Practical assignment: SME Borrowing History

Yu Chen – yu.chen2@liverpool.ac.uk

DUE: Thursday, November 16, 2023

---

1 Introduction

This is an accompanying report, along with the Jupyter Notebook file, aimed at explaining the procedures and final thoughts for the assignment. While some of the content can be found in the Notebook in the Markdown comments, this report serves as a clean read without being "disturbed" by codes. The committee may find this report easier to follow.

2 Library code to be re-used

Throughout this analysis, I meant to write `library codes` that can serve as `recipes`, for instance, to solve a certain data problem, create a model, generate a figure, etc. Most importantly, these library codes can be re-used by others (*e.g.* team members or clients) later on, as in being adopted in `user code`. Please refer to Section: Library code in the Notebook for the library functions created for this analysis. Notably, these library codes can be easily packaged into a Python package, facilitating the sharing and usage by others.

3 Tasks

3.1 Data exploration, cleaning, pre-processing

The raw data needs some cleaning due to the existence of errors implicit in the manual entry process. I have written specific Python functions that deal with each certain data problem respectively. These functions are clear to its purpose and can be re-used even with a larger data set.

Most importantly, these functions can be chaining together using Pandas's **chaining** mechanism. As such, a one-stop function can be created, by chaining, to process the raw data and then save into the '/data/processed' directory (see the GitHub repository). Later on, modellers can directly import this processed clean dataset from disk, saving the trouble for processing data each time.

Particularly, in response to data problems such as duplicates, missing data, incorrect dates, the corresponding functions are written:

1. `inspect_date_format()`
2. `inspect_missing_data()`
3. `inspect_duplicates()`

3.2 Data visualisation

Four types of visualisation (see Fig. 1 - Fig. 4) are created and implemented by the functions below. They include the three required figures, as shown below.

1. `boxplot_annual_averge()`
2. `plot_monthly_debt_history()`
3. `plot_total_debt_history()`
4. `plot_which_lender_history()`

