POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Homework report

Homework 3

Scientific computing tools for advanced mathematical modelling

Authors: Lorenzo Galvan, Randeep Singh, Daniela Zanotti

Academic year: 2021-2022

---

1. Mathematical formulation of the problem

The COVID 19 pandemic required continuous monitoring by authorities due to several changes in the scenario that occurred. Predictive models of contagion evolution may be pivotal to making weighted decisions regarding actions to be implemented.
The goal of this homework is set up an automatic tool that accesses, updates and organizes the COVID-19 epidemiological data of Italy (on a regional basis) and implement a strategy to update daily the prediction of the trend for the following days.
The prediction involves 4 categories:

- new daily infections, amount of positive cases registered that day;
- hospitalized, total amount of hospitalized patients until that day;
- recovered, amount of people recovered until that day;
- deceased, amount of people deceased until that day.

We are required to predict the values of the four chosen features in 7 days from the date of gathered data:

$$y_{i,t+7} = RNN(Data, t)$$

where $y_{i,t+7}$, $i = 1, .., 4$, is the feature at time $t+7$ (identified by the days) obtained through a Recursive Neural Network using the data available up to day t.

2. Methods

2.1. Data organization

In order to make our prediction we need data about the current trend of infections, which are available for all the Italian regions at *this link*. From the raw data we store only the selected features and create a `.csv` file in which we can easily identify the region where data were collected day by day. We also decide to keep data only from a certain date on, since at the beginning of the pandemic trends, habits and restrictions were very different from the recent ones. To correctly identify the current trend we consider only data from May 2021 on, month in which a relevant number of people were already vaccinated.
Our initial data are reported in Figures 1, 2, 3.

