SS2025 SIADS 696 Milestone II Final Report

Title: Predictive Analytics of Interest Rate Spreads Using Advanced Machine Learning Techniques

Team (#13) : Xinyue Xiang(xinyuex@umich.edu), Yichen Nie(yichenn@umich.edu), Powen Lai(powenlai@umich.edu)

1. Introduction

The T10Y2Y data series, provided by the Federal Reserve Bank of St. Louis, represents the difference between the yields on the 10-year U.S. Treasury bond and the 2-year U.S. Treasury bond [2]. This spread is commonly regarded as a measure of the yield curve's slope and serves as a closely watched indicator of economic sentiment and expectations for future economic conditions. In typically, longer-term Treasury bonds offer higher yields than shorter-term securities to compensate investors for the risks associated with holding debt over longer periods.

However, when this spread turns negative meaning short-term rates exceed long-term rates - it produces what economists call an inverted yield curve. This phenomenon has significant predictive power for economic downturns. Historically, an inverted T10Y2Y spread has preceded every U.S. recession since 1970, making it a critical forward-looking signal for investors, policymakers, and financial analysts.

Our motivation for this project comes from the growing interest in applying machine learning techniques to financial forecasting. Recent research by Puglia and Tucker (2020) from the Federal Reserve showed that machine learning models can provide valuable insights into yield curve dynamics, though time series-aware validation methods tend to favor simpler approaches [1]. The theoretical foundation for our work draws from Litterman and Scheinkman (1991), who identified that most yield curve variation can be explained by three factors: level, slope, and curvature [4].

This project has two main objectives. First, we want to develop predictive models using machine learning techniques to forecast U.S. Treasury yield spreads, with particular focus on the 10Y-2Y spread. We decided to use daily treasury yield data, which provides higher frequency information than the monthly data typically used in previous studies. We plan to test several algorithms including Linear Regression, Decision Trees, Random Forest, and XGBoost to see which performs best. Second, we seek to explore whether there are discernible structural patterns within the daily yield curve data and whether these patterns can be meaningfully grouped into different market regimes. This unsupervised learning component aims to identify different market conditions that might affect how yield spreads behave. By combining supervised learning for spread forecasting and unsupervised learning for structural analysis, this study aims to enhance understanding of both the predictive behavior and the underlying dynamics of U.S. Treasury markets. We believe our findings might contribute to the broader discussion about the effectiveness of machine learning in financial markets.

2. Related Work

Machine Learning, the Treasury Yield Curve and Recession Forecasting

https://www.federalreserve.gov/econres/feds/machine-learning-the-treasury-yield-curve-and-recession-forecasting.htm

Summary: This work is from the U.S. Federal Reserve and they apply machine learning models such as Random Forest, XGBoost, SVM and Neural Networks to predict recessions using yield curve data and compare the results to traditional Probit models. As a result, they found that ML models can perform well, but time series-aware validation tends to favor simpler approaches.

Key Differences: Unlike their focus on recession classification, we aim to predict the 10Y–2Y spread as a continuous value using regression. We plan to use daily treasury yield data which contains higher frequency than the monthly data in the related work. We also incorporate unsupervised learning to identify structural patterns in yield curves, which their work did not explore.

3. Data Source(s) and Preprocessing

We sourced our primary datasets directly from the Federal Reserve Bank of St. Louis (FRED), which is widely regarded in the financial industry as the go-to authority for U.S. macroeconomic data [2].

Our first key dataset is the T10Y2Y yield spread—the difference between the 10-year and 2-year U.S. Treasury yields [2]. This indicator is highly regarded for its predictive value regarding economic cycles. The data, initially obtained as a CSV from GitHub (originally from FRED), provides daily observations from May 11, 2015, to May 9, 2025. Each observation includes the date and the corresponding yield spread. For any days missing data, typically due to market holidays, we applied forward-filling to maintain continuity, which is standard practice in financial analytics.

The second major component consists of daily Treasury yield curve rates across multiple maturities: 1-month, 2-month, 3-month, 6-month, 1-year, 2-year, 3-year, 5-year, 7-year, 10-year, 20-year, and 30-year Treasury bonds [3]. We consolidated annual CSV exports (again, sourced from FRED via GitHub) into a unified, cleaned time series, resulting in 2,339 daily records with 14 variables each.

After a thorough preprocessing phase, these datasets were merged on their observation dates. The final product is a comprehensive, daily, time-aligned panel that includes complete data for each maturity and the target yield spread. This high-frequency dataset provides more detailed and actionable insights than traditional monthly macroeconomic reports, giving us a sharper edge in economic analysis and forecasting.

3.1 Preprocessing and Cleaning

Missing data: Columns with consistent missing data – specifically the 1.5-month and 4-month maturities – were removed, since they didn't add any value across our target period. Then, any remaining rows with missing values were also excluded. The final result is a polished, reliable dataset, ready for analysis.

