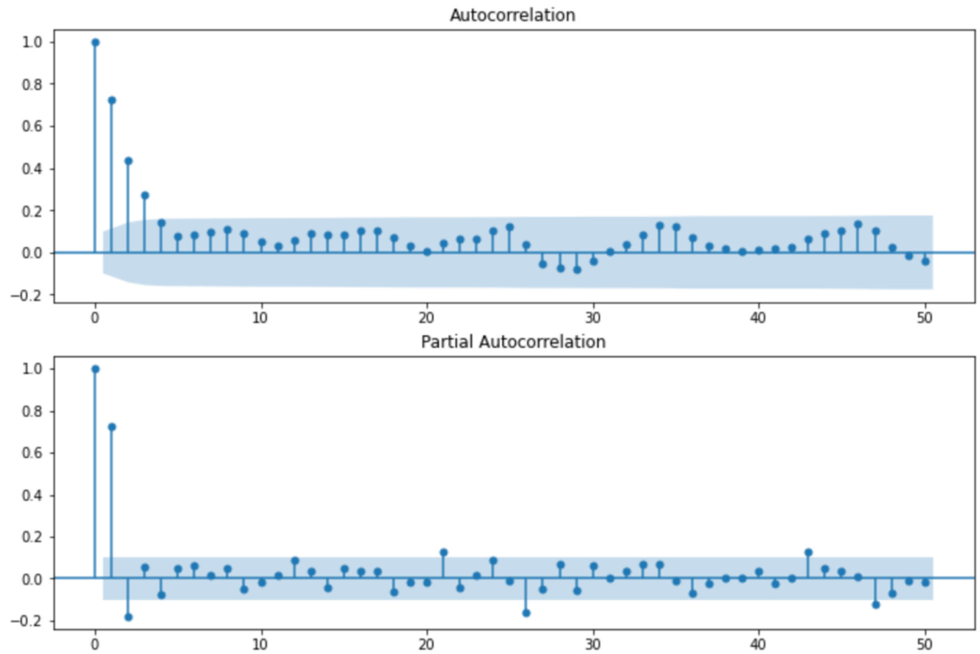
```
adf_test = adfuller(year['CO(GT)'])
# Output the results
print('ADF Statistic: %f' % adf_test[0])
print('p-value: %f' % adf_test[1])
```

```
ADF Statistic: -8.748258
p-value: 0.000000
```

We can further explore this using the adfuller test in python that statistically shows that the data is stationary as the p-value is very low.

2.

```
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
sm.graphics.tsa.plot_acf(year['CO(GT)'], lags = 50, ax=ax1)

ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(year['CO(GT)'],lags=50,ax=ax2)
```



For p we can look at the significant partial autocorrelation values to see how many previous values we need to look at to make a prediction on the observation. In this case we can see that the first 3 values can be deemed significant, the third value is very close so we can start with 2.

For d we look at the need for differencing. We showed that there is no need for differencing so this value will be 0.

For q we look at the autocorrelation values to see how far back we need to look to calculate the moving average. In this case there are 4 significant values that we can see. For the first model I will use 2 as they are the two most significant values, but in the follow up model I will experiment with using 3 or 4.

3.

