ARIMA for the forecasting of Covid-19 metrics in Spain

An academic essay on regression

Eduardo Sánchez López
Escuela Superior de Informática
Universidad De Castilla La Mancha
Ciudad Real España
eduardo.sanchez00@outlook.es

Maira Torres Medina
Escuela Superior de Informática
Universidad De Castilla La Mancha
Ciudad Real España
maria.torres@alu.uclm.es

0 Introduction

This essay is the result of an academy study on different coronavirus metrics for all the regions in Spain. It covers an overview on ARIMA and the methodology behind the correct parametrization of it and a practical application with python autoarima module.

1 Objectives

The main objective is to predict the number of daily cases, daily deaths, daily recoveries, hospitalizations, and critical hospitalizations during the next seven days from a given day point. For the verification of the forecasts validity three error metrics are obtained, the error on the first and seventh day and the Mean Absolute Percentage Error (MAPE).

These predictions are done using an Autoregressive integrated moving average model. This model is autoregressive, does not depend on other features to fit or predict. A prediction is done for each feature in each Spanish region separately. ARIMA methodology and the way of working would be discussed in the following sections.

For the model explanation and for the better understanding of the study, the training would be done with data from the second of February of 2020 up to the thirtieth of April of the same year. The prediction would then be from the first seven days of May 2020 and it would be compared with the real data in that time range.

Finally, this model is designed and developed for a collaborative endeavour. Following the principles and techniques previously introduced, an automatic machine learning algorithm is presented. This algorithm performs all the predictions for the following 7 days starting the 15th of April until the 30th. This means that for each of those days, a prediction for all the selected features is done. Then, it saves that data into comma separated values files, one for each prediction. These files are then ready to be transferred to the assembly data processing system. The collaborative model takes all the predictions that were sent by different sources to make its own forecast. This model is not covered in the essay.

2 Considerations

The motive behind this study is academical, a project undertaken by the students on the *master's degree in Computer Science* of *Escuela Superior de Informática* in Ciudad Real, Spain. Thus, there are some limitations that needed to be imposed for all the students to stand on the same ground.

· Same data source for all the models. [1]

· Data cleaning should be kept to a minimum. The only needed changes are transform the total cases, recoveries, and deaths into daily data and add the 'PCR+' column data to their respective row in the column 'CASOS'. The 'TestAc+' column includes the data of antibodies tests, which is a less trustful test that can give false results. The 'TestAc+' were not included because it can cause that the same new infected person could appear duplicated in the total amount of cases. Also, many people do the two types of tests at the same time. This is because the antibodies test returns the result very fast while PCR needs some days to be processed. If one person that follows this behaviour is infected and correctly identified by both tests, the antibodies test result would appear on the next day in the official data and the PCR in three or four more days, causing a duplication that's hard to detect.

The conversion from total amount to daily increments is done by doing a subtraction for each row. Being **x** a row index in the training dataset, the formula is as follows.
Dataset[x][feature] = dataset[x][feature] – dataset[x – 1][feature]

For a better quality of the model there are some adjustments that could be performed that did not make into this project.

Weekend data comes with a lag. Usually every Monday has an increment in all the parameters in contrast with past data. This is because the data recollection is slower on weekends and adds part of the updates on Mondays or Tuesdays. This is especially obtrusive in ARIMA because it can be seen as a seasonality problem, which is not. The proposed solution is to estimate according to the data on the current week how many cases should be in Saturday and Sunday. If the gap between the expected and the obtained is bigger than a threshold previously stated, balance those days with cases reported on Mondays and Tuesdays.

Many regions changed the way that they report the data during the pandemic. Also, in some cases the data was initially wrong. This had to be reverted by correcting the data at some point in the timeseries, which caused artificial peaks. The solution for this problem is to analyse the veracity of the data region by region, since each one has different problems and a common solution is not possible.

3 Auto regressive integrated moving average

Auto regressive integrated moving average, ARIMA from now on, is a class of models that takes into consideration the past values of a timeseries, its own lags, and the lagged forecast errors to produce predictions.

This makes ARIMA an autoregressive model. It does not need external features that explain the characteristic that is going to get forecasted. It's only used for regression in timeseries problems, which means that the **x** value in the problem is always a measure of time (days, weeks, months…) and the **y** value is a continuous quantitative variable.

Time series forecasting is different from simple regression forwarding, there are some factors that must be considered before proceeding. Two of the most important characteristics of a timeseries, especially for ARIMA forecasting, are the **stationarity** and **seasonality.**

Stationarity and seasonality

A stationary series is one where the values of the series is not a function of time. [2] This means that the mean, variance, and autocorrelation are constant over time. The seasonality defines how influenced is the function by specific regular intervals of, usually, less than a year. It is usually related with **cyclic** behaviour, although the main difference is that a strong seasonality means that there are going to be variances in a **fixed** interval and cyclic in a **non-fixed** interval. As an example, sales of a local restaurant are seasonal during the year, because they are usually going to be higher on the summer months and in the fall. On the other hand, a videogame company could see their revenue increased each 5 months because it is the needed development time for a new update in one of their products. This increase in revenue could happen on January, June, and November, but next year would happen in different months. By definition, a stational timeseries cannot be seasonal.

