ECON 4305 Term Paper

Forecasting US Interest Rate

Authors:

PANG, Tsz Kin (20763658)

LYU, Bochen (21162069)

SIDHU, Gurjot Singh (20798251)

CHAN, Kin Fai (20758562)

Contents

1 Objective

This report presents the methodology of ECON 4305 group X on making use of machine learning methods to predict interest rate. By leveraging on machine learning, its goal is to make accurate prediction on the future interest rate based on macro data.

2 Abstract

US interest rate has always been one of the most important economic indicators to the global financial market. It impacts the flow of capital, corporate funding cost foreign exchange market, etc. Recently, the improvement in machine learning methods provided an opportunity to apply the machine learning techniques in predicting future interest rate. While we are seeing existing models such as the Vasicek Interest Rate Model (Vašíček O, 1977) and the Taylor'rules (Taylor, 1993). They applied techniques in time series modeling and knowledge of macroeconomics. Nonetheless, machine learning can capture more insignificant factors that help to make interest rate decision. In recent years, academic has tried various methods of applying machine learning to predict interest rate or other macro-economic indicators. Neural networks are proved to be useful in predicting shoter-term interest rate and interest rate spread.

3 Introduction

Every FOMC meeting, investors, businesses, banks, and consumers anxiously wait for the FED's speech that can greatly impact their lives. People try their best at predicting the results in advance as it can give them a huge advantage over others. The Federal Funds Rate decided by the Federal Reserve of the USA has wide-reaching influence in not just USA but also the entire world due to globalisation. It impacts the flow of capital, corporate funding cost foreign exchange market, etc. While machine learning has existed for a while. Improvements in computing hardware, availability of "big data" and new machine learning theories has allowed us to make strides in the field of machine learning. And, has allowed us to make breakthroughs in many different areas such as cancer recognition, speech to text and generative AI. It has also been used in the field of economics to predict various different economic indicators such as inflation, GDP, etc. There have been many attempts at predicting the interest rates because of their immense importance. Methods include using surveys, futures data, and no-change forecasts. Also traditional statistics models like ARIMA, BVAR, GARCH, etc. However these methods may face challenges like having to deal with the complex nature of historical interest rates. Errors in data collection, time lags in data, the interconnected nature of input variables. And, the need for expression of linguistics variables reduces their prediction ability. Additionally, these methods are not good at dealing with the non-linearity, nonstationary and the dynamic environment of interest rate markets. Artificial Neural Networks have shown to work well with nonlinear problems and irregular sampling. We wanted to see if it extends to other machine learning methods. Classification methods we will be looking into Logistic Regression, Decision Tree Classifier, SGD Classifier, SVM and XG Boost Classifier. Logisitc Regression uses a weighted sum of inputs and a bias term and outputs 0 and 1 using the logistic function. If the output is less than 0.5 it belongs to the negative class. And if the output is equal to or more than 0.5 it belongs to the negative class. SGD classifier utilizes SGD training but can behave like logistic regression if we utilize a log-loss function or linear SVM if we use the hinge function. Decision trees utilize tree induction to create purer subsets from the starting node with the help of the greedy algorithm. SVM tries to find a optimum hyperplane to separate two classes. XGboost combines multiple weak decision trees while using regularization.

4 Methodology

This paper will study how to predict the interest rate accurately by the following approach. The first step will involve data cleaning and processing. The second step will be variables grouping to reduce the dimensionality of the predictors. The third steps will involve classifications on whether the signals given by the groups are positively affecting the interest rate. Then, regression will be carried out on predicting the actual variables. Last but not least, we will seek for improvement to improve our model by incorporating more factors and apply reinforcement learning in order to achieve better results.

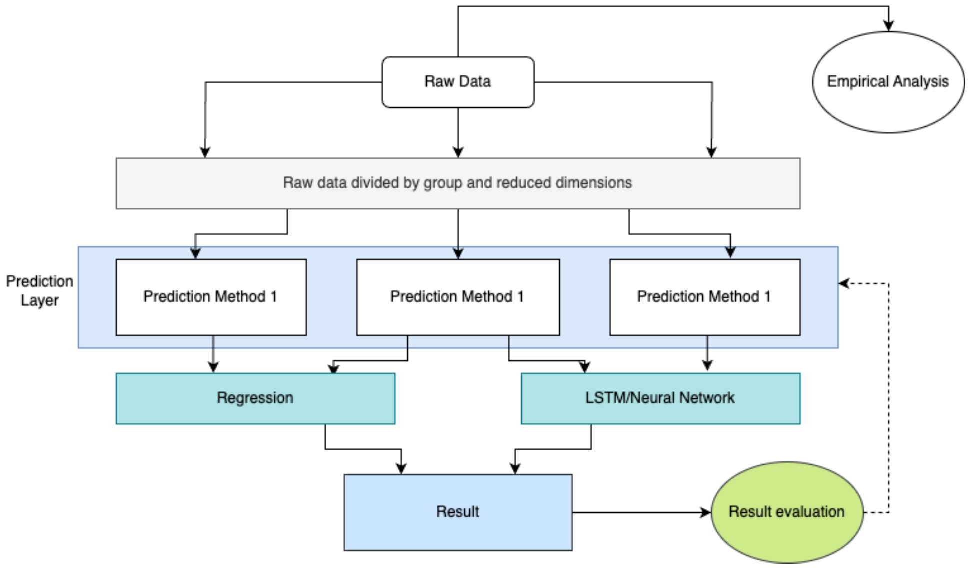

Figure 1: Flow Chart for the whole prediction approach

5 Data Cleaning and Preprocessing

As we could observe, the last (latest) row – 5/1/2023 – has quite a lot of missing value. Hence, we would drop it. Also, for ACOGNO (New Orders for Consumer Goods), TWEXAFEGSMTHx (Trade Weighted U.S. Dollar Index), the missing value would be too much, so we would consider dropping them.

For UMCSENTx (Consumer Sentiment Index), it has an interesting pattern: from 11/1/1959 to 11/1/1977, for each 3 row, i to i+2, i would have value, while the coming next 2 have missing value. As a result, we would use i to fill i+1 to i+2; While others would just use the mean of all data in CSI. For the rest, we could just use fillna with the mean.

For ANDENOx (New Orders for Nondefense Capital Goods), it is noticeable that the number is much larger when time increases. As a result, I suggest we should just drop it instead of using the mean for fillna.

For VIXCLSx (VIX CBOE Volatility Index), we could just use fillna, assuming it does not have a significant fluctuation (See during COVID vs rest), same for permit(s), S&P div yield, etc.