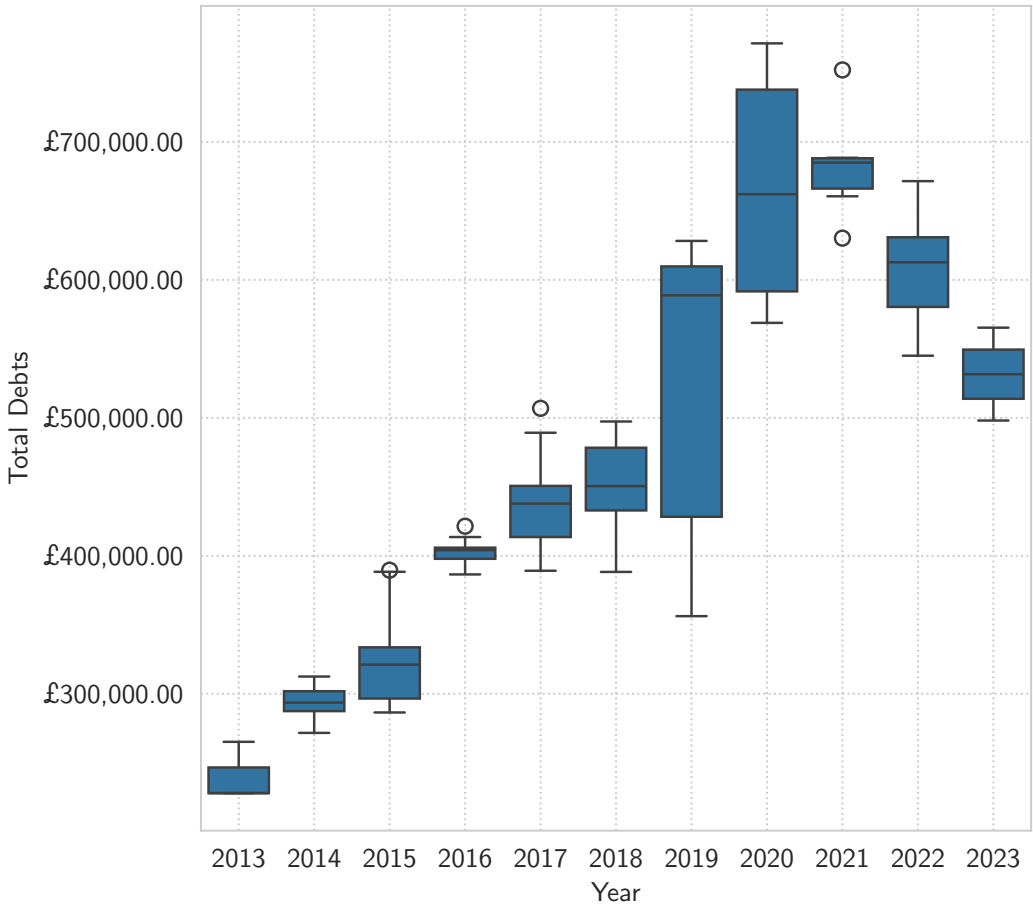

Figure 1: Average debts over years

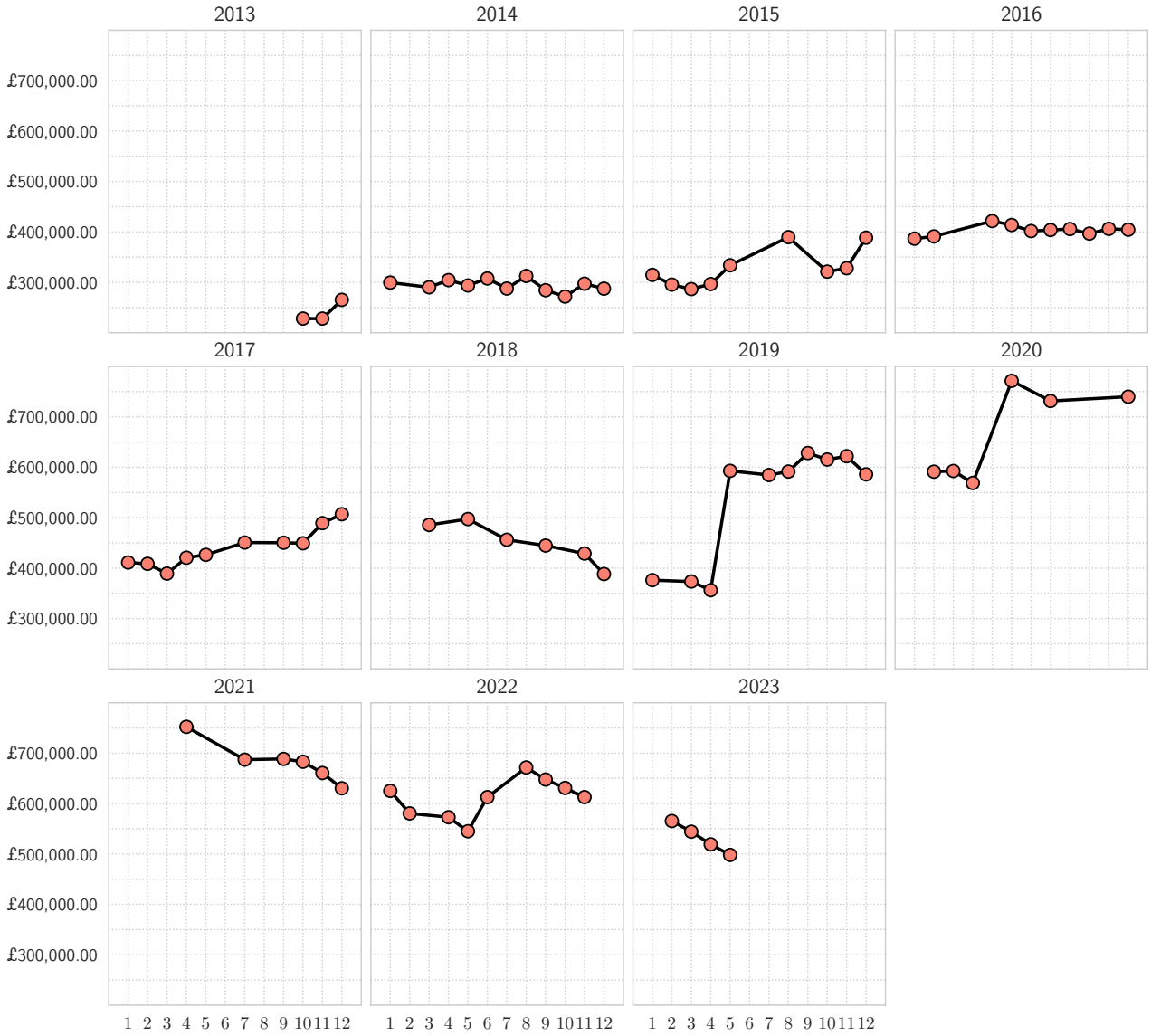

Figure 2: The total debts from all lenders over months and years

3.3 Predictive modelling

Given the borrowing history data, I would create uncertainty-aware Machine Learning models for forecasting future trends and imputing the missing values in the history. One key component of this analysis is **uncertainty modelling**.

3.3.1 Uncertainty decomposition in predictive modelling

Specifically, both *aleatoric* uncertainty and *epistemic* uncertainty will be accounted for in this analysis. Typically, in a regression setting, we are interested in inferring a function/hyposis, parameterised by ω, that are likely to have generated the observed data $\mathcal{D} : (\mathbf{x}_i, y_i)_{i=1}^N$ of N observations.

