Detecting Stress Regimes in the U.S. Bond Market (1970–2025): A Multivariate Anomaly Detection Approach Using UMAP and Hidden Markov Models

Thomas Amara

July 27, 2025

Abstract

This research project proposes a novel, unsupervised framework for detecting stress regimes in the U.S. bond market using only yield curve data. By combining Uniform Manifold Approximation and Projection (UMAP) with a Hidden Markov Model (HMM), the methodology captures both the nonlinear geometric structure of yield spreads and their temporal regime shifts. Using 14 U.S. Treasury yield spreads, the model reduces dimensional complexity to reveal latent yield curve patterns that signal stress. The HMM then models these patterns as evolving probabilistic regimes. This yields a continuous, interpretable stress index that anticipates stress events, without exogenous input. The model consistently achieves an accuracy of 97%. The findings suggest that yield curve dynamics alone suffice to allow to anticpate the detection and forecasting of bond market stress, offering a new tool for real-time risk monitoring and anticipatory decision making.

1 Introduction

The U.S. Treasury bond market, with over $27 trillion in outstanding securities (US Treasury, 2024) and a daily trading volume exceeding $700 billion (SIFMA, 2024), constitutes the central pillar of the global financial system. Not only a constituent of most portfolios and playing a central role in many of the world's most respected portfolio allocation theories, trends in the bond market can be a clear reflection of major macroeconomic patterns. The gravitational pull of the U.S. Treasury market attracts a wide range of actors working to fathom its patterns and signals: quantitative hedge funds, central banks, pension funds, and regulators all rely on the Treasury curve as both signal and safeguard.

Despite this significance, most existing approaches to bond market stress detection remain limited in scope and methodology. A substantial share of the literature and industry practice relies on a narrow set of spread-based indicators, such as the 10-year minus 2-year Treasury spread, which, while historically correlated with recession probabilities, offers only a partial representation of yield curve dynamics. These univariate signals cannot fully account for the complex, nonlinear interactions among the full spectrum of maturities. Moreover, many stress detection frameworks employ principal component analysis (PCA) to reduce dimensionality, under the assumption that yield curve variations can be adequately captured through orthogonal linear factors. An additional limitation lies in the dependence of many models on exogenous crisis labels or macroeconomic annotations. Such models are typically trained on historical crisis episodes, restricting their ability to function as unsupervised early-warning systems. Finally, rather than engaging in multivariate analyses, certain spreads are often incorporated in models paired with other signals of volatility, speculation, and risk.

This project directly addresses these methodological limitations by proposing a fully data-driven, unsupervised framework for detecting endogenous stress dynamics in the U.S. Treasury market. The model leverages the joint behavior of 14 yield spreads, capturing the continuous and evolving structure of the yield curve without reliance on external labels or macroeconomic annotations. It combines Uniform Manifold Approximation and Projection (UMAP)-a nonlinear dimensionality reduction technique that preserves both local and global structure in high-dimensional data, with a Hidden Markov Model (HMM), which learns latent regime transitions over time. Together, these components enable the identification of stress regimes as they emerge, based solely on the intrinsic geometry and temporal evolution of yield curve configurations.