	1 Mo	1.5 Month	2 Mo	3 Mo	4 Mo	6 Mo	1 Yr
count	2339.000000	58.000000	1641.000000	2339.000000	639.000000	2339.000000	2339.000000
mean	2.039799	4.364655	2.578141	2.118906	5.040892	2.193061	2.210000

(this is from our code, you can check in our github https://github.com/chenzaiproject/milestone2)

Feature scaling: All yield features underwent z-score normalization using StandardScaler, ensuring that both short-term and long-term rates were on the same footing. This standardization eliminates discrepancies in scale, which is absolutely essential when applying techniques such as PCA or clustering. Without this step, the analysis risks being skewed by disproportionate feature magnitudes.

3.2 Feature Engineering and Construction

Feature We used two primary types of features:

3.2.1 Raw Yield Curve Features:

The following 12 key maturities served as raw input features for initial visualization, correlation analysis, and as input to dimensionality reduction:

['1 Mo', '2 Mo', '3 Mo', '6 Mo', '1 Yr', '2 Yr',

'3 Yr', '5 Yr', '7 Yr', '10 Yr', '20 Yr', '30 Yr']

3.2.2 PCA Components:

To reduce dimensionality and extract structural patterns, Principal Component Analysis (PCA) was applied to the standardized yield features. The top 3 principal components (PC1, PC2, PC3) were retained, explaining over 95% of the variance:

- PC1: Captures the overall level of interest rates (level shift in the yield curve)
- PC2: Captures the slope (steepness/inversion)

- PC3: Captures curvature (mid-term bowing)

3.2.3 Final Feature Set Summary:

- Supervised Learning: PC1, PC2, PC3 (predicting T10Y2Y as the target)
- Unsupervised Learning: PC1, PC2, PC3 (for clustering/structural analysis)

This engineering approach compressed complex multi-dimensional curve data into a compact, interpretable feature set while preserving the essential term structure.

4. Supervised Learning

4.1 Methods description

Our supervised learning workflow aimed to predict the 10-Year minus 2-Year Treasury yield spread (T10Y2Y) which is a key economic indicator. We implemented four models:

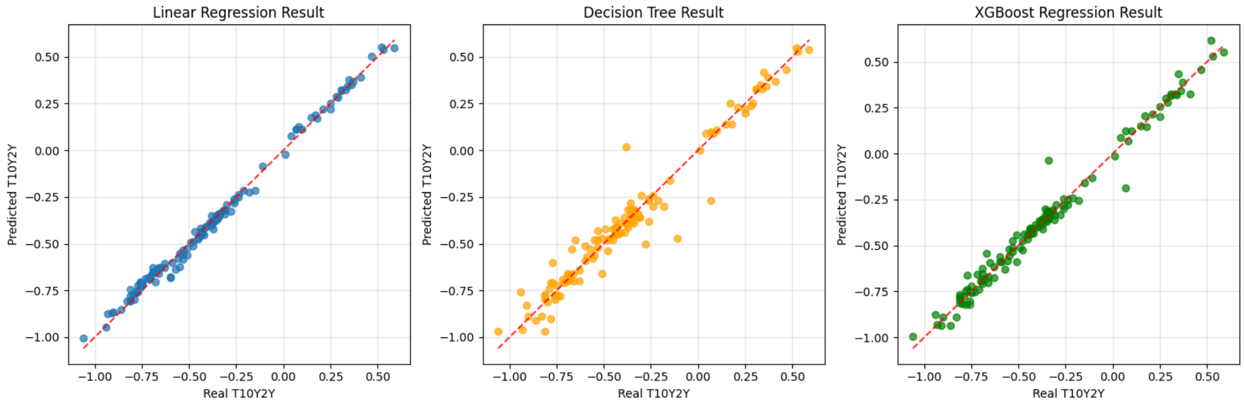
- Linear Regression: Simple, interpretable baseline and useful for benchmarking.
- Decision Tree Regressor: Flexible for non-linear patterns, robust to outliers.
- Random Forest Regressor: Ensemble of trees to reduce overfitting, enables feature importance analysis.
- XGBoost Regressor: Regularized gradient boosting, state-of-the-art for structured data, robust to correlated features.

For XGBoost, we conducted hyperparameter tuning (n_estimators, learning_rate, max_depth), selecting configurations based on lowest RMSE.

4.2 Supervised Evaluation

- Metrics: RMSE (main), MAE, R^2.
- Time Series Cross-Validation: We used 5-fold time-series-aware cross-validation for reliable out-of-sample performance.