In Figure 1 there is an example on how seasonality and stationarity affect the data. ARIMA works best when the stationarity and seasonal component is low, so the more the data looks like in the first graph, ARIMA would give a better result. How to deal with stationarity is discussed in the next section.

The other problematic component, seasonal, is less critical. Part of it is because of the existence of Seasonal auto regressive integrated moving average or simply SARIMA. In short words, is the ARIMA version for seasonal problems, which will not be discussed.

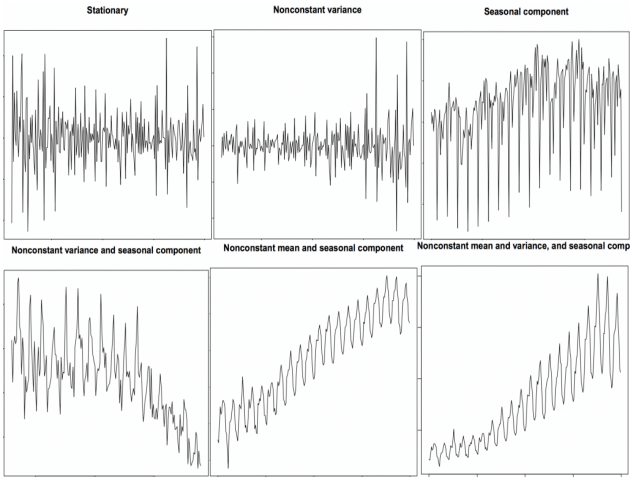

Figure 1 Stationarity and seasonality example [3]

The reason behind not considering SARIMA for this problem is the nature of the data. In only two and a half months of data recollection is hard to detect any type of seasonality. Some viruses are more prevalent in the cooler and humid seasons, while in the other seasons are not as prevalent. This could, in theory, be also true for the covid-19, [4] although at least 1 year of data recollection is needed.

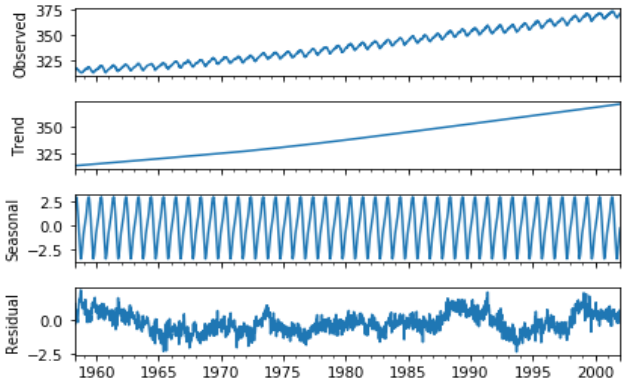

Figure 2 Seasonality decomposition [6]

Previously it was mentioned that seasonality could be introduced because of the lag in the weekends that is reflected in a higher number of casualties in Mondays and Tuesdays. This is because of faulty data, not because it is the real behaviour. The virus shouldn't discriminate by the day of the week, especially because the time range between the infection and the discovery, incubation period, of the virus in a patient is not always the same and can vary a lot. [5] This fact makes it difficult to perform investigations and conjectures about the possibility that some weekdays produce more contagions than others.

In Figure 2 an example of seasonality decomposition is shown. Note that the decrements on the real data doesn't always mean that there's also a decrement on the real data.

Like in the restaurant sales example, overall sales on the rest on the months that are not in summer or in the fall are going to be lower in comparation and thus looking that there's a decrement on the economic growth of the company. To make sure that that decrement is not a symptom of business degradation, the data should be compared with past years in the same seasons. A good indicator for the detection of seasonality are the **seasonal indices**.

In the Figure 3 the same trend and seasonality decomposition is shown for the covid-19 new daily cases in the region of Madrid. This data would be referenced throughout the document as reference for procedures.