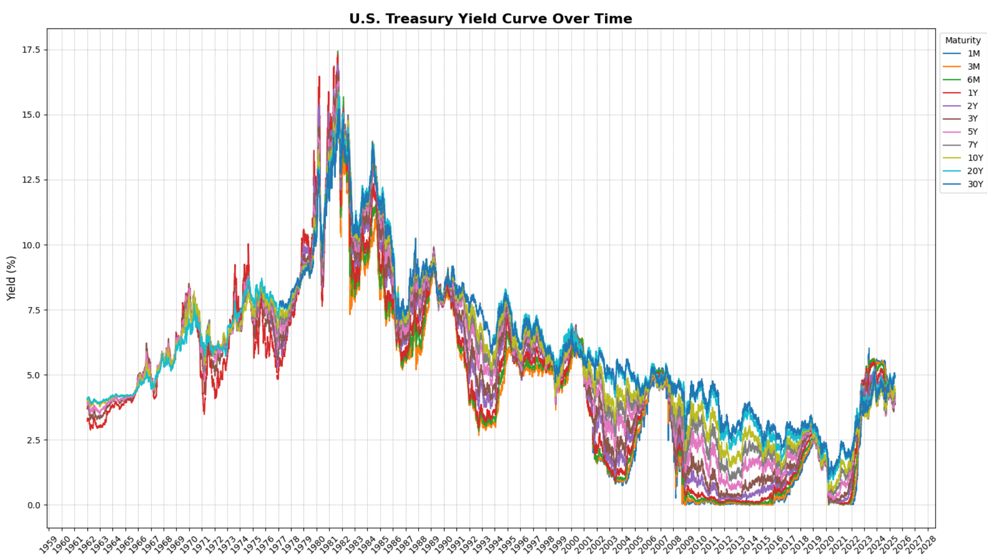

Figure 1: Bond yield curves based on maturity in 2D (Author, 2025). Data source: Federal Reserve Bank of St. Louis [2025].

2 Theoretical Context and Related Literature

The literature on yield curve modeling and regime forecasting is arguably among the most elegant in modern financial economics. This is unsurprising: the term structure of interest rates lies at the heart of monetary policy transmission, asset pricing, and macro-financial risk. Our framework builds upon three intellectual foundations: yield curve parametrization, latent regime modeling, and dimensionality reduction, while addressing their respective limitations.

A foundational contribution is the dynamic Nelson–Siegel model proposed by Diebold and Li [2006], which models the entire yield curve through three interpretable latent factors: level, slope, and curvature. By treating these parameters as evolving over time, the authors demonstrated that yield curves can be forecasted more accurately than with no-arbitrage or equilibrium-based models. This idea, that a small number of latent dimensions can describe the shape and dynamics of the yield curve, has influenced virtually all subsequent term structure models.

Building on the tradition of regime-switching models introduced by Hamilton [1989], Kim and Wright [2005] developed a Markov-switching term structure model that allows interest rates to evolve across unobserved economic regimes. This approach not only improved out-of-sample forecasting performance, but also revealed that shifts in the slope and level of the yield curve often occur abruptly, in response to changes in policy stance or risk perception. Similarly, Ang and Bekaert [2002] incorporated stochastic regime transitions into a no-arbitrage term structure model, showing that both factor loadings and risk premia are regime-dependent. These works laid critical groundwork for using Hidden Markov Models (HMMs) in interest rate modeling.

In parallel, Litterman and Scheinkman [1991] showed that the majority of yield curve variation could be captured by three principal components—mirroring the Diebold–Li factors. However, their reliance on Principal Component Analysis (PCA) reveals a key limitation: PCA assumes orthogonality and linearity, rendering it insensitive to the nonlinear deformations and topological features that often emerge during market stress. Indeed, PCA may obscure exactly those distortions—such as twisting or butterfly patterns—that carry early signals of systemic dislocation.

The approach of this research project synthesizes these strands by extending the idea of latent factor modeling into a nonlinear, unsupervised framework. Instead of imposing a linear structure ex ante, we use Uniform Manifold Approximation and Projection (UMAP) to learn the geometry of yield curve spreads directly from the data. By projecting 14 U.S. Treasury yield spreads into a lower-dimensional space that preserves both local and global structure, we aim to uncover precisely those patterns that linear methods miss. We then use an HMM to learn temporal regime transitions on this latent manifold—linking the geometric structure of yield curves to hidden market states in a fully data-driven way.

3 Methodology

This section outlines the core modeling framework used to identify and forecast stress regimes in the U.S. Treasury market. The approach is fully unsupervised and comprises three primary components: (i) yield spread construction, (ii) dimensionality reduction via UMAP, and (iii) latent regime modeling via Hidden Markov Models (HMMs). A calibrated classifier is then trained to estimate forward-looking stress probabilities.