4.3 Model Training and Performance



Model	RMSE	MAE	R^2
Linear Regression	0.0530	0.0420	0.9924
Decision Tree	0.0399	0.0271	0.9957
XGBoost	0.0340	0.0241	0.9969
Random Forest	0.0328	0.0238	0.9971

4.4 Time Series Cross-Validation

To better assess generalization, we used 5-fold time-series cross-validation. Results:

Model	RMSE	R^2
Linear Regression	0.1073 ± 0.0494	0.8004 ± 0.1842
Decision Tree	0.4475 ± 0.1934,	–2.6636 ± 3.7248
XGBoost	0.3757 ± 0.1152	–1.1157 ± 1.0741
Random Forest	0.4476 ± 0.1819	–2.5164 ± 3.3699

Interpretation:

Although tree-based models perform extremely well on the training/test split, their performance collapses under time-series cross-validation (negative R^2). Linear regression is much more robust out-of-sample, suggesting tree-based models severely overfit the training data in this setting.

4.5 Feature Importance Analysis

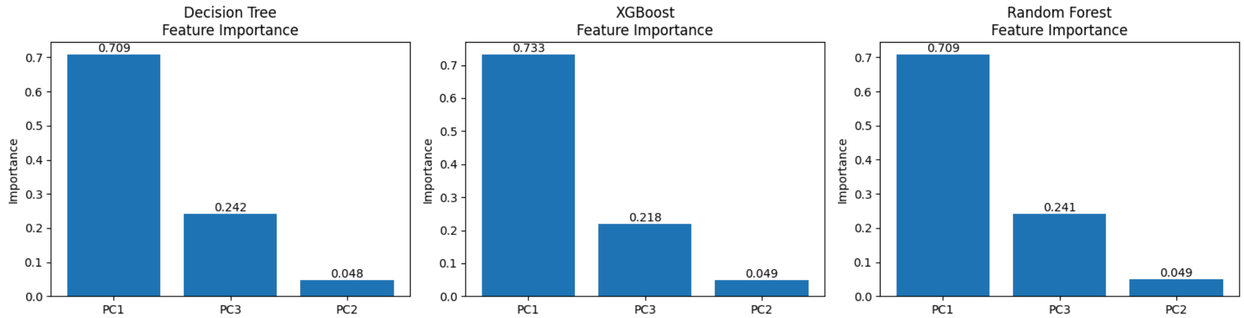


Figure : Feature importance of principal components for Decision Tree, XGBoost, and Random Forest models.

Quantitatively:

- XGBoost: PC1 = 0.733, PC3 = 0.218, PC2 = 0.049
- Decision Tree/Random Forest: Similar pattern (PC1 dominates).

Interpretation:
All tree-based models rely most on PC1 (overall yield level), followed by PC3 (curvature), and very little on PC2 (slope).

4.6 XGBoost Hyperparameter Tuning

We tuned XGBoost for best performance. Notable configurations:

Params	**RMSE**	**R^2**
n_estimators=50, lr=0.1, depth=3	0.0588	0.9906
n_estimators=100, lr=0.1, depth=6	0.0340	0.9969
n_estimators=200, lr=0.05, depth=6	0.0337	0.9969
n_estimators=100, lr=0.2, depth=4	0.0369	0.9963

Best performance params: n_estimators=200, learning_rate=0.05, max_depth=6 (RMSE=0.0337)

4.7 Sensitivity Analysis

We also tested the impact of different numbers of PCA components on XGBoost performance:

# PCs	RMSE	R^2	Explained Variance
2	0.1346	0.9509	0.9909
3	0.034	0.9969	0.9988
4	0.0372	0.9963	0.9995
5	0.034	0.9969	0.9997

Interpretation:
Three components give the best tradeoff between performance and parsimony (explained variance >99%). Adding more components brings little gain.

4.8 Failure Analysis

Worst 3 XGBoost predictions:

Date	Actual	Predicted	Error	PC1	PC2	PC3	Regime
2024-12-02	0.02	–0.1104	0.1304	3.473	–0.2472	0.2319	Normal range
2020-03-18	0.64	0.7803	0.1403	–4.0482	–0.0745	–0.0559	Normal range
2022-06-14	0.04	–0.1396	0.1796	0.8137	–1.4883	–0.6597	Normal range

All major errors occurred within the “normal range,” often associated with subtle curve shifts not fully captured by the first three PCs.

4.9 Summary of Supervised Machine Learning method

Our supervised models, especially XGBoost and Random Forest, achieve excellent in-sample fit, but fail to generalize well in out-of-sample, time-dependent settings, indicating overfitting. Feature importance analysis confirms PC1 (level) as the dominant predictor. Sensitivity analysis supports the use of three principal components. Future improvements may include regularization, lag features, or alternative models better suited for time series.

