# Forecasting Using Time Series Analysis

Here, I use time series analysis to forecast the future value of the stock price of the firm I worked for for 8 years, EnerNOC, a global leader in energy management and demand response. I test for stationarity, select and AIRMA model, identify conditional heteroscedasticity, select and ARCH model, and interpret the results.

# Testing for Stationarity

First, I use Quandl, a tool that pulls publically available finical data from around the world, to get the list price of ENOC stock for everyday that it has been traded publically. I bring this data into python as a pandas data frame.

After renaming the dataframes columns and index, I take the log of the daily the closing value. This new column is called logENOC. I take a log because this ensures that level induced volatility does not interfere with the stationarity of my series. Logging the series prevents overall changes in volume from distorting the behavior of the stock.

Next, I take the difference of the closing values. This differencing is another way to get my time series to be stationary. I don’t care about the absolute value of the stock. That metric is not stationary (it in theory should go up forever!). Instead I am interested in forecasting the return on the stock. This is the day to day change in value. I want to forecast what the change in the stock will be tomorrow, not what the stock will be tomorrow.

One can see by visual inspection that the level closing price and the log of the level closing price are not mean reverting and are therefore not stationary. Stationarity is a necessary condition for reliable forecasting. Visually, the difference of the closing price, the stock returns, do look stationary.

When one plots the autocorrelation and partial autocorrelation function for each type of data, the log shows clear partial autocorrelation, while the differenced value does not.

The real statistical test of stationarity, the Dickey-Fuller test, yields a very small P-value for the differenced time series proving that it is indeed stationary.

Using Python’s stats model package, and various parameters of an ARIMA (Auto-Regressive-Integrated-Moving-Average) model are thrown at the data. The model with the lowest Akaike Information Criterion (AIC) a measure of how probable the model’s explanatory power, is selected as the most accurate. Here the AIRAM model used a p=2, d =1, q= 0.

The model is then used to predict values at each step of the series. The difference between these values and the actual time series is found and squared. This is the square error in the model.

The model is then used to forecast a number of steps ahead. This also produces a 95% confidence interval for this prediction.

This prediction at first seems reasonable. However, on closer inspection the squared residuals show autocorrelation!

# Autocorrelation Found in the Squares of the Residuals!

The residuals showed little to no autocorrelation or partial autocorrelation. This is expected. The squares of the residuals DID show autocorrelation and partial autocorrelation. This indicates that there is some structure remaining in the data. As the ARIMA model accounts for linear behavior, this structure can be thought of as behaving no-linearly as the residuals are the difference between the linear model and reality. The model does a bad job at predicting the behavior of the data at particular times. These particular times are structured in such a way that once the model is shocked into being significantly different than an actual value, it has a hard time recovering. This ‘hard time’ recovering is uncorrelated, but not independent, behavior.

This conditional heteroscedasticity justifies the selection of an ARCH model. A script was written that fit an ARCH model for q=0 to q=10. Again, the AIC criterion was used to select the best performing ARCH model which turned out to be a q=1 or ARCH(1). Additionally, a GARCH model was fit as well. The ARCH model’s lower AIC criterion made it a better selection than the GARCH model.

# Final Prediction

As expected the ARCH model accounts for the conditional heteroscedasticity. Therefore, the ARCH model is used to make the prediction. The resulting model is used to predict values and again compare them against the historically accurate time series. The difference between the time series and the predictions are the errors. Comparing the performance of the model against the actual performance over time yields a relatively accurate picture.

# Interpreting Results

Something VERY strange happened at the tail of my data sample. The stock collapse of late 2015 was totally missed by the model. The model does a pretty remarkable job, except at the end of the time series, matching the volatility associated with the great recession. Only when the volatility was most recent, at the worst moments of the stock collapse of 2015, was the model not able to keep up with reality. This is an example of knowing what was going to happen with 20, 20 hindsight. The tail of the model, form 2010 onwards, is so good at predicating behavior because the memory of the system is still fresh. We know this is an artificial circumstance of having all the data available at once. I am overall pretty impressed with the model’s moving window. It did a good job given the worse finical crisis of the last half century. Missing the volatility of the past year gives me pause however. The dynamics of the firm obviously changed, yet the weight of the model’s distant past prevented this change from being forecast. This came from shocks that were novel to the system. The regulatory uncertainty and perception of poor management did not effect the price of the stock until the collapse of 2015. These pieces of information ended up being more important than the information that drove the system over most of the life of the firm. This ARCH model did not appreciate that fact and therefore did not weight these critical developments in its prediction of stock performance.