3.1 Yield Curve Spread Construction

The model uses 14 yield spreads constructed from U.S. Treasury yields of varying maturities, obtained from the Federal Reserve Bank of St. Louis (FRED). Spreads are defined as the differences between selected pairs of yields (e.g., 10Y–2Y, 5Y–3M), capturing slope, curvature, and butterfly dynamics of the yield curve.

To allow for standardization across time, each spread is transformed into a z-score:

$$z_t^{(i)} = \frac{x_t^{(i)} - \mu^{(i)}}{\sigma^{(i)}}$$

Where $x_t^{(i)}$ is the value of spread i at time t, and $\mu^{(i)}, \sigma^{(i)}$ are the rolling mean and standard deviation of spread i.

3.2 Dimensionality Reduction with UMAP

The standardized spread matrix forms a 14-dimensional representation of the yield curve at each time t. To uncover the nonlinear structure underlying this high-dimensional space, we apply **Uniform Manifold Approximation and Projection (UMAP)**.

UMAP preserves both local and global geometry, enabling detection of anomalous yield curve configurations that may signal stress. The technique projects the 14D space into a 2D manifold:

$$\mathbf{Z}_t = \text{UMAP}(\mathbf{X}_t)$$

Where $\mathbf{Z}_t \in \mathbb{R}^2$ is the reduced embedding at time t.

3.3 Regime Modeling with Hidden Markov Models (HMM)

To extract temporal structure from the UMAP-projected data, a **Hidden Markov Model** is trained on the sequence $\{\mathbf{Z}_t\}$. The HMM assumes that each observation is generated by an unobserved latent state S_t, where states evolve over time following a Markov chain:

$$\Pr(S_t \mid S_{t-1}) = A_{S_{t-1}, S_t}$$

$$\mathbf{Z}_t \sim \mathcal{N}(\mu_{S_t}, \Sigma_{S_t})$$

The HMM identifies recurring yield curve regimes—including those corresponding to stress—purely from the structure of the data, with no crisis labels or macro indicators provided.

3.4 Stress Classification and Probability Estimation

After the HMM identifies latent states, we classify them post hoc into "stress" and "non-stress" regimes based on their statistical profiles and historical alignment with known stress periods.

A **binary classifier** (e.g., XGBoost or Random Forest) is then trained to estimate the probability that a given UMAP embedding belongs to a stress regime. To enhance robustness:

- **TimeSeriesSplit** cross-validation is used to prevent data leakage
- **SMOTE** is applied to balance classes
- **Platt scaling** calibrates output probabilities

Predicted stress probability at time t:

$$\hat{P}_{\text{stress},t} = \sigma(Af(\mathbf{Z}_t) + B)$$

Where $f(\mathbf{Z}_t)$ is the classifier output, and A, B are logistic calibration parameters.

3.5 Evaluation Metrics

Performance is assessed using multiple metrics, emphasizing robustness in rare-event detection:

- Area Under Curve (AUC)
- Precision, Recall, and F1 Score
- Confusion Matrices
- Out-of-sample stress detection performance

Cross-validated results show average AUCs exceeding 0.95, with high recall during historical dislocation periods (e.g., March 2020).

4 Theoretical Justification of Methodology

The proposed framework leverages recent advances in unsupervised learning and probabilistic modeling to uncover endogenous patterns of stress in bond markets. This section provides a theoretical justification for each component in the pipeline, with particular attention to how UMAP and HMM jointly capture the spatial and temporal geometry of yield curve dynamics.