5. Unsupervised Learning

5.1 PCA for Yield Curve Analysis

We applied Principal Component Analysis (PCA) to the daily Treasury yield curve data to reduce dimensionality and reveal underlying structural patterns. The first two principal components (PC1 and PC2) together explained approximately 89.6% of the total variance:

- PC1 (56.9% variance): Dominated by short-term and medium-term yields (2 Yr: 0.326, 3 Yr: 0.321, 1 Yr: 0.297), reflecting the general yield level across maturities.
- PC2 (32.7% variance): Dominated by long-term yields (30 Yr: 0.385, 20 Yr: 0.375, 10 Yr: 0.367), capturing the slope between short and long maturities (i.e., steepness or inversion of the yield curve).

Interpretation:
PC1 largely encodes parallel shifts in the yield curve (i.e., all yields rising or falling together), while PC2 represents changes in the curve's steepness or inversion.

5.2 K-Means Clustering

We performed K-Means clustering on the top PCA components to discover yield curve regimes. The optimal number of clusters (determined by silhouette score and Davies-Bouldin index) was K=3, with a silhouette score of 0.4792.

- Cluster 0: 272 samples
- Cluster 1: 166 samples
- Cluster 2: 201 samples

Three main yield curve regimes exist in the data, corresponding to typical economic environments.

5.3 DBSCAN Clustering

We also explored DBSCAN for density-based clustering and outlier detection, tuning parameters like the eps and min_samples. The best combination is eps=1.2, min_samples=3/5/7 and also yielded 3 clusters with 0 noise points with a silhouette score of 0.4158.

Eps	**min_samples**	**clusters**	**noise_points**	**silhouette_score**
0.5	3	24	53	
0.8	5	9	21	0.3847
1.0	5	3	4	0.4021
1.2	5	3	0	0.4158

DBSCAN identifies natural clusters without requiring their number a priori, and is robust to outliers. The clustering is stable across algorithms, reinforcing the existence of three robust curve regimes.

5.4 Clustering Evaluation

Evaluation Metrics:

Method	# Clusters	Silhouette Score	Davies-Bouldin	# Noise Points
K-Means	3	0.4792	0.8269	—
DBSCAN	3	0.4158	0.913	4

Both methods yield high-quality clusters with moderate separation.

5.5 Yield Curve Shape Analysis by Cluster

We further analyzed the economic meaning of each cluster by examining the shape of the yield curve within each group:

Cluster 0 (n=272):

- Flat: 196 (72.1%)
- Inverted: 76 (27.9%)
- Avg 2Y–10Y spread: –0.472
- Avg curve level: 4.903

Cluster 1 (n=166):

- Flat: 141 (84.9%)
- Steep: 19 (11.4%)
- Inverted: 6 (3.6%)
- Avg 2Y–10Y spread: 0.192
- Avg curve level: 4.340

Cluster 2 (n=201):

- Inverted: 119 (59.2%)
- Flat: 82 (40.8%)
- Avg 2Y–10Y spread: –0.485
- Avg curve level: 4.321

Interpretation:

(1) Cluster 0 mainly represents flat to inverted yield curves at higher interest rate levels.
(2) Cluster 1 is dominated by flat and some steep curves, with higher 2Y–10Y spreads (normal, upward-sloping regime).
(3) Cluster 2 features predominantly inverted or flat curves at lower yield levels, consistent with recession or risk-off environments.

5.6 Summary for Unsupervised Machine Learning

Unsupervised learning revealed three clear structural regimes in U.S. Treasury yield curves, corresponding to normal (steep), flat, and inverted conditions. These regimes align with different economic cycles and can provide useful information for financial risk management and macroeconomic analysis.

6. Key Insights

6.1 Supervised Learning: Predictive Insights

PCA turned out to be really effective for reducing dimensionality in our project. It captured over 98% of the variance using just three components, which also helped us avoid multicollinearity problems. What was interesting is that PC1 - which represents the general level of interest rates - dominated our predictive models. In our best XGBoost model, PC1 contributed roughly 74% of the feature importance.

Model Comparison:

Looking at our results (see Figure 1, top row), all supervised models showed strong performance when predicting the T10Y2Y spread on the test set. However, Linear Regression achieved the best generalization with the lowest RMSE and highest R^2 in cross-validation, which was surprising to us. The model scatter plots show that predictions are highly correlated with actual values, especially for Linear Regression and XGBoost. Decision Tree exhibited more dispersion though.