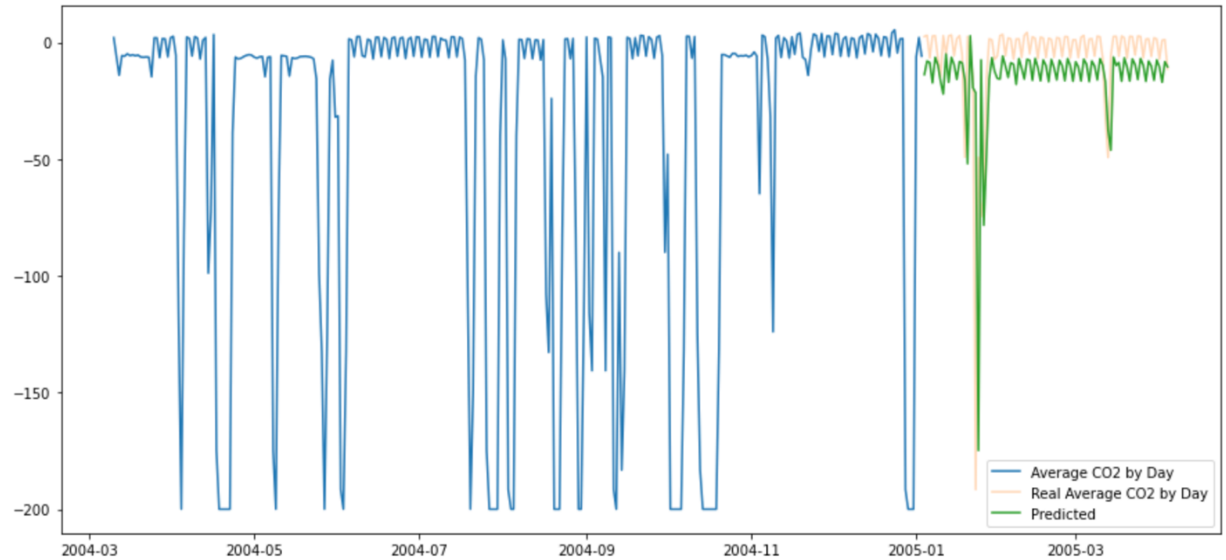
```
mod = ARIMA(year['CO(GT)'], order = (2,0,2))
fit = mod.fit()
print(fit.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                 CO(GT)   No. Observations:                  391
Model:                 ARIMA(2, 0, 2)   Log Likelihood               -2042.572
Date:                Wed, 03 Jul 2024   AIC                           4097.144
Time:                        19:04:04   BIC                           4120.957
Sample:                    03-10-2004   HQIC                          4106.583
                         - 04-04-2005
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const        -34.0107     14.572     -2.334      0.020     -62.571      -5.450
ar.L1         -0.3085      0.548     -0.563      0.573      -1.382       0.765
ar.L2          0.5399      0.305      1.768      0.077      -0.058       1.138
ma.L1          1.1786      0.551      2.138      0.033       0.098       2.259
ma.L2          0.2423      0.186      1.303      0.193      -0.122       0.607
sigma2      2014.1686    150.742     13.362      0.000    1718.720    2309.618
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):               580.66
Prob(Q):                              0.95   Prob(JB):                         0.00
Heteroskedasticity (H):               0.68   Skew:                            -1.45
Prob(H) (two-sided):                  0.03   Kurtosis:                         8.22
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

```
# Split the data into train and test
train_size = int(len(year) * 0.8)
train, test = year[0:train_size], year[train_size:len(year)]

# Fit the ARIMA model on the training dataset
model_train = ARIMA(train['CO(GT)'], order=(2, 0, 2))
model_train_fit = model_train.fit()
```

4. In the model we can see that the second autoregressive feature is more significant then the first with a p-value of .07 vs .573. This might lend to us wanting to add more autoregressive features.

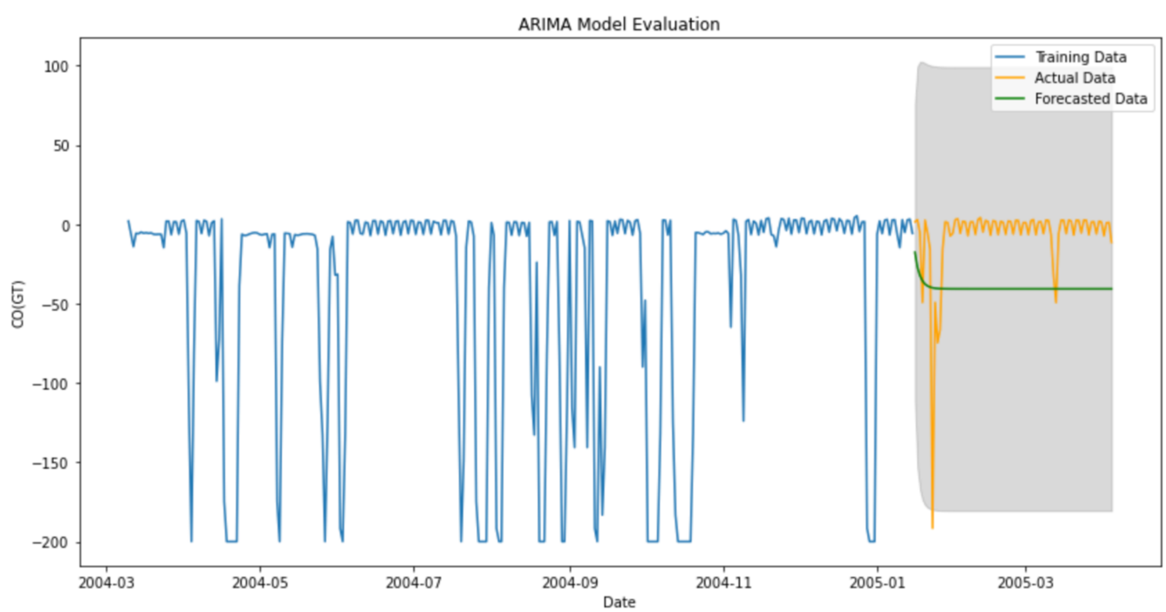
The first moving average factor is more significant than the second so we can utilize those.