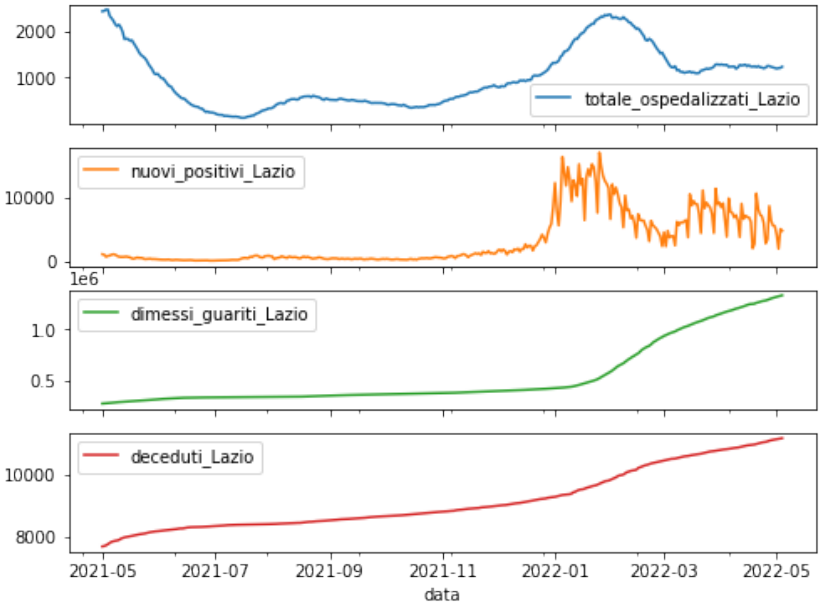

Figure 1: Initial data Lazio

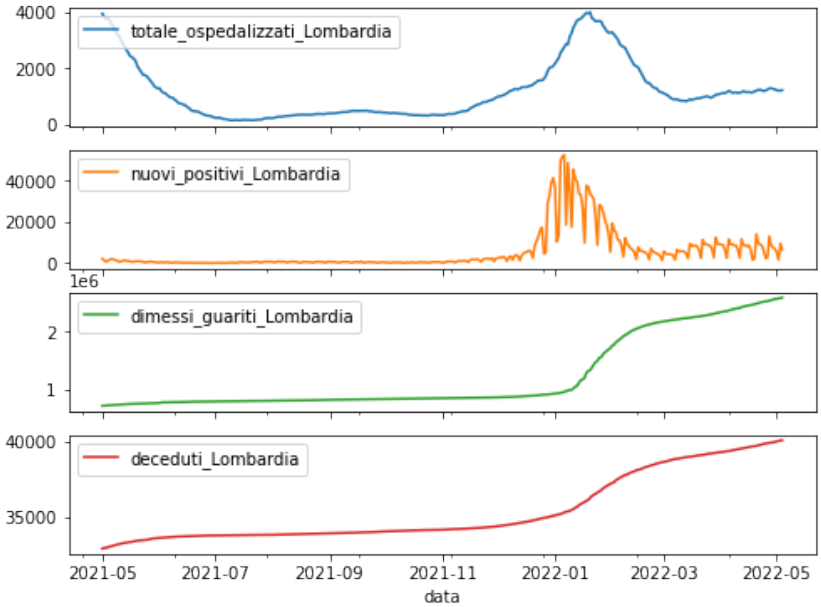

Figure 2: Initial data Lombardia

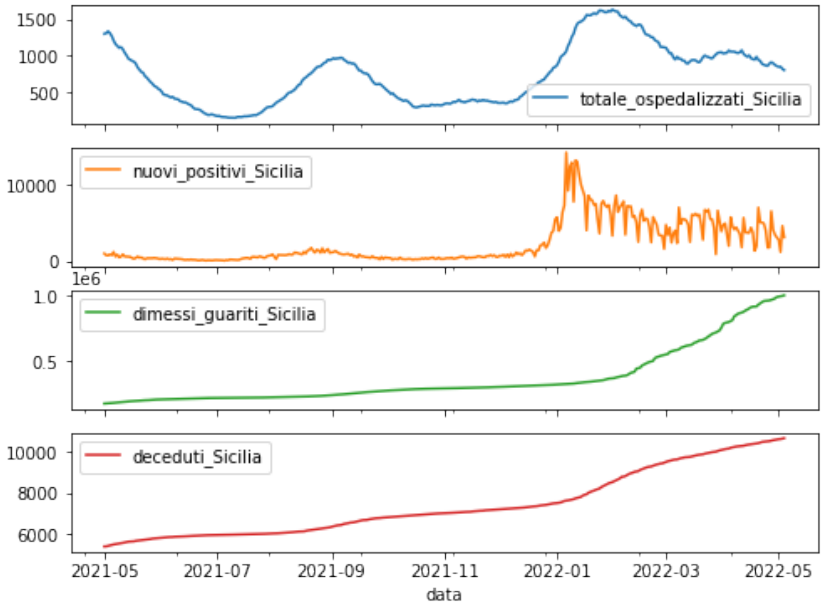

Figure 3: Initial data Sicilia

2.2. Preprocessing

At first, we pre-process the data doing two main steps:

1. we apply smoothing on data using Savitzky–Golay filter since there are many oscillations, especially in *new positive*, attributable to the less testing on Sundays and festivities. Moreover, this filtering process cleans data from noise, making all the workflow more robust;
2. to train the network we don't use raw data values but we work with percentage changes of the features between two consecutive days. This brings two advantages: the first is that transformed data are continuous and live in an interval around 0; the second is that in this way it is easier for the network to learn the trend rather than the absolute values (e.g., passing from 100 to 110 will be the same to passing from 1000 to 1100 infected, since the percentage change is the same).

As last thing before starting to model the neural network, we divide the data into training and test set, considering as test the last 7 days.
Below (Figures 4, 5, 6) are reported the plots of the training sets.

