Repo rate curve - machine learning

I – Introduction

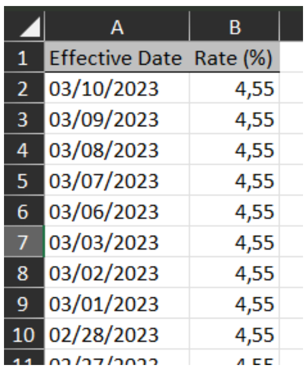
In this paper, we are focusing on one metric which is repo rate, here SOFR. Repo rates have an impact on daily economic activities and for those several goals, please refer to 5 - Repo rate curve – building paper. Being able to give a reliable forecasting of repo rates is crucial in decision making in many businesses. There are difficult to estimate but lately, with the development of machine learning, we can use deep learning models for prediction purpose.

II – Overview and data

This paper aims to present an analysis and results about regressors for repo rates, embedding polynomial and neural network models, to be able to give a reliable forecasting of time-series using long-term repo rates as input. To accomplish our analysis, we will be following some steps, first we'll highlight data, then we'll focus on the simple polynomial regression, and we'll dig into neural network regression details. A final chapter will be dedicated to model discussions, warnings and next research axis.

One of pain point of the whole analysis is repo rates data we can rely on. As mentioned in 5 - Repo rate curve – building paper, looking at market without already published secured overnight rate force us to build a repo curve/repo rate through most liquid instruments. For this paper, we'll then rely on US market, with SOFR rate, as by definition repo rates are already available. This choice/assumption will be discussed in final chapter, commonly with term repo rate linked topic.

Mentioning that, used data are SOFR historical rates starting 2nd of April 2018 until today (10th of March 2023):



	A	B
1	Effective Date	Rate (%)
2	03/10/2023	4,55
3	03/09/2023	4,55
4	03/08/2023	4,55
5	03/07/2023	4,55
6	03/06/2023	4,55
7	03/03/2023	4,55
8	03/02/2023	4,55
9	03/01/2023	4,55
10	02/28/2023	4,55
11	02/27/2023	4,55

III – Polynomial regression

Past repo rates highlight a curve which is clearly not linear. Hence, we introduce polynomial regression to overcome this problem, which helps identify the curvilinear relationship between independent and SOFR variable.

One important note about independent variable here: SOFR rate is a time-series, then by construction each repo rate is linked to one date, which is not, strictly speaking, a numerical variable. To bypass this "issue", we'll simply encode dates from 0 to full number of days.

Saying that, the polynomial regression model is:

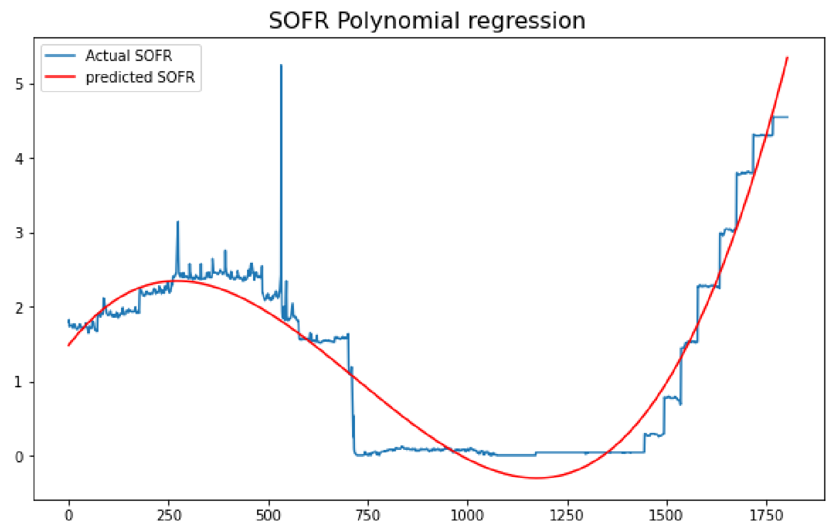
$$\begin{cases} SOFR_i = P^n(x_i) + \varepsilon_i \\ P^n(x) = \sum_{k=0}^{n} \beta_k x^k \end{cases}$$

$$\Rightarrow \begin{pmatrix} SOFR_1 \\ \vdots \\ SOFR_M \end{pmatrix} = \begin{pmatrix} 1 & \cdots & x_1^n \\ \vdots & \ddots & \vdots \\ 1 & \cdots & x_M^n \end{pmatrix} \cdot \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_M \end{pmatrix}$$

$$\Rightarrow \boldsymbol{SOFR} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$$

$$\Rightarrow \boldsymbol{\hat{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{SOFR}$$

