Semester Paper in Machine Learning

Liudmila Kolesnikova
Department of Computer Science and Mathematics
University of Applied Sciences
Munich, Germany
kolesnikova.hm@gmail.com

I. Introduction

This document is our attempt to analyse the development and the current state of the Covid-19 pandemic and make predictions. Throughout the analysis it is assumed that the data contains true numbers. No unregistered cases are taken into account.

II. Data Visualization

For initial inspection of the history and the current state of the Covid-19 pandemic two metrics are chosen: numbers of the infected adjusted to population figures and mortality rate. Two countries were selected for comparison purposes due to their unique approaches to handling the pandemic: China as the country that managed to "flatten the curve" and the US as the country where the numbers of confirmed cases skyrocketed in the middle of the year and keep rapidly growing.

Although the numbers of confirmed as well as infected cases keep growing, the mortality rate plot (Fig.1) shows that all three countries, after a short period of rapidly growing mortality rates (one-two months), have managed to bring the pandemic under control. The plot exhibits a steady decline of the mortality rate since May, 2020.

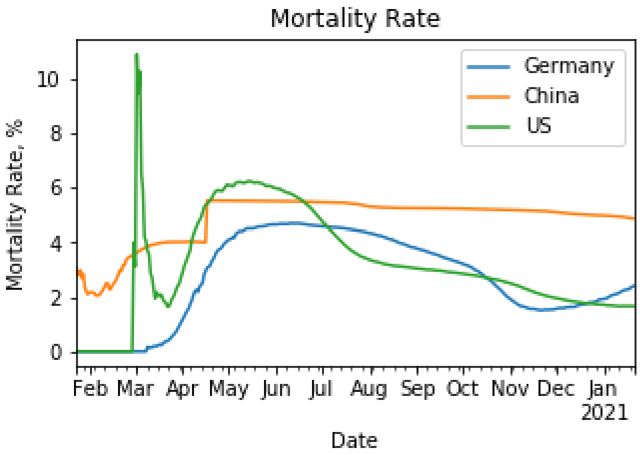

Fig. 1. Mortality Rate from Covid-19 in Germany, China and the US in 2020.

III. Clustering

As preparation for clustering certain data had to be added to the dataframe (e.g. country population). We also got rid of unnecessary entries, for instance Covid-data on board of cruise ships.

We performed clustering of time series according to the relative numbers of the infected [1]. As clustering algorithm *KMeans* was chosen due to its simplicity. The initial dataset has 366 features, which means conducting PCA is necessary to be able to visualize the clustering results. First two PCA-components explain 0.95 of the variance. As metrics for the number of clusters serve Inertia and Silhouette Score [2]. Both indicate that 3 clusters deliver best results (Fig.2).

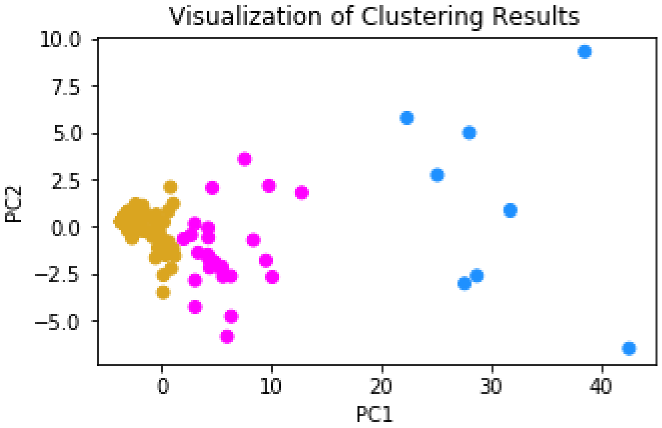

Fig. 2. Three Clusters identified by Kmeans.

As a result 188 countries were clustered according to the shapes of their curves and the percentage of the infected. An average representative of each cluster was calculated and plotted along with real time series of representative countries (Fig.3).

IV. Modelling

A. Linear Regression with basis-change (or Polynomial Regression)

First attempt to analyze the data was using linear regression with train-test-split appropriate for time series (train and test splits are consecutive segments of the data). The results yielding huge errors on test-sets turned out to be inappropriate for the complex data that we are trying to predict.

Second attempt was using apprippriate for time series train-test and cross-validation splits (*TimeSeriesSplit* in sklearn). These models delivered uninterpretable results and a negative R-squared score.

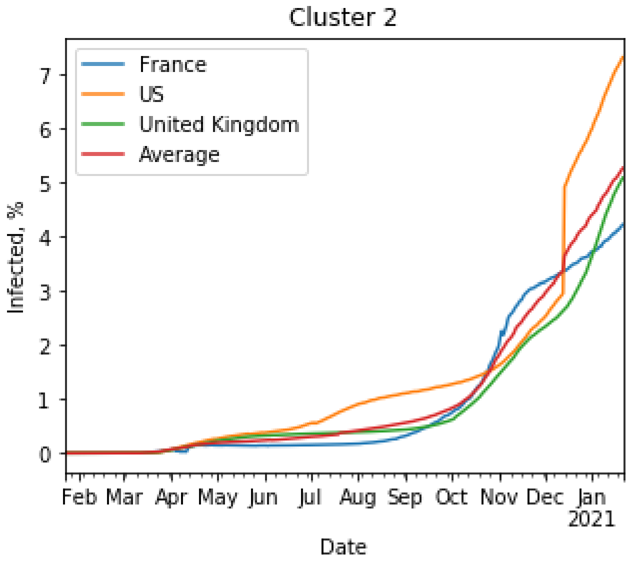

Fig. 3. Cluster 2 representing countries with exponential growth in the relative numbers of the infected. It is also the cluster with the biggest percentage of the infected ranging from 5 to 7 percent of the population.