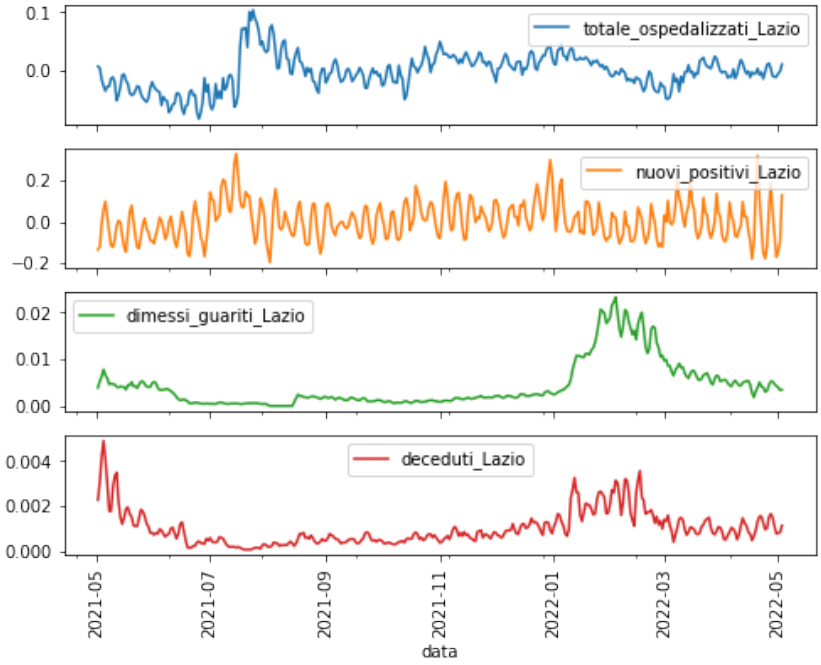

Figure 4: Training set Lazio

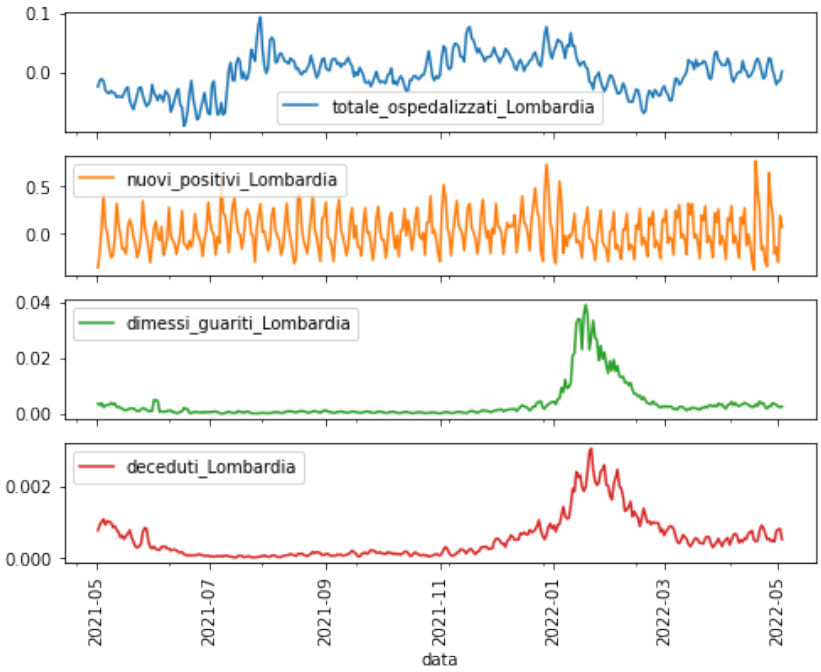

Figure 5: Training set Lombardia

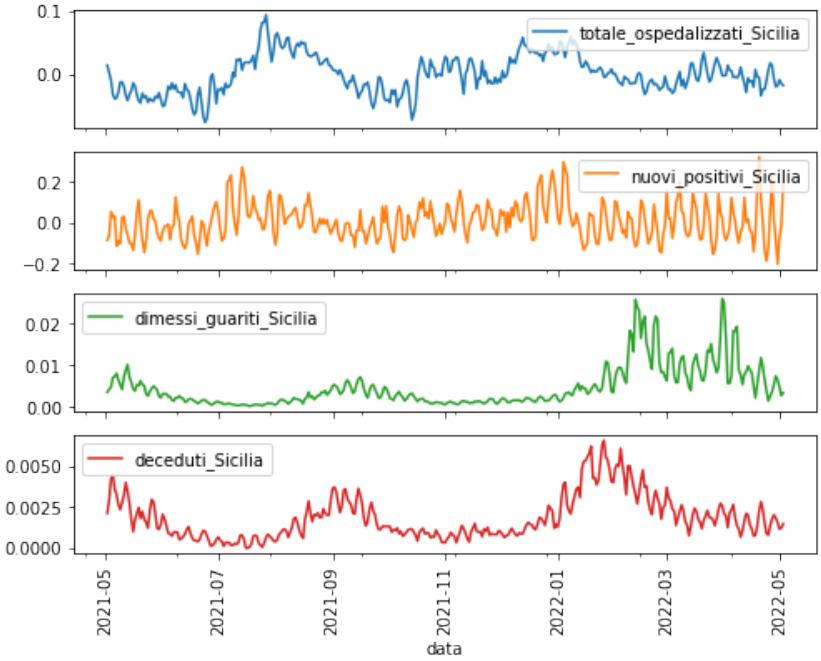

Figure 6: Training set Sicilia

2.3. Recurrent Neural Network - LSTM

Since we are dealing with data series, we decide to use Recurrent Neural Networks, in order to carry information over different time steps rather than keeping all the inputs independent of each other. However RNNs suffer of vanishing/exploding gradients due to the continuous matrix multiplications during the back-propagation process. As a consequence to these issues, RNNs are unable to work with long sequences and hold on to long-term dependencies, making them suffer from "short-term memory".
Consequently, we decide to rely on Long Short-Term Memory networks (LSTMs), a particular variant of RNNs which is are more complex, so that the learning is improved, but it requires more computational resources.

Each LSTM is composed of:

- **hidden state and new inputs**: hidden state from a previous timestep h_{t-1} and the input at a current timestep x_t are combined before passing copies of it through various gates;
- **forget gate**: this gate controls what information should be forgotten. Since the sigmoid function ranges between 0 and 1, it sets which values in the cell state should be discarded, remembered, or partially remembered;
- **input gate**: it helps to identify important elements that need to be added to the cell state.

The LSTM cell works like this:

1. the previous cell state c_{t-1} gets multiplied by the results of the forget gate and we add new information from the input gate multiplied by the cell state candidate to get the latest cell state c_t;
2. the hidden state is updated: the latest cell state c_t is passed through the tanh activation function and multiplied by the results of the output gate;
3. the latest cell state c_t and the hidden state h_t go back into the recurrent unit, and the process repeats at timestep t+1;
4. the loop continues until we reach the end of the sequence.