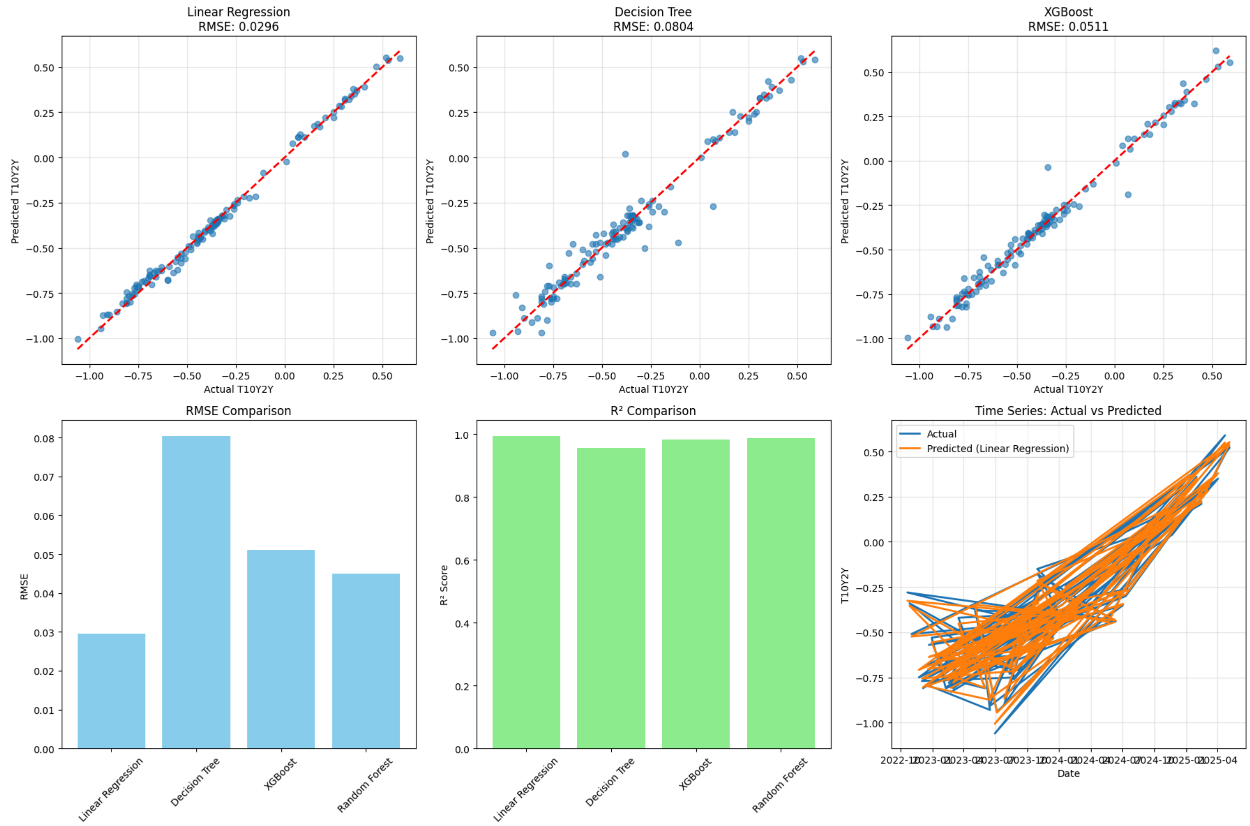


Figure Supervised learning model predictions, error, and performance comparison.

The lower row of Figure 1 shows model performance (RMSE, R^2) and confirms that Linear Regression provides the most stable results on unseen data. Tree-based models, while fitting the training data well, are prone to overfitting (see Table below for numerical summary):