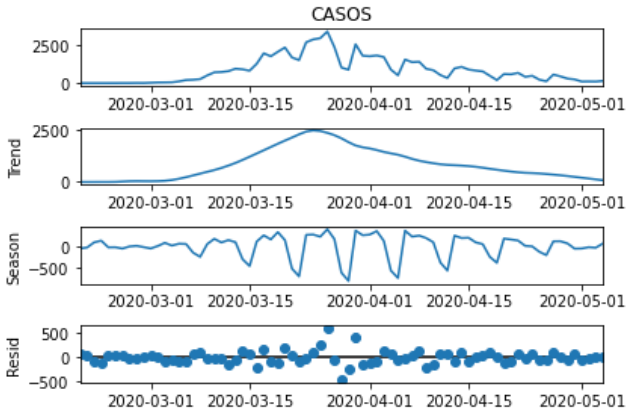

Figure 3 Decomposition on Madrid new cases

This plot shows graphically the "weekend effect" in the data recollection in Spain. There are seasonal periods with 7 days in which one or two of them present a local maximum (workdays) and other two present a local minimum (weekends). This seasonal effect is not as strong as the number of cases goes down, possibly because the workforce is not as oversaturated on the weekends to process and publish the data on time.

One solution for this fake seasonal component without having to use SARIMA is to use the trend for the fitness on the model. The residual indicates that the seasonality is correctly decomposed in most time periods. The only exception is the peak, which could be attributed to the high workload that makes even more difficult publishing the data on time. For the sake of working with the same data with the rest of the students, as previously mention, this solution is not implemented, and the study will carry on assuming that there's no seasonality.

Model construction

ARIMA models have three important components **p**, the order of the AR term, **q** the term of the MA term and **d** which is the number of differencing required to make the time series stationary. AR refers to a purely Auto Regressive mode, one which only depends on its own lags. MA on the other hand can be seen as the opposite, caring only about the lagged forecast errors. ARIMA formula is shown in Figure 4.

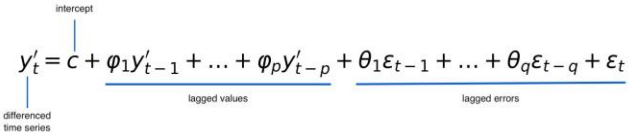

Figure 4 ARIMA formula [7]

The predicted y' value in a time position t is equal to the intercept plus the linear combination of up to p lags plus the linear combination of up to q lags. To find these two parameters, the first step is to find d, the order of differencing.

The number of times that one time series have to be differentiated is the closest value required in order to get a defined mean and for the auto-correlation function to reach 0 quickly. If two orders of differentiating give a similar result, the one that results with the least deviation in the differenced series is usually the more adjusted.

The Augmented Dickey Fuller test, ADF from now on, is a statistical test which states in its null hypothesis that the timeseries is non-stationary. With a significance level of 0.05, if the p-value is less than it, the null hypothesis is rejected, and the time series is considered stationary. This test is useful to avoid hand-by-hand calculations and for the automatization of the fit of an ARIMA function. The p-value for the daily new cases in Madrid area without doing differentiation is 0.1593. With only one differentiation the p-value goes below 0.05. The null hypothesis was rejected and thus, d is 1.

The AR term, p, can be calculated by looking at the partial autocorrelation graph of the resulting dataset after differentiating d times the original. This graph is shown in Figure 5.

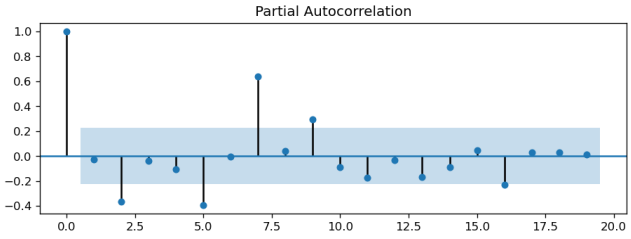

Figure 5 First differentiating partial autocorrelation

The PACF lag 1 is almost zero for this data, which means that it is a very good candidate for the p value. Lag number 2 is substantially worse and number 3 is almost as close to zero as 1, so they are discarded and the final value of p is 1. Having a value of 1 means that the model is going to get forecasted using the values of the previous day.

Finally, getting the MA value, q, is very similar as the process to obtain p using an autocorrelation plot instead of a partial variant.

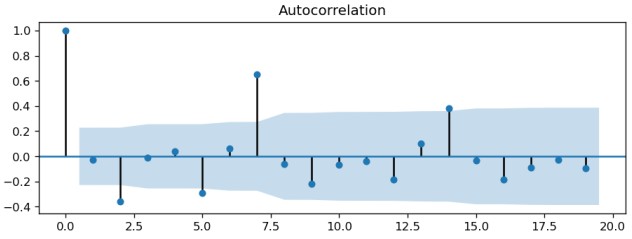

Figure 6 First differentiating autocorrelation

The best value for q is again 1. Now that all the values are found the Figure 7 shows a prediction of the number of cases compared with the observations from the 1st of May up to the 7th.