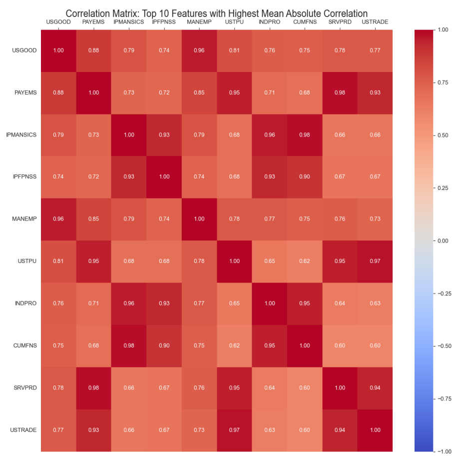

Figure 2: Variable Correlation Matrix

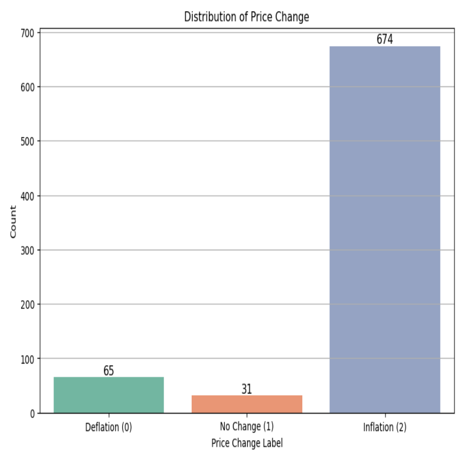

Figure 3: Interest Rate Change Distribution

Given the strong multicollinearity of the data demonstrated in the heat map, we adopt PCA method

Table 1: Missing Values Summary

Variable	**Missing Values**	**Percentage**
ACOGNO	398 missing values	51.49%
TWEXAFEGSMTHx	168 missing values	21.73%
UMCSENTx	154 missing values	19.92%
ANDENOx	109 missing values	14.10%
VIXCLSx	42 missing values	5.43%
PERMIT	12 missing values	1.55%
PERMITNE	12 missing values	1.55%
PERMITMW	12 missing values	1.55%
PERMITS	12 missing values	1.55%
PERMITW	12 missing values	1.55%
S&P div yield	2 missing values	0.26%
CMRMTSPLx	1 missing value	0.13%
HWI	1 missing value	0.13%
HWIURATIO	1 missing value	0.13%
BUSINVx	1 missing value	0.13%
ISRATIOx	1 missing value	0.13%
NONREVSL	1 missing value	0.13%
CONSPI	1 missing value	0.13%
S&P PE ratio	1 missing value	0.13%
CP3Mx	1 missing value	0.13%
COMPAPFFx	1 missing value	0.13%
DTCOLNVHFNM	1 missing value	0.13%
DTCTHFNM	1 missing value	0.13%

to reduce the dimension of the variables. These predictors can be categorized into 8 types according to their nature: Output and income, Labor market, Housing, Consumption, Money and credit, Interest and exchange rates, Prices and Stock market.

Then we will subtract suitable number of main components from each category. By preprocessing the dataset, our model will gain more economic explanation power.

6 Dimension Reduction Method

We would use OLS regression to be the baseline model, to compare the reduction performance with A. PCA, B. RFE C. SelectKBest. In baseline scenario where no dimension reduction is used, we get following prediction results. (Blue represents actual and yellow represents predicted data)

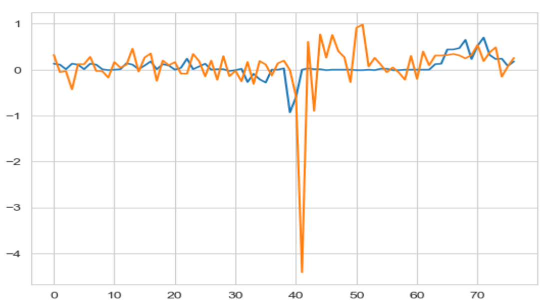

Figure 4: Baseline: Prediction Result

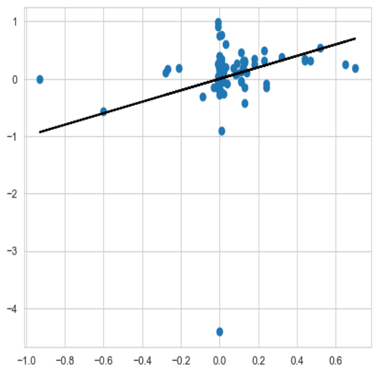

Figure 5: Baseline: Scatter Plot

6.1 PCA

In order to preserve the unique features and to avoid over generalization, an individual analysis on PCA performance based on the grouping is provided. The RMSE result based on the grouping is in Figure 6.

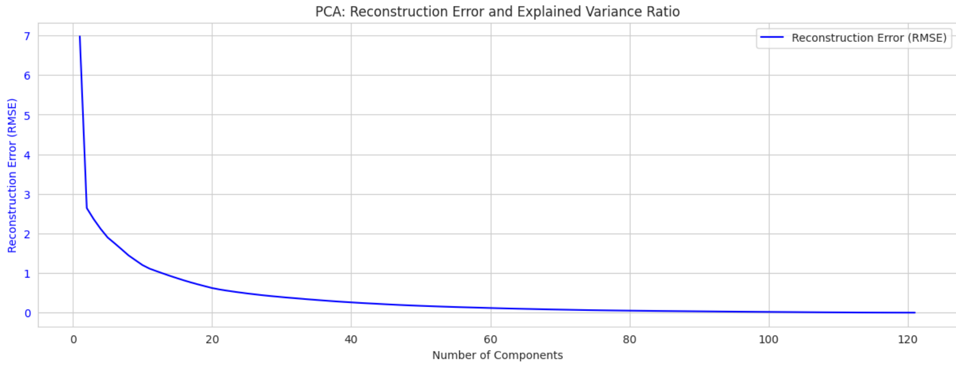

Figure 6: PCA: RMSE

OLS model:

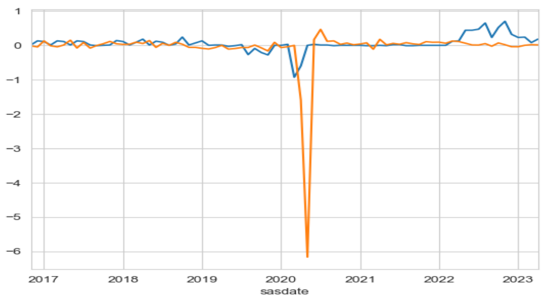

Figure 7: OLS + PCA: Prediction Result

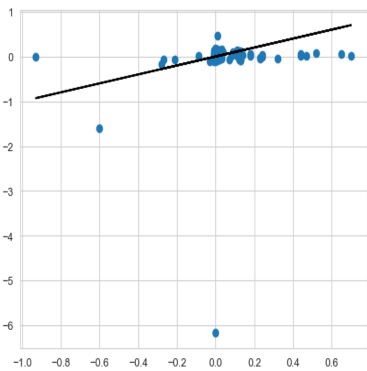

Figure 8: OLS + PCA: Scatter Plot

Elastic Net + PCA:

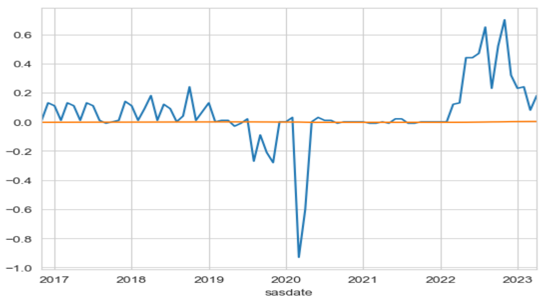

Figure 9: EN + PCA: Prediction Result

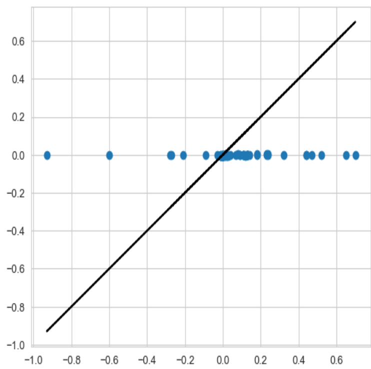

Figure 10: EN + PCA: Scatter Plot

Elastic Net + Sliding Time Window + PCA:

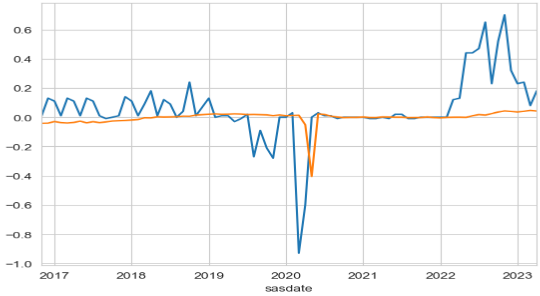

Figure 11: EN + SW +PCA: Prediction Result

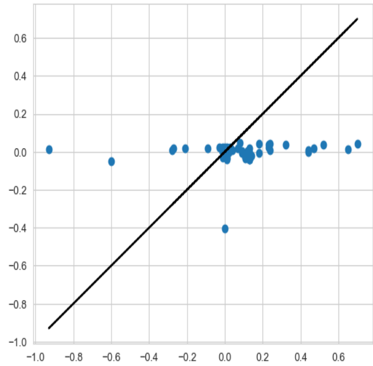

Figure 12: EN + SW + PCA: Scatter Plot

6.2 Recursive Factor Elimination (RFE)

Recursive Factor Elimination is also applied to Linear regression. The configuration to verify the effectiveness of REF is as follows:

- **train %**: 70 % (537)
- **valid %**: 15 % (115)
- **test %**: 15 % (116)

Grid search results (Root mean Squared Error score) are as follows:

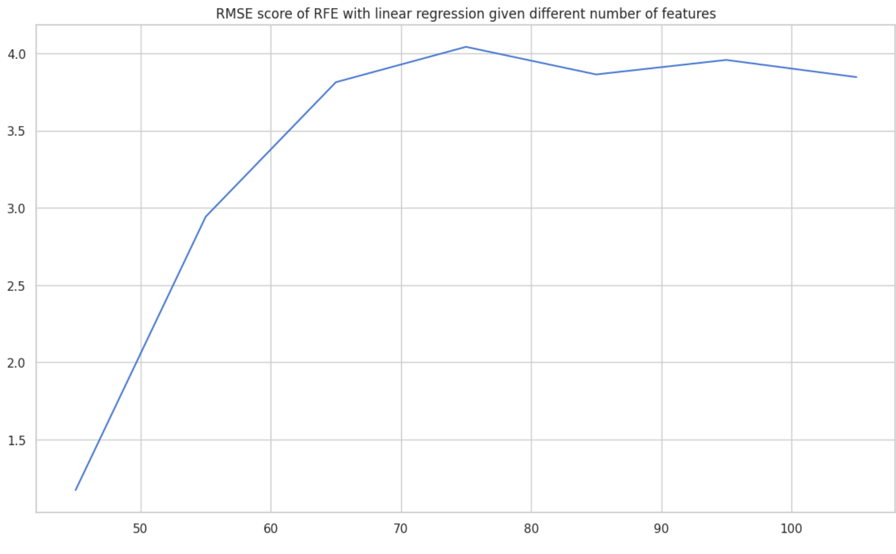

Figure 13: RFE: RMSE score

The result implies that 45 features could be an ideal choice of n given that it contributes to the lowest Root Mean Squared Error value for linear regression.

OLS + RFE:

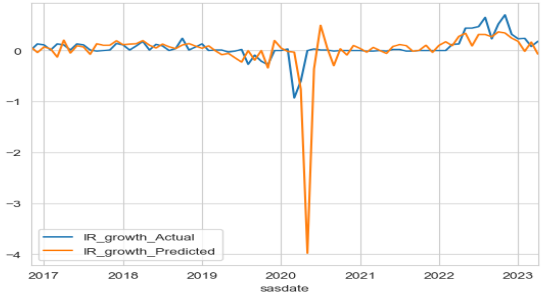

Figure 14: OLS + RFE: Prediction Result

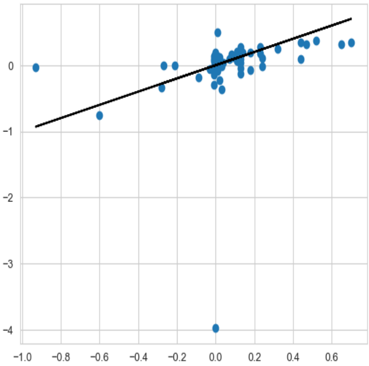

Figure 15: OLS + RFE: Scatter Plot

Elastic Net + RFE:

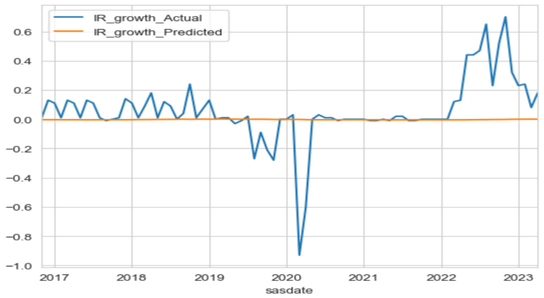

Figure 16: EN + RFE: Prediction Result

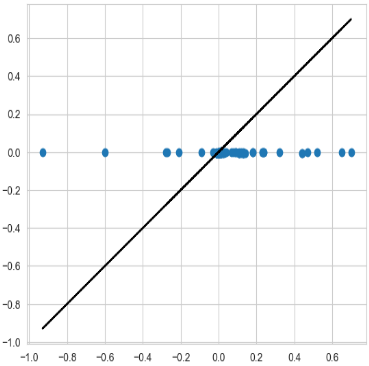

Figure 17: EN + RFE: Scatter Plot

Elastic Net + Sliding Time WIndow + RFE:

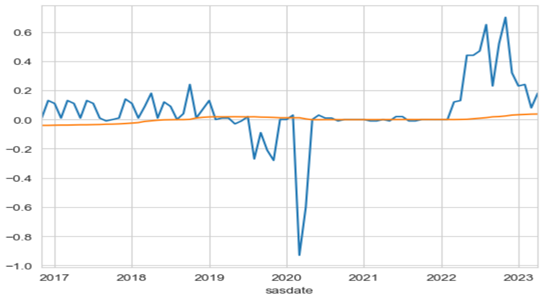

Figure 18: EN + SW + RFE: Prediction Result

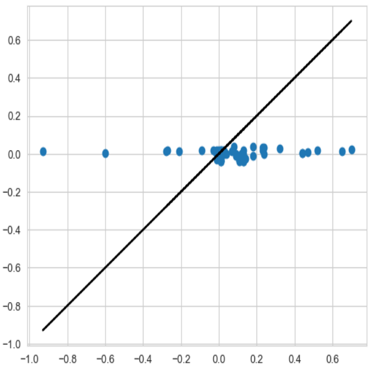

Figure 19: EN + SW + RFE: Scatter Plot

6.3 SelectKBest

By conducting SelectKBest method we get top 3 variables in each group:

- Group 1: INDPRO, IPMANSICS, CUMFNS
- Group 2: DMANEMP, MANEMP, USGOOD
- Group 3: PERMIT, HOUST, HOUSTS
- Group 4: AMDMUOX, BUSINVX, CMRMTSPLX
- Group 5: M2SL, M2REAL, NONBORRES
- Group 6: CP3MX, TB3MS, TB6MS

- Group 7: CPIULFSL, CUSR0000SAS, CUSR0000SA0L5
- Group 8: VIXCLSX, SP DIV YIELD, SP PE RATIO

6.4 Dimension Reduction Method Selection

Table 2: Dimension Reduction Result

Model	R-squre	MSE	Time
OLS	-6.84	0.60	2.32
OLS with PCA	-10.95	0.73	0.66
OLS with PCA in EN Reg.	-0.07	0.22	2.58
OLS with PCA in EN Reg. in S.W.	-0.07	0.22	1.37
OLS with RFE	-4.13	0.48	106.51
OLS with RFE in EN Reg.	-0.07	0.22	107.44
OLS with RFE in EN Reg. in S.W.	-0.06	0.22	37.97

According to the training time, RFE takes the longest time; While PCA and SelectKBest are similar. Also, in terms of the multicollinearity issue, RFE might not be the best approach since RFE mainly focus on finding the features with highest attribute e.g. coefficient estimates, which does not contribute to eliminating the multicollinearity issue in each procedure, until n features have been selected. Hence, if there are 3 features with really high, or even perfect multicollinearity issue, but having high contribution to the y prediction, RFE suppose cannot help with this issue. Only the least important features are pruned. So, considering the computational cost and ability to tackle multicollinearity issue, we do NOT use RFE.

Alternatively, for SelectKBest, despite the computational cost being less high, for our regression task, it is using F-stat, which measures how well a feature explains the variance in the target variable relative to the variance explained by other features. Besides, while feature selection methods like SelectKBest may help identify important features, they do not resolve underlying multicollinearity issues.

Therefore, we are only using PCA to tackle the multicolinearity issue, instead of RFE and SelectKBest.

7 Classification Models

We recognize that the dataset has imbalanced distribution, thus SMOTE is leveraged to diminish bias in our dataset to get better results.

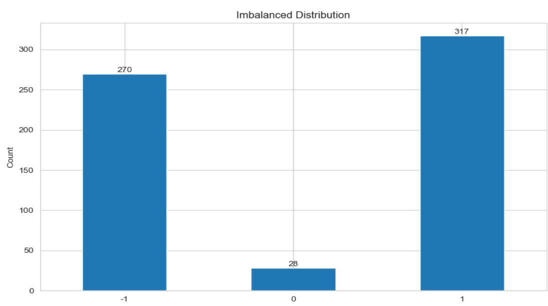

Figure 20: Imbalanced Distribution

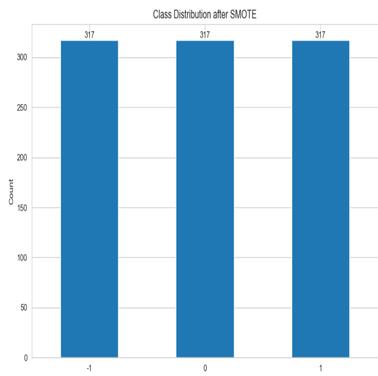

Figure 21: Distribution after SMOTE

We conducted empirical studies on using regression models, RNN and multilayer neural network to make prediction on interest rates change. The model settings and results before SMOTE are as follows:

7.1 Logistic Regression: Before SMOTE

Best parameters after grid search:

- C: 0.01
- L1 ratio: 0.2
- Best Score(F1): 0.5460

Performance of the model:

Table 3

Model	Accurary	Precision	Recall	F1 Score	ROC
Logistic Regression	0.487013	0.41959	0.487013	0.43032	0.595458