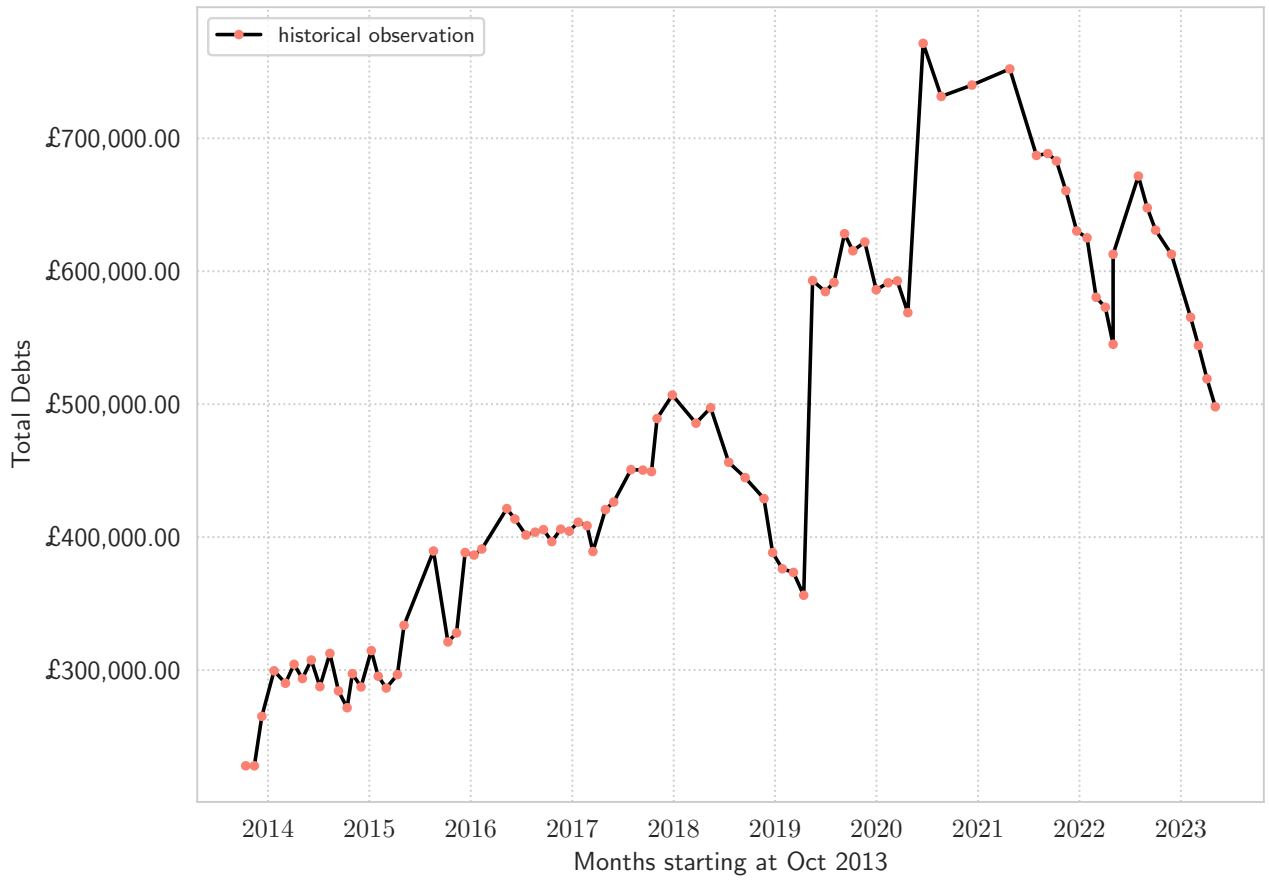

Figure 3: Total monthly debts from all lenders starting from October 2013

$$y_i = f_{\boldsymbol{\omega}}(\mathbf{x}_i) + \epsilon_i \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^K$ and ϵ suggests the noise term. A typical assumption of Gaussian noise enables the model fitting via an approach of maximum likelihood estimation. This gives rise to a probabilistic interpretation of the data generating process. The likelihood, under the model, of seeing the data with known σ^2 is written as:

$$p(y_i|\boldsymbol{\omega}, \mathbf{x}_i, \sigma^2) = \mathcal{N}(f_{\boldsymbol{\omega}}(\mathbf{x}_i), \sigma^2) \tag{2}$$

Optimising with the likelihood as the loss function, shown below, gives the inferred model, which formulates the conditional distribution $p(y^*|\mathbf{x}^*, \boldsymbol{\omega})$ given an unseen $\mathbf{x}^*$.