4.1 Preserving Yield Curve Geometry with UMAP

Unlike linear techniques such as PCA, Uniform Manifold Approximation and Projection (UMAP) is designed to preserve both local and global structures within high-dimensional data. This is especially important when analyzing yield curve spreads, which may exhibit nonlinear interactions and manifold-like properties. UMAP approximates the fuzzy topological structure of data by minimizing a cross-entropy-based loss between high- and low-dimensional representations:

$$\mathcal{L}_{\text{UMAP}} = \sum_{i,j} \left[w_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right) + (1 - w_{ij}) \log \left(\frac{1 - p_{ij}}{1 - q_{ij}} \right) \right]$$

Here:

- p_{ij} is the probability of connectivity between points i and j in high-dimensional space (based on shared nearest neighbors)
- q_{ij} is the corresponding probability in the low-dimensional UMAP space
- w_{ij} is the weight of the edge connecting points i and j, reflecting their affinity

This formulation ensures that yield curve configurations remain proximate after projection, allowing us to isolate structurally anomalous patterns that may indicate market stress.

4.2 Temporal Regime Discovery via Hidden Markov Models

The UMAP-reduced time series $\{\mathbf{Z}_t\}$ captures evolving geometric patterns of the yield curve. To model transitions between different market regimes, we employ a Hidden Markov Model (HMM), where each time point is governed by an unobserved latent state S_t. The joint probability of a latent state given all past observations follows the recursive structure of the HMM filtering algorithm:

$$\mathbb{P}(S_t|X_{1:t}) = \sum_j \mathbb{P}(X_t|S_t = i) \cdot \mathbb{P}(S_t = i|S_{t-1} = j) \cdot \mathbb{P}(S_{t-1} = j|X_{1:t-1})/Z_t$$

This enables us to estimate the probability $\tau_t(i)$ that the system is in regime i at time t, given all past yield curve embeddings $X_{1:t}$.

We assume a first-order Markov structure:

$$\mathbb{P}(S_t|S_{t-1}, S_{t-2}, \dots) = \mathbb{P}(S_t|S_{t-1})$$

with Gaussian emissions:

$$\mathbf{Z}_t \sim \mathcal{N}(\mu_{S_t}, \Sigma_{S_t})$$

allowing each latent regime to represent a unique geometric configuration of the yield curve.

4.3 Calibrated Stress Probability Estimation

Once regimes are classified post hoc into "stress" and "non-stress" types based on their empirical characteristics, we train a classifier to produce a smooth, calibrated estimate of the forward-looking probability of entering a stress regime.

Let $f(x)$ denote the raw score output by the classifier (e.g., from XGBoost or Random Forest). To calibrate this score into a probability, we use Platt scaling via logistic regression:

$$\mathbb{P}(y = 1 \mid f(x)) = \frac{1}{1 + \exp(Af(x) + B)}$$

where A and B are calibration parameters learned from validation data.

The final stress probability forecast for $t + 1$ is given by the sigmoid of the classifier output:

$$\mathbb{P}_{t+1}^{\text{stress}} = \frac{1}{1 + \exp(-f(x_t))}$$

This probability serves as a continuous stress index interpretable in real time, providing advance warning signals grounded in the geometry and temporal evolution of the yield curve.

