Learning the link: Exploring US Market Dependencies with Machine Learning

Alexandra Liu Olsen, plj661

Carla Simonÿ Mc Nair, xjr742

Nathalie Aaes-Jørgensen, sbr482

Stine Breiner Pedersen , flg771

We contributed equally to the project.

Outline

1. Data presentation
2. Clustering: Are there any links to be learned?
3. Learning the link:
    3.1. US yield data → US yield data *(XGBoost, Gru)*
    3.2. All data → Stock data *(XGBoost)*
    3.3. All data → Interest rates *(XGBoost, NN)*
4. Detecting rate date changes *(VAE)*
5. Conclusion: Did we learn any link?

Naming conventions

“Type of yield X”: Government bond with maturity of X years

“FED(-rate)”: The Federal Reserve i.e. US Central bank, promoting stability by i.e. determining the interest rate

“ZCB”: Zero Coupon Bonds

“S&P”: Leading american index of stock prices

Data - **Introduction**

Tabular data, time series

Challenges!

28 features excl. date

10.528 time observations

Data - Introduction

Tabular data, time series

28 features excl. date

10.528 time observations

US yield data:

10.073 time observations and 20 features
20 different yield types of zero coupon bonds
NA's removed (4,3%)

Interest rate data:

8.558 time observations and 3 features
FED's interest rates and meeting decisions
NA's removed (4,69%)

Stock data:

8.558 time observations and 5 features
Historic data of S&P 500 index
NA's removed (4,69%)

Data - Introduction

FED data:

Interest rate data:

8.558 time observations and 3 features
FED's interest rates and meeting decisions
NA's removed (4,69%)

US yield data:

10.073 time observations and 20 features
20 different yield types of zero coupon bonds
NA's removed (4,3%)

Training, Validation and Test sets

Tabular data, time series

Training: **82%**

- 1985/01/02 - 2017/31/12
- 8234 obs.

Validation: **17%**

- 2018/1/1 - 2024/31/12
- 1750 obs.

Test: **1%**

- 2025/1/1 - 2025/9/5
- 89 obs.

Seen as a forecasting period

Motivation - Learning the link

US yield data:

10.073 time observations and 20 features
20 different yield types of zero coupon bonds
NA's removed (4,3%)

Interest rate data:

8.558 time observations and 3 features
FED's interest rates and meeting decisions
NA's removed (4,69%)

Stock data:

8.558 time observations and 5 features
Historic data of S&P 500 index
NA's removed (4,69%)

Boosting decision trees

- ➔ robustness on tabular data
- ➔ non-linear relationships
- ➔ little preprocessing
- ➔ not particular for sequential dependencies

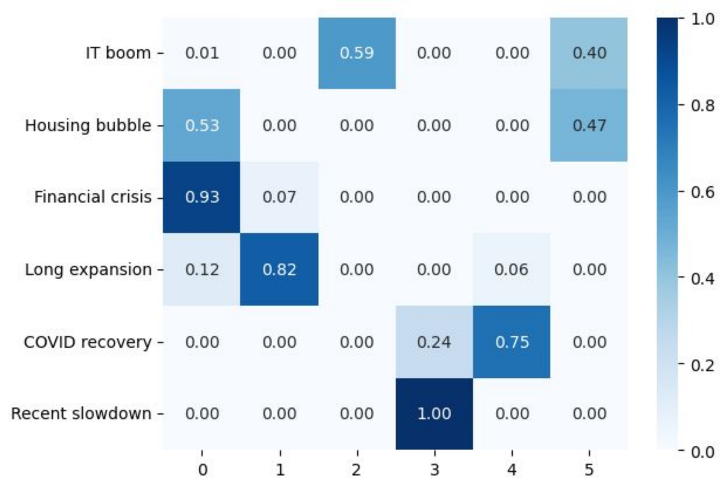
Detecting movements on the financial markets

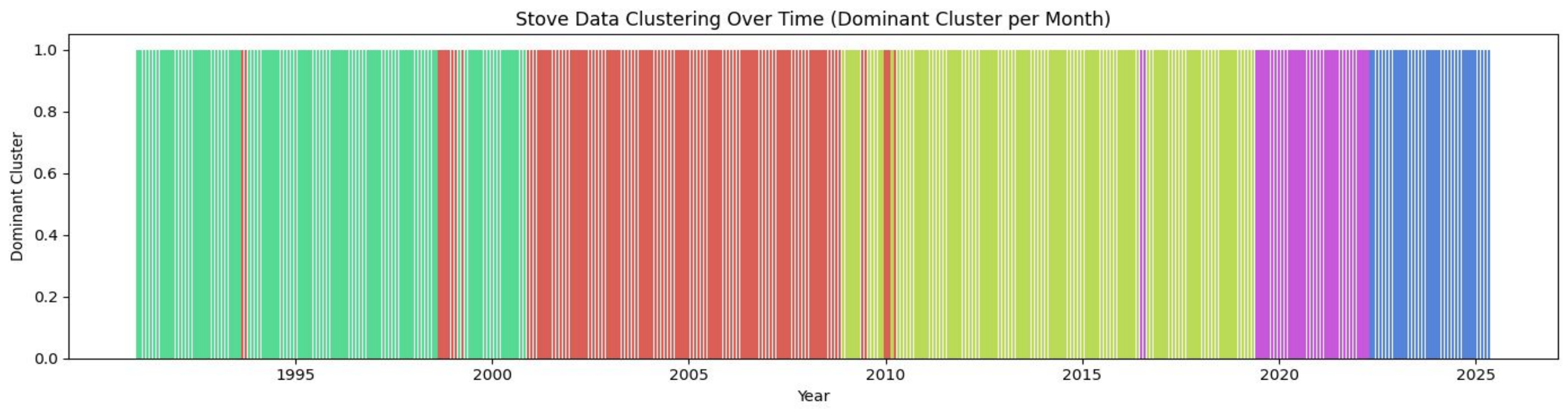
Objective: Time periods of cyclical problems

Clustering

Optimal amount of clusters - 4 or 6?

- ➔ Aim: fit economic cyclical difficulties (6 major)
- ➔ Clustering fitting fairly well after 2008 (financial crisis)
- ➔ IT boom and Housing bubble developing similarly and has difficulty separating

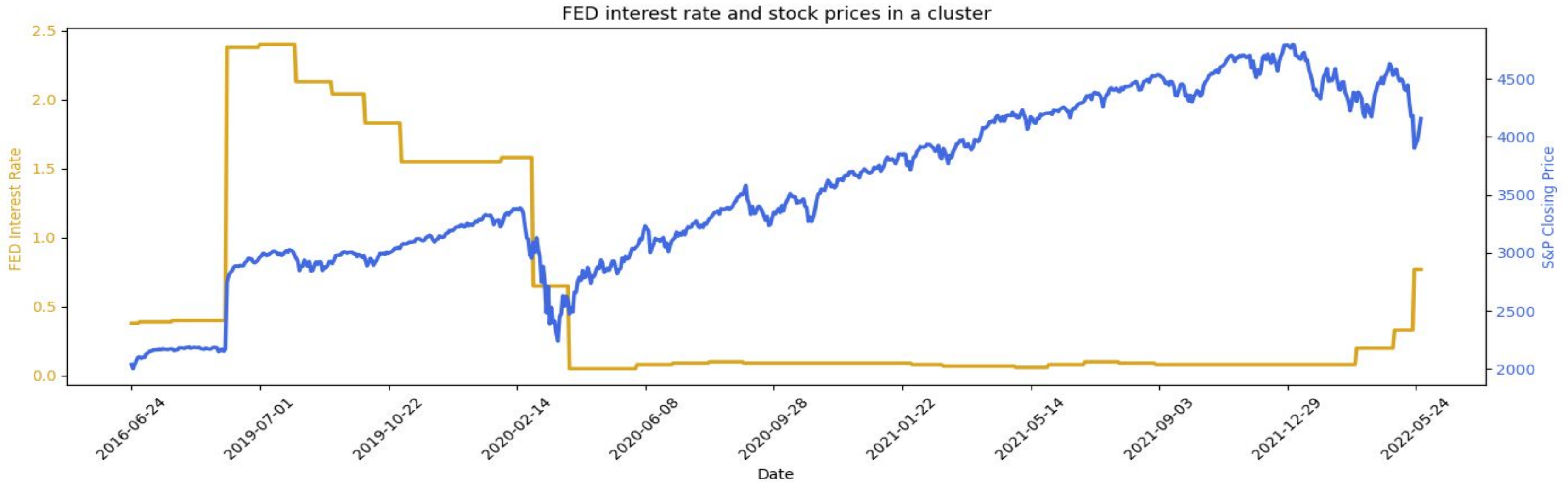




Example finding: Bust

Economy 101: when economy busts, FED corrigates interest rates down to facilitate prices eventually can go up again.

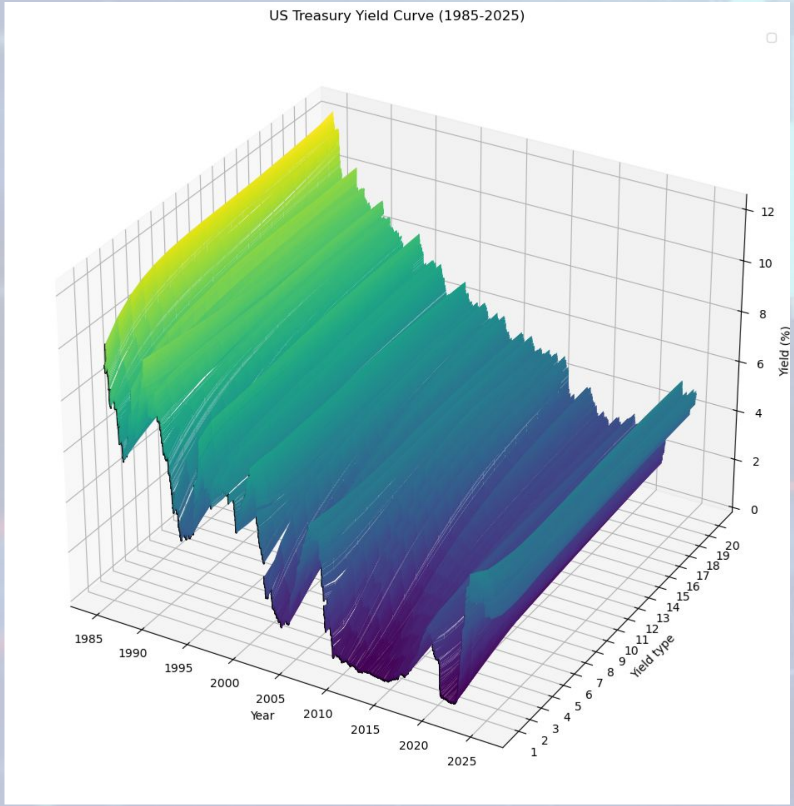
Example: Covid-recession

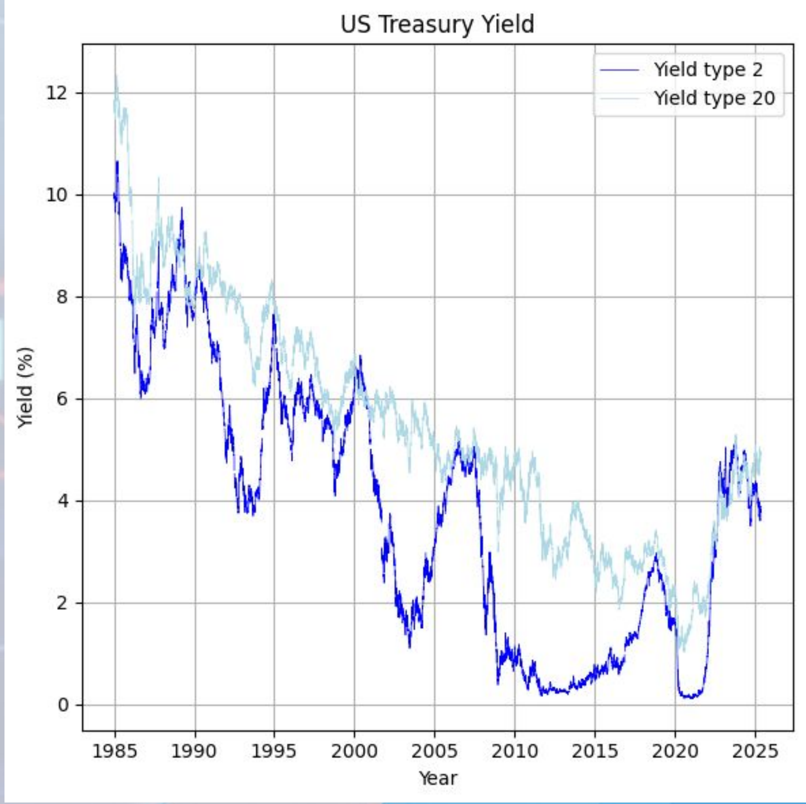


Predicting yields - FED

Target: Yield type 2 and yield type 20
Objective: Performing better than baseline model