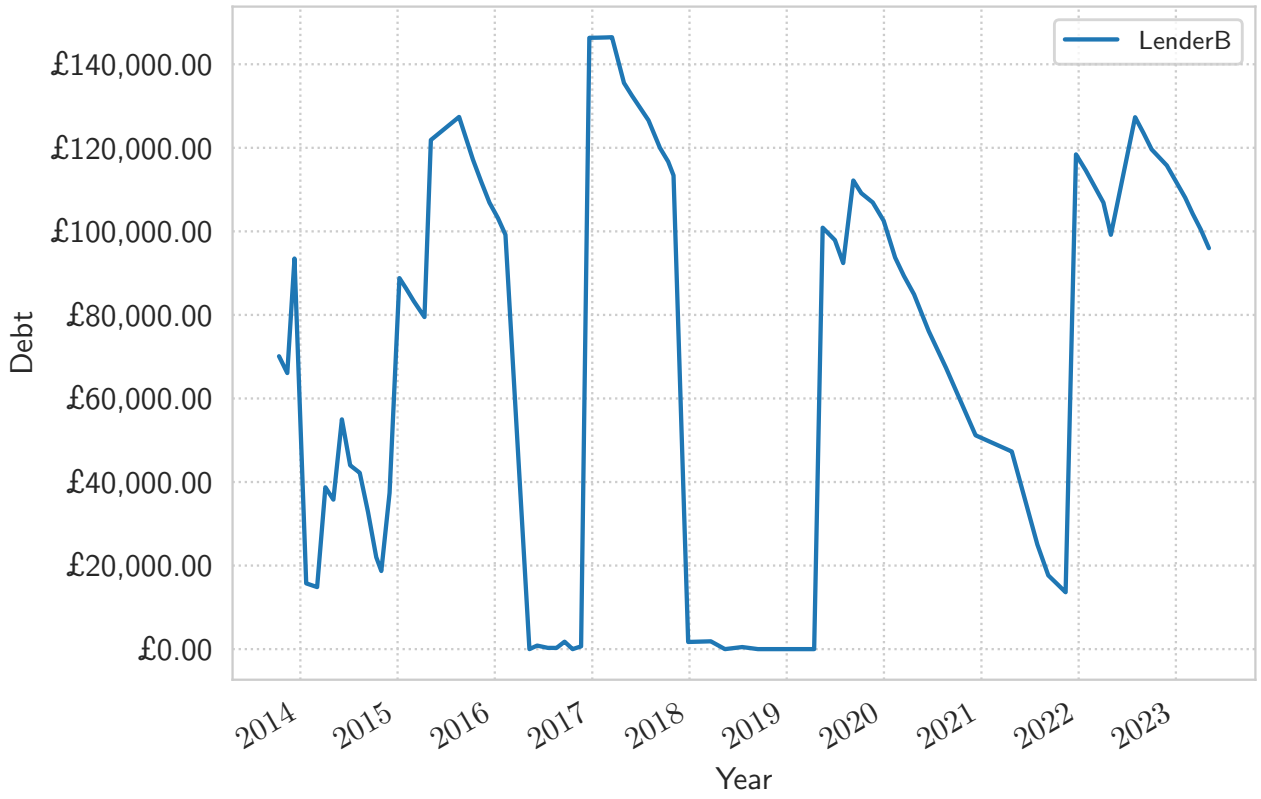

Figure 4: Debt history a certain lender

$$
\begin{aligned}
\boldsymbol{\omega}^{MLE} &= \arg\max_{\boldsymbol{\omega}} p(\mathcal{D}|\boldsymbol{\omega}) \\
&= \arg\max_{\boldsymbol{\omega}} \log p(\mathcal{D}|\boldsymbol{\omega}) \\
&= \arg\max_{\boldsymbol{\omega}} \log \Big[\prod_i p(y_i|\mathbf{x}_i, \boldsymbol{\omega})\Big] \\
&= \arg\max_{\boldsymbol{\omega}} \sum_i \log p(y_i|\mathbf{x}_i, \boldsymbol{\omega})
\end{aligned} \tag{3}
$$

However, implicit in the above MLE procedure is the ignorance of model uncertainties (*epistemic uncertainty*). The absence of enough data has restricted Machine Learning models from effectively learning the true relationship between features and labels (i.e. the underlying data generating process). Significant uncertainties exist on the model configurations that may have explained such limited data. Consequently, such uncertainties further compromise the generalization power of learned models in that predictions from uncertain/unrepresentative models can still be unreliable and over confident, especially when doing extrapolation on unseen situations.

3.3.2 Gaussian process modelling

With the importance of uncertainty quantification in mind, given a very small data set subject to missing data [1], I will use Gaussian process for three reasons: (i) it accounts for the model uncertainty; (ii) it is efficient to compute on a small data set with analytic solution; (iii) it naturally deals with missing data;

Consider a generic regression setting as shown in Eq. (1). In order to account for model uncertainty, we infer a distribution over functions $p(f|\mathcal{D})$ given the data $\mathcal{D} : (\mathbf{X}, \mathbf{y})$ in a compact notation. Note that I dropped the parametric representation of the function f for a generalised notation. We are then interested in estimating the posterior distribution after seeing the data.

Gaussian Process (GP) represents a distribution of functions by, practically, an infinite dimensional multivariate Gaussian distribution. It defines a data generating process as:

$$f \sim \mathcal{GP}(m(\mathbf{x}), \mathcal{K}(\mathbf{x}, \mathbf{x}')) \tag{4}$$
$$y_i \sim \mathcal{N}(f(\mathbf{x}_i), \sigma_y^2) \tag{5}$$

where $\mathcal{K}(\mathbf{x}, \mathbf{x}')$ denotes the covariance function (also known as kernel) and implicitly we assume Gaussian likelihood $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$. As noted, the sampled function from GP is impossible to compute as it requires evaluation at an infinite number of points. Instead, one considers finite samples from a multivariate Gaussian. The joint distribution of the observed data and the latent noise-free function values on the test points can be written as:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \mathbf{K}_y & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{pmatrix}\right) \tag{6}$$

