A dive into US Treasury bonds forecasting with machine learning

A project made by:

Zakaria El Yassini

Peijin Chen

A project submitted for:

Statistical Learning (MAST 679 H) given by Professor Simone Brugiapaglia

Acknowledgments

We would like to begin this project by respectfully acknowledging that Concordia University is located on unceded Indigenous lands whose custodians are eh Kanien”keh’a:ka Nation.

This project was further inspired by course materials from MAST 679H and MAST 679G (Neural Networks course taught by Professor Cody Hyndman). I would also like to acknowledge the occasional use of Grammarly to assist in correcting minor grammatical errors throughout the preparation of this report.

Contributions

Throughout this project, I – Zakaria El Yassini – investigated of yield curve forecasting using both a one-day and a five-day lag predictor for yield rates across different maturities to predict the subsequent time step’s yield curve. For that purpose, I implemented several methodologies, including PCA Analysis, Linear regression, polynomial regression, and spline methods (both quadratic and cubic) with various regularization approaches ranging from smoothing constraints to a ridge smoothing constraint. Additionally, I have implemented a neural network using an LSTM architecture, that have been explored through a standardization of the data as well as applying a weight sharing constraint to reduce the complexity of the network. The impact of reduced data sampling frequency on these methodologies was also explored, forming the basis for a comprehensive comparative analysis.

Throughout this project, I, Peijin Chen, investigated how changing the features and lags as well as the forecast horizon changed predictive performance. To this end, I created different versions of the data that each had a unique lag amount and forecast horizon. In addition, I implemented several linear regression type models and tested their performance on these variants of the dataset. Lastly, I analyzed the results to see how the performance of the models varied as a function of the lags and the forecast horizon.

MOTIVATION

A yield curve [1] can be viewed as a graphical representation of the variation of yields on debt instruments – specifically, in this project, US Treasury bonds – across their years remaining to maturity.

The yield curve has widely served as a benchmark for various debt instruments in financial markets, from mortgage rates to bank lending rates, and functions as a critical indicator of short-term interest rates. Pension funds and insurance companies study yield curves extensively due to their significance in portfolio management and liability matching. Additionally, this curve provides a comprehensive snapshot of the bond market's condition, making it valuable for investors as it effectively predicts future economic activity and inflation levels- which directly impact daily goods valuation, financial assets, and real estate [2].

Throughout this project, the focus will be on predicting US Treasury bond yield rates using multiple methodologies: Linear models, higher-degree polynomials, spline methods, and models incorporating penalty and regularization terms. We will explore how is taking multiple models to predict each yield independently different from a multivariate modeling. We will also analyze how varying the numbers of time lags affect yield prediction at time t, while exploring the balance between model complexity and effectiveness. Essentially, we approach yields as time series data to forecast future yields using our selected methods. We will also evaluate the potential application of neural networks to forecasting yield curves.

These approaches have been extensively researched in financial literature, with interest significantly growing after McCulloch [3] introduced splines method to estimate interest rate by fitting a smooth discount function to observed Treasury Bond prices. His approach divides the term structure into intervals with cubic polynomial functions joined smoothly at knot points. This method in the core, translates heterogenous treasury security prices into a continuous discount function allowing for accurate interpolation of zero coupon yields across the entire maturity spectrum.

Yield Curve modeling has also been studied from another perspective by Van Deventer, Imai and Mesler [4]. Kamakura's study of yield curve smoothing [4] emphasizes deriving the most economically realistic and mathematically consistent curves by formally defining optimality for the smoothing. One key result is how models like the Nelson-Siegel yield has limitations in fitting complex yield structures, especially in specific types of bonds such as found in the US Treasury yields.

Recent work by Xiang Gao [5] integrate Heath-Jarrow-Morton (HJM) arbitrage-free models with neural networks for Treasury yield prediction. This approach combines stochastic calculus with deep learning techniques.

The HJM model describes the dynamics of instantaneous forward rates under a risk-neutral measure: $df(t,\tau) = \mu(t,\tau)\,dt + \sum_{i=1}^{d} \eta_i(t,\tau)\,dW_i(t)$, where μ is the drift term, η_i the volatility factors, and W_i Brownian motions.

This framework uses dynamic Nelson-Siegel factors (level X_1, slope X_2, curvature X_3) that follows mean-reverting process Stochastic Differential Equations (SDE):
$dX_t = \kappa_t(\theta_t - X_t)\, dt + \sigma_t \, dW_t$.

These SDEs are discretized to enable Kalman filtering. The forward rate is then expressed as an affine structure: $f(t, \tau) = \beta_\tau \, X_t$ where β_τ are deterministic loading parameters (e.g. Nelson-Siegel Basis) and X_t the state variables.

At the core of the neural network used, an LSTM architecture parametrize κ_t, θ_t, and σ_t. The HJM arbitrage free constraint, represented by:
$\Lambda = \frac{1}{2} B_\tau \Sigma_t B_\tau^T - B_\tau \kappa_t (\theta_t - X_t) \beta_0 X_t = 0$ where $B_\tau = \int_0^\tau \beta_u$ du and $\Sigma_t = \sigma_t \, \sigma_t^T$ is be enforced via a composite loss function $L = \left\| Y_t - \hat{Y}_t \right\|_2 + \lambda \cdot \Lambda$.

The Kalman filters provide state estimation, while particle filters handle non-Gaussian noise. Results show improved short-term accuracy with arbitrage constraints which offer a combination between a use of a model with macroeconomic criteria (through the dynamic Nelson-Siegel model), with fundamental market principles (through the non-arbitrage theory), creating a more robust predictive framework for Treasury yields.

Data Exploratory Analysis

Our project analyses the historical daily Treasury par yield curve rates [6] from the US Department of the Treasury website. We collected data spanning 17 years (2007-2024) including daily yields across different maturity points. These yields readings are at standardized maturities: 1, 1.5, 2, 3, 4, and 6 months, as well as 1, 2, 3, 5, 7, 10, and 30 years.

To prepare our dataset, the raw Treasury yield data had undergone several preprocessing steps. The US Department of Treasury website publishes data daily, organized in annual files, requiring us to merge these separate annual datasets into a comprehensive multi-year collection. Consolidating this process, we identified and addressed several data quality issues to ensure analytical integrity:

1. The 1.5-month maturity point showed negligible presence throughout the study period and was consequently excluded from the analysis
2. Yields for maturities under 6 months had inconsistencies and considerable noise; they were generating missing values for multiple subperiods. We have consequently excluded from our analysis this category
3. Rows with missing values were removed to ensure data completeness.
4. The Covid period (2020-2021) has generated a totally frozen and crashed bonds markets, which is why this period had also been excluded from the analysis, for simplicity.

Our redefined dataset thus focuses on yield predictions for maturities ranging from 6 months to 30 years, which defines our 9 predictors. The time series forecasting we will first approach deals with predictions are made at time t for these 9 maturities, based on yield observations at time $t-1$. Later sections will provide a more formal mathematical formulation of this prediction framework.

Additionally, we plan on investigating the consequences of extending the predictor lag- incorporating multiple consecutive days prior to time t - to potentially enhance the accuracy of our prediction problem. A similar investigation has been conducted on lowering the data capture frequency.