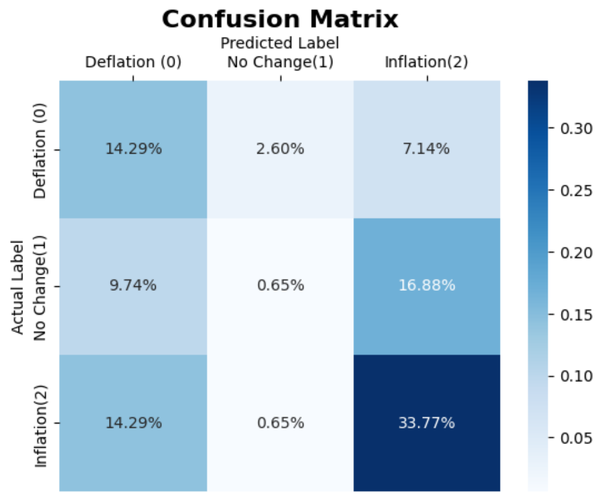

Figure 22: Confusion Matrix of logistic regression

7.2 Decision Tree Regression: Before SMOTE

After hyper parameters optimization, we obtained the following optimized parameters:

- max depth: 3
- max leaf nodes: 5
- min samples split: 3
- Best Score (F1): 0.6428

Performance of the model:

Table 4

Model	Accurary	Precision	Recall	F1 Score	ROC
Logistic Regression	0.38961	0.285561	0.38961	0.329518	0.544689

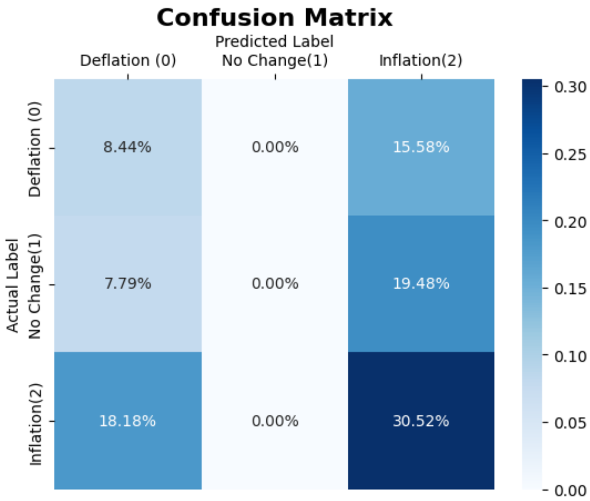

Figure 23: Decision Tree Confusion Matrix

7.3 SGD Classifier: Before SMOTE

After hyper parameters optimization, we obtained the following optimized parameters:

- alpha: 0.1
- eta 0: 0.1
- l1 ratio: 0.25
- learning rate: adaptive
- Score (F1): 0.6393

Performance of the model:

Table 5

Model	Accurary	Precision	Recall	F1 Score	ROC
Logistic Regression	0.519481	0.377195	0.519481	0.437035	0.610604

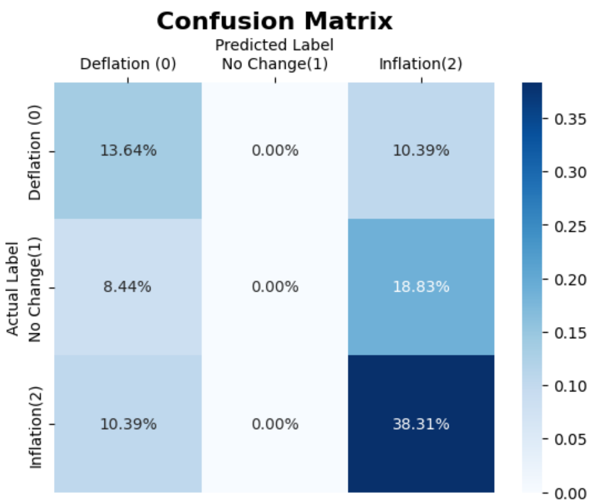

Figure 24: SGD Classification Confusion Matrix

7.4 Support Vector Machine: Before SMOTE

The fourth approach is a Support Vector Machine. The hyper-parameters tuning is as follows:

- **svm C**: 0.001
- **svm kernel**: linear
- **Best Score (F1)**: 0.6477

The model performance:

Table 6

Model	Accurary	Precision	Recall	F1 Score	ROC
Logistic Regression	0.519481	0.377195	0.519481	0.437035	0.610604

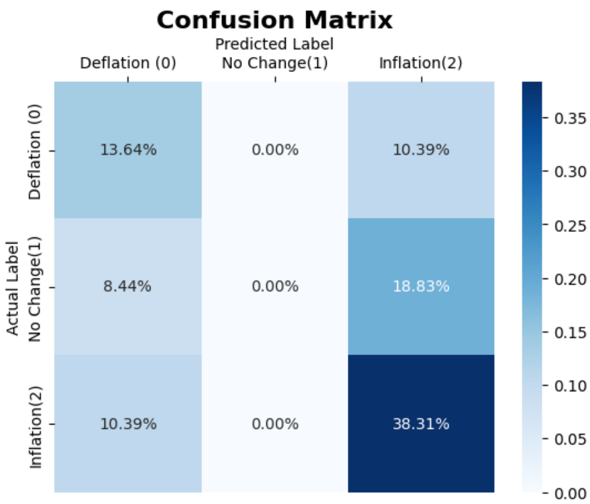

Figure 25: SVM Confusion Matrix

7.5 XG Boost Classifier: Before SMOTE

The fourth approach is the XG Boost Classifier. The hyper-parameters tuning is as follows:

- **XGB learning rate**: 0.01
- **XGB max depth**: 5
- **XGB n estimators**: 100
- **XGB subsample**: 0.5
- **Best Score (F1)**: 0.6726

The model performance:

Table 7

Model	Accurary	Precision	Recall	F1 Score	ROC
Logistic Regression	0.506494	0.366176	0.506494	0.42491	0.650393

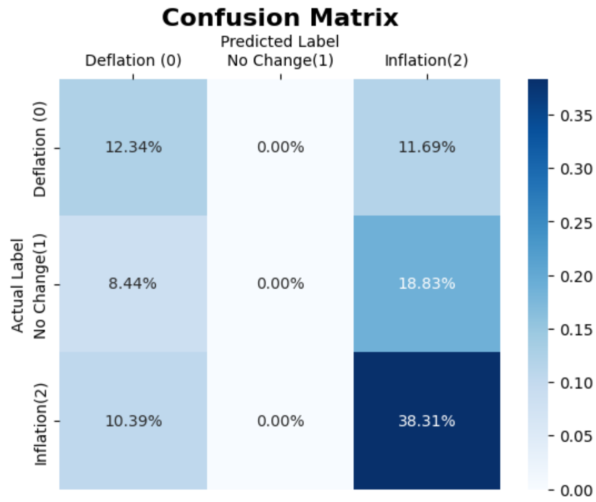

Figure 26: XGBoost Classification Confusion Matrix

7.6 All Model Results: After SMOTE

Table 8: Metrics of Classification Result after Result

Model	Accuracy	Precision	Recall	F1 Score	ROC
Logistic Regression	0.42	0.45	0.42	0.43	0.58
Decision Tree Classifier	0.40	0.47	0.40	0.42	0.58
SGD Classifier	0.45	0.48	0.45	0.45	0.58
SVM Classifier	0.45	0.48	0.45	0.46	0.58
XGBoost Classifier	0.53	0.54	0.53	0.53	0.69

We can see that basically all metrics in all models have improvement after SMOTE, and XGBoost is the best performing model.