One of the perks with GP is that, within the Gaussian world, the posterior predictive density, given unseen test points $\mathbf{x}_*$, can be analytically written as:

$$f|\mathbf{x}_*, \mathbf{X}, \mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \tag{7}$$

where

$$\boldsymbol{\mu}_* = \mathbf{K}_*^T \mathbf{K}_y^{-1} \mathbf{y} \tag{8}$$
$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}_y^{-1} \mathbf{K}_* \tag{9}$$

The next challenge is then to infer the hyperparameters $\boldsymbol{\theta}$ of the designated kernel, which will be done by maximizing the marginal likelihood $y(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$. Similarly, within a Gaussian world, such marginal likelihood can also be written analytically:

$$\arg\max_{\boldsymbol{\theta}} \ \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) \tag{10}$$

[1] It's been found that in the Jupyter Notebook that 33 months of records are missing from 2013 Oct - 2023 May and currently the number of data points is N=84.

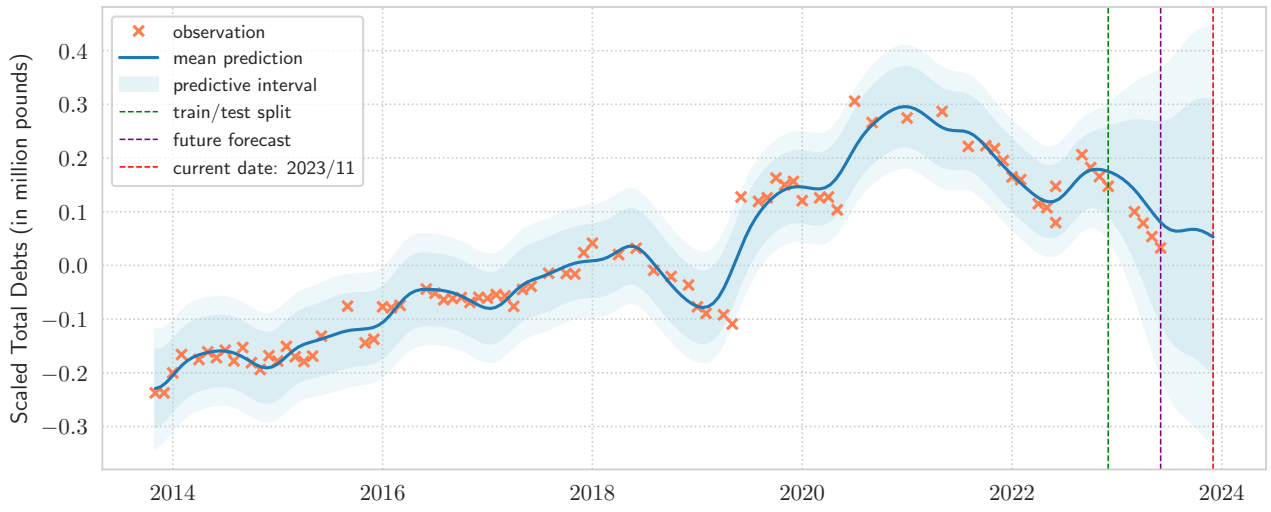

Figure 5: Gaussian process modelling on the total debt. The left of the vertical green dash line denotes the training data. Observation between the green line and purple line suggests the test data, which is dated until 2023 May. Starting from the purple to the red line suggests the future forecasting, where we don't have ground truth, until the current date: Nov 2023. Shared area suggests the uncertainty in the forecasts. Darker area corresponds to a 95% confidence interval while lighter area corresponds to a 99% confidence interval.