Throughout our analysis, the dataset the data set was initially partitioned into training (70%), validation (15%), and testing (15%) subsets. Given the structured nature of the data – its time series characteristics- we implemented a rolling window cross-validation to tune model hyperparameters and identify best model configuration and coefficients.

In this project, the Root Mean Squared Error (RMSE) was used as the primary evaluation metric due to its straightforward interpretation. The RMSE has the ability to penalize larger errors more heavily, and the method provides a clear result of the average magnitude of error across all statistical learning algorithms implemented.

Initial exploratory data analysis revealed substantial linearity among yields of differing maturities, particularly the maturities with their lagged maturity, e.g. 1 year maturity remaining with 6 months maturity remaining. The scatterplot visualization in ***Figure1*** confirms this claim, which become even more evident in the correlation matrix of these variables as seen in ***Figure2***.

Figure1 : Scatterplot of the US Treasury Bonds yields corresponding to various maturity remaining

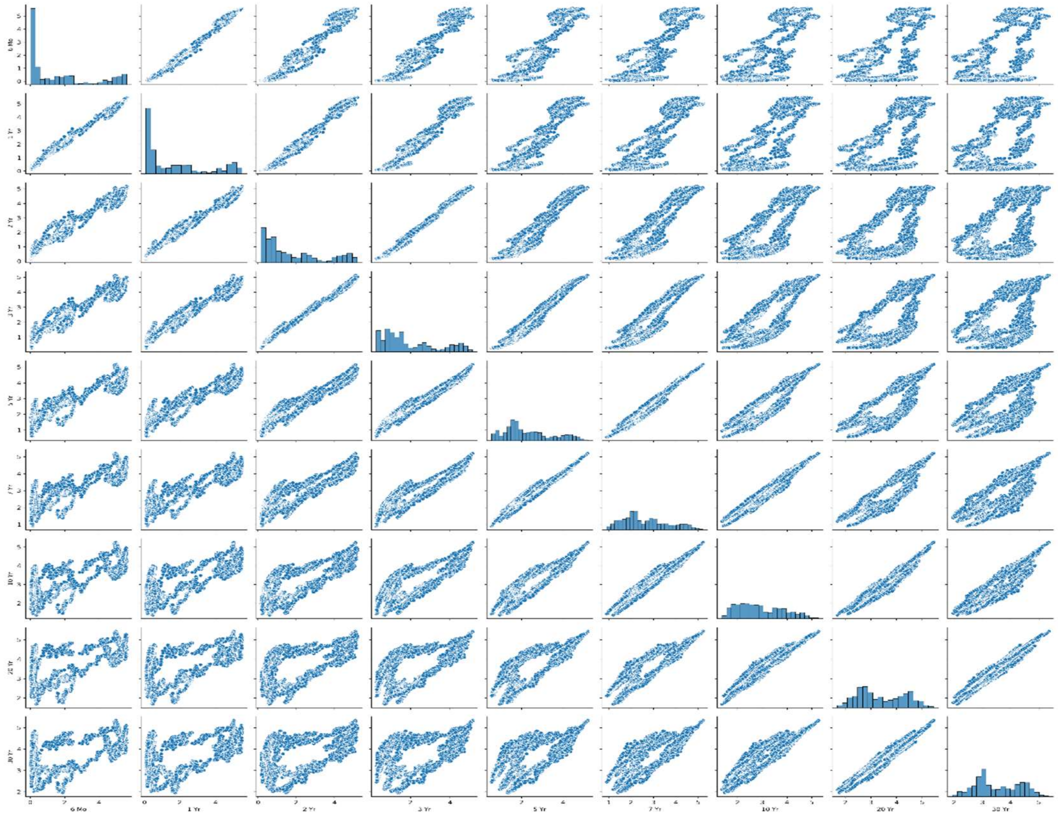
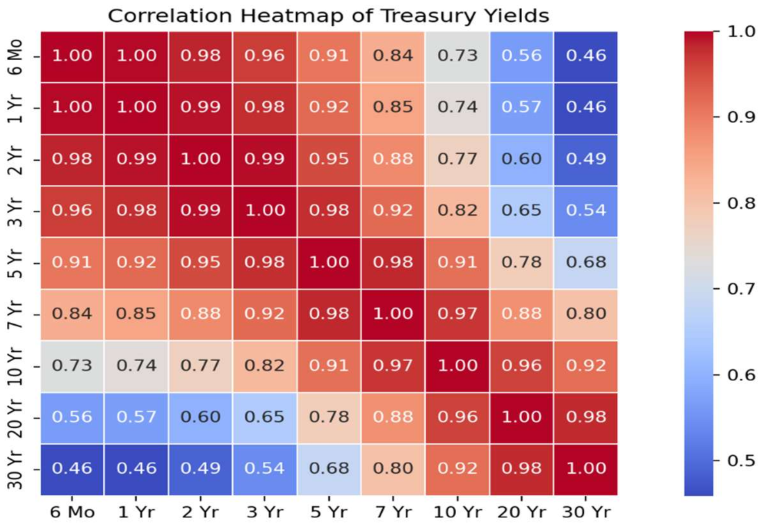


Figure2: Correlation Heatmap of US Treasury Bonds yields corresponding to various maturity remaining



Key Takeaway:

- This high degree of correlation suggests that Principal Component Analysis (PCA) would be an appropriate dimension reduction technique for our predictors, which we will explore in section 1 of Analysis.
- Short-term yields (under 6 months), and certain time periods demonstrate excessive volatility and inconsistency, and were excluded to improve model stability
- The time series nature of the data suggests that lagged predictors will be valuable for forecasting.

Analysis

Section 1: Principal Component Analysis

The Principal Component Analysis (PCA) serves as a dimension reduction technique that projects our Treasury yield data onto a new orthogonal coordinate system, where the k^{th} greatest variance lies on the k^{th} coordinate (the k^{th} principal component).

In our implementation of this method, we apply this spectral decomposition to provide a lower-dimensional structure that preserves 99% of the variance in the original yield rates data. Thus, allowing us to address the multicollinearity fundamentally present in the original yield curve data, while maintaining their essential structural characteristics.

Our analytical approach of the method proceeds as follows:

1- Transform the original yield data into the PCA space

2- Execute our prediction methods on this transformed orthogonal representation
3- Revert the predictions back to the original yield curve space