Predicting yields - the data

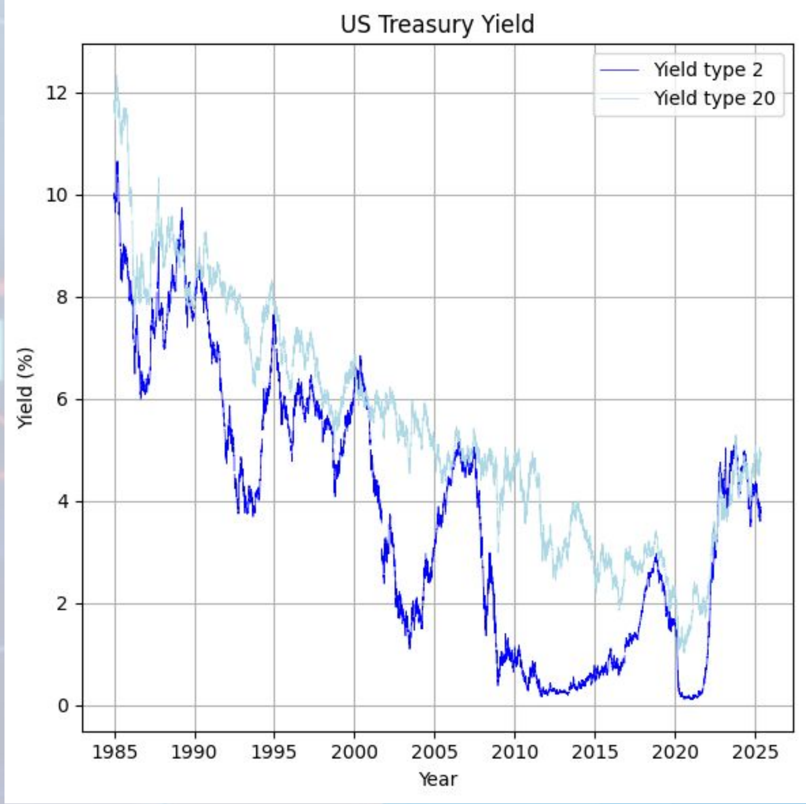




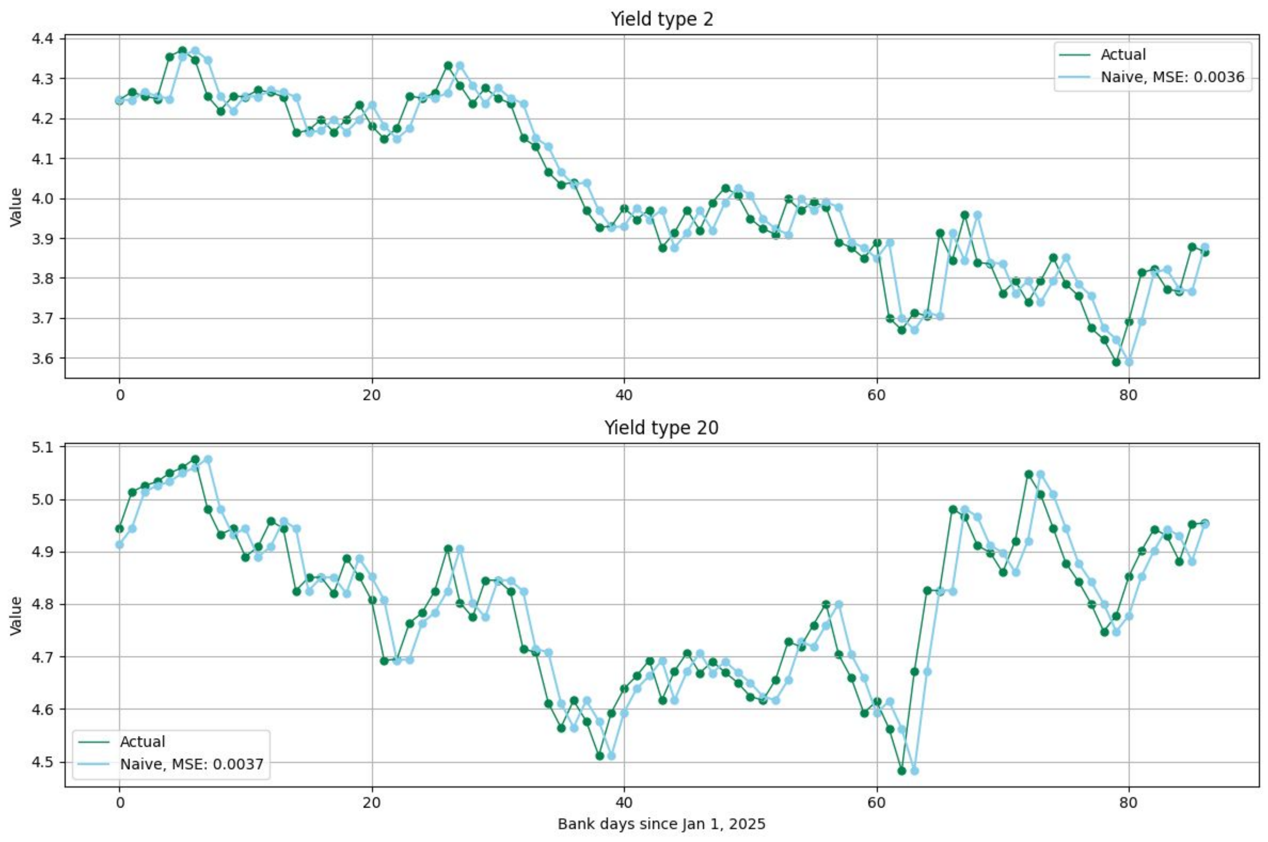
Predicting yields - the data

Date	Yield type 2
2025-01-01	4.2189
2025-01-02	4.2462
2025-01-03	4.2453
2025-01-04	4.2657
2025-01-05	4.2549

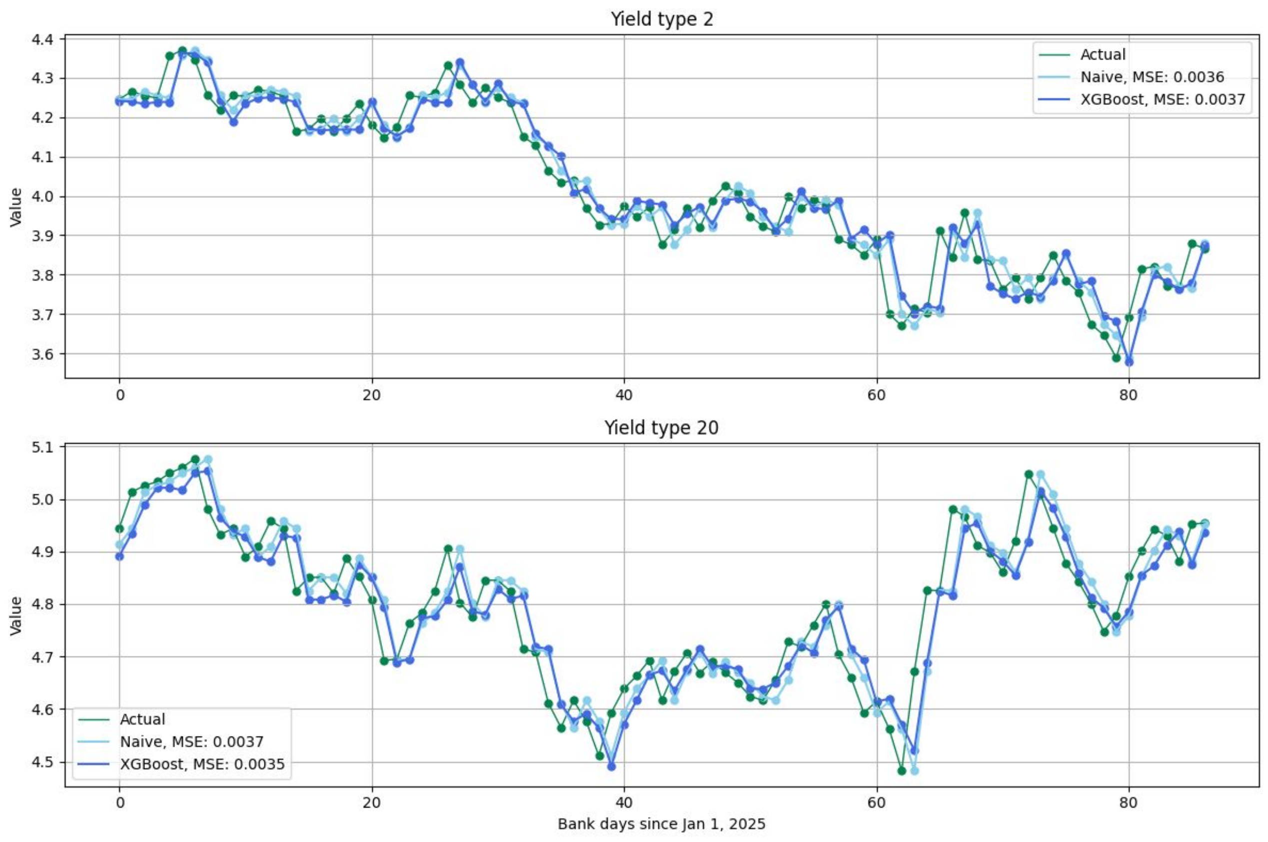
Target	Lag 1	Lag 2	Lag 3	Lag 4
4.2189	…	…	…	…
4.2462	4.2189	…	…	…
4.2453	4.2462	4.2189	…	…
4.2657	4.2453	4.2462	4.2189	…
4.2549	4.2657	4.2453	4.2462	4.2189



Predicting yields - baseline model



Predicting yields - XGBoost



Model architecture

Number of boosting rounds = 1000
Early stopping = 20

Hyperparameters*

Yield type 2:

- number of lags = 30
- learning rate = 0.2
- max depth = 2

Yield type 20:

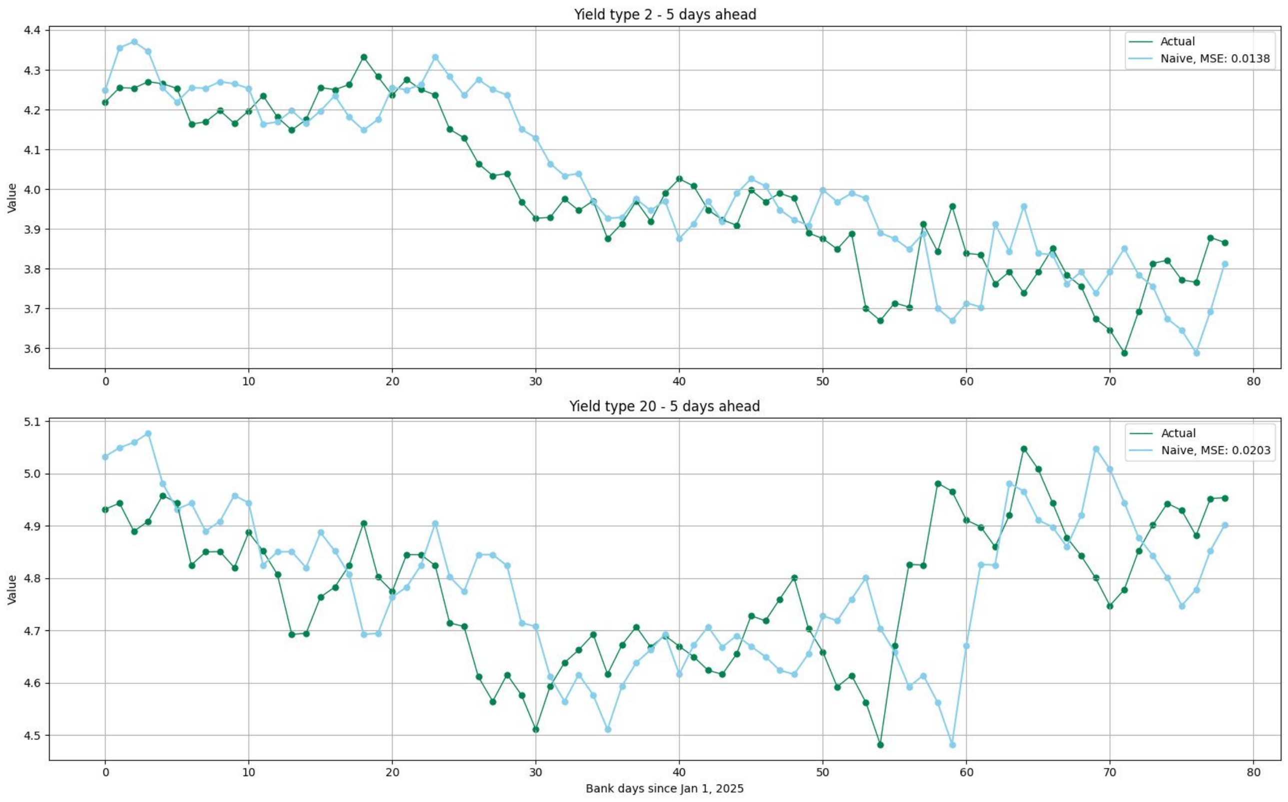
- number of lags = 103
- learning rate = 0.05
- max depth = 3

Take aways

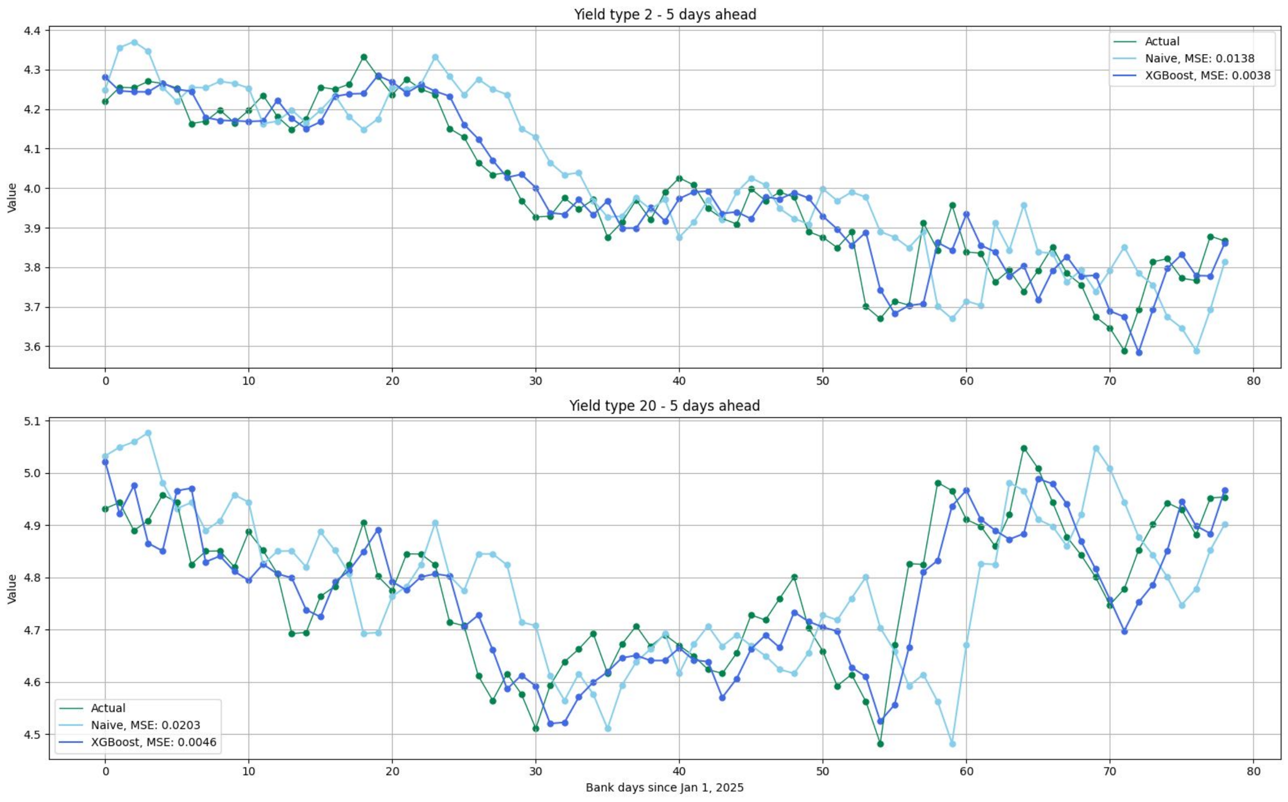
- unable to outperform baseline
- short term yields predicted with fewer lags