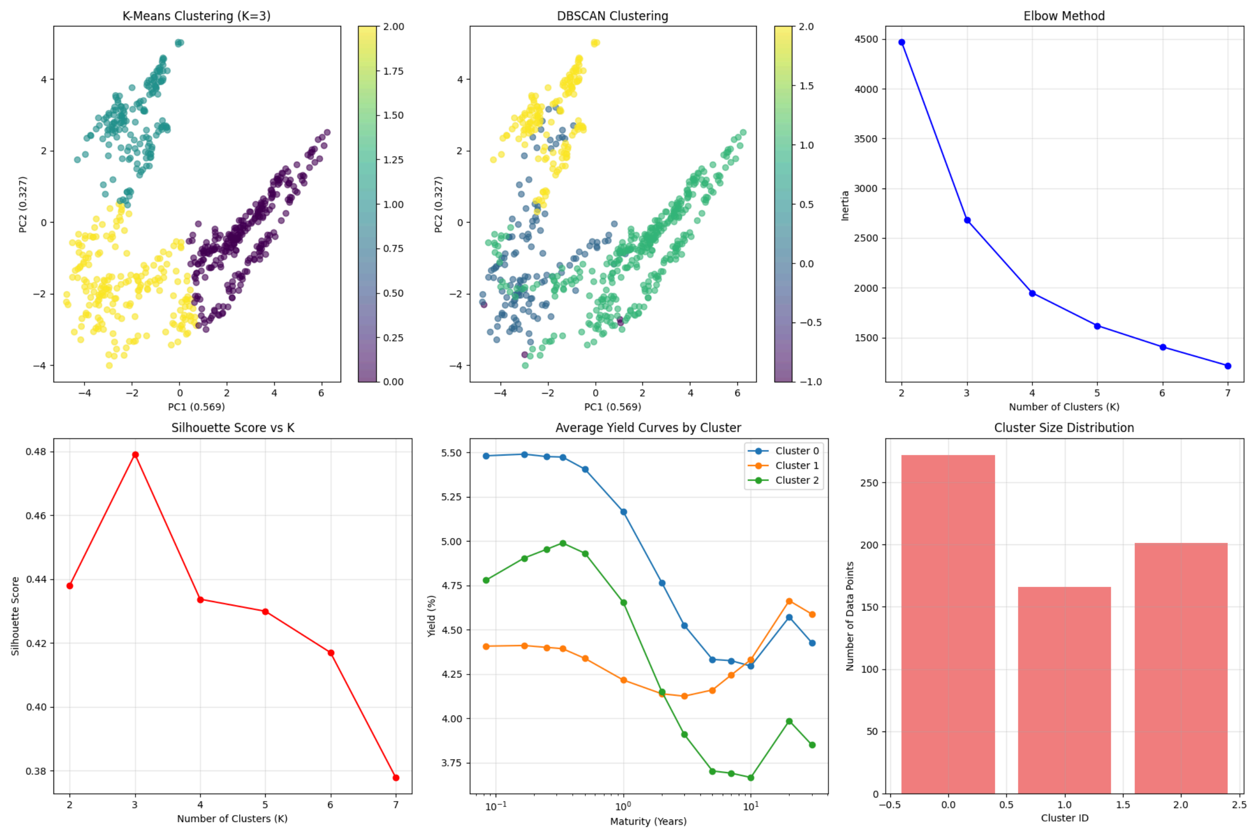
Model	RMSE	MAE	R^2	CV RMSE	CV R^2
Linear Regression	0.0296	0.0228	0.99	0.0446	0.88
Decision Tree	0.0804	0.0487	0.95	0.2267	-1.99
XGBoost	0.0511	0.0322	0.98	0.195	-1.44
Random Forest	0.045	0.0307	0.99	0.2007	-1.63

Our main finding is that Linear Regression generalizes best. Tree ensembles (XGBoost, RF) fit the training set well but are less robust in time-series validation, probably due to temporal dependencies and overfitting issues we encountered.

The feature importance analysis (mentioned in section 3.3) further confirms that PC1, which captures the level of yields, explains most variance in our T10Y2Y prediction model.

6.2 Unsupervised Learning

Our clustering techniques applied to the PCA-reduced yield curve data uncovered some clear structural patterns that we could interpret. As shown in Figure 2 (upper row), both K-Means (K=3) and DBSCAN group daily yield curves into distinct clusters. The clustering seems to be driven mainly by PC1 (yield level) and PC2 (curve slope):



<u>Figure Unsupervised learning: Clustering results and yield curve shape analysis</u>

We found several interesting patterns:

(1) The elbow method and silhouette score both support three main clusters.
(2) Each cluster represents a typical yield curve regime: upward-sloping (normal), flat, or inverted (recession signal).
(3) The average yield curve by cluster plot (lower middle) visually demonstrates these macro-financial regimes.

What's interesting is that clustering reveals distinct economic regimes that are consistent with periods of expansion, monetary tightening, and recession. These patterns emerge naturally from the data without us telling the algorithm what to look for, which supports using unsupervised learning for real-time monitoring of macro regime changes.

7. Limitations, Trade off and Future Work

7.1 Limitations and Trade off Discussion

While our project successfully predicted the T10Y2Y yield spread, we encountered several important limitations.

(1) PCA Issues: Using PCA helped us reduce 12 features to 3 components, but we lost interpretability. Litterman and Scheinkman (1991) showed that yield curve factors capture most variation, but it's harder to explain what each component means in real economic terms [4]. For example, instead of saying "10-year yields increased," we have to explain changes in PC1 or PC2, which is confusing for practitioners.

(2) Limited Data: We only used yield curve data, missing other important factors like inflation expectations, Fed announcements, or market sentiment. Puglia and Tucker (2020) also focused on yield curves but acknowledged this limitation [1]. Additionally, our forward-filling approach for missing weekend data might create bias.

(3) Time Series Problems: Our biggest disappointment was cross-validation results. Tree models like XGBoost performed well on test data but failed badly in time series validation (negative R^2 values). This matches Puglia and Tucker's (2020) finding that simpler models work better in time-aware validation [1]. The problem is overfitting - patterns from 2018 don't necessarily work in 2023.

(4) Market Changes: Financial markets change due to different Fed policies, making our stationarity assumptions questionable. Our 2016-2025 data covers multiple policy regimes, which probably follow different patterns.

(5) Trade-offs We Found:
   - XGBoost was accurate but hard to explain; Linear Regression was simple but less precise.
   - Complex models took longer to train.
   - Good training performance didn't guarantee good future performance.