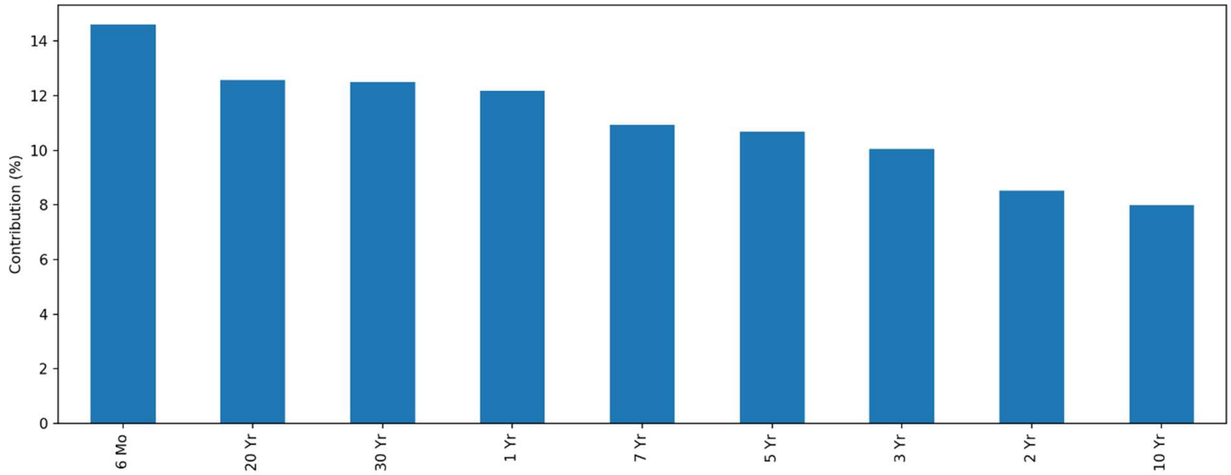
The eigen-decomposition of our covariance matrix reveals that the dominant yield rates dynamics is captured using considerably fewer variables than the original dimension of the predictor. Throughout our experiment, we have found that only 3 principal components were able to capture 99% of the variance that explains our original 9-dimensional predictor. The variance explained by each principal component is summarized in the following table, where values are rounded for 3rd and 4th columns; unrounded values result in a cumulative variance of 100% until the 9th principal component.

Principal Component	Eigenvalue	Variance Explained (S)	Cumulative (S)
1	8.7628	87.13	87.13
2	1.1608	11.54	98.67
3	0.1149	1.14	99.82
4	0.0092	0.09	99. 91
5	0.0062	0.06	99.97
6	0.0015	0.01	99.98
7	0.0007	0.01	99.99
8	0.0005	0.01	100
9	0.0004	0.00	100

These results show alignments with established findings in fixed income literature where the yield curve is commonly modeled by three underlying factors: level, slope, and curvature [7].

The visualization of factor loadings in ***Figure3*** demonstrates how each maturity contributes to the top three principal component (accounting for 99% variance), providing clarity on which maturities influence these dominant factors.

Figure3 : Contribution of each maturity to the top principal components



Having established our top principal components, we proceed with our forecasting framework. Our process employs a one-period lag structure, where we use top 3 principal components from time $t-1$ to predict yield curve behavior at time t.

We initially considered three different regression methodologies:

1- *Standard Linear Regression:* Serving as our baseline approach, modeling a direct linear relationship between lagged principal component and future yields. It is appropriate in this context as yield curve dynamics often exhibit linear dependencies over short time horizons. Additionally, the orthogonality of principal components helps mitigate multicollinearity issues that we typically find in yield curve forecasting models.

2- *LASSO Regression:* This was considered but omitted from our analysis since the Standard Linear Regression did generalize well. Moreover, the principal components themselves have already performed a dimensionality reduction that LASSO would typically help with.

3- *Polynomial Regression:* This was considered to capture potential nonlinear relationships between lagged principal components and future yields, as interest rate dynamics often exhibit asymmetric responses to economic shocks.

<u>Key results:</u>

Our implementation of these regression methods revealed a rather interesting pattern. The best polynomial model (degree 2) effectively converged to a linear model, attributing weights very close to zero for most higher-order terms and interactions. This behavior is clearly demonstrated in the coefficient values for each feature of this model. In each component i, we observe that PC_i has a dominant coefficient, with other terms being negligible in the regression method.

Our results shows that while PCA is an effective tool for dimensionality reduction in linear models, its application to non-linear models such as polynomial regression or even cubic splines can lead to the loss of the very complexity that we are trying to capture. PCA analysis might not be suitable for polynomial or cubic splines, which require flexibility and expressiveness to perform well in complex settings.

Following the implementation of our linear regression model in the PCA space, we obtained the coefficient matrix that characterizes the temporal dynamics of the yield curve components. This matrix is as follows: $\begin{bmatrix} 0.9976 & -0.0045 & 0.0022 \\ 0.0001 & 0.9971 & -0.0089 \\ -0.0003 & -0.0006 & 0.9930 \end{bmatrix}$, where row i represents the coefficients for predicting principal component i (PC_i) at time t, and the column i being the influence of PC_i at time $t-1$. Hence the element (i, j) quantifies the effect of the j^{th} principal component at the previous time step on the i^{th} principal component at the current time step.

The diagonal entries indicate that each component's value at time t is heavily influenced by its own value at the previous time step, suggesting stability in the underlying yield curve dynamics.

The model's performance was evaluated using the mean squared error (RMSE) across training, validation and test sets. With the two models implemented showing identical results, due to the convergence of the polynomial case to the linear model. Our training RMSE was of 6.89%, while the validation error metric was of 9.91%, with the testing set recording 10.44% error.

This shows an overall very excellent generalization. These results lead us to a known theoretical model in yield curve literature, with the first principal component corresponding to the “level” of the yield curve, the second principal component corresponding to “slope”, and the third principal component representing “curvatures” [7].

Section 2: Moving beyond PCA analysis

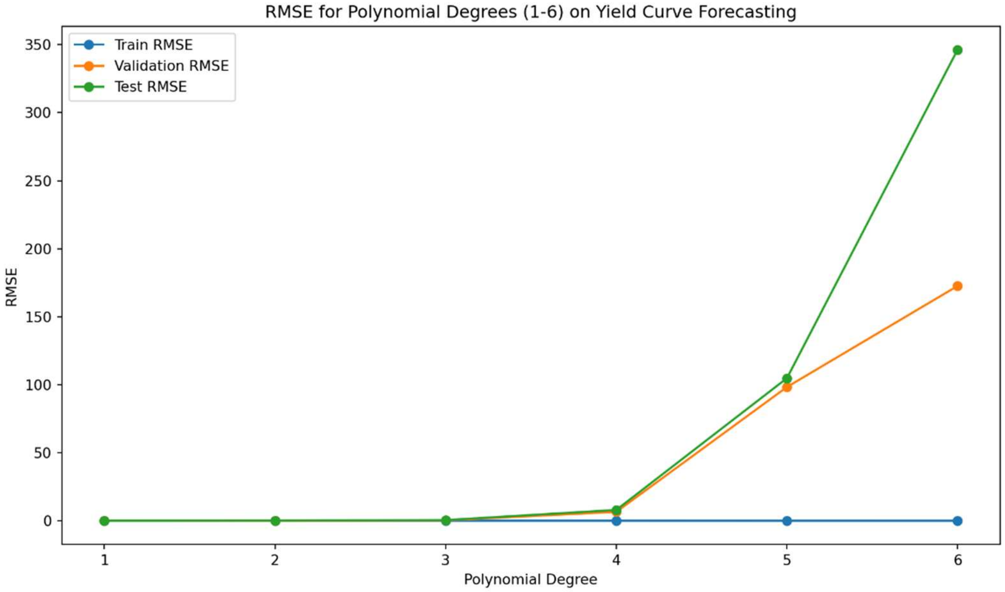
Polynomial Regression

Following our implementation of the dimension reduction technique, namely PCA, our focus shifted towards other modeling techniques commonly used in the context of statistical learning. Our first methodology included running a polynomial regression with varying degrees ranging from 1 to 6. This approach is naturally appealing in terms of yield curve forecasting. As it will allow flexibility of fitting a larger number of models with higher degrees polynomials but also treat it the same as a linear model for simplicity at degree 1.