The python library scikit learn enables us to highlight this kind of polynomial regression. Due to actual SOFR rate historical curve shape, a polynomial degree superior or equal to 3 is a proper, in first approach, choice, and we set this parameter at 3. Here is the result:



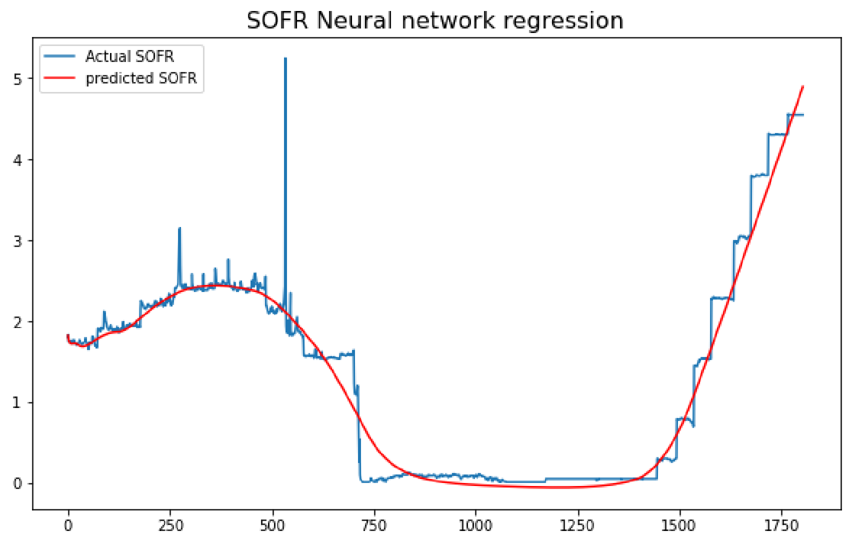
And the mean squared error is almost equal to 0,12.

IV – Neural network regression

For the second regression model, we chose the deep learning approach, through neural network scheme. Keras library is the perfect technical tool for that purpose and following steps had been followed:

- Neural network structure is divided into sequential and fully connected dense layers:
    - A first one which captured the single SOFR input for each $t_i \to x_i$, with activation function considered as linear.
    - Two nested layers with exponential linear unit activation functions.
    - A final one with linear activation function.
- The optimizer is adam, which embeds gradient descent algorithm for minimization.

Graphical results are:



And the mean squared error is almost equal to 3, which is quite disappointing compared to more “classical” polynomial regression, which has a far better score then. It doesn’t automatically mean the NN regression method is worst as prediction is the most important, not the past fitting.

V – Model assumptions, limitations, and next research axis

- User should be very careful about SOFR forecasting using those kind of regressors : a simple glance at predicted curve will lead us to foresee, as of t, very high repo rates in coming days/weeks, due to very steep curves. It’s of course totally meaningless for medium term SOFR, and even for close SOFR forecast, number of days should be limited.

- In addition, quite obviously, the regression process should be performed daily, in order to capture the last SOFR, which can heavily impact predicted curve if this last one is very different than previous one (jumps).
- The forecast should be back-tested then, comparing predicted futures repo rates and futures actual repo rates, it’s another part of the analysis.
- As this kind of regression should also be done on every market which highlights different repo rates (here the regression is only done on US market with SOFR), in order to have a perhaps more accurate overview and in order to handle possible performance issue, perhaps the range of historical rates should be reduced (basically, even with the current analysis, the curvature linked to far past data is quite useless, only the close uprising of SOFR is really conclusive).
- The analysis was done with daily SOFR, but once again it should be generalized to other markets, with, as pre-requisite, the capture of historical repo rates through market objects (5 - Repo rate curve – building).
- Analysis should be done using another regression methods, to catch the “better”.
- The analysis should also be extended, this time, to term repo curve, which leads to a more complex regression: for each past date, not only a single quantity like SOFR, but a full repo curve.