**Optimized with Bayesian optimization. See appendix for further details*

Predicting yields 5 days ahead - baseline model



Predicting yields 5 days ahead - XGBoost



Model architecture

Number of boosting rounds = 1000
Early stopping = 20

Hyperparameters*

Yield type 2:

- number of lags = 455
- learning rate = 0.01
- max depth = 5

Yield type 20:

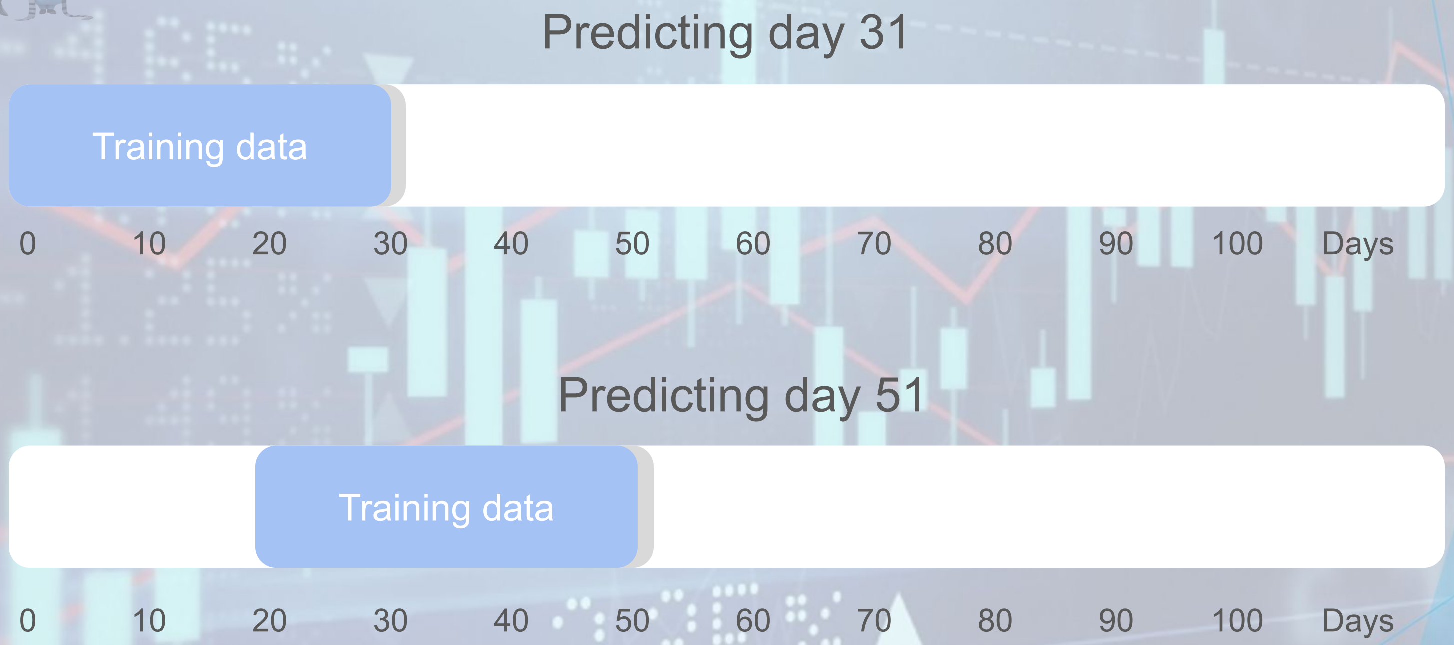
- number of lags = 398
- learning rate = 0.23
- max depth = 4

Take aways

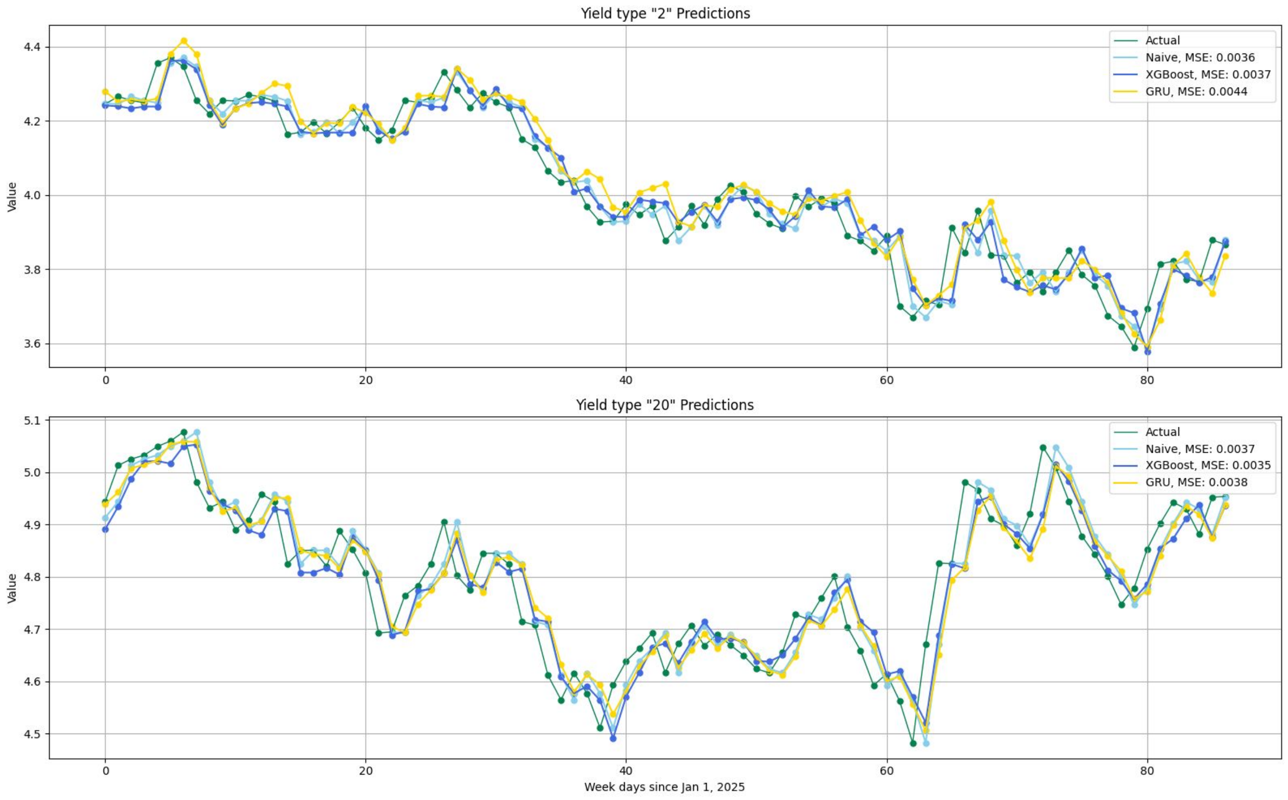
- able to outperform baseline
- more complicated models than the former

**Optimized with Bayesian optimization. See appendix for further details*

GRU - Sliding Window



Predicting next day yields - GRU



GRU – Crash Course

(*Gated Recurrent Unit*)

- A type of Recurrent Neural Network (RNN)
- Similar to LSTM but with a simpler architecture
- Remember previous inputs for future predictions

➔ Able to capture time-dependencies

Model architecture

epochs: 30

batch: 32

GRU layers: (128, 64)

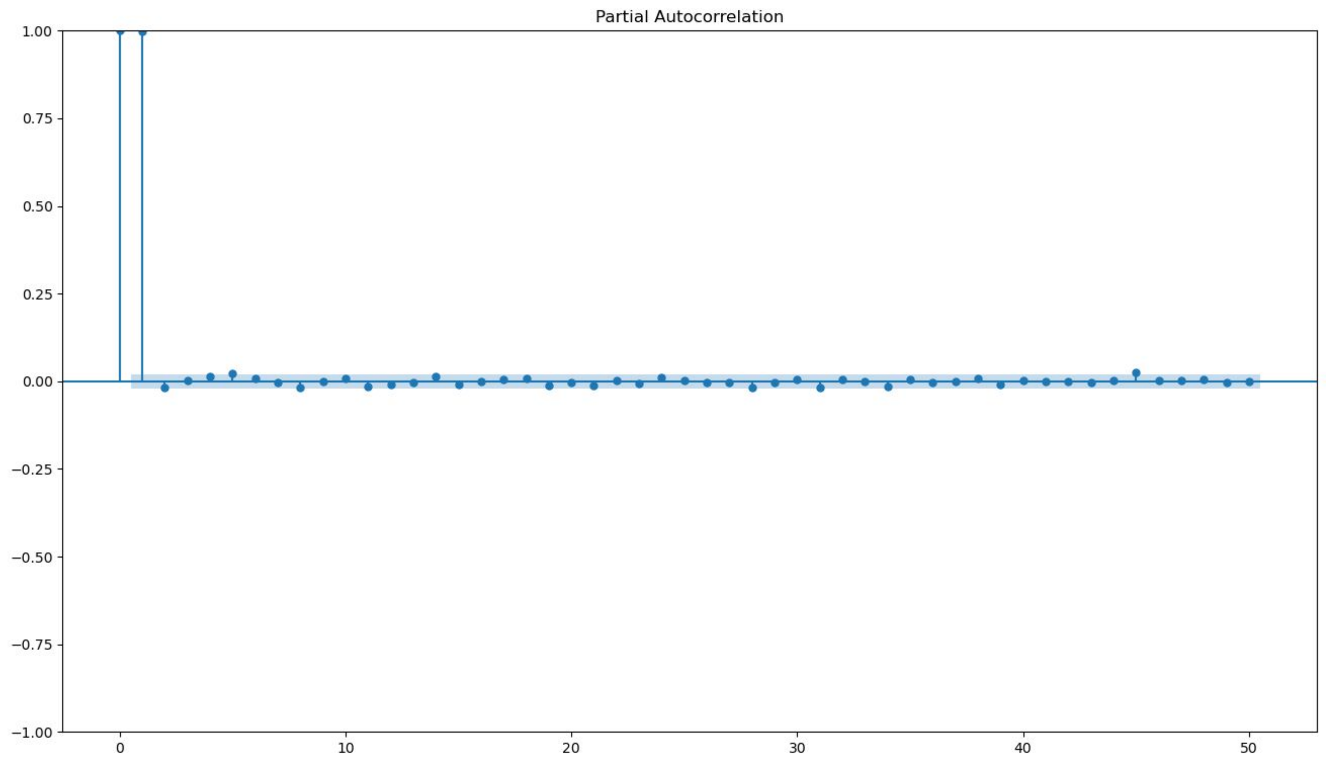
Activation: LeakyRelu (alpha = 0.1)

Take aways

Performing worse than baseline

Not able to capture an underlying pattern of our data …

Predicting next day yields - Evaluation of GRU



Partial Autocorrelation

- Shows the correlations within different lags
- 1-2 lags are significant
- Short term dependency
- Most useful information is the recent observations

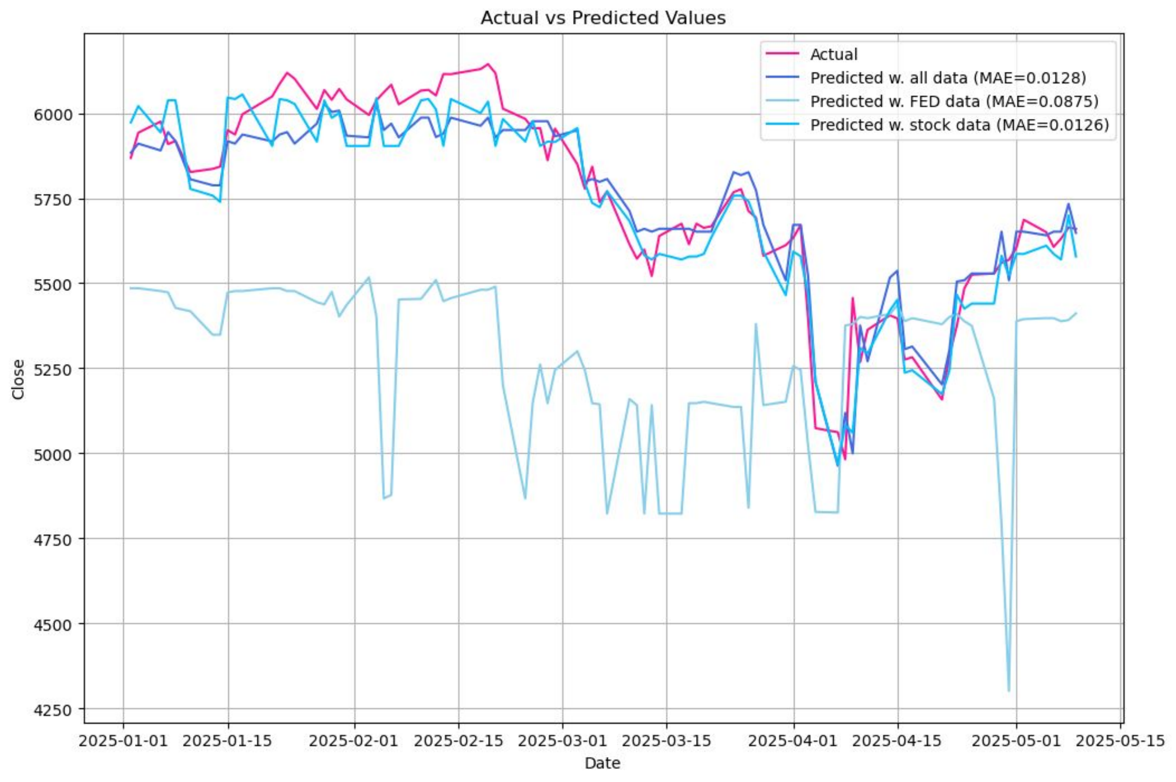
Conclusion

- GRU is not ideal for predicting on out type of data
- Maybe a simpler model would be better. Hence XGB