For a polynomial of degree p, the model for each maturity i, is modeled as follows: $\hat{y}_t^{(i)} = \sum_{j=0}^{p} \sum_{\tau \in M} \beta_j^{(\tau)} \left(x_{t-1}^{(\tau)}\right)^j$, with $x_{t-1}^{(\tau)}$ representing the yield rate at maturity τ at time $t-1$, $\beta_j^{(\tau)}$ is the coefficients of the feature j of degree j at maturity τ, and M being the set of maturities. Note that when we approached this exact problem with a higher lag (5 days lag), we predict with $x_{lag\ 5}^{(\tau)}$, which represent the different yield rates for the previous 5 days. Each maturity i, has their own set of coefficients, but take all the previous maturities of the previous time step as predictors.

For the one day-lag prediction problem, ***Figure4*** compares the performance of the different degrees of polynomials considered

Figure4 : Evolution of RMSE metrics in function of polynomial degrees



Interestingly, the linear model demonstrated superior performance compared to the other models, followed by degree 2. This suggests the apparent complexity of yield curves, their day-to-day movement might follow a relatively simple pattern, with excessive complexity only serving as a shortcut towards overfitting rather than improving forecasting. On this note, the linear model produced a remarkable training RMSE of 5.28%, a validation RMSE of 5.73% ,and a testing RMSE of 6.98%.

Spline methods

Beyond the polynomial regression, our implementation of machine learning algorithms included a quadratic spline method, which theoretically should provide more flexibility in capturing the varying curvatures across the maturity segments considered. This model can be represented as:

$$\hat{y}_t^{(i)} = \sum_{\tau\in M}\sum_{j=1}^{k}\left[a_j^{(\tau)} + b_j^{\tau}\left(x_{t-1}^{(\tau)} - \kappa_j\right) + c_j^{\tau}\ \left(x_{t-1}^{(\tau)} - \kappa_j\right)^2\ \right] I_{\{\,x_{t-1}^{(\tau)}\in[\kappa_j\,,\,\kappa_{j+1}]\,\}}$$

Where k is the number of knots, k_j are the knot points, I is the indicator function, and $a_j^{(\tau)}$, $b_j^{(\tau)}$,$c_j^{(\tau)}$ being the spline coefficient for each segment and maturity.

Now another method that was considered for the splines methods is the cubic spline model. It was a natural progression as it offers greater flexibility while maintaining continuity in the second derivative. The cubic spline can in be represented in the same way via

$$\hat{y}_t^{(i)} = \sum_{\tau\in M}\sum_{j=1}^{k}\left[a_j^{(\tau)} + b_j^{\tau}\left(x_{t-1}^{(\tau)} - \kappa_j\right) + c_j^{\tau}\ \left(x_{t-1}^{(\tau)} - \kappa_j\right)^2\ +\ d_j^{\tau}\ \left(x_{t-1}^{(\tau)} - \kappa_j\right)^3\right] I_{\{\,x_{t-1}^{(\tau)}\in[\kappa_j\,,\,\kappa_{j+1}]\,\}}$$

In the context of yield curves, a natural thought that arises is that it needs to be smooth, as this creates a more reliable and consistent representation of market expectations for future interest rates. It also helps address potential inaccuracies caused by noise the US Treasury bond market. For both methods, we have implemented a smoothing constraint, to penalize the wiggling of the curves predicted by our methods.

This introduces a parameter controlling the trade-off between fitting the data and minimizing curvature. Our objective function in the smoothing spline model is $argmin_{\beta}\ RSS + \lambda \int\ (f''(x))^2\ dx$, with λ the smoothing penalty, associated with the roughness penalty in the integral

Another variant of the model that we had considered, which helps in case of overfitting the data was the ridge smoothed spline method, which is essentially a smoothed spline with a ridge component added to it. This model was selected due to the closed form solution for the vector of spline coefficient $\beta =\ (\ X^T\ X + \lambda\, D^{(2)^T} D^{(2)} + \alpha\ I\ )^{-1}\ \ X^T Y$, with $D^{(2)}$ the second-difference matrix which is the discrete approximation to the second derivative, λ being the smoothing penalty, X the spline-transformed feature matrix, and α the ridge penalty hyperparameter. A more formal derivation of this formula will be presented in an attached document.

After performing the rolling window sequential cross validation to tune the parameters k, λ and α, the model that performed the best out of the six models considered above for the one-day lag prediction problem was the smoothed cubic spline method, with ridge penalization, which comes in agreement with how McCulloch [3] introduced Splines methods rather for yield curve modeling.

Our findings show that the tuned parameters were $k = 3$, $\lambda = 0.001$ and $\alpha = 0.0215$. This configuration yields a training RMSE of 6.24%, a validation RMSE of 8.66%, and a testing RMSE of 19.23%. These results still show a considerable amount of overfitting, as evidenced by the large gap between the test and training error metrics. It is also worth nothing that this method still lags behind the linear model considered previously.

<u>Neural networks: LSTM architecture</u>

Venturing beyond traditional statistical learning methods, we have explored the application of deep learning into yield curve forecasting. The choice of LSTM was particularly relevant to our case for a few compelling reasons:

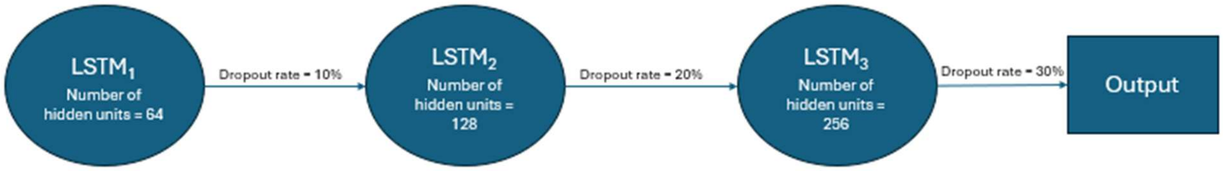
One of which has to do with our conceptualization of yield rates as a time series data with temporal dependencies. We are in a case of structured data, where permutation of the data points order could tremendously make the dataset lose some of the key component that they represent.

Secondly, LSTM showcases the ability of transforming the ability to selectively pass information through their memory cells, it preserves long-term dependencies while avoiding the vanishing or exploding gradient problems that regular RNN architectures have [8].

This capability is particularly valuable for yield curve forecasting, where capturing long-term trends influenced by shifts in major macroeconomic factors and other important information over time is crucial.

In our model, we incorporated several key elements to enhance the model performance and also to prevent overfitting. Our architecture in short was a three stacked LSTM neural network with increasing complexity in each layer (represented in the number of hidden units) and an increased dropout rate regularization, that matches the increasing complexity, to randomly deactivate a fraction of neurons during training, thus promoting generalization. A simple diagram explaining our most simple LSTM architecture can be viewed in ***<u>Figure5</u>***

<u>Figure5 :</u> Neural Network architecture simplified



The Adam optimizer was chosen as our way to implement stochastic gradient descent, to efficiently update the weights, leveraging learning rates - which was constant for the sake of stability updates across training epochs and for simplicity- and momentum for faster convergence. The momentum stochastic gradient descent which was implemented through the 32-batch sized Adam optimizer helps dampens oscillations and accelerate convergence by adding inertia to the parameter updates [9].