8 Federal Funds Rate Prediction

8.1 Federal Funds Rate and Rate Spread

In this chapter, we conduct regression methods to predict the change of Federal Funds Rate and its spread. We use ElasticNet as the baseline model and RNN, LSTM as advanced models.

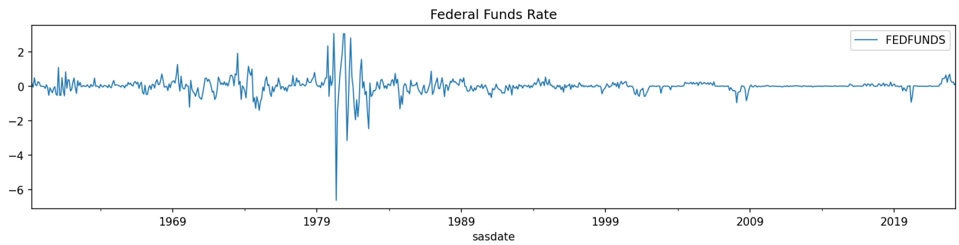

Figure 27: Federal Funds Rate Change

8.2 Baseline Model: Elastic Net

As there are a relatively large number of variables in the regression, we use ElasticNet for regularization. However, it is discovered that even in the circumstance that hyperparameters are set to pose low restric-

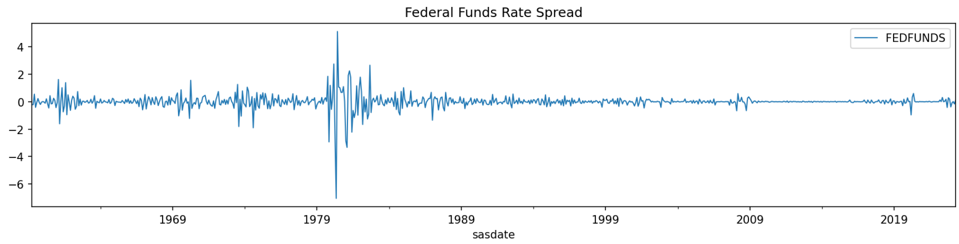

Figure 28: Rate Change Spread

tion, results are shown as constant in two scenarios. It is helpless for us to predict the further Federal Funds Rate movement. Therefore, more advanced neural network models need to be introduced.

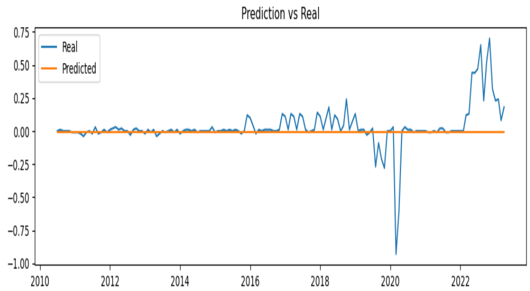

Figure 29: Elastic Net: Rate Change

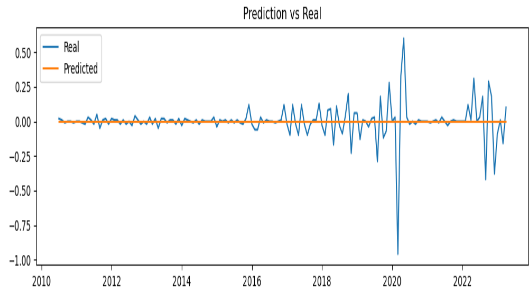

Figure 30: Elastic Net: Spread

8.3 Basic Machine Learning Modles

First, we use basic machine learning methods to solve the regression problem.