Predicting stock prices - S&P 500

Target: Closing price

Predicting S&P 500 - Predicting test set (2025)



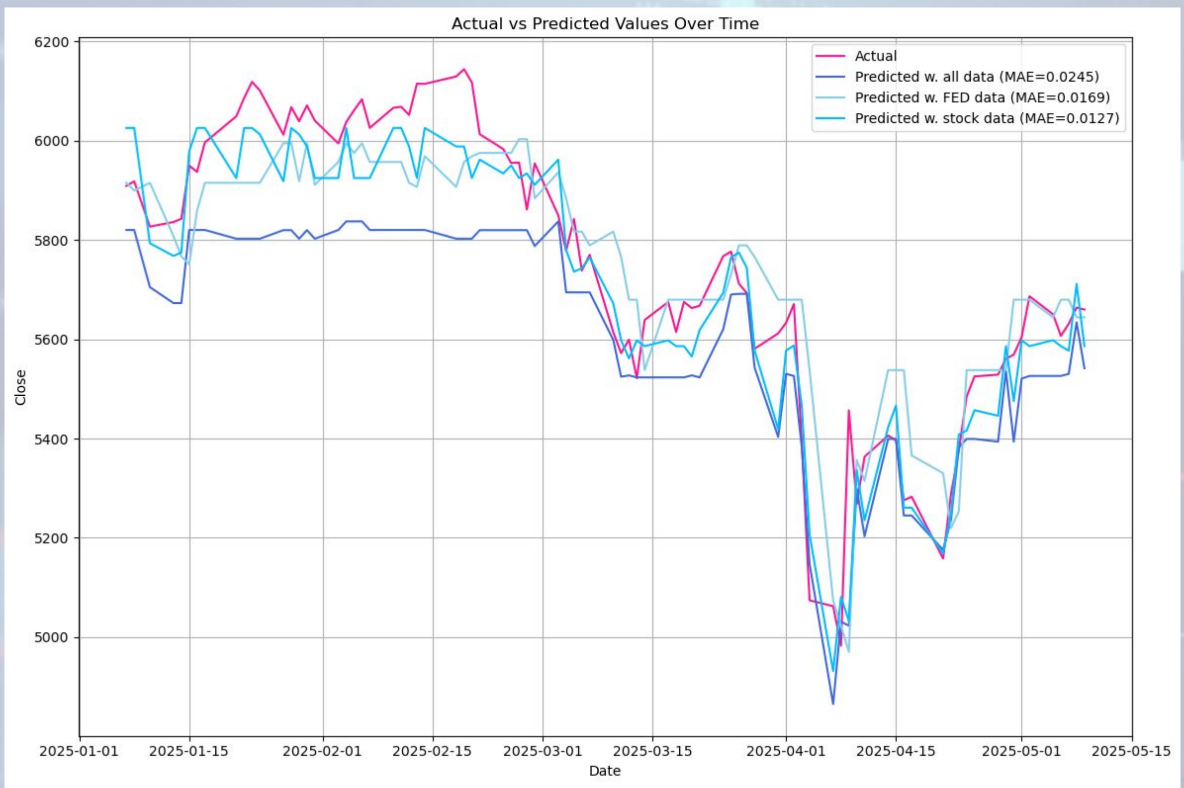
Model architecture:

- ➔ XGboost
- ➔ Booster: `gbtree`
- ➔ Default hyper parameters

Take aways:

- ➔ Generally good fit
- ➔ FED data can't predict level but captures some dips in the stock price
- ➔ 'Low' and 'High' are the most important features evaluated using SHAP values

Predicting S&P 500 - Predicting test set (2025)



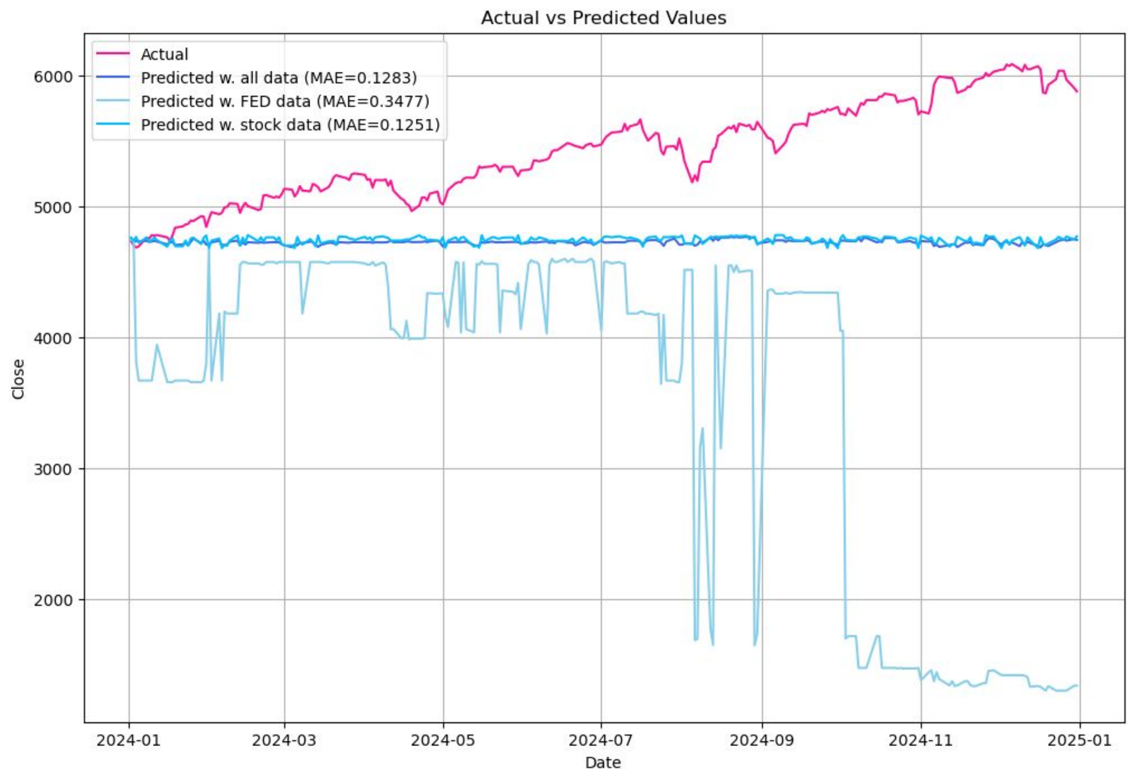
Model architecture:

- ➔ XGboost
- ➔ Time series, 1,2,3 day lag
- ➔ Booster: `gbtree`
- ➔ Default hyper parameters

Take aways:

- ➔ Overall a well fit w. stock data
- ➔ FED data creates noise
- ➔ ‘Low’ and ‘High’ are the most important features evaluated using SHAP values

Predicting S&P 500 - Predicting year 2024



Model architecture:

- ➔ XGboost
- ➔ Booster: `gbtree`
- ➔ Default hyper parameters

Take aways:

- ➔ Single year like year 2025
- ➔ Why is the predictions in year 2024 so much worse than 2025?
- ➔ Perhaps the stock market is too volatile to predict

Predicting S&P 500 - Expanding Window Cross-Validation

Predicting year 2000

Training data

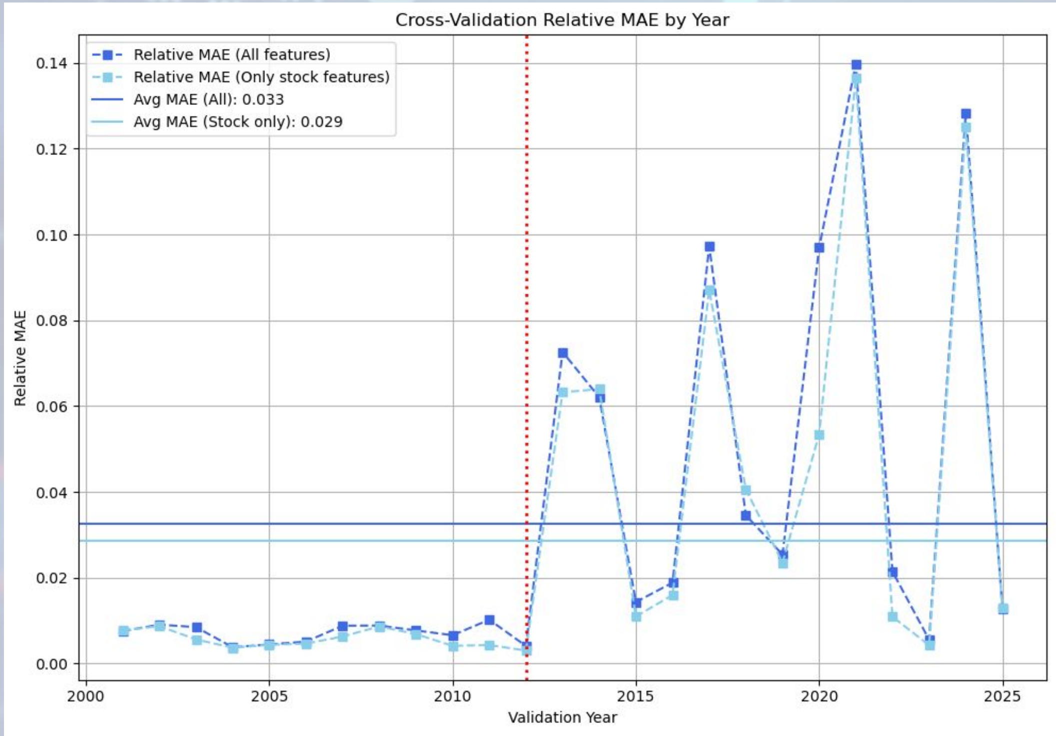
1991 2000 2010 2015 2020 2025

Predicting year 2015

Training data

1991 2000 2010 2015 2020 2025

Predicting S&P 500 - Expanding Window Cross-Validation



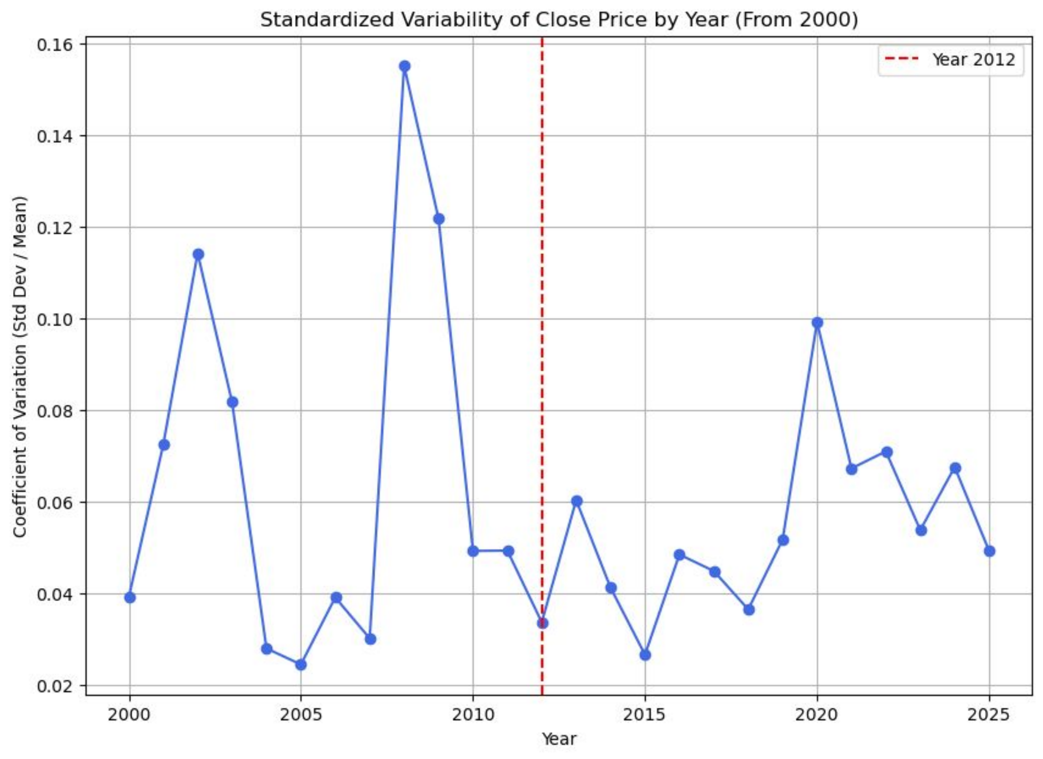
Model architecture:

- ➔ XGboost
- ➔ Booster: gbtree
- ➔ Default hyper parameters

Take aways:

- ➔ Unstable relative MAE's from 2012, why?
- ➔ Change in the structure of the target
- ➔ Optimizing the model using random search *(see appendix)* doesn't improve the loss