Additionally, early stopping was implemented by monitoring the validation loss and halting training when no improvement over a 5-epoch interval was achieved on the validation set. This helps with the computational cost of the method, helps with overfitting and preserves the model's ability to generalize new data.

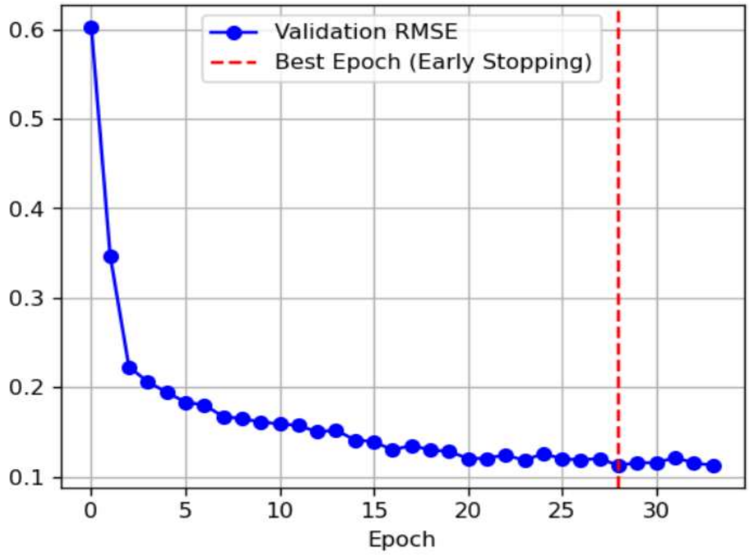
Furthermore, we have considered standardizing the data before running it through the network based on the training set only to control information leakage occurring across validation and test sets.

Another improvement model considered was by having a weight sharing constraint, to account for our previous result of the maturities being correlated one with another, weight sharing assumes they evolve according to a common dynamic. This constituted the idea behind an addition added to standardized Neural network architecture we have considered.

Our implementation of these three models showcased a performance comparable to our best-found model, namely the linear model, in terms of performance. We have found that although the initial model did in fact overfit, the standardization of the data did help reduce the overfitting the validation and testing RMSE, whilst the weight sharing constraint did in fact eradicate that problem.

The performance of this best model – when applied to the one-day lag prediction problem- exhibited a training RMSE of 6.32%, a validation RMSE of 11.86% and a testing RMSE of 11.67%. The early stopping was triggered at Epoch 34. This concept can be visualized through ***Figure5.***

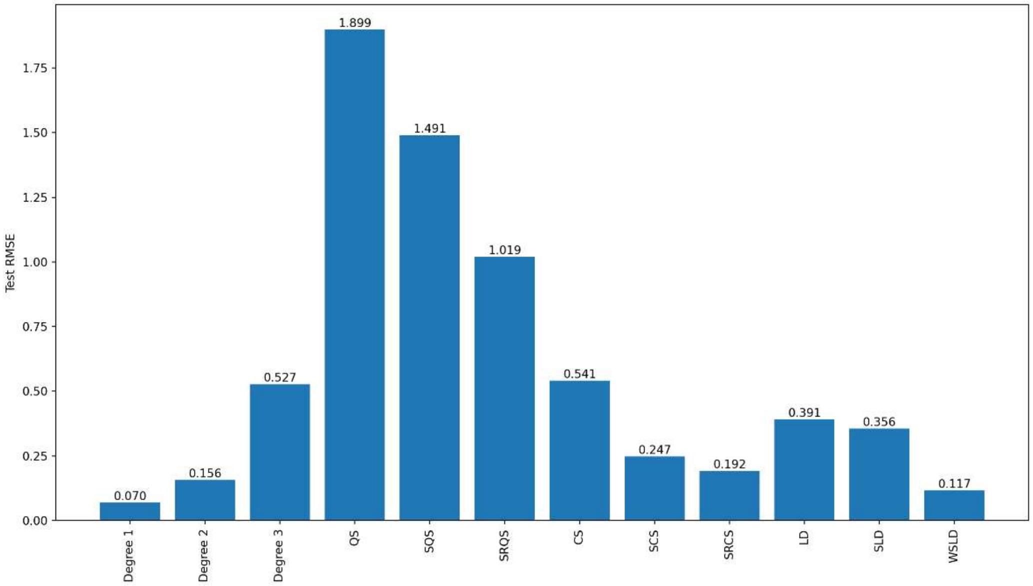
Figure5 : Early stopping representation for the standarized stacked LSTMs with weight sharing



Benchmarking Model Performance: A Cross-Methodological Evaluation

In order to systematically evaluate the relative merits of each of the approaches on our prediction problem, a comparative analysis using a bar chart plot of the root mean squared error distribution for the testing set was conducted. This visualization in ***Figure6*** clearly illustrates which method was able to generalize the best on unseen data.

Figure6 : Algorithm testing error comparison plot for the one-day lag prediction problem



This resulting graph showcases the superiority of the linear regression, the weight-shared and standardized stacked LSTM model, and the quadratic polynomial model, with the former showing a marginally lower error than the neural network model, in the one-lag prediction problem. A detailed discussion of these findings is provided in the conclusion.

Temporal Horizon Effect: Performance across extended Time Lags

In this section, the effect of increasing the lag horizon on the predictors used for forecasting US Treasury bond yields is explored. Instead of using only a one-day lag, where target values would be predicted from the previous day, a 5-day lag is being used instead. This means that for time step t, the model uses yields from the previous 5 days.

The rational behind this change is to assess the degree of influence that incorporating a larger set of historical data has on the prediction accuracy, as well as if the model that generalized best on the unseen data (the test set) would change with that in mind. The same statistical learning algorithms were performed as before - polynomial regression with varying degrees, Splines methods (both with and without penalization terms), and stacked LSTM with dropout rates (in their basic form, with standardization and with combined standardization and weight sharing) - maintaining the same reasoning for model selection and comparison.

The best polynomial regression method was a first-degree polynomial just as before, showcasing incredible accuracy, and generalization capabilities in contrast to higher degree polynomials that tended to overfit the data. With a training RMSE of 10.85% , a validation RMSE of 13.29% , and a testing RMSE 15.78%, the linear model performed the best out of all the models considered in the section. The parameters used in this model are as follows:

Interestingly so, some results in the splines method showcased an unusual pattern with the validation RMSE being lower than the training RMSE. Several factors may account for this unexpected behavior:

1- The presence of more complex samples in the training data
2- Validation set being inherently easier to predict
3- Possibility of intensive hyperparameter tuning
4- Data leakage between training and validation sets

The first two point would be less likely to have been the case as other models behaved normally for that matter.

Increasing the lag of the predictor made the regularization hyperparameters penalize more so than the one-day lag case, for some of the models considered in the spline methods, namely the smoothed cubic spline with added ridge penalty, with a smoothing penalty of $\lambda = 1438$.