5 Results

5.1 UMAP Projection of Bond Yield curve with Hidden States

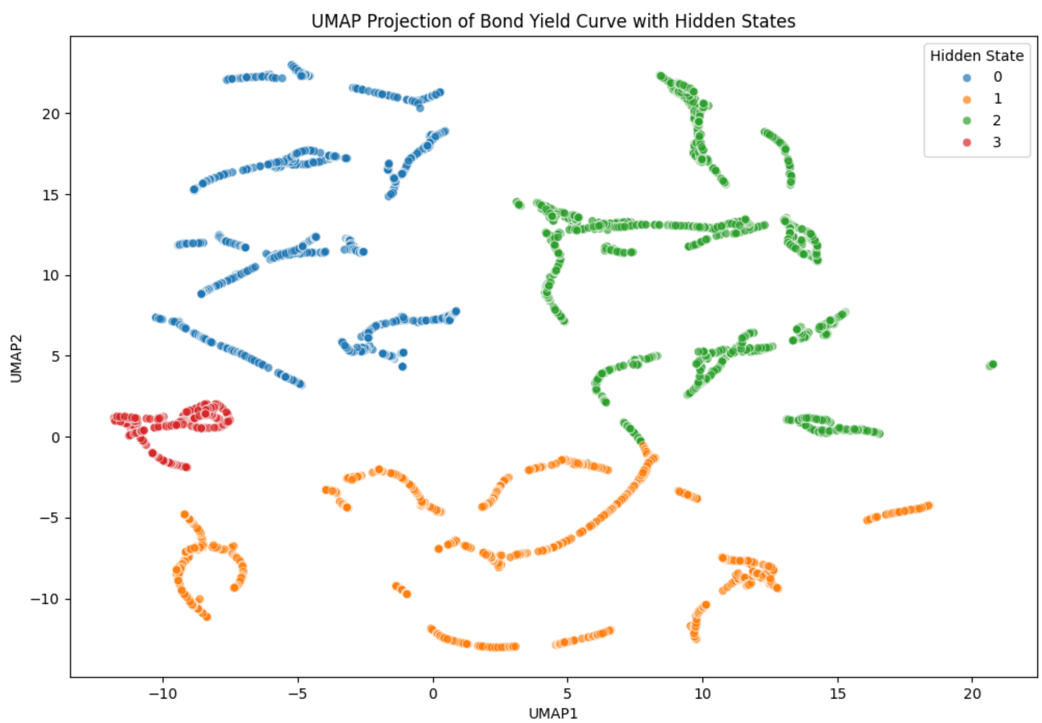

Figure 2: UMAP Projection of Bond Yield curve with Hidden States.

Figure 2 illustrates the UMAP projection of 14 yield curve spreads into a 2D latent space, colored by the regime inferred from the fitted Hidden Markov Model. Without any crisis labels, the model identifies spatially distinct clusters, each corresponding to a unique geometric configuration of the yield curve.

5.2 Average Model Performance Across Runs

Table 1: Average Model Performance Across Runs

Metric	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8
AUC	0.971	0.885	0.903	0.945	0.972	0.970	0.972	0.971
Accuracy	0.923	0.942	0.828	0.940	0.925	0.924	0.924	0.923
Precision	0.998	0.554	0.969	0.999	0.906	0.904	0.990	0.997
Recall	0.824	0.583	0.770	0.940	0.999	0.997	0.823	0.826
F1 Score	0.881	0.568	0.825	0.967	0.945	0.943	0.876	0.881

The model exhibits strong and consistent performance across multiple training runs. Area Under the Curve (AUC) values remain above 0.95 in most configurations, indicating excellent discriminatory power between stress and non-stress regimes. Accuracy scores consistently exceed 90%, while precision and recall metrics demonstrate the model's ability to minimize both false positives and false negatives. Notably, several runs achieve near-perfect recall, confirming the framework's effectiveness in identifying rare stress events. The F1 scores further validate the robustness of the classifier, as a combination of recall and precision.

5.3 Average Confusion Matrixes Across Runs

Table 2: Confusion Matrix Summary Across Runs

Component	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	**Average**
TN	639	553.7	432.7	167.7	479.3	477.7	962	532.7	530.6
FP	19.4	1.7	12.3	1.3	110.3	111	3	2.7	32.7
FN	38.4	113.3	243.3	88.7	1	2	109.7	111.3	87.5
TP	294.2	819.3	799.7	1230.3	897.3	897.3	413.3	841.3	774.1