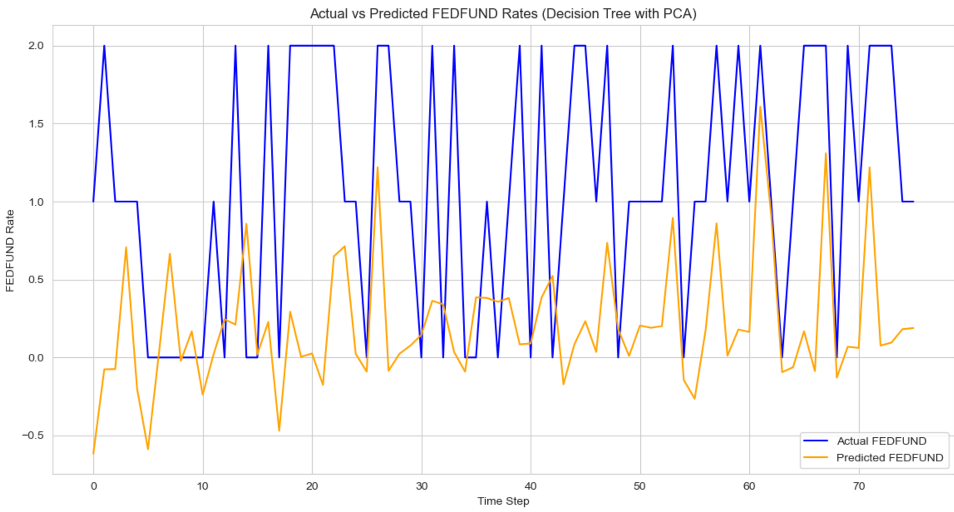

Figure 31: Decision Tree

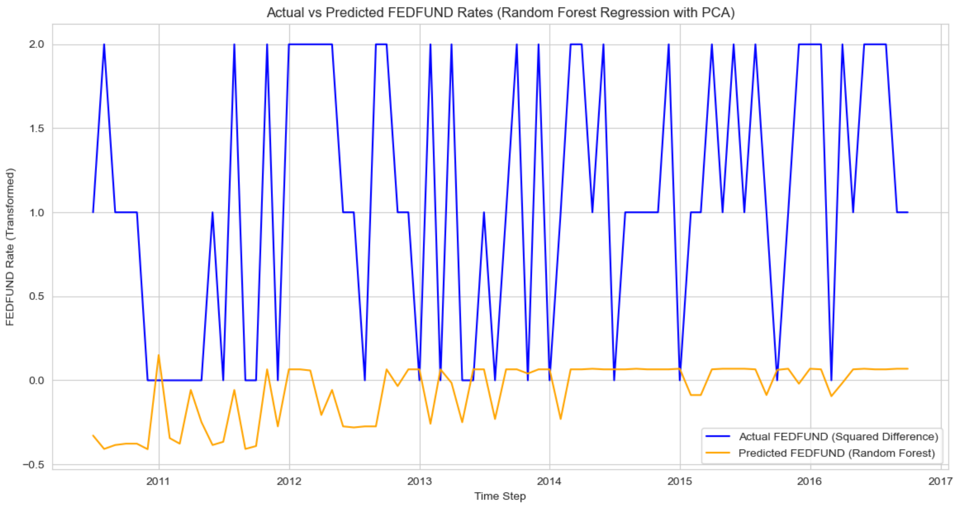

Figure 32: Random Forest

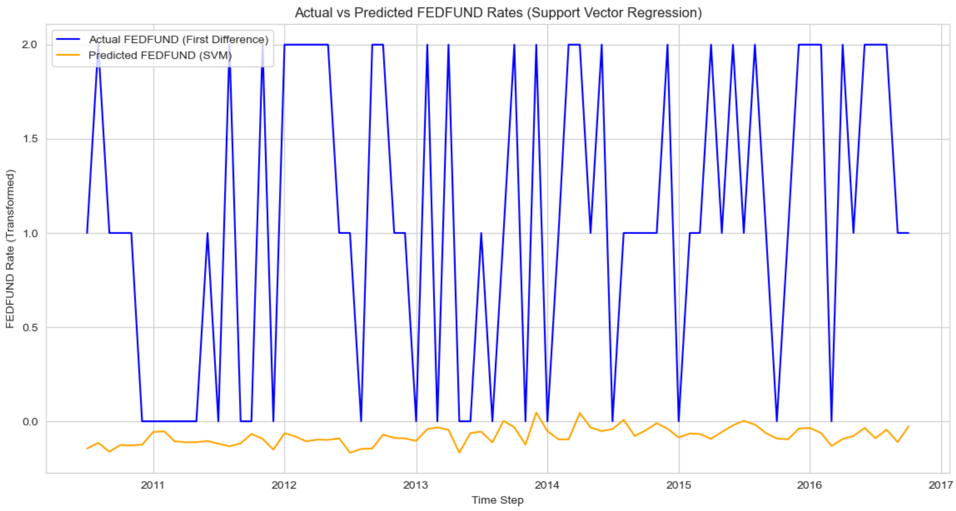

Figure 33: SVM

8.4 RNN-Garch

The RNN-GARCH method leverages the nonlinear modeling capabilities of RNNs with the volatility clustering properties captured by GARCH models. By integrating these two approaches, the hybrid model can better capture the complex dynamics of Federal Funds Rate, including nonlinear relationships and time-varying volatility.

8.5 LSTM

LSTM is a type of Recurrent Neural Network (RNN) that is specifically designed to address the issue of long-term dependencies in sequence data. Unlike standard RNN, LSTM can capture and utilize long-term dependencies due to their unique architecture, which includes a gating mechanism and a cell state that can store information across time steps. Interest rates can be influenced by long-term economic tren.ds and policy decisions. LSTM's capability to remember and utilize information from earlier time steps makes them effective in identifying and leveraging these long-term patterns.