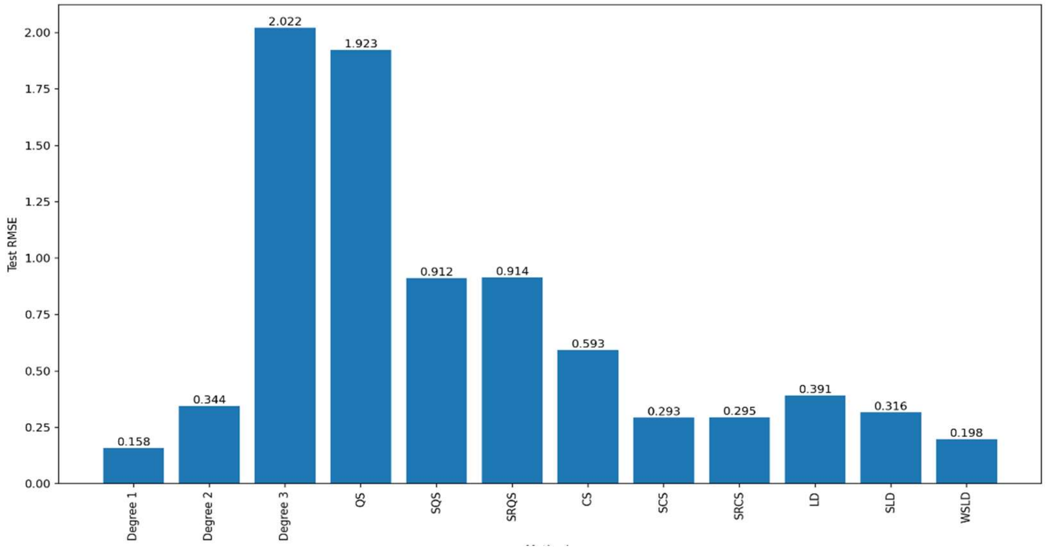
The best spline method that generalized the best on the testing dataset was the smoothed cubic spline. The hyperparameters obtained through a rolling window cross validation was $k = 3$, and $\lambda = 0.0127$. This method gave an 11.29% training RMSE, an 12.61% validation RMSE and a 29.32% testing RMSE.

Moreover, for the neural network model we have considered, it has also showcased the best result with a combined standardization of the data and weight sharing constraint added. The performance metrics resulting were: 11.79% on the training dataset, a 17.17% on the validation dataset and 19.84% on the testing dataset.

The visualization of these results is clearly represented in ***Figure7*** , where a comparative bar chart showed that the five-day lag prediction problem had the same top 2 best models on the generalization over the testing set (namely Linear model following by the weight-shared and standardized stacked LSTM model). However, a notable change occurred in the 3rd position, with the smoothed cubic spline claiming this spot. This shift incorporates how certain modeling techniques respond in a different manner to extending the temporal lags, suggesting that the cubic spline methods with smoothing constraints better capture yield curve dynamics at longer prediction horizons compared to the previously higher-ranked model - the quadratic regression-.

Overall, the increase of the prediction lag did not increase the performance over the test set, aside from the smoothing quadratic spline and the standardized neural network. This is adherent with what would be expected from the behavior of increasing the lag on a small horizon (5 days). This reflects the complex nature of yield curve modeling, where information relevance decays non-uniformly across different time horizons. This is precisely why the LSTM networks were included in our methodology – their architectural design specifically addresses long term dependencies.

Figure7 : Algorithm testing error comparison plot for the five-day lag prediction problem



Temporal Resolution Effects: Analysing Lower-Frequency Data Capture

This section examines how the granularity of temporal data sampling affects the predictive power performance in US Treasury data. By transitioning from daily to semi-monthly data capture (The first day and mid-month points approximately), we investigated the effects of altering the frequency of the observations to the best model captured, as well as the overall prediction accuracy based on what we know from the approximate behavior of bonds.

Taking note of consistency in this section of our analysis, we have employed the same statistical learning methods as in previous sections, adhering to identical evaluation criteria and comparative frameworks.

Our implementation for the polynomial regression framework showed consistency with patterns observed in both the one-day lag and the five-day lag analysis. Amongst the six comparative methods evaluated, the linear model demonstrated superior generalization capability, with a training RMSE of 30.81%, validation RMSE of 52.74%, and testing RMSE of 57.21%. Higher degree polynomials models exhibited characteristic overfitting behavior, with performance degradation on out-of-sample data despite improved training set accuracy.

The splines methods exhibited patterns consistent with out first analysis. Among these approaches the smoothed cubic spline with a ridge regularization showcased superior generalization capabilities, achieving a testing RMSE of 62.93%, while maintaining competitive training accuracy (RMSE of 35.67%) compared to the linear model. The validation RMSE of 29.42% suggests that this performance resulted from an extensive hyperparameter tuning. These results highlight the importance of judicious regularization implementation when modeling stochastic processes such as the US Treasury bonds, particularly when employing rolling-window cross validation.

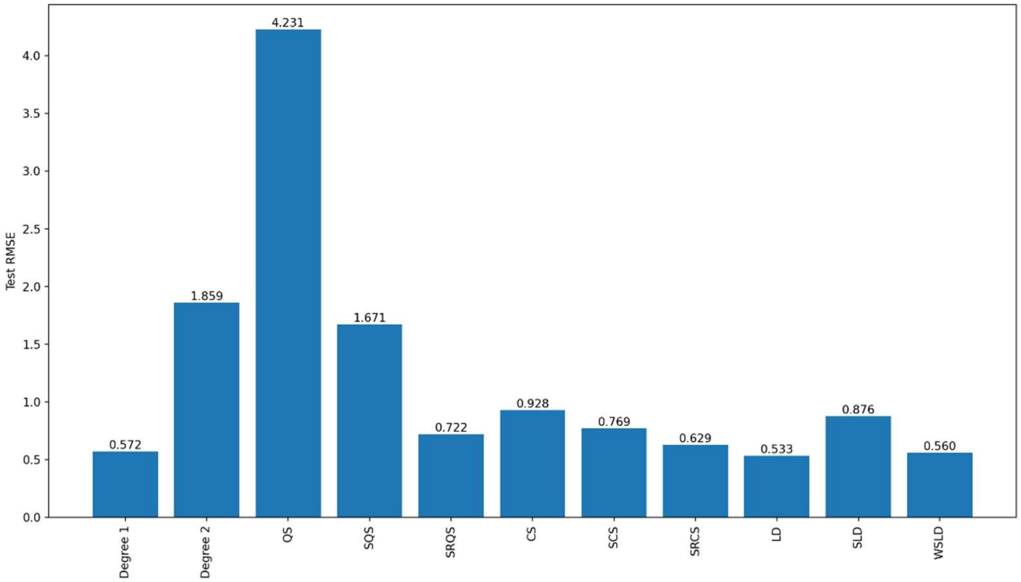
To enhance the model robustness, more effective hyperparameter optimization strategies should be explored. Techniques such as randomized search, and Bayesian optimization could potentially lead to better parameter configurations while reducing computational overhead. Our current search method incorporated logarithmic spaced values for λ and α, and number of knots varying from 3 to 10. The optimal configuration obtained via this method was: number of knots of $k = 5$, a smoothing hyperparameter of $\lambda = 206.9$, and a ridge parameter of $\alpha = 0.0048$; this is a case of a more penalizing smoothness parameter.