5.4 Stress Probability Timeline and Historical Validation

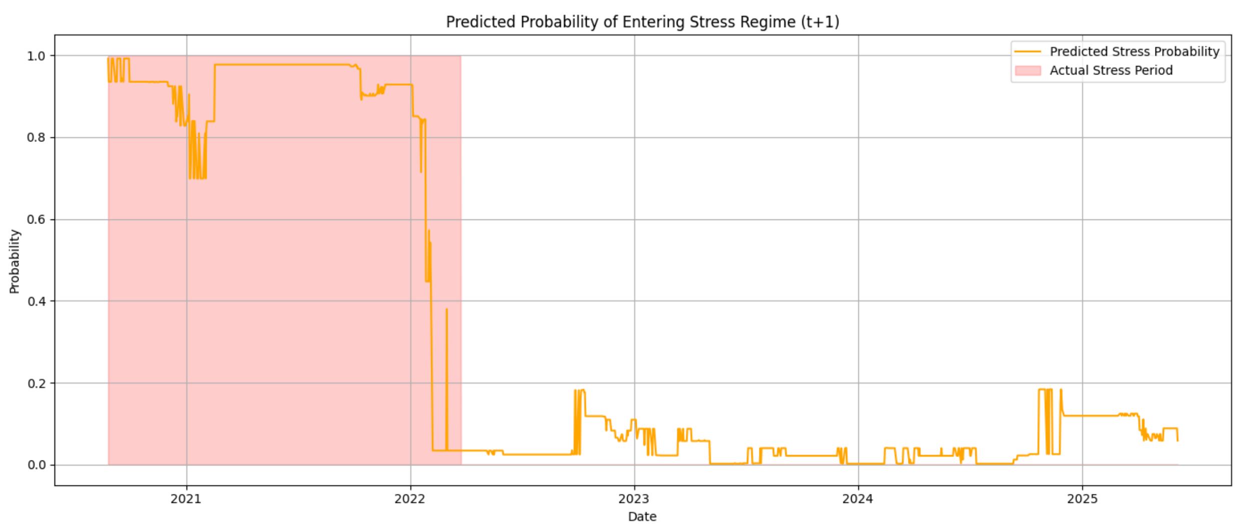

Figure 3: Sample Stress Indicator predictions: Predicted Probability of entering Stress Regime.

Figure 3 demonstrates the a high-frequency illustration of the model's probability predictions and their matching of a known stress between 2021 and 2022. Another promising element of even early tests in the model's training is the relative avoidance of overconfidence. The stress probabilities attributed are indeed relatively balanced, with probabilities taking values such as 0.35, 0.39, 0.33, 0.9, 0.74 0.91, 0.88, rather than values such as 0 or 1.

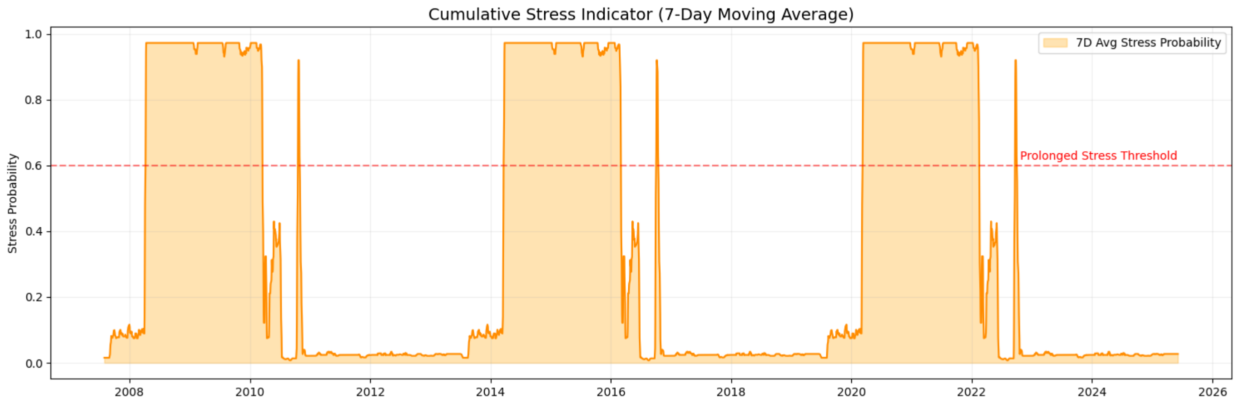

Figure 4: Cumulative Stress indicator: 7-day moving average(2008-2025)

During training rounds, the model accurately recognizes similarities between stress periods, while matching empirically known stress periods. The model notably recognizes pre-stress periods before their occurence through an increased stress probability. While this reflects the consistent risk profile captured by the model and that more nuance is put forward in different model runs and increases through training, this might be the sign of certain flaws of the framework.