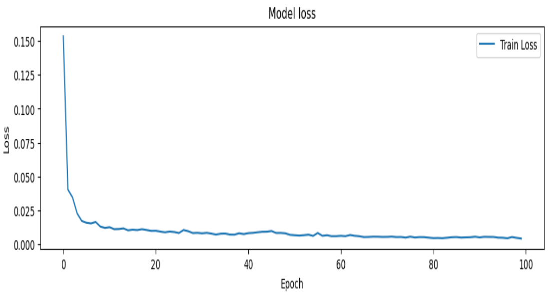

Figure 34: RNN Rate Change: Loss

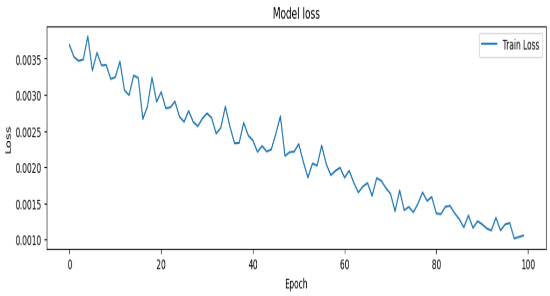

Figure 35: RNN Spread: Loss

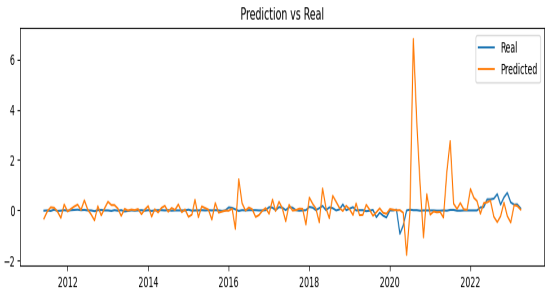

Figure 36: RNN: Rate Change

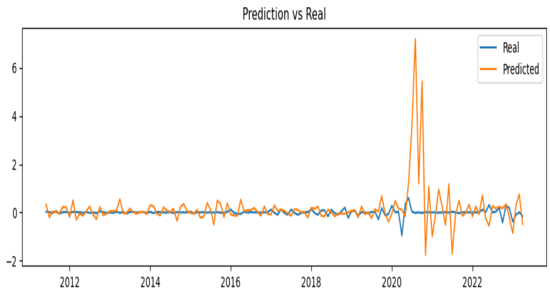

Figure 37: RNN: Spread

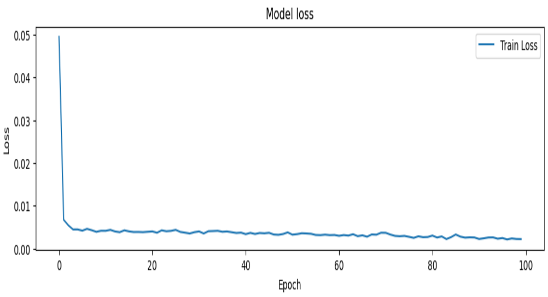

Figure 38: LSTM Rate Change: Loss

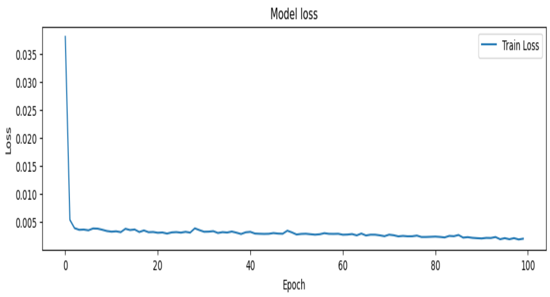

Figure 39: LSTM Spread: Loss

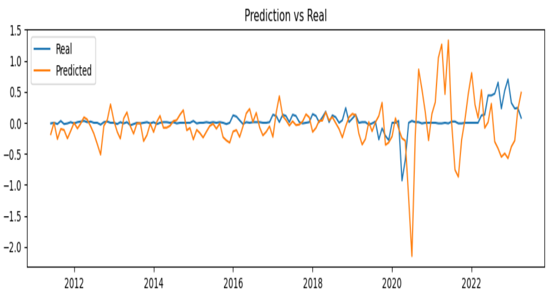

Figure 40: LSTM: Rate Change

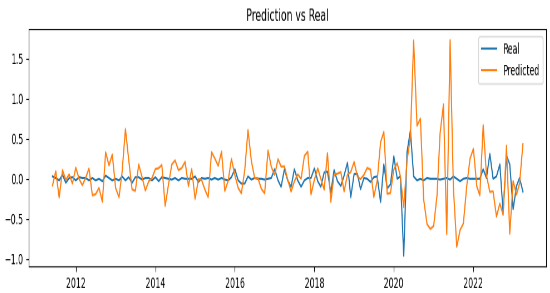

Figure 41: LSTM: Spread

8.6 Result Comparison

From Table7, it can be seen that ElasticNet method performs better than RNN and LSTM in prediction accuracy in terms of RMSE and MAE. However, it is because most of the change of Federal Funds Rate does not have too much deviation from 0 from 2011 to 2023. For RNN and LSTM, because they can behave well in time series data, they tend to be affected by the extreme value around 1980 as indicated in Figure 11 and Figure 12.

But it can still be discovered that LSTM can cope with long-term economic data. The RMSE and MAE values of LSTM model are significantly small. Besides, RNN performs better when processing the change, while LSTM when processing the first difference data.

Table 9: Statistical Summary of Regression Result

Method	**Data Type**	**RMSE**	**MAE**
ElasticNet	Change	0.16	0.07
ElasticNet	Spread	0.13	0.06
RNN+Garch	Change	0.78	0.34
RNN+Garch	Spread	0.91	0.37
LSTM	Change	0.42	0.26
LSTM	Spread	0.36	0.24

Besides the precision, winning rate is a better matrix to measure the performance of models. We define winning rate as times of model when it predicts the accurate movement of Federal Funds rate.

Table 10: Winning Rate of Models

Method	**Data Type**	**Winning Rate**
ElasticNet	Change	0.52
ElasticNet	Spread	0.53
RNN+Garch	Change	0.61
RNN+Garch	Spread	0.59
LSTM	Change	0.72
LSTM	Spread	0.76

From Table 8, it can be identified LSTM is the most suitable model to predict Federal Funds Rate when it is measured by winning rate.

9 Potential issues of the models setup

9.1 Data

The data consists of different dimensions. From the perspective of data frequency, a higher frequency data can capture a more comprehensive variation within the data. But more data may also contribute to a higher error in making predictions that affect the choice of optimal models. Meanwhile, the data may not be comprehensive enough to capture all the explaining factors. It is impossible to capture all the explaining factors.

10 Result Improvement

10.1 Decision Tree Regression

Potential Problems
When using Principal Component Analysis (PCA), the decision tree model's predictions for the FED-

FUND Rate appear excessively flat. Although it captures some movements, the adjustments based on fluctuations are minimal, resulting in a prediction curve that resembles a flat line. The R-squared value for this model is -32.8.

In contrast, when PCA is not applied, the R-squared score improves to -3.2; however, the model fails to capture fluctuations entirely, leading to a prediction that is also flat and unhelpful for future forecasting.

The poor performance with PCA may stem from its inherent assumptions, as noted by Protopapas et al. (2019):

- **Mean/Variance Sufficiency:** PCA assumes that means and the covariance matrix adequately describe the distributions of the predictors.
- **High Variance Indicates Importance:** PCA posits that components with higher variance are more significant for explaining data structure. This assumption may not hold true, particularly if essential patterns exist in lower-variance components.
- **Orthogonality of Principal Components:** PCA assumes that intrinsic dimensions are orthogonal.
- **Linear Change of Basis:** PCA assumes that the data lie on a lower-dimensional linear manifold. If the data are non-linear, linear methods may be ineffective.

Proposed Solutions
The model with PCA appears to be underfitting due to its conservative predictions. This could be attributed to the nature of our financial time series dataset; reducing dimensionality may have removed important features. To address this, we could increase the maximum depth from 7 to a more aggressive value, such as 10.

Similarly, for the decision tree regressor without PCA, a maximum depth of 3 seems insufficient. Exploring depths greater than 10 may help capture the most essential features.

10.2 Random Forest Regression

For the Random Forest Regression (RFR) model, the R-squared value with PCA is -65.098186, while without PCA it improves to -0.980383, which is the best performance among all models so far. The RFR without PCA significantly outperforms the decision tree regressor, which is expected since ensemble learning typically enhances predictive performance by combining multiple decision trees.

The RFR successfully captures most fluctuations but exhibits slight overestimation of the actual FEDFUND rate. This overfitting in individual decision trees can help the random forest learn from errors, improving its predictions. We could experiment with increasing the number of trees beyond the current values of 100 and 200, potentially trying 300 or more to achieve better results.

While the RFR without PCA performs well, the version with PCA performs the worst among all models. This may be due to PCA's significant removal of features from the financial time series data, resulting in flat predictions and underfitting. For ensemble learning, retaining more dimensions could allow the RFR to learn more effectively.

10.3 Support Vector Machine (SVM)

The predicted curve from the SVM model is less volatile than the actual curve, indicating that the model is insufficiently capturing the relevant information from the features and is experiencing underfitting. Increasing the regularization parameter C may help reduce underfitting and create a less smooth SVM model, as C aims to minimize training error (Academy, 2023). Allowing for a higher chance of error in this case could lead to improved performance.

References

[1] Vašíček, O. (1977). An equilibrium characterization of the term structure. Journal of Financial Economics, 5(2), 177-188.

[2] Taylor, J. B. (1993). Discretion versus policy rules in practice. Carnegie-Rochester Conference Series on Public Policy, 39, 195-214.

[3] Choudhry, T., D. M. (2010). Machine learning analysis and modeling of interest rate curves. ResearchGate.