Predicting S&P 500 - Expanding Window Cross-Validation



Standardized variability:

$$Standardized\,Variance_y = Variance_y / Mean_y$$

For each year y

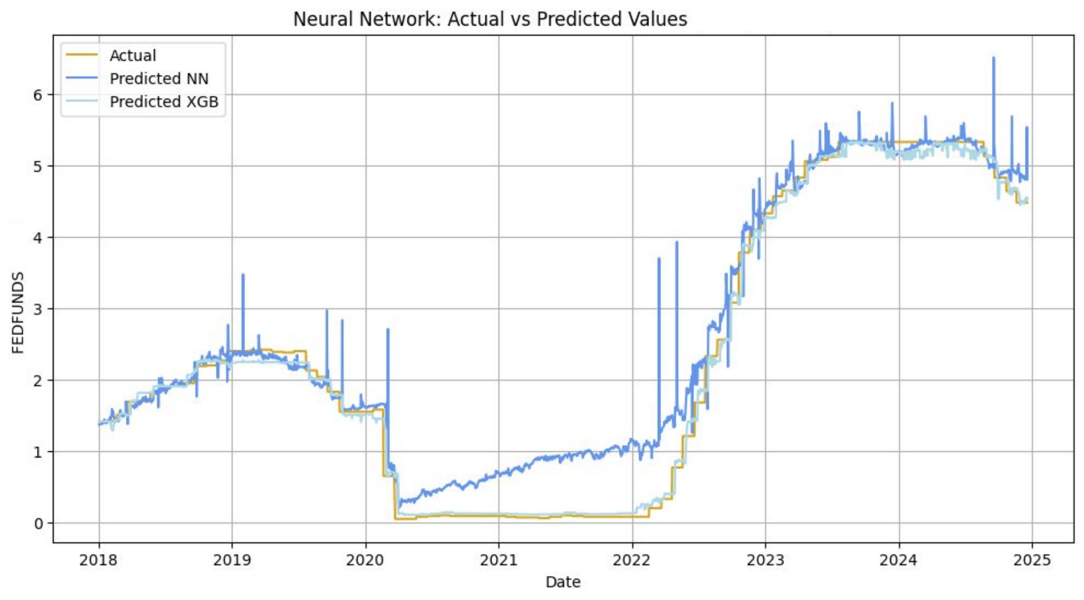
Take aways:

- ➔ Change in variance structure in 2012
- ➔ Overall high variance
- ➔ A lot of exogenous factors affect the stock prices

Predicting the interest rate - FED

Target: Interest Rate

Predicting the interest rate - Using train-val-test split



Boosting decision tree

- ➔ XGBoost
- ➔ Default HP
- ➔ MAE: 0,5421

Neural network

- ➔ Sequential
- ➔ Default HP (3 layers)
- ➔ MAE: 1,8036

Take aways

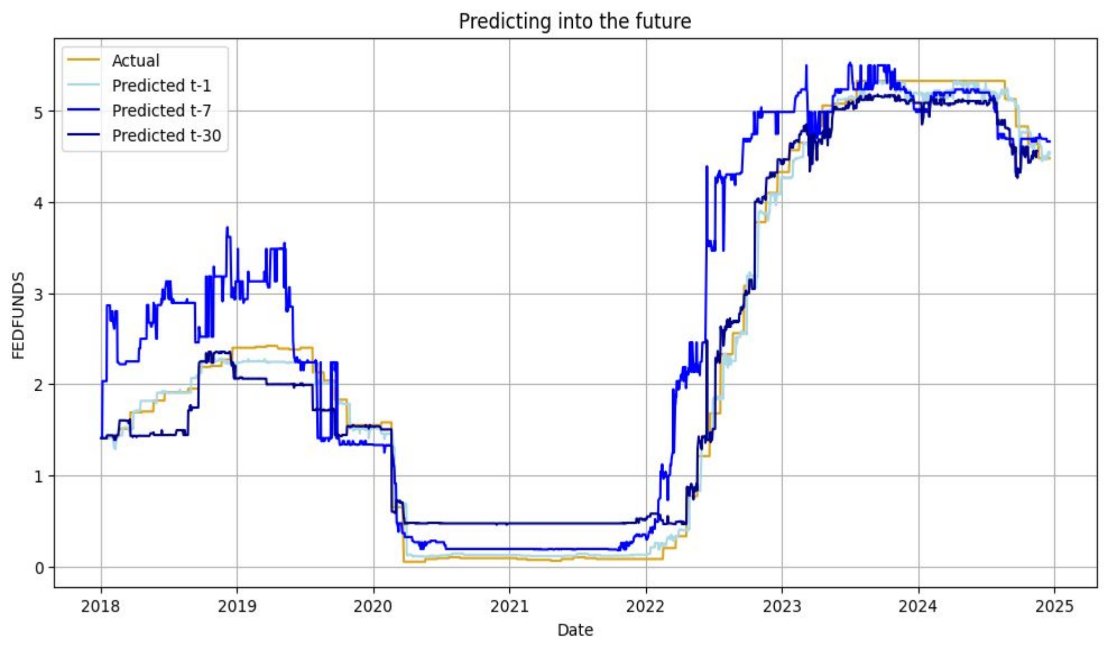
- ➔ Interest rates are much easier to model
- ➔ BDT seems to be less volatile in prediction

Predicting the interest rate - Shifting target to predict

input 1	input 2	target
x.1	y.1	z.1
x.2	y.2	z.2
x.3	y.3	z.3
x.4	y.4	z.4

input 1	input 2	target
x.1	y.1	z.2
x.2	y.2	z.3
x.3	y.3	z.4

Predicting the interest rate - Shifting target to predict



Hyperparameter chosen with Bayesian optimization

MAE does not change linearly

- ➔ t-1: 0,0596
- ➔ t-7: 0,4929
- ➔ t-30: 0,1257

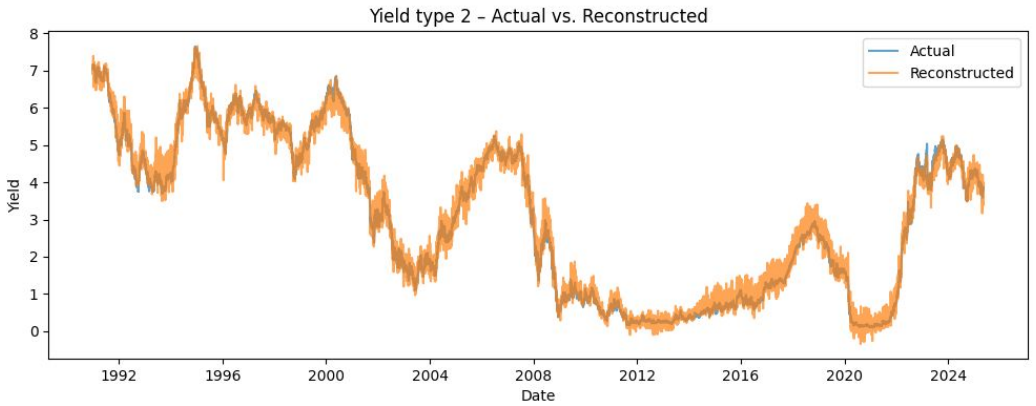
Take aways:

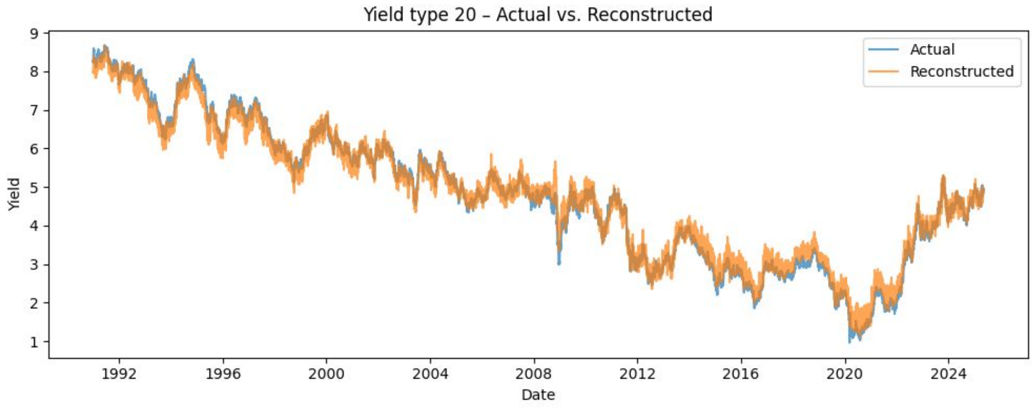
- ➔ Interest rates are possible to model with longer scope

Variational Autoencoder - FED

Target: Rate change dates

Anomaly detection - Variational Autoencoder

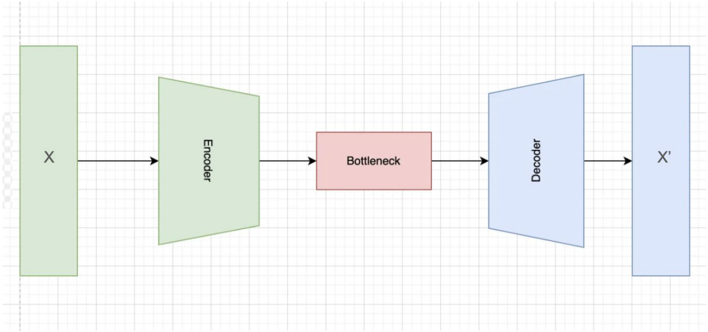




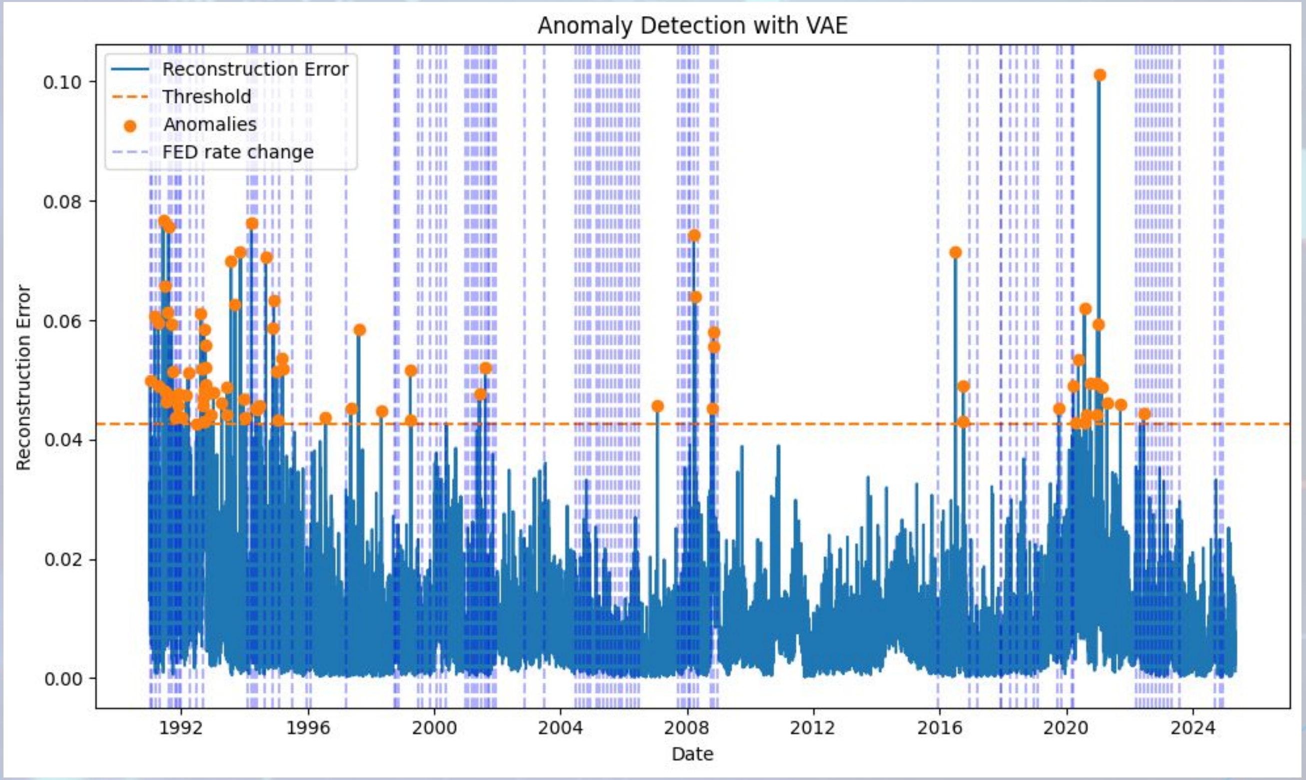
Summary

- 2-dimensional latent space
- *loss = reconstruct loss + $\beta \cdot$ KL-divergens*
- 5 fold cross validation using `sklearn.model_selection.TimeSeriesSplit()`
- 100 epochs w. early stopping (patience = 10)
- Train loss: 653.768 → 18.002
- Validation loss: 158.547 → 34.944

  Achieved at 75 epochs



Anomaly detection - Reconstruction error



Top 1% reconstruction error:

- 86 instances
- 96 changes in FED
- 12 overlapping

Reasons for not detected rate changes:

- 2020-2021: Covid
- 1993-1994: U.S. Budget Agreement, inflation slowdown

Hypergeometric distribution:

$$\mathbb{P}(X = 12) = \frac{\binom{96}{12}\binom{8964-96}{86-12}}{\binom{8964}{86}} = 8.4985 \cdot 10^{-11}$$

Conclusion - Did we learn any links?

US yield data:

XGBoost outperforms the baseline for longer horizons.
GRU is not very well performing.

Interest rate data:

Rate is a fairly easy target to predict with XGBoost.
VAE anomaly detection captures interest rate decision dates.

Stock data:

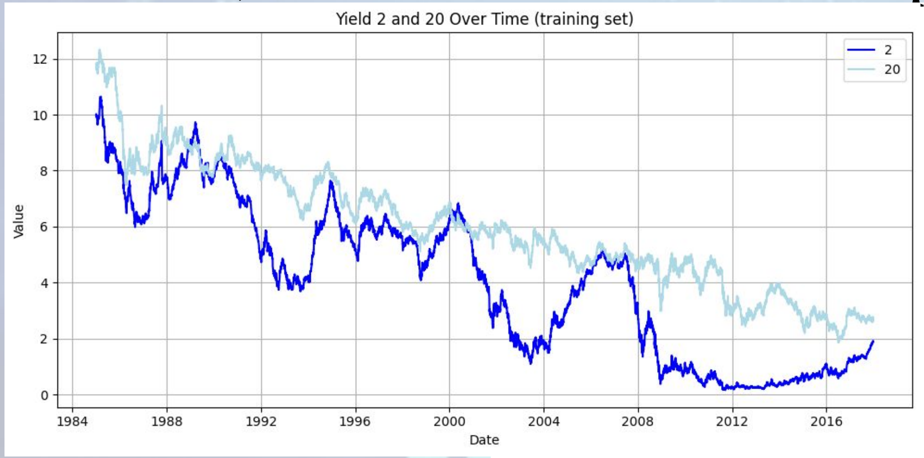
Closing price is primarily dependent on intraday prices.
Stock market in general is too volatile to predict.