3.3.3 modelling settings

train/test split For a supervised task, people are mostly concerned with the model's generalisation power (*i.e.* predictive performance on unseen data). Such ability is typically evaluated, approximately, using testing data based on a train/validation/testing split. Cross validation is a common way to approximate such expected loss. However, it is in debate if it is appropriate for time series analysis to adopt cross validation as it jeopardize the temporal structure. It is more reasonable to adopt backtesting, which is time consuming per se.

Particularly, in this analysis, for demonstrating purposes, I will simply train the model on the training split and quantitatively evaluate its performance on the testing split to illustrate the procedures.

time series modelling Typically a full-fledged time series forecasting analysis using Machine Learning/Deep Learning involves temporal effects and covariates (both static or dynamic). Please refer to the discussion in Section 4.1 for what can be done in a real-world setting. Note that the existence of missing data will make the autoregressive modelling scheme challenging. But it can be solved by additionally add an imputation step to fill in missing values first before creating a time-series forecasting model, please refer to Section 4.2 for future possibilities.

Currently, in this analysis, I will create a GP model that predicts the total debt as a function of time. The current setting is deemed static as it ignores the impact of past debt values.

Table 1: Performance evaluation of the GP model on test data

Mean absolute error	Interval width	Log-likelihood	Prediction Interval Coverage Probability
0.0410	0.2608	1.6377	100%

3.3.4 results

Figure 5 shows the Gaussian process modelling on the total debt. The left of the vertical green dash line denotes the training data. Observation between the green line and purple line suggests the test data, which is dated until 2023 May. Starting from the purple to the red line suggests the future forecasting, where we don't have ground truth, until the current date: Nov 2023. These forecasts had been done with access to only data as of 2023 May and can now be further validated at current moment to signal the generalisation power of the trained model. This actually reflects the concept of backtesting. Most importantly, we have given the predictive interval to prepare for uncertainty-informed decision making of the company, such as better managing the debts with risk in mind.

Quantitatively, I have tabulated the performance quantitatively through a few metrics in Table 1 This allows the quantitative comparison between different modelling techniques. For example, a full-fledged time series model (such as Transformer) in future studies.

missing data imputation With the GP model, the values at the 33 missing months can be easily imputed, along with a predictive interval. The exact values of past missing months may not directly be interested. However, a complete past record enables more advanced models considering temporal effects, such as Transformer. These additional incorporation of temporal effects are key characteristics of state-of-the-art modelling techniques and can be significantly informative.

4 Final thoughts and future possibilities

4.1 Full-fledged modelling incorporating temporal effects and covariates

Typically, the state-of-the-art time series modelling techniques will involve temporal effects and covariates. Consider the quantity-of-interest (y) as the target to be forecasted, an autoregressive scheme is commonly used to formulate the effects of past values on the future value.

$$y_t = f(\mathbf{x}_t; \boldsymbol{\omega}) + \epsilon, \text{with } \mathbf{x}_t = [y_{t-1}, \ldots, y_{t-p}] \tag{11}$$

This reflects an univariate situation. More commonly, there exist covariates (static or dynamic) that provide extra features to the forecast, i.e. $y_t = f(\mathbf{x}_t, \mathbf{z}; \boldsymbol{\omega})$ where $\mathbf{z}$ denotes covariates. Dynamic covariates may include other related variables, or temporal effects such as seasonal periodicity (yearly, monthly, etc). Static covariates may include calendar events, holidays that may affect the the y_t considerably within a short time period. For example, Black Friday may affect the forecast of sales.

In this analysis, a simplified analysis is conducted to forecast total debt as a function of time only. Given more data, I can deliver a more advanced analysis using the state-of-the-art Transformer model.

4.2 Knowledge-informed missing data imputation

As noted above, depending on the task, to find out the exact value of the past missing data may not be directly interested to modellers. However, the goal of many imputation methods is to deliver a more accurate model based on a complete data set. Gaussian process can probabilistically impute the missing value as suggested in this analysis. But such reconstructed values do not deliver more information in the forecasting model as it is indeed the same model.

Therefore, some other methods that can incorporate other knowledge into the modelling process may better inform the learning process and, subsequently, the forecasting. See this paper for further details.