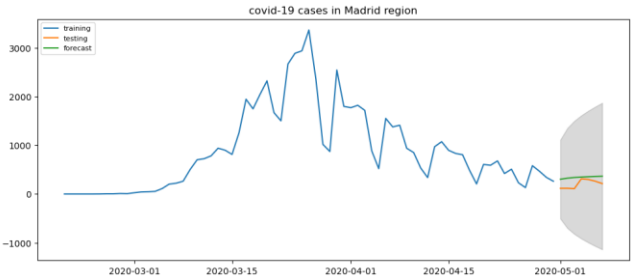

Figure 7 covid-19 cases in Madrid region

The error of the forecast versus the observations in day 1 of the prediction is 186 cases and in the 7th day 152. The day with least error is the fourth of may (fourth day into the prediction as well) with 36 cases of difference. The forecast is pessimistic and not optimal. This is expected with a high seasonal function as previously stated.

Autoarima

Autoarima is a popular module written for many languages, including Python and R. It automatizes all the process explained in this document to obtain automatic forecasts. One of the tasks for this project is to obtain a prediction for the next seven days for each of the days from the fifteen to the thirtieth of April for each region for each feature. This task would take a high amount of time done by hand.

Presented in the following graphs are the new daily cases, recoveries, death and hospitalized from the 1st to 7th of May in Madrid. The number of critical hospitalized is not shown because is almost the same as the regular ones. These are a subset of the total forecasts that needed to be done for the collaborative predictor.

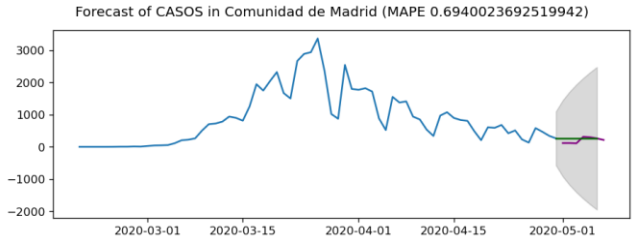

Figure 8 Daily cases forecast

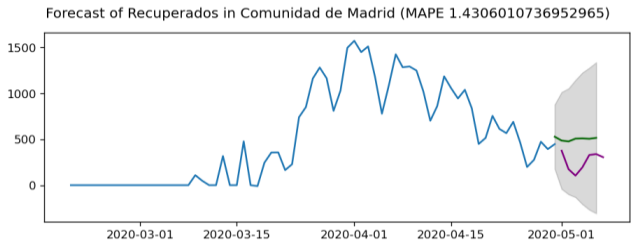

Figure 9 Daily recoveries forecast

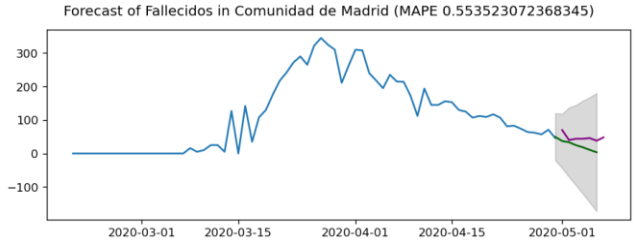

Figure 10 Daily death forecast

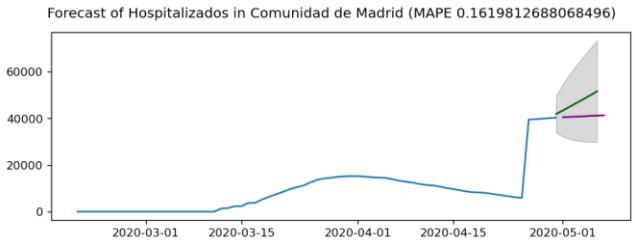

Figure 11 Daily hospitalizations forecast

As shown, the trend is captured decently. The number of hospitalizations is data that need to be heavily transformed to get forecasting.

Conclusions

ARIMA is a useful model that can be useful for this type of problem, where seasonality is not a factor. There are multiple improvements that could be made to upgrade the quality of the forecasts, mostly on the data cleaning stage.

There is also a lot of value to perform a time series analysis after a year of data recollection. This could help understand if the virus is seasonal, like the flu, and how to combat it with better and more concise forecasts.

ANNEXES

REFERENCES

[1] https://covid19.isciii.es/resources/serie_historica_acumulados.csv
[2] https://www.machinelearningplus.com/time-series/time-series-analysis-python/
[3] https://www.machinelearningplus.com/wp-content/uploads/2019/02/stationary-and-non-stationary-time-series-865x569.png
[4] https://www.webmd.com/lung/coronavirus-heat
[5] https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200402-sitrep-73-covid-19.pdf?sfvrsn=5ae25bc7_4#:~:text=The%20incubation%20period%20for%20COVID,occur%20before%20symptom%20onset.
[6] https://github.com/jrmontag/STLDecompose/blob/master/STL-usage-example.ipynb
[7] https://towardsdatascience.com/time-series-forecasting-in-real-life-budget-forecasting-with-arima-d5ec57e634cb