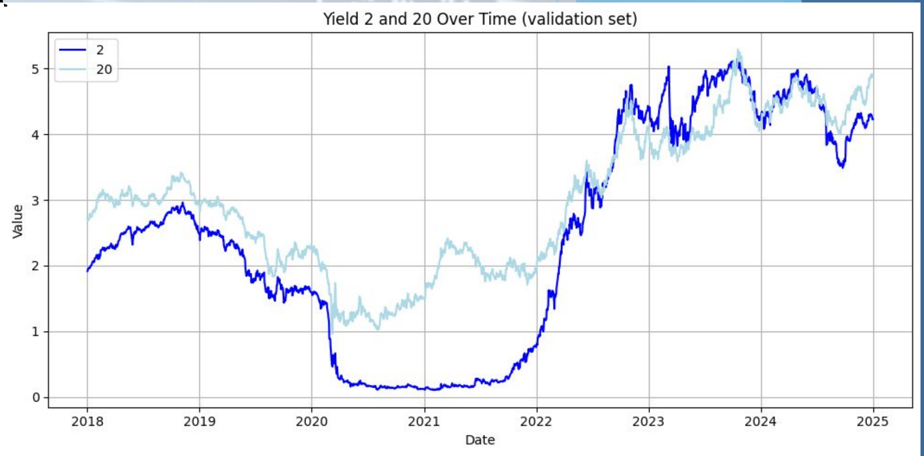
“An economist is an expert who will know tomorrow why the things he predicted yesterday didn't happen today.”

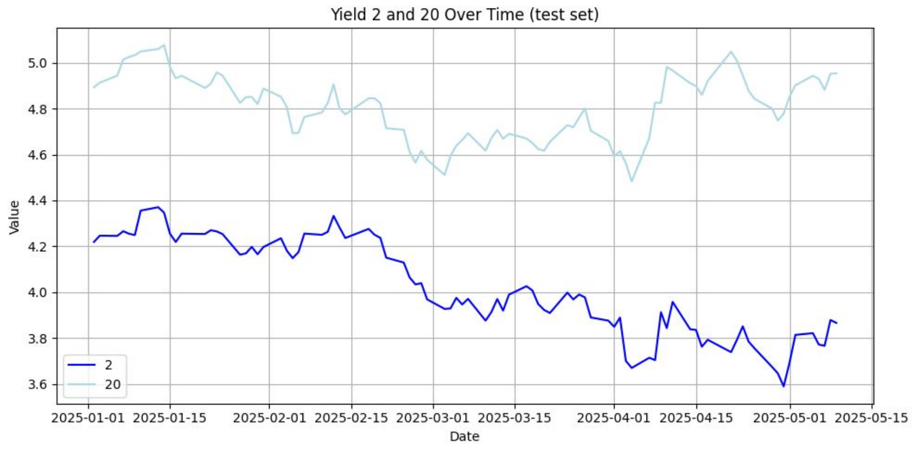
- Laurence J. Peter

Appendix

Train, validation and test split





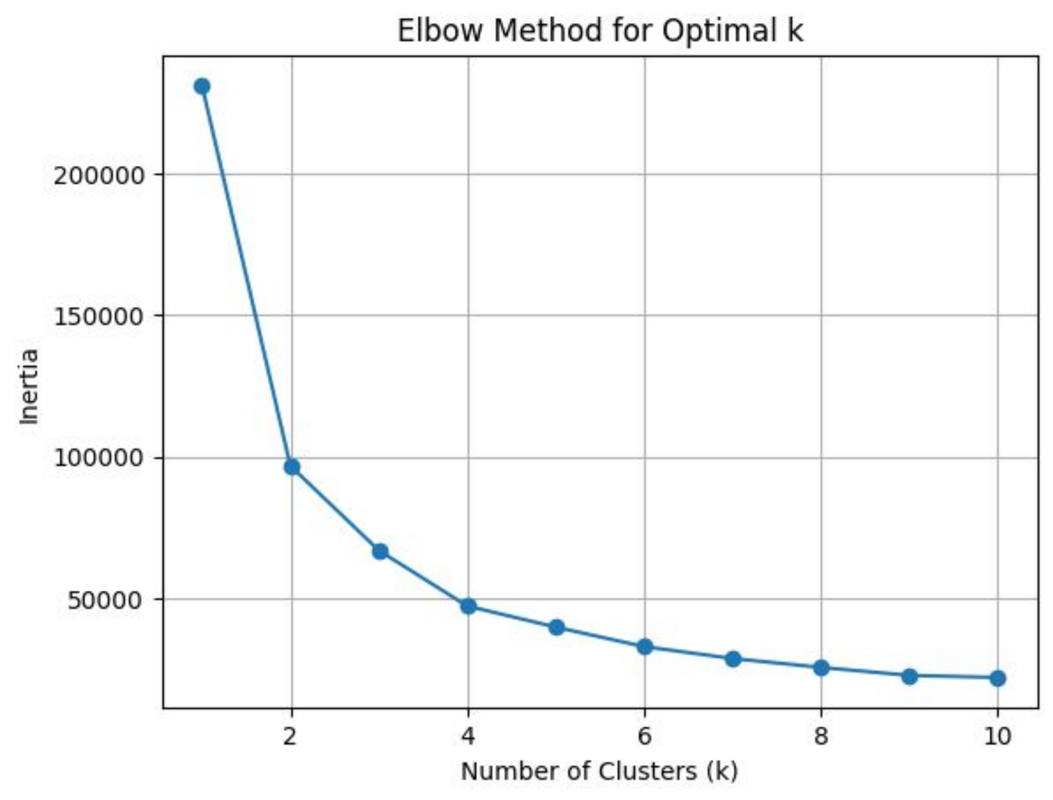


Cluster - Elbow plot

Choosing optimal k

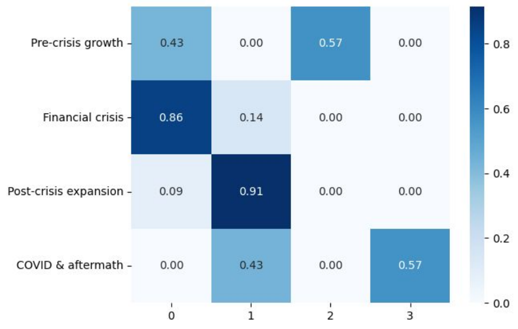
Could argue optimal amount of clusters were 2, 4 or 6. We chose 6 as it aligns with economic theory.

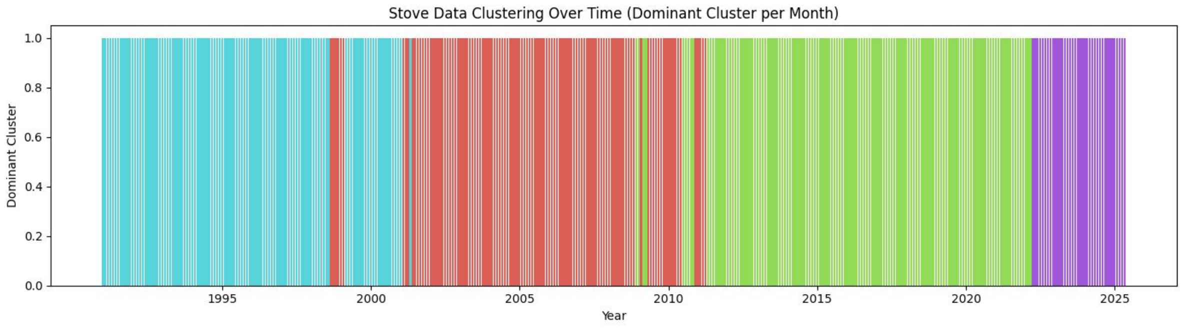
We tried fitting 4 clusters, to see how well it fits reality.

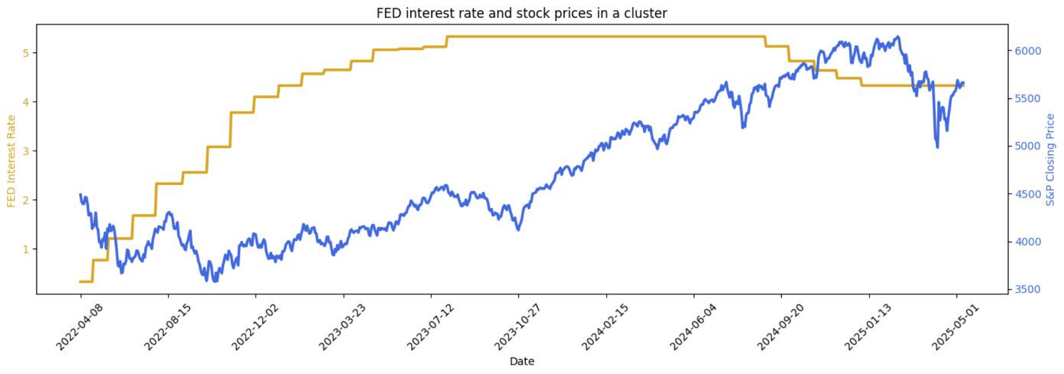


4 clusters - optimal?

Attempt of fitting and “constructing” 4 major cyclical developments.







left: Example of boom within a cluster in the 4-cluster split. Here the economy is in the end of a boom, hence the interest rates increase and stock prices starts to drop.

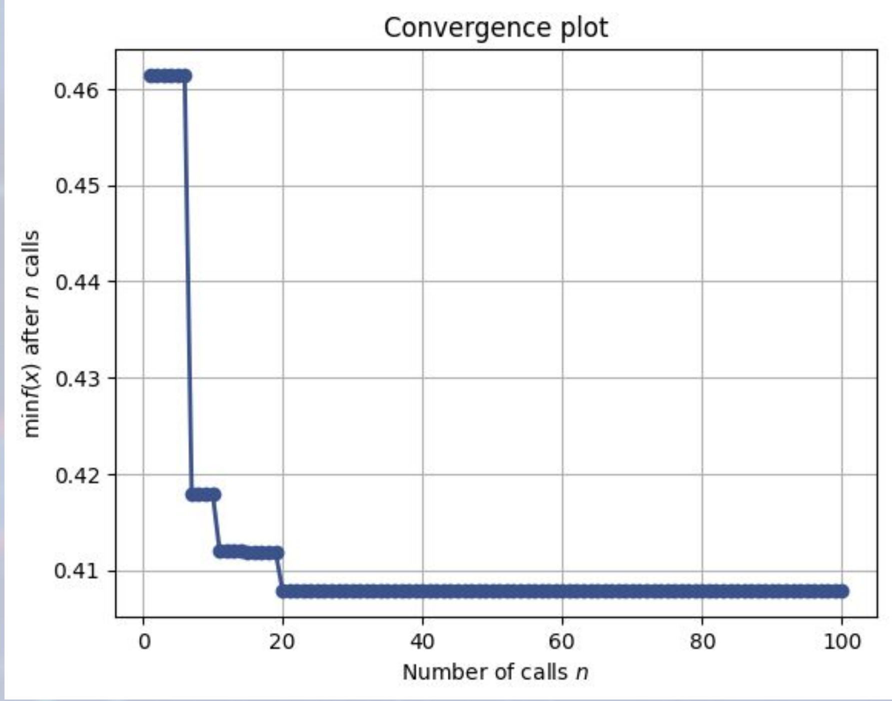
Increasing prices provoke the rates to drop in the end, and the economy in 2024 enters recession.

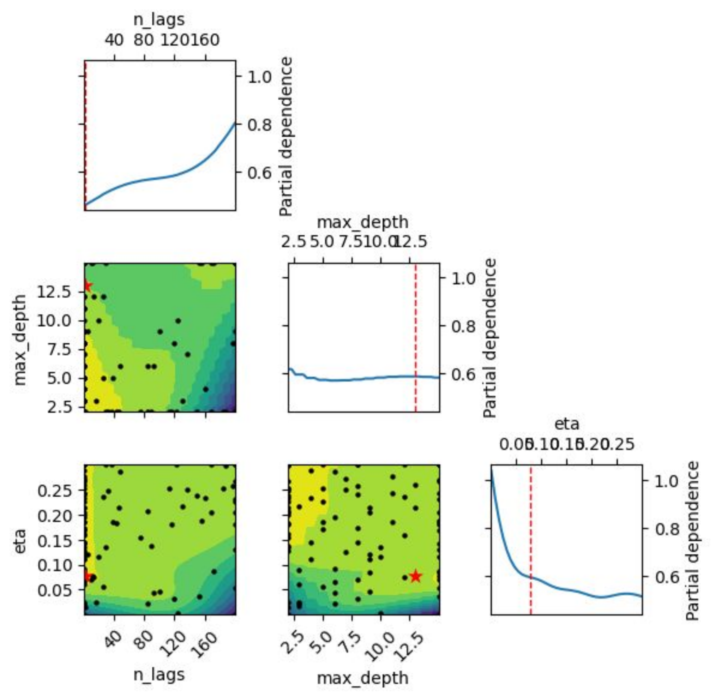
Hyperparameter tuning - Yield type 2, 1 day

Search space:

- number of lags: (1, 200)
- learning rate: (0.001, 0.3)
- max depth: (2, 15)