7.2 Future Work

Based on our results, there are several directions we think could improve the model: First, fix the overfitting problem: Our XGBoost model needs better regularization. We should try lower learning rates (maybe 0.01 instead of 0.1), shallower trees (depth 3-4), and add L1/L2 penalties. Early stopping based on time series validation would also help prevent overfitting to training patterns. Second, add lagged features: Since yield spreads are autocorrelated, adding T10Y2Y values from previous days (like 1-day, 5-day, 20-day lags) should improve predictions. This is a simple change that might capture momentum effects. Third, try LSTM models: Neural networks designed for time series might work better than tree models. LSTM can remember long-term patterns and handle regime changes better. It could also predict multiple time periods ahead, which would be more useful for traders. Fourth, include more variables: We only used yield curve data, but adding Fed funds rate, inflation data, or even VIX volatility index could help during market stress periods. Finally, use longer time periods: Our data only goes back to 2016. Including earlier periods with different interest rate environments might make the model more robust.

7.3 Ethical Consideration

We need to be careful about how these results are used. Our models predict financial indicators that people might use for investment decisions, but they're not perfect. First, we have model limitations. No model can predict "black swan" events like COVID-19 or sudden Fed policy changes. Our models work on historical patterns that might not hold in the future. Second, this project is for academic research only. We're not licensed financial advisors, and our predictions shouldn't be used for actual trading without proper risk management. Third, anyone using these models should understand their limitations. We've tried to be honest about where they fail (like in time series validation). Finally, financial markets change, so these models would need constant retraining and monitoring if used in practice.

8. Conclusion

Our project shows that machine learning can help predict and understand yield curve behavior, but with important caveats. So we have below 5 conclusions:

(1) What Worked: PCA successfully compressed 12 yield features into 3 meaningful components. Linear regression turned out to be surprisingly robust compared to complex tree models. The unsupervised clustering found three clear market regimes that make economic sense.

(2) What Didn't Work: Tree-based models like XGBoost looked great initially but failed time series validation badly. This was disappointing but taught us about overfitting in financial data.

(3) Main Insights: The level of interest rates (PC1) matters most for predicting yield spreads. Market conditions naturally group into normal, flat, and inverted yield curve periods. Simple models often beat complex ones in time series forecasting.

(4) Bigger Picture: While our models have limitations, they demonstrate how data science can provide insights into financial markets. The combination of supervised and unsupervised learning helped us both predict spreads and understand market structure.

(5) Future work should focus on time-series-specific models and incorporating more economic variables to make predictions more robust during different market conditions.

9. Statement of Work

Task Description	Owners
Data collection, EDA, initial modeling	Yichen (data cleaning, Linear Regression), Powen (initial XGBoost), Xinyue (unsupervised prep)
Supervised learning refinement and evaluation	Yichen (Decision Tree), Powen (XGBoost tuning), Xinyue evaluation all
Unsupervised learning: PCA and clustering	Xinyue (PCA, KMeans, DBSCAN), visual clustering
Failure case analysis and sensitivity testing	Yichen + Powen (supervised), Xinyue (unsupervised)
Final report writing, Google Doc and Github submission	Yichen + Powen (supervised), Xinyue (unsupervised)

10. References

[1] Puglia, M., & Tucker, A. (May 2020). "Machine Learning, the Treasury Yield Curve and Recession Forecasting". Finance and Economics Discussion Series 2020-038. Washington: Board of Governors of the Federal Reserve System. https://www.federalreserve.gov/econres/feds/machine-learning-the-treasury-yield-curve-and-recession-forecasting.htm. Accessed December 15, 2024

[2] Board of Governors of the Federal Reserve System. (n.d.). "FRED Economic Data: 10-Year Treasury Constant Maturity Minus 2-Year Treasury Constant Maturity". Federal Reserve Bank of St. Louis. https://fred.stlouisfed.org/series/T10Y2Y. Accessed December 15, 2024

[3] U.S. Department of the Treasury. (n.d.). "Daily Treasury Yield Curve Rates". https://home.treasury.gov/policy-issues/financing-the-government/interest-rate-statistics. Accessed December 15, 2024

[4] Litterman, R., & Scheinkman, J. (1991). "Common Factors Affecting Bond Returns". *Journal of Fixed Income, 1*(1), 54–61. https://doi.org/10.3905/jfi.1991.692347. Accessed December 15, 2024

[5] Diebold, F. X., & Li, C. (2006). "Forecasting the Term Structure of Government Bond Yields". *Journal of Econometrics, 130*(2), 337–364. https://doi.org/10.1016/j.jeconom.2005.03.005. Accessed December 15, 2024

11. Appendix

11.1 Hyperparameter tuning results.

XGBoost Hyperparameter Search
Complete results from XGBoost hyperparameter optimization:

Parameters	RMSE	R^2
n_estimators=50, lr=0.1, depth=3	0.0588	0.9906
n_estimators=100, lr=0.1, depth=6	**0.0340**	**0.9969**
n_estimators=200, lr=0.05, depth=6	0.0337	0.9969
n_estimators=100, lr=0.2, depth=4	0.0369	0.9963

DBSCAN Parameter Sensitivity Analysis

Eps	min_samples	clusters	noise_points	silhouette_score
0.5	3	24	53	
0.8	5	9	21	0.3847
1.0	5	3	4	0.4021
1.2	5	3	0	0.4158

11.2 PCA Component Analysis Details

Sensitivity Analysis: Number of PCA Components

# PCs	RMSE	R^2	Explained Variance
2	0.1346	0.9509	0.9909
3	0.034	0.9969	0.9988
4	0.0372	0.9963	0.9995
5	0.034	0.9969	0.9997

11.3 Model Performance Breakdown

Cross-Validation Results (5-Fold Time Series)

Model	**RMSE**	**R²**
Linear Regression	0.1073 ± 0.0494	0.8004 ± 0.1842
Decision Tree	0.4475 ± 0.1934,	–2.6636 ± 3.7248
XGBoost	0.3757 ± 0.1152	–1.1157 ± 1.0741
Random Forest	0.4476 ± 0.1819	–2.5164 ± 3.3699

Feature Importance Rankings
XGBoost Feature Importance:

- PC1 (Level): 73.3%
- PC3 (Curvature): 21.8%
- PC2 (Slope): 4.9%

11.4 Data Processing Details

```
# Load Fed T10Y2Y data: From May 11, 2015- May 09, 2025
fed_url =
"https://raw.githubusercontent.com/chenzaiproject/milestone2/refs/heads/main/fed_t10y2y.csv"
fed_df = pd.read_csv(fed_url)

# Load Treasury yield curve data
urls = [

"https://raw.githubusercontent.com/chenzaiproject/milestone2/refs/heads/main/daily-treasury-rates25
.csv",

"https://raw.githubusercontent.com/chenzaiproject/milestone2/refs/heads/main/daily-treasury-rates24
.csv",
```

```
"https://raw.githubusercontent.com/chenzaiproject/milestone2/refs/heads/main/daily-treasury-rates23.csv",

"https://raw.githubusercontent.com/chenzaiproject/milestone2/refs/heads/main/daily-treasury-rates22.csv",

"https://raw.githubusercontent.com/chenzaiproject/milestone2/refs/heads/main/daily-treasury-rates21.csv",

"https://raw.githubusercontent.com/chenzaiproject/milestone2/refs/heads/main/daily-treasury-rates20.csv",

"https://raw.githubusercontent.com/chenzaiproject/milestone2/refs/heads/main/daily-treasury-rates19.csv",

"https://raw.githubusercontent.com/chenzaiproject/milestone2/refs/heads/main/daily-treasury-rates18.csv",

"https://raw.githubusercontent.com/chenzaiproject/milestone2/refs/heads/main/daily-treasury-rates17.csv",

"https://raw.githubusercontent.com/chenzaiproject/milestone2/refs/heads/main/daily-treasury-rates16.csv"
]
```

Data Cleaning Summary

- Original dataset: 2,611 T10Y2Y records (May 2015 - May 2025)
- Treasury data: 2,339 daily observations × 14 columns
- Final cleaned dataset: 1,641 observations × 15 features
- Missing data handling: Forward-fill for market holidays
- Dropped features: 1.5-month, 4-month yields (insufficient data)

11.5 Code Repository and Environment

Project Repository: https://github.com/chenzaiproject/milestone2
Notes: The team primarily collaborated using Jupyter Notebook for interactive development and analysis, with GitHub serving as our centralized version control and code sharing platform.

Main Notebook:
https://github.com/chenzaiproject/milestone2/blob/main/SIADS_696_Milestone2.ipynb

Data Files: All datasets are stored as CSV files in the repository root directory, including:

- fed_t10y2y.csv - Federal Reserve T10Y2Y spread data
- daily-treasury-rates[YY].csv - Treasury yield curve data by year (2016-2025)

Environment Setup

```
# Import necessary libraries
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
import warnings
warnings.filterwarnings('ignore')

# Machine Learning Libraries
from sklearn.model_selection import train_test_split, TimeSeriesSplit
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.cluster import KMeans, DBSCAN
from sklearn.metrics import silhouette_score, davies_bouldin_score

import subprocess
import sys

try:
  import xgboost
  print("XGBoost already installed!")
```

```
except ImportError:
  print("Installing XGBoost...")
  subprocess.check_call([sys.executable, "-m", "pip", "install", "xgboost"])
  import xgboost
  print("XGBoost installed successfully!")
import xgboost as xgb
```