As the final version serves a model fitted with degree 8 polynomial for Germany and degree 6 polynomial for China and random train-test and cross-validation splits. As goodness-of-fit measures both MSE and R-squared are used. Cross-validation scores are plotted to visually inspect which model yields best results. According to both models the number of confirmed covid-cases will keep growing. For illustration purposes the polynomial fit for China is chosen as having the most expressive score- (Fig.4) and fit-plots (Fig.5).

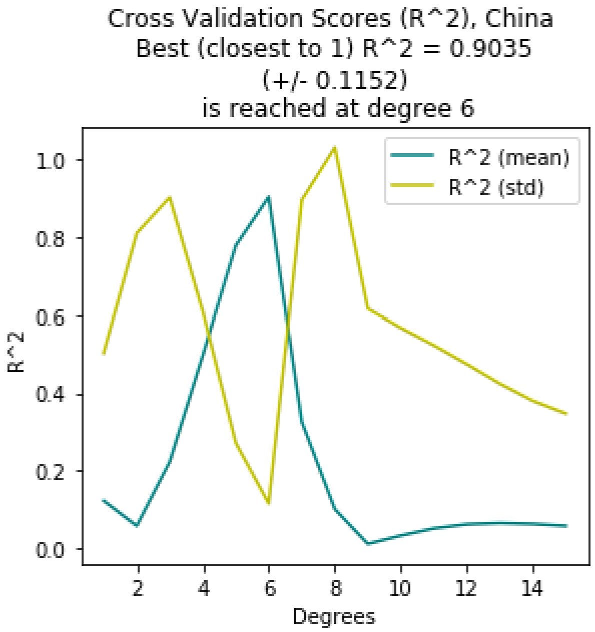

Fig. 4. Cross-validation R-squared scores.

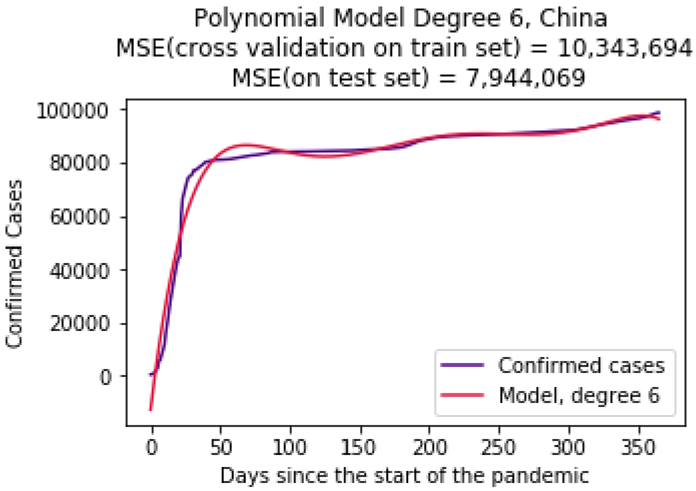

Fig. 5. Confirmed cases data fitted with degree 6 polynomial.

B. Autoregression

Optimal lag is closen with the help of the Autocorrelation plot, which in the case of Germany states that 26 days before the predicted day preserve significant amount of correlation, necessary for the prediction. For the modelling the library called *statsmodels* is used. The model-parameter *dynamic* determines whether the real or predicted values are used as an input for the model.

If real time series values are used as an input (*dynamic=False*), predicted values closely follow the real values up until the point when the model reaches the test-set. On the test set we don't have an option to use real values anymore. Since our model has never seen the train set, the only option it has is to use predicted values every time, meaning that prediction starts to deviate significantly compared to the real train-set.

However this deviation is miniscule compared to the divergence that the model experiences using predicted values from the very beginning (*dynamic=True*). If we use predicted values for forecasting, with each prediction the model accumulates error. By the time the model reaches the test set, it has already accumulated a significant error which results in predicted values having huge MSEs. To illustrate this behaviour we measured MSEs on test-sets for Germany and China. In case of China the MSE amounts to 57,894,399 (real values as input) and 2,198,106,445 (predicted values as input). The results are consistent for Germany [3].

REFERENCES

[1] Aleszu Bajak, "How to use hierarchical cluster analysis on time series data", December 13, 2019. (https://www.storybench.org/how-to-use-hierarchical-cluster-analysis-on-time-series-data/)

[2] Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn and TensorFlow," O'Reilly Media, 2017. (https://github.com/ageron/handson-ml)

[3] Jason Brownlee, "Autoregression Models for Time Series Forecasting With Python", August 15, 2020. (https://machinelearningmastery.com/autoregression-models-time-series-forecasting-python/)