Moreover, the neural network architecture evaluation revealed distinctive performance patters across our LSTM variants. The base LSTM with dropout regularization demonstrated the strongest performance, for notably both the training accuracy, and generalization capability. Achieving a training RMSE of 28.24%, an RMSE of 50.12% on the validation set, and an RMSE of 53.33% on the testing set. This method was only slightly better than the standardized neural network with weight sharing. This suggests that while the weight sharing did introduce beneficial bias, it did not fully overcome the challenges introduced by the standardization. These challenges are in fact, the loss of valuable scale information related to the lower frequency of data capture. Hence, leading standardization to obscure some of the valuable information in the data.

The effects of reducing the temporal resolution are clearly depicted in ***Figure8***, where a comparaison across models shows a distinct shift in the predictive hierarchy. This transition to semi-monthly observations favored more complex architectures like the LSTM models. In particular the stacked LSTM with increased dropout rates emerged as the top performer, followed closely by the standarized, weight-shared, LSTM architecture, and then the simpler linear model.

The results confirms that even with sparser data, deep recurrent architectures retain their advantage in capturing the underlying dynamics of Treasury yields, especially due to their ability to leverage long-term dependencies.

Figure8 : Algorithm testing error comparison plot for the semi-monthly observed, 5-observation lag prediction problem



Section 3: Forecast Horizons in US Treasury Yield Prediction Models

Published research on forecasting commonly employs distinct temporal horizons when analyzing US Treasury bond yields, each chosen based on the specific application and the trade-off between accuracy and the forecast's time span [10].

- Ultra-short-term forecasts (e.g., daily or weekly) are utilized in trading strategies and risk management, where immediate price movements are critical [11].
- Medium-term forecasts, ranging from one to six months, are relevant for portfolio adjustments and understanding cyclical trends in interest rates [12].
- Longer-term forecasts, extending from one to two years or even longer, are crucial for strategic asset allocation and macroeconomic analysis, particularly in anticipating shifts in monetary policy and economic growth that significantly impact Treasury yields [13], [14].

For instance, models incorporating macroeconomic factors have been used to forecast 10-year Treasury yields over one to two-year horizons [14]. The accuracy of these forecasts generally decreases as the horizon lengthens, reflecting the increasing uncertainty associated with predicting future economic conditions and market sentiment [10].

Our analysis focuses on not only on the forecast horizon but the number of lags (autoregressive order). Specifically, we tested combinations of lags in the set $\{10,30,60\}$ (in days) and forecast horizons in the set $\{1,5,10,30\}$ (days). The models employed were Ridge Regression, LASSO, and ElasticNet, which use L_2, L_1, and a combination of L_1 and L_2 regularization. The output of the models is multi-valued, that is, 9 values for each of the selected maturity levels.

We should also mention that the changes in Treasury yields change over time: that is, it is typical for yields to change 2-10 basis points ($.02\%$ to 0.10 %) over a day, but changes over the course of 1-6 months could be as much as 100 basis points (1%).

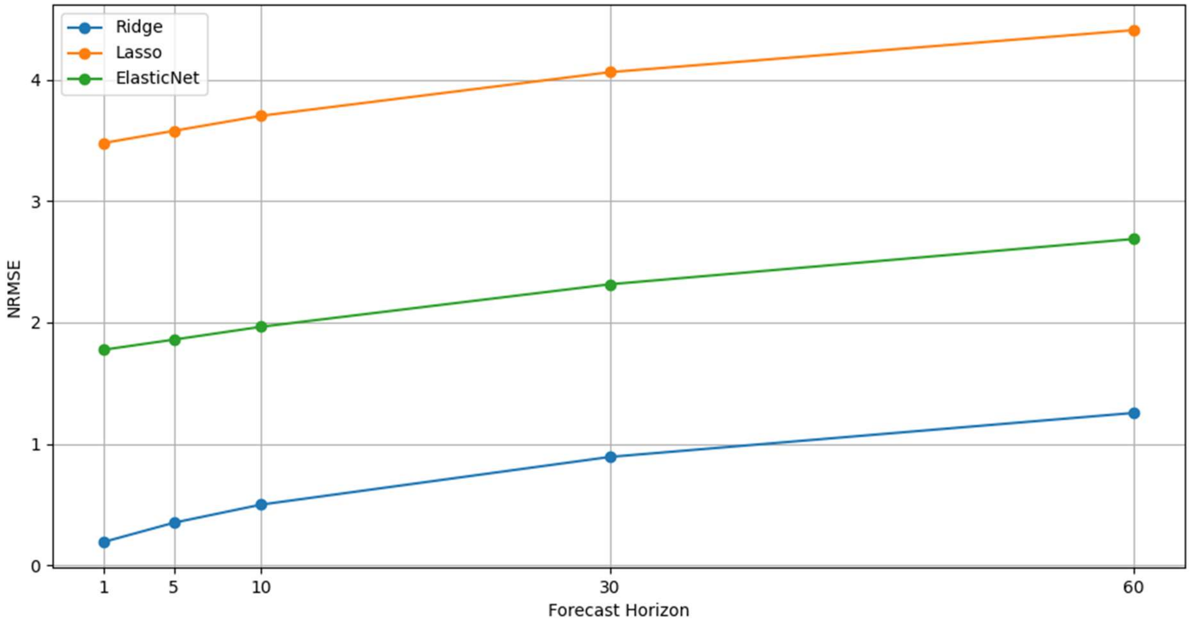
One difficulty in assessing the forecasting errors of bond yields is that errors accumulate over longer forecast horizons. It is typical for bond yields to change 2-10 basis points ($.02\%$ to $.10$ %) over the course of a day, while over the course of 3-6 months, they might change as much as 100 basis points (1%). This makes comparing errors between 1-day ahead forecast and 30-day ahead problematic.

For this reason, we chose to use the Normalized Root Mean Square Error (NRMSE), a performance metric that evaluates the prediction accuracy of a model by scaling the root mean square error (RMSE) of predicted Treasury yields relative to the volatility (standard deviation) of the actual observed yields. This normalization helps assess the model's performance in the context of how volatile the yields are, making it easier to compare models across different periods with varying yield volatility. It is defined as follows:

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{\text{true},i} - y_{\text{pred},i}\right)^2}}{\sigma_{\text{true}}}$$

In the above, σ_{true} represents the standard deviation or volatility of the yields, that changes with respect to the forecast horizon, and predictor lag.

Figure 9 : Normalized RMSE of Linear Models over various Forecast Horizons for 10-day lag



Where Results show that Ridge Regression models outperformed the LASSO and ElasticNet models. We can see that model performance tends to degrade as the forecast horizon increases, especially at the 30 and 60 day forecast horizon, suggesting that longer-term forecasting might benefit the most from more advanced statistical and machine learning methods. It also suggests that exogenous macroeconomic variables that offer different sources of information might be necessary to improve forecasting in the intermediate and longer-term time range.

Conclusion

This study systematically explored the prediction of US Treasury yields under varying temporal settings and modeling strategies, revealing important insights on both model performance and the underlying bond market dynamics.

Future work should incorporate more robust hyperparameter optimization methods such as randomized search, or Bayesian optimization to enhance model robustness and prevent extensive hyperparameter tuning. Moreover, while excluded the data from the 2020-2021 period, which characterized a freezing of the Treasury bond data, we acknowledge that a more sophisticated temporal interpolation method-like the dynamic or vanilla Nelson Siegel Model- would have provided a more principled approach to handling this period and missing data.