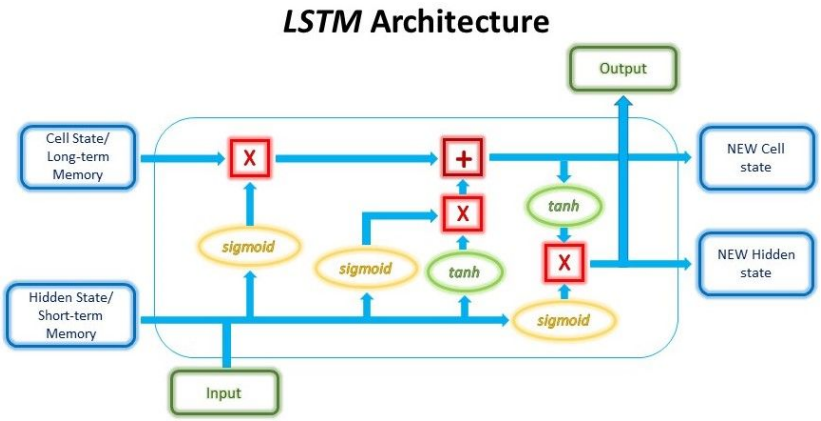

Figure 7: LSTM

2.4. The model

Looking at the plots of the data we can see that the regions are very different from one another, so we define different models (one for each region), all with the same network structure but with different model parameters and the hyper-parameters.
The structure of the models is the following:

- an LSTM layer, with regularizer L1L2
- a Dropout layer, with parameter = 0.1
- a Dense layer with 4 neurons
- a Reshape layer

The model uses the past 28 days to make a prediction in 7 days. However, it cannot just be applied once for the day we are interested in to get the desired result, because we obtain percentage changes and so we need to go back to the original scale. In order to do that the final predictions are obtained by predicting the values for each one of the 7 days after the last datum and then making the inverse of the percentage change.
In this way we can more accurately predict what will happen in 7 days' time, based on what is predicted in the in-between days.

3. Numerical results

Once defined the model, we can start with predictions and simulations about the future values of the selected features, based on the available information obtained after the preprocessing phase.
All the predictions are performed with the before mentioned structure in section 2.4.

As an example, we report the data prediction for the three selected regions made for 10/05/2022, with data available only up to 7 days before. Our prediction and the values founded are reported in the tables below, respectively in Table 1 and 2.
From the tables, we can see that the model actually predicts pretty well for all the three regions, with what we find to be a small prediction error.

Table 1: Predictions

	new daily infections	hospitalized	recovered	deceased
Lazio	4423.656487	1239.254169	1373091.879	11248.55932
Lombardia	9176.208612	1218.240761	2661991.656	40329.90421
Sicilia	4712.198525	813.0720985	1029766.1	10786.97705

Table 2: True values

	new daily infections	hospitalized	recovered	deceased
Lazio	4864	974	1362679	11201
Lombardia	9481	1122	2635044	40186
Sicilia	3763	740	1030621	10722

The errors we obtain are reported in table 3 and 4.

Table 3: Absolute prediction errors

	new daily infections	hospitalized	recovered	deceased
Lazio	440	-265	-10413	-48
Lombardia	305	-96	-26948	-144
Sicilia	-949	-73	855	-65

Table 4: Percentage prediction errors

	new daily infections	hospitalized	recovered	deceased
Lazio	9.05%	−27.20%	−0.76%	−0.42%
Lombardia	3.22%	−8.55%	−1.02%	−0.35%
Sicilia	−25.22%	−9.86%	0.08%	−0.60%

4. Conclusions

Making predictions on Covid-19 data is challenging: from the beginning of the pandemic a lot of several trends have taken place for many different reasons, such as the different variants of the virus or the various kind of restriction adopted. Dealing with these predictions using neural networks is even more challenging. In fact, a neural network needs a lot of data to work properly, but in our situation, even considering all the data at our disposal, we only have around 800 days of measurements.
Moreover, we could not use all of this data, since during the first period of the pandemic testing and measurements were not as accurate and structured as in more recent times.
Taking this into account for building our model, we tried to balance the need of data with the desire of considering only the last and more meaningful trend. This approach seems to work since the predictions are quite good, in particular the ones of recovered and deceased patients.
The main drawback of this model is that it is dependent on the trend used to build it: if we try to apply it on data gathered after the spread of a new variant or after a change of the restrictions, we will probably obtain results not as close to reality.

References

[1] Dipartimento della Protezione Civile Presidenza del Consiglio dei Ministri. Repository covid-19. `https://github.com/pcm-dpc/COVID-19`.

- Our code repository: Repository
- Explanation and example of LSTM models: Towards Data Science