```
MAE: 15.271274917815404
```

Looking at the MAE we can see that there on average the model is 15.27 points off the true value. In the grand scheme of the model that is not that large of a difference.

5.



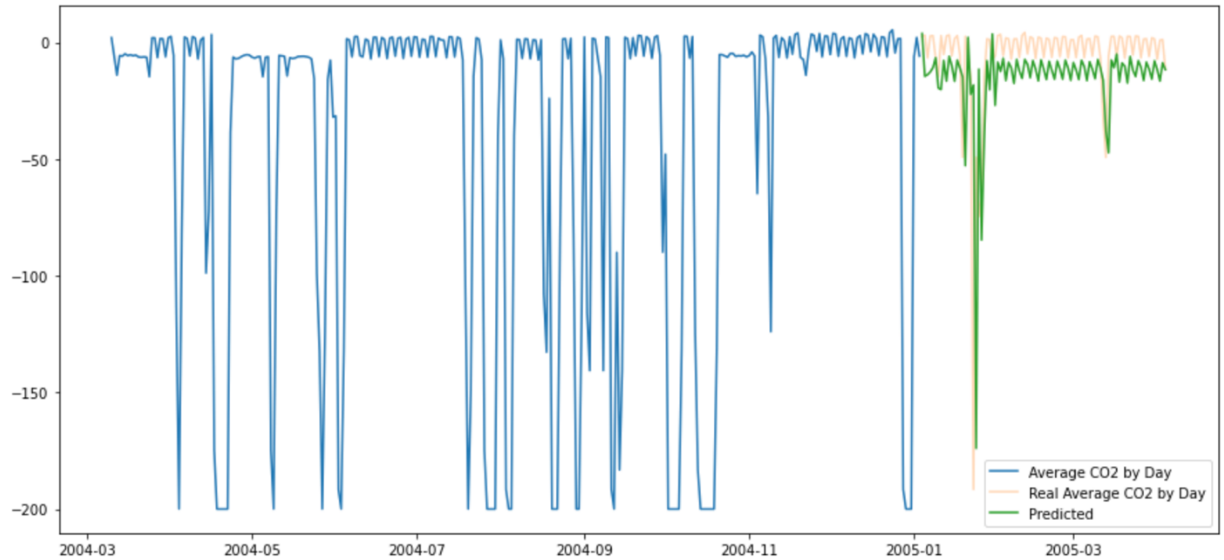

```
RMSE: 41.310347660980455
```

When predicting the model does not perform very well. This is most likely do to the lack of trends in the data. The model forecasts a decrease in the average, but the 95% confidence interval contains a large majority of the possible outcomes. The predictor has a RMSE of 41.3 so the root mean square error is about 41 points.

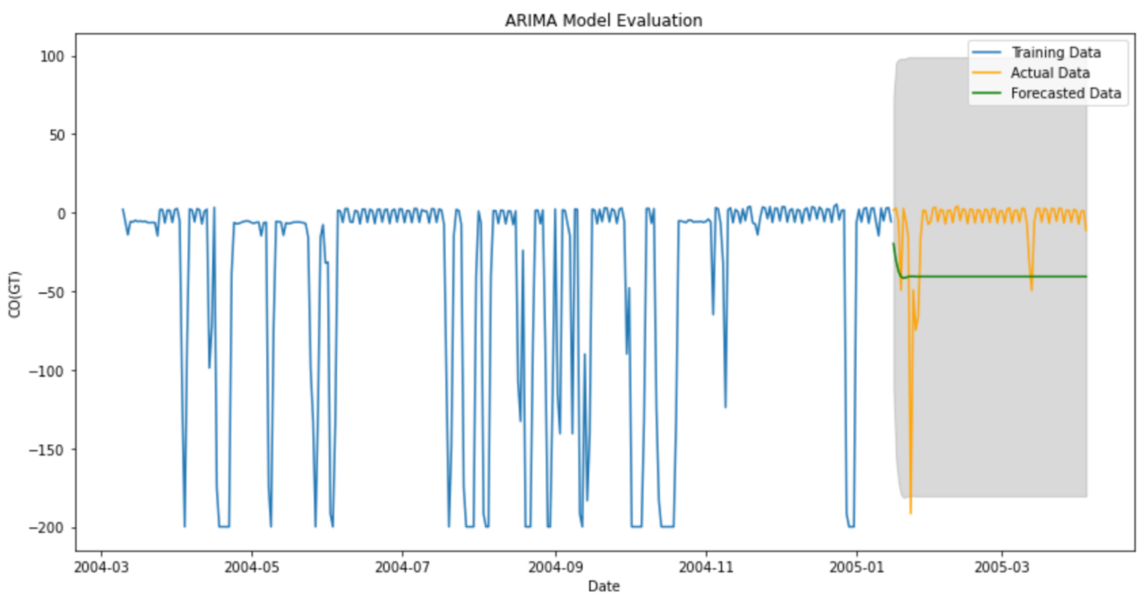
6.