We can acknowledge that a more comprehensive of each model's performance can also be conducted by considering other error metric, such as the Mean Absolute Error (MAE), R^2 (coefficient of Determination), and Adjusted R^2. Different conclusion could be made based on each of these metrics, as MAE provides insights into average prediction error without overemphasis on larger deviations, R^2 allows us to capture the proportion of variability in the data captured by the model, and the adjusted R^2 is useful when comparing models with different numbers of features, which could be an excellent metric in our case.

Throughout our analysis of various prediction problems, we have opted not to include the model coefficients in the report due to their volume. For instance, a linear model fitting a nine-dimensional predictor plus its bias to 9 targets independently, will yield 90 coefficients (with biases included), this introduces unnecessary cluster. Instead, we have included the coefficients in the accompanying code file *"main"*.

Our analysis began with the principal component analysis, which revealed that 99% of the variance in the original nine-dimensional predictor space could be captured by just three principal components. This aligns with fixed income theory, where the yield curve dynamics is characterized by level, slope, and curvature factors [7]. While this approach proved its effectiveness for dimensionality reduction especially in linear context, we believe that applying this method to the most mathematically, and financially sound model- particularly over of very long horizon and lag interval- may be suboptimal, as it risks oversimplifying the problem.

The Principal Component Analysis can oversimplify yield curve dynamics by neglecting important non-linearities and structural constraints. This is especially relevant when incorporating stochastic no-arbitrage constraints, where assumptions underlying PCA may not fully align with the economic structure of the model.

In the one day-ahead forecasting setting, a simple linear model achieved outstanding performance across all methodologies evaluated. This performance is consistent with the quasi-martingale behavior of short-term US Treasury Bonds, in which the most accurate predictor of tomorrow's rate is today's. Although we have introduced more sophisticated LSTM architectures -enhanced through standardization and weight sharing- theses models delivered competitive yet slightly inferior results. These findings suggest that in the ultra-short-term settings, additional model complexity provides limited, if any, tangible benefits.

When increasing the prediction horizon to five days, we observed a consistent increase in the forecast error across relative to the one-day-ahead setting. This outcome likely reflects the behavior of the yield curve, where daily fluctuations tend to be modest – typically a few basis point for the short maturity remaining, and bigger for higher maturity remaining- but accumulates over multiple days, leading to a greater prediction uncertainty. Despite the increased volatility, the linear model continued to outperform its more complex counterparts, reinforcing the idea that model simplicity remains with the advantage in the very short-term forecasting.

This project demonstrated a clear pattern in the relationship between model complexity and temporal horizon in yield prediction. Simpler models, namely the linear model, tended to perform exceptionally well on the short-term horizon due to the martingale-like nature of yield rates. However, more complex models did increasingly well as the horizon was enlarged. For instance, the ridge regularized, smoothed cubic spline started outperforming the quadratic polynomial model as the prediction lag increased. Moreover, the stacked LSTM model did outshine the linear model, when the observation frequencies decreases.

These findings emphasize the necessity of careful model selection, to ensure model complexity aligns with the structure of the underlying data.

A common application of our model is to forecast the next observation that we would like to capture. For example, using the linear model for the one-day lag and five-day lag prediction problems, and using the best LSTM model for the semi-monthly case, our forecast for the first observation of 2025 can be shown in the table:

	Maturities								
	6 Mo	**1 Yr**	**2 Yr**	**3 Yr**	**5 Yr**	**7 Yr**	**10 Yr**	**20 Yr**	**30 Yr**
One-day Lag	4.25	4.17	4.24	4.26	4.38	4.48	4.58	4.86	4.78
Five-day lag	4.28	4.21	4.21	4.24	4.35	4.45	4.57	4.87	4.80
LSTM (semi-monthly)	4.34	4.30	4.19	4.12	4.34	4.45	4.58	4.86	4.80

A result check from the US Department of Treasury website [6] shows that these results are consistent with the real-world data, with the one-day lag providing the closest results to the actual curve. The LSTM model for semi-monthly did remarkably well, especially for the higher time remaining to maturity, which showcases the model to use long term dependencies.

And to finish with, our study also showcased that with simple models, the relative ability to generalize well on unseen data seem to diminish as the lag of predictor increases, and the forecast horizon does as well. We also noticed that more complex model shows relatively a better performance in their use hence the necessity of careful model selection depending on the set-up of the prediction problem.

Application of these methods is especially valuable in the context of risk management, high-frequency trading, hedging strategies and forecasting the markets movement- including recession prediction.

Bibliography

[1] F. S. Mishkin, "Yield Curve," *NBER Working Paper Series,* vol. 3550, 1990.

[2] H. Tatsat, B. Lookabaugh and S. Puri, Machine Learning and Data Science Blueprints for Finance, Sebastopol: O'Reilly Media, Inc., 2021.

[3] J. H. McCulloch, "Measuring the Term Structure of Interest Rates," *The Jounrnal of Business,* vol. 44, no. 1, pp. 19-31, 1971.

[4] D. R. van Deventer, K. Imai and M. Mesler, Advanced Financial Risk Management: Tools and Techniques for Integrated Credit Risk, Market Risk, Liquidity Risk and Interest Rate Risk Management, Hoboken, NJ: Wiley, 2013.

[5] X. Gao, "Stochastic control, numerical methods, and machine learning in finance and insurance," Concordia University, Montreal, 2021.

[6] U. D. o. t. Treasury, "Interest rate statistics," U.S. Department of the Treasury, 24 Apr. 2025. [Online]. Available: https://home.treasury.gov/policy-issues/financing-the-government/interest-rate-statistics. [Accessed 24 Apr. 2025].

[7] R. Litterman and J. Scheinkman, "Common Factors Affecting Bond Returns," *The Journal of Fixed Income,* vol. 1, no. 1, pp. 54-61, 1991.

[8] R. C. Staudemeyer and E. M. Rothstein, "Understanding LSTM – A Tutorial into Long Short-Term Memory Recurrent Neural Networks," Schmalkalden & Singapore, Schmalkalden University of Applied Sciences & Singapore University of Technology and Design, 2019.

[9] C. M. Bishop and H. Bishop, Deep Learning: Foundations and Concepts, Cham: Springer Nature Switzerland AG, 2024, p. 220.

[10] C. D. F. Li, "Forecasting the term structure of government bond yields,," *J. Econometrics,* vol. 130, no. 2, pp. 631-661, 2006.

[11] M. K. K. Brandt, "Price discovery in the US Treasury market: Evidence from on-the-run and off-the-run securities,," *Rev. Financ. Stud.,* vol. 17, no. 3, pp. 733-768, 2004.

[12] T. C. R. M. E. J. Adrian, "Pricing the term structure with linear regressions," *J. Financ. Econ.,* vol. 110, no. 1, pp. 110-138, 2013.

[13] A. P. M. Ang, "A no-arbitrage vector autoregression of term structure and inflation: Evidence from the US," *J. Econometrics,* vol. 113, no. 1, pp. 115-151, 2003.

[14] J. P. M. Cochrane, "Bond risk premia," *Am. Econ. Rev.,* vol. 95, no. 1, pp. 138-160, 2005.