100 iterations



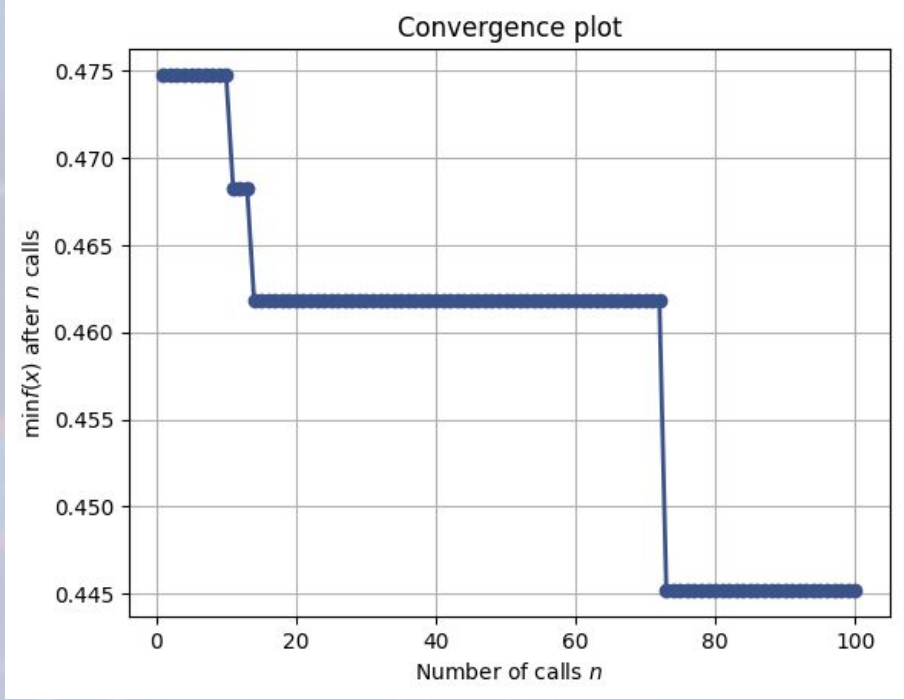


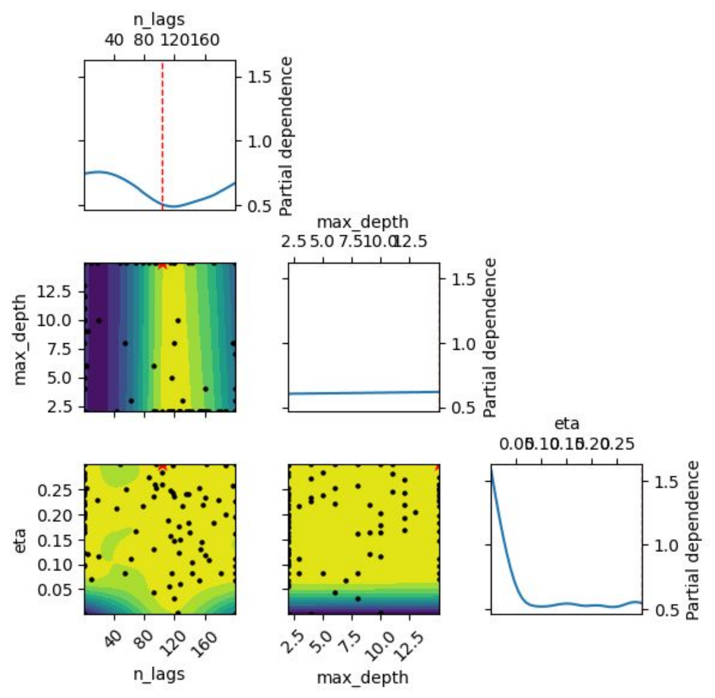
Hyperparameter tuning - Yield type 20, 1 day

Search space:

- number of lags: (1, 200)
- learning rate: (0.001, 0.3)
- max depth: (2, 15)

100 iterations

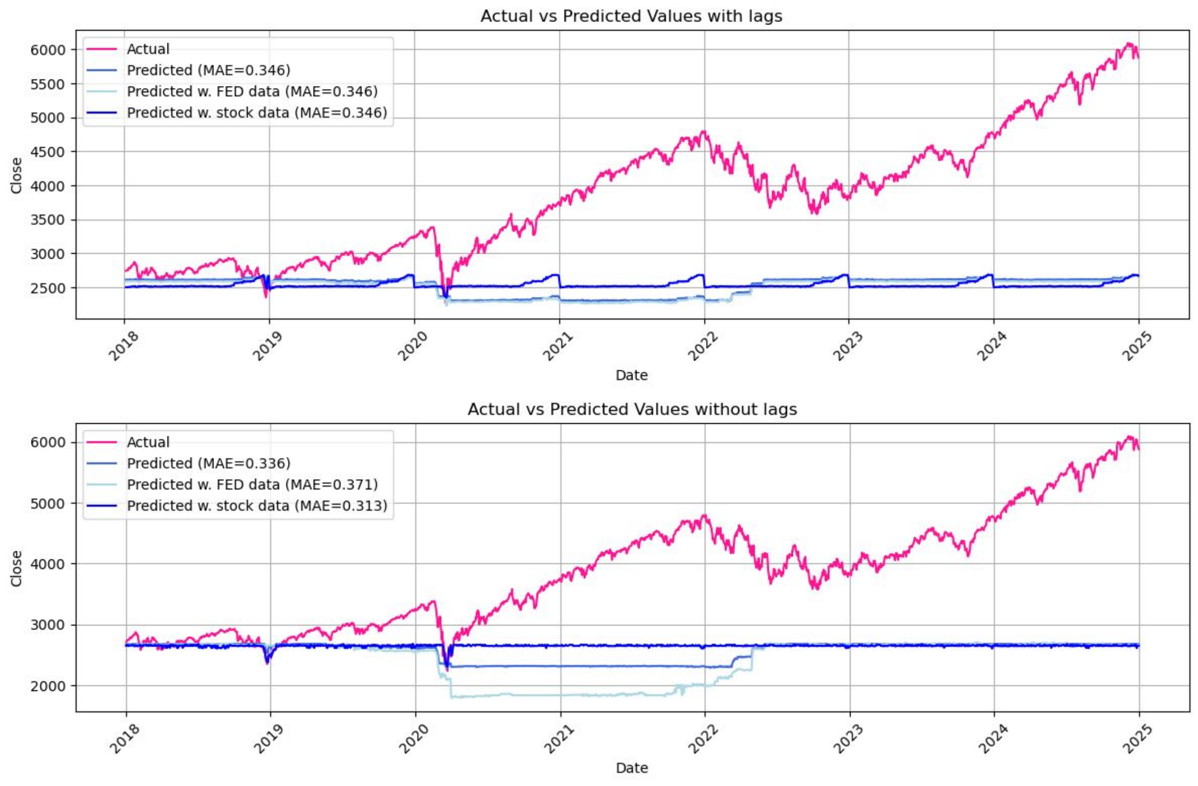




Predicting yields - Evaluation of XGBoost

Yield	Horizon	Baseline MSE (95% CI)	XGBoost MSE (95% CI)
2	1 day	0.0036 (0.0025 - 0.0051)	0.0037 (0.0024 - 0.0051)
20	1 day	0.0037 (0.0027 - 0.0051)	0.0035 (0.0025 - 0.0046)
2	5 days	0.0138 (0.0099 - 0.0176)	0.0038 (0.0026 - 0.0054)
20	5 days	0.0203 (0.0132 - 0.0292)	0.0046 (0.0034 - 0.0060)

Predicting S&P 500 - Using train-val-test split



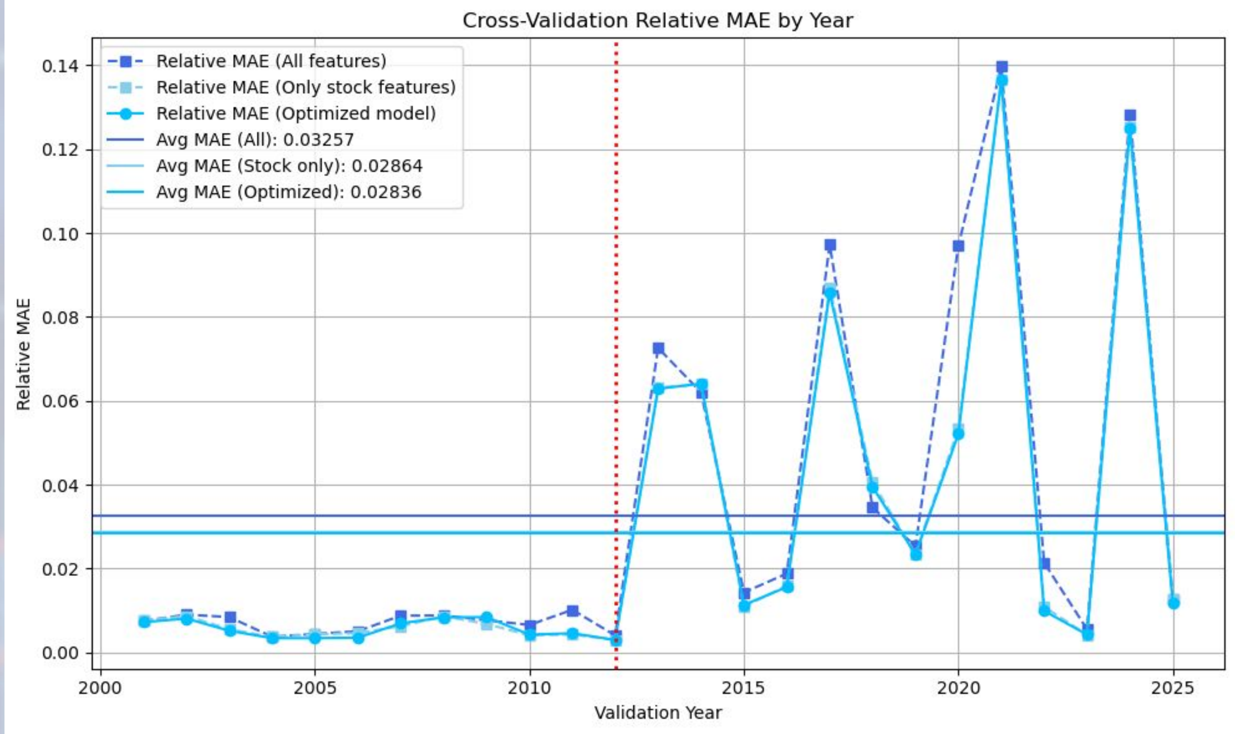
Model architecture:

- ➔ XGboost
- ➔ Booster: `gbtree`
- ➔ Default hyper parameters

Take aways:

- ➔ Using 1, 2 and 3 days lag on target
- ➔ Something is very wrong
- ➔ Can't predict far out into the future with old training data
- ➔ What is the solution?

Optimal expanding window model



Model architecture:

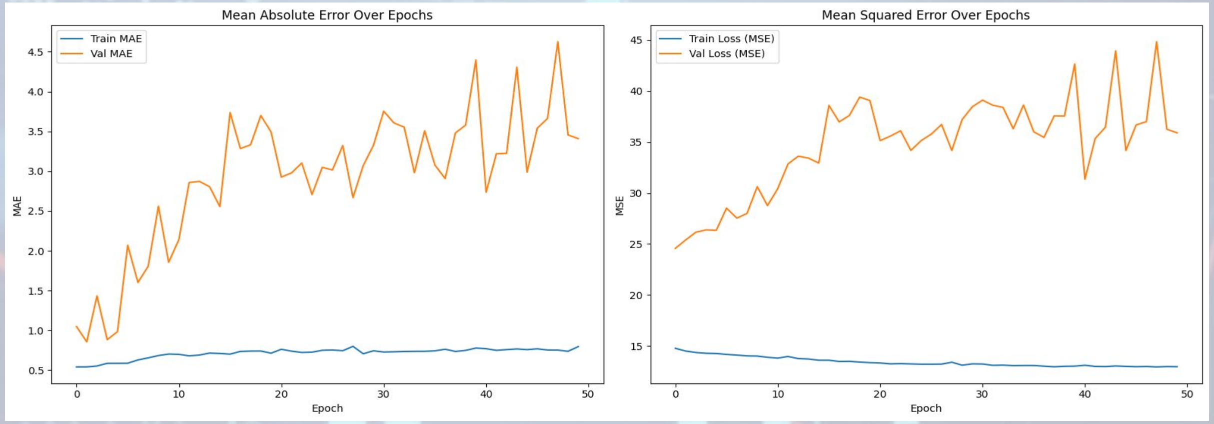
- ➔ XGboost
- ➔ Booster: gbtree
- ➔ Hyper parameters:

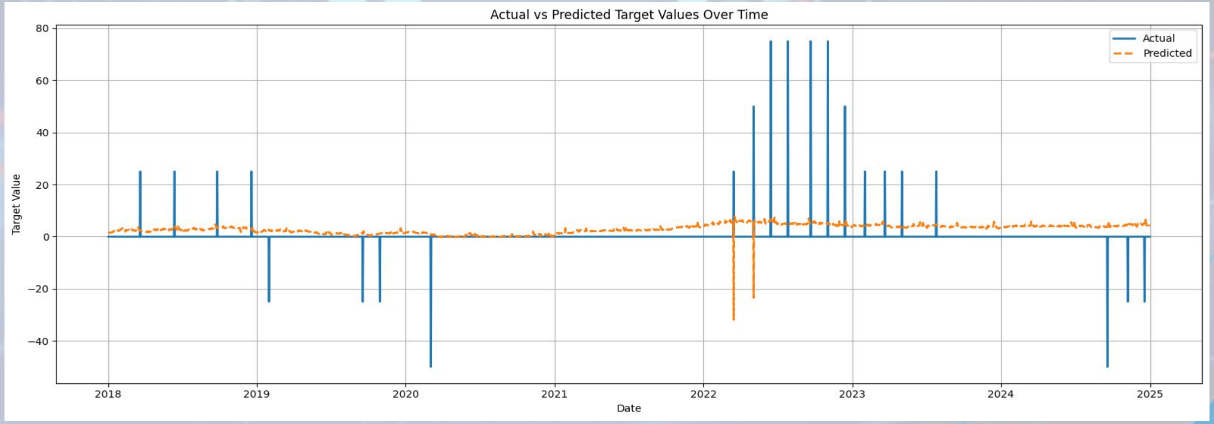
```
'max_depth': 5,
'learning_rate': 0.1478,
'subsample': 0.6858,
'colsample_bytree': 0.8885
```

Optimization:

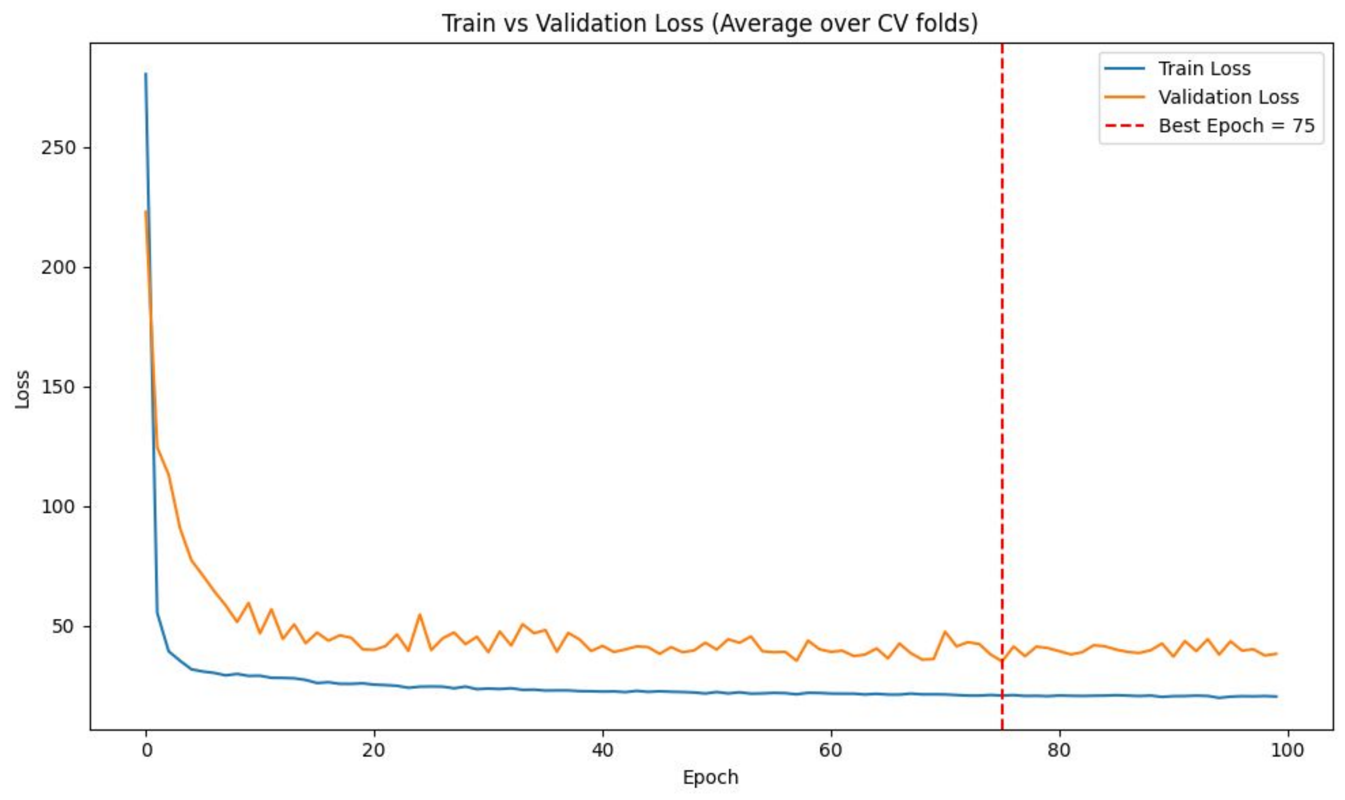
- ➔ Random search
- ➔ 30 trials

Predicting Interest Rate Decision - NN (very bad model)





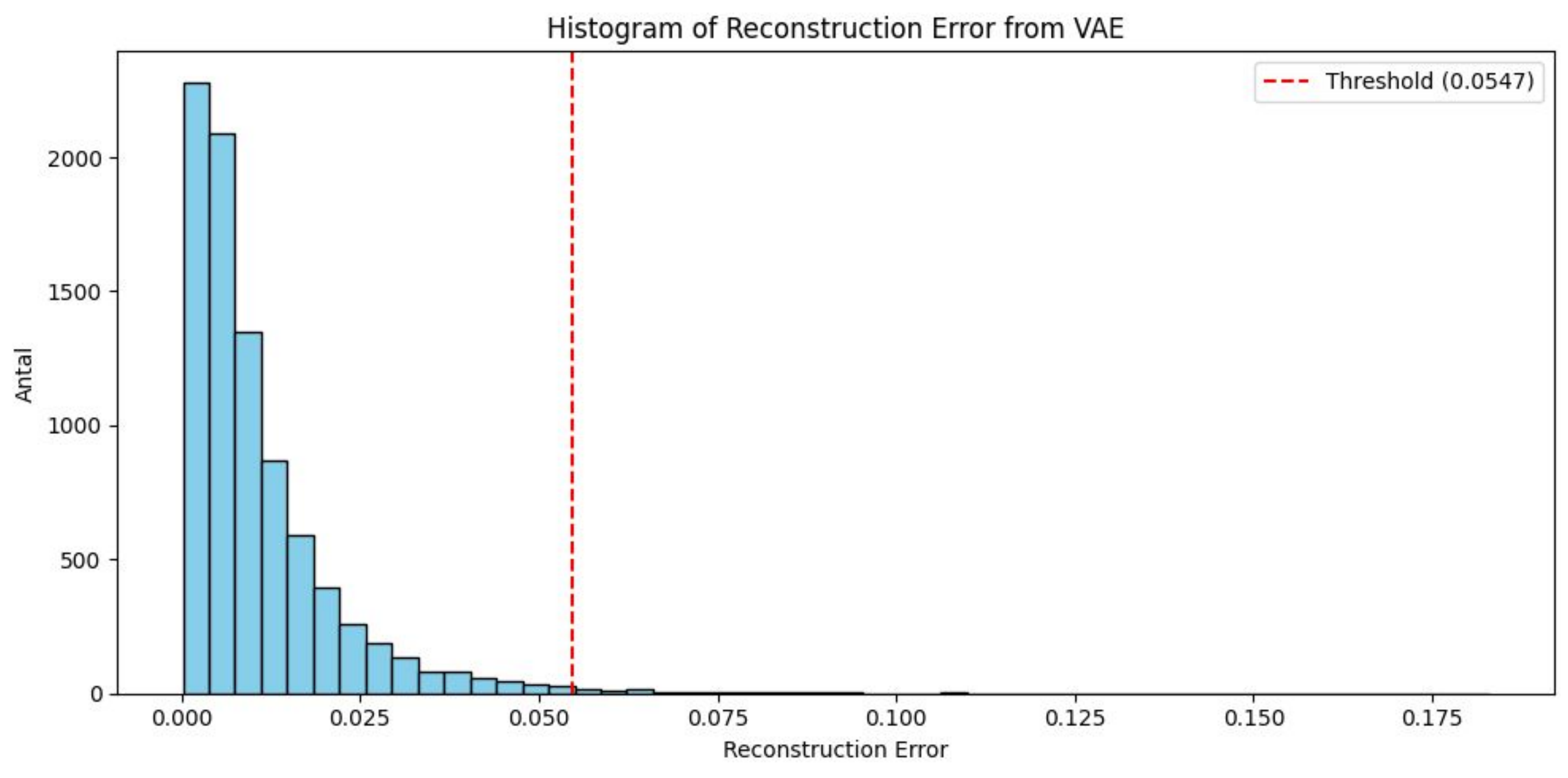
Variational Autoencoder - Train and validation loss



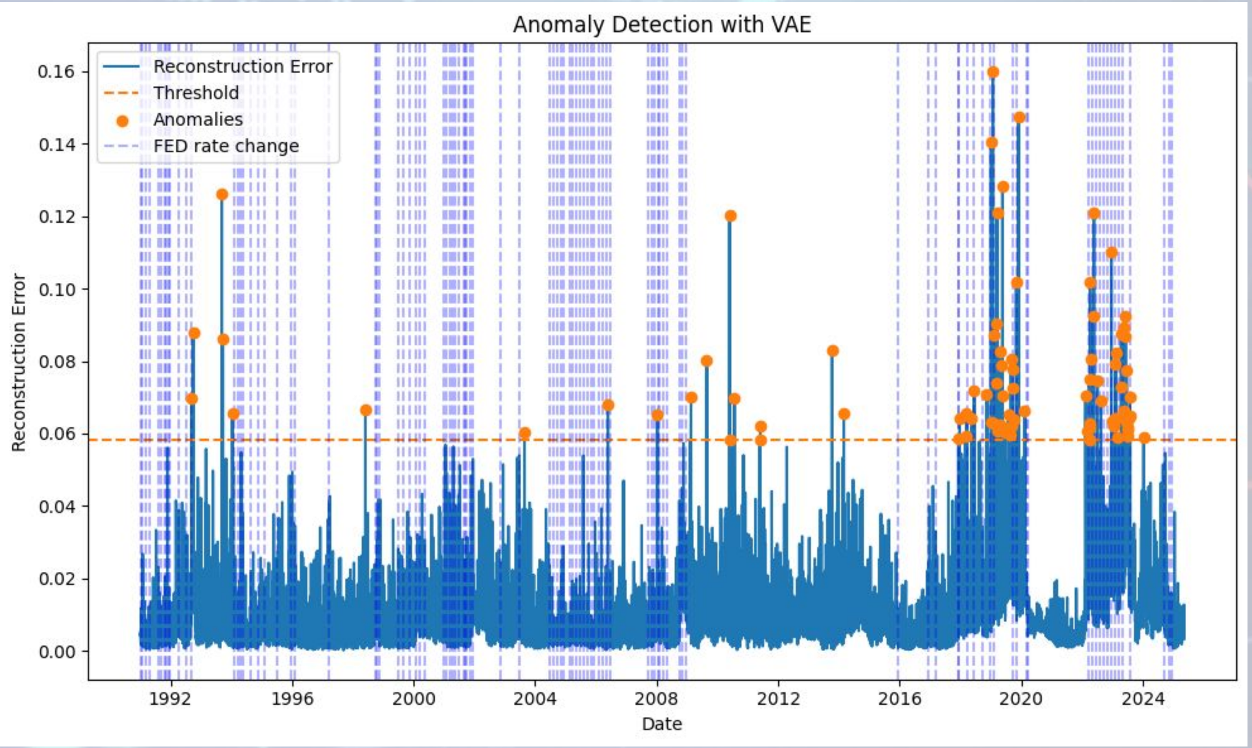
Comments

The plot shows train- and validation loss when training the VAE on 100 epochs, each of batch size 32, with early stopping time with a patience of 10 epochs. Validation loss is computed using 5-fold cross-validation via `sklearn.model_selection.TimeSeriesSplit()` which preserves the temporal structure of the time series data.

Variational Autoencoder - Histogram of Reconstruction error



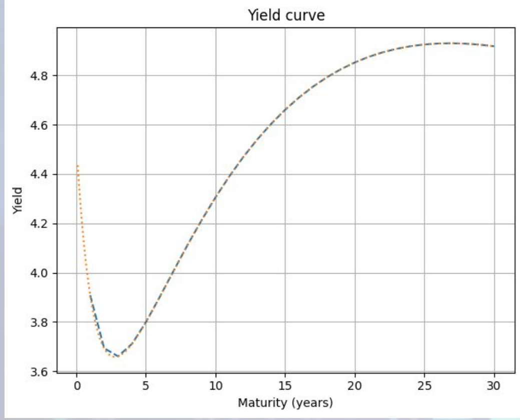
Variational Autoencoder - Another training approach

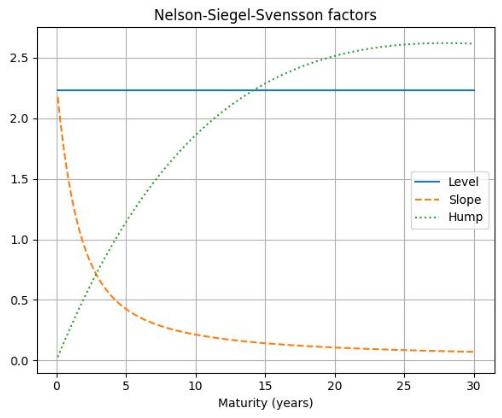


Comments

When training and choosing the number of epochs based on the validation set (01-01-2018 - 01-01-2025) the model clearly generalizes best on the validation set and is thus not good for finding anomalies in the whole time frame.

PCA - Analysis of latent space

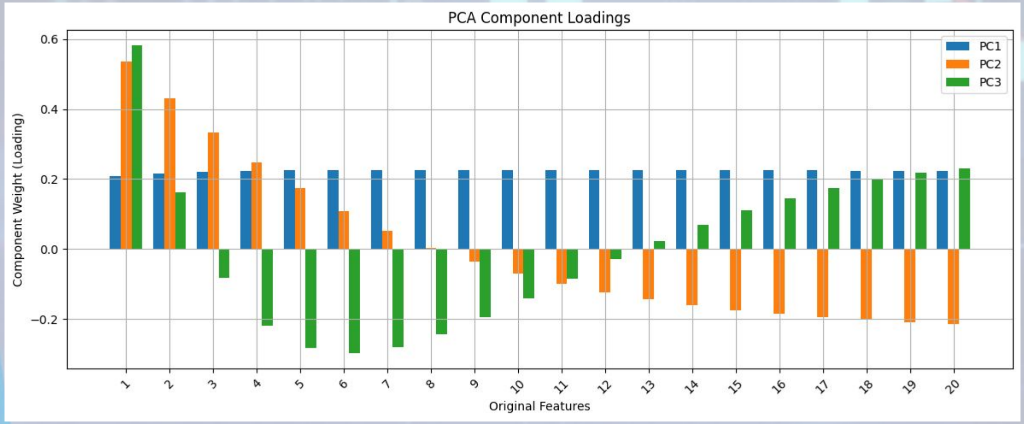




This plot shows the decomposition of the yield curve using the betas from the Nelson-Siegel-Svensson (NSS) model, which is explicitly designed to estimate and smooth yield curves across maturities:

$$r(T) = \beta_0 + \beta_1 \left[\frac{1 - exp(-T/\lambda)}{T/\lambda}\right] + \beta_2 \left[\frac{1 - exp(-T/\lambda)}{T/\lambda} - exp(-T/\lambda)\right]$$

Level = β_0, slope = β_1, hump = β_2.



This plot shows the loadings of the first three principal components (PC1, PC2, PC3) obtained from applying PCA. Each bar represents how much a given maturity contributes to a specific principal component. Interestingly, the shape of the loadings - especially for PC1 and PC2 - resembles the factor structure of the NSS model above.

Sources

Data:

https://www.federalreserve.gov/econres/feds/the-us-treasury-yield-curve-1961-to-the-present.htm

Articles:

https://towardsdatascience.com/multi-step-time-series-forecasting-with-xgboost-65d6820bec39/

https://medium.com/@manthapavankumar11/anomaly-detection-in-time-series-data-with-the-help-of-lstm-auto-encoders-5f8affaae7a7

https://www.frbsf.org/wp-content/uploads/wp07-20bk.pdf