```
mod = ARIMA(year['CO(GT)'], order = (2,0,4))
fit = mod.fit()
print(fit.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                 CO(GT)   No. Observations:                  391
Model:                 ARIMA(2, 0, 4)   Log Likelihood               -2040.848
Date:                Wed, 03 Jul 2024   AIC                           4097.695
Time:                        19:27:55   BIC                           4129.445
Sample:                    03-10-2004   HQIC                          4110.280
                         - 04-04-2005
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const        -34.0910     13.885     -2.455      0.014     -61.305      -6.877
ar.L1         -0.1952      0.305     -0.640      0.522      -0.793       0.403
ar.L2          0.2474      0.253      0.976      0.329      -0.249       0.744
ma.L1          1.0642      0.305      3.490      0.000       0.466       1.662
ma.L2          0.4311      0.203      2.120      0.034       0.033       0.830
ma.L3          0.2445      0.150      1.635      0.102      -0.049       0.538
ma.L4          0.1480      0.068      2.170      0.030       0.014       0.282
sigma2      1996.2661    144.407     13.824      0.000    1713.234    2279.298
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):               534.87
Prob(Q):                              0.97   Prob(JB):                         0.00
Heteroskedasticity (H):               0.68   Skew:                            -1.37
Prob(H) (two-sided):                  0.03   Kurtosis:                         8.04
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```



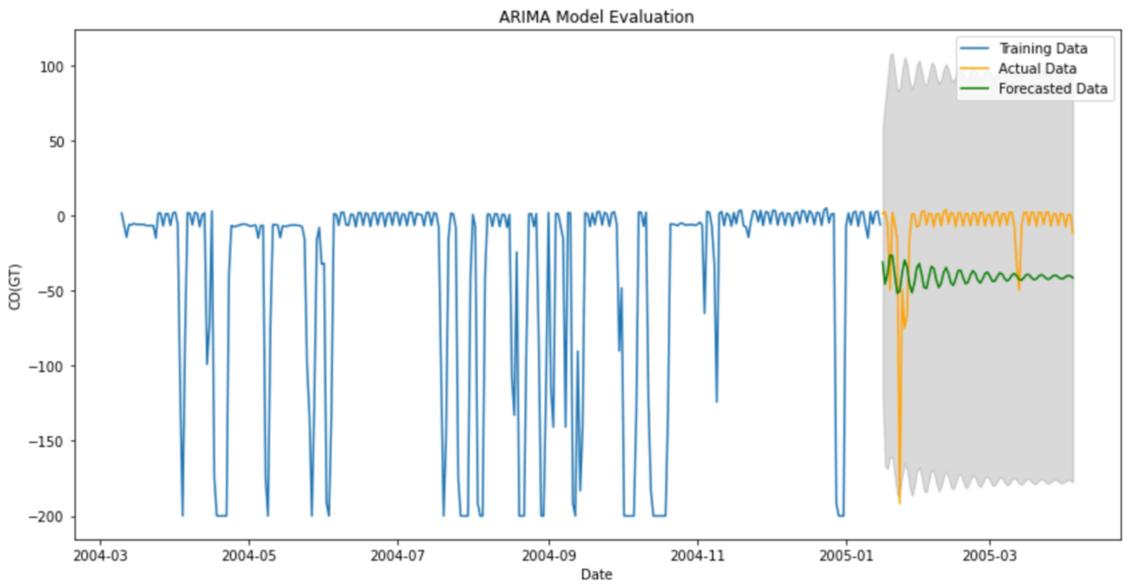

```
MAE: 15.750089920820542
```




```
RMSE: 41.4507992420051
```

By increasing q the model was only marginally changed. The MAE value increased slightly so it is performing slightly worse, and the RMSE also increased slightly. The increase in complexity of the model was not worth it as the model did not improve.

When increasing p to 3 the model fails to converge so the results are misleading, but the prediction does begin to match the pattern of the original data. The RMSE and MAE values once again increase from the original model.




```
RMSE: 41.70866896770278
```

As far as the most accurate model the metrics point towards the first model as being the most accurate overall. When selecting a model, I would select the first model, but none of them perform very well at all on the data. This data does not follow specific trends or seasonality, so this data is very hard to forecast. The data is also stationary across the period, so there is no major change to forecast at all. This all leads to poor performance and uncertainty in the model.

Reference

Bhatt, D. (2023, November 3). *Time series analysis and forecasting with Arima in python*. Medium. https://medium.com/datainc/time-series-analysis-and-forecasting-with-arima-in-python-aa22694b3aaa

Vito,Saverio. (2016). Air Quality. UCI Machine Learning Repository. https://doi.org/10.24432/C59K5F.