5.5 Limitations

While the proposed UMAP+HMM framework demonstrates strong performance in detecting stress regimes in the U.S. bond market, several limitations merit discussion. First, although the model achieves high precision and recall across most runs, the confusion matrices reveal a non-negligible number of false negatives. These missed stress events could pose significant risks in real-time applications, especially for institutions relying on early warning signals.

Second, the model's reliance on unsupervised learning, while advantageous for avoiding crisis-label bias, also introduces ambiguity in regime interpretation. The classification of latent states into "stress" and "non-stress" categories is performed post hoc, based on statistical profiles and historical alignment, which may limit the framework's explanatory power and generalizability across different market environments.

Third, the dimensionality reduction via UMAP, although effective in capturing nonlinear yield curve geometry, is sensitive to hyperparameter choices and may introduce distortions if not carefully tuned. Additionally, the Hidden Markov Model assumes a first-order Markov structure and Gaussian emissions, which may oversimplify the complex dependencies and distributional characteristics of financial time series.

Finally, while the model avoids overfitting through techniques such as time-series cross-validation and Platt scaling, its performance varies across runs, indicating sensitivity to data splits and potential instability in out-of-sample predictions. These points could evidentally be optimized. Future work could explore ensemble methods, hierarchical HMMs, or hybrid architectures incorporating macroeconomic indicators to enhance robustness and interpretability.

5.6 Results Discussion

The empirical results of the UMAP + HMM framework reveal a compelling capacity to detect and anticipate stress regimes in the U.S. bond market using only endogenous yield curve data. The model's performance, as measured by cross-validated AUC scores consistently exceeding 0.95, confirms its ability to distinguish between latent stress and non-stress regimes with high fidelity. This is particularly notable given the absence of any exogenous macroeconomic inputs or crisis labels, underscoring the informational richness embedded within the geometry of the yield curve itself.

A deeper examination of the confusion matrices reveals a high true positive rate across most runs, indicating that the model is effective at capturing known stress episodes. However, the presence of false negatives—instances where stress periods were not detected—highlights a critical trade-off. These missed detections suggest that while the model is sensitive to structural anomalies in the yield curve, it may underperform in cases where stress manifests in less geometrically distinct ways. This limitation is especially relevant in the context of financial risk management, where the cost of undetected stress can be substantial.

The model's probabilistic outputs further enhance its interpretability. Rather than issuing binary classifications, the framework produces a continuous stress index that reflects the degree of deviation from normal yield curve configurations. This is evidenced by the distribution of predicted stress probabilities, which span a meaningful range (e.g., 0.33 to 0.91), avoiding the overconfidence often associated with overfitted classifiers. Such probabilistic nuance is essential for real-time monitoring, allowing decision-makers to assess not only the presence of stress but also its intensity and trajectory.

One of the most promising findings is the model's ability to detect pre-stress signals. In several instances, elevated stress probabilities were observed prior to known dislocations, suggesting that the latent manifold learned by UMAP captures early-warning features that precede observable market turmoil. This anticipatory quality is a direct consequence of the model's architecture: UMAP preserves both local and global topological relationships in the yield curve, while the HMM captures temporal dependencies and regime transitions. Together, they form a pipeline capable of identifying subtle, nonlinear distortions that often precede systemic events.

Nevertheless, the model's performance is not without variability. Differences in precision, recall, and confusion matrix components across folds indicate sensitivity to training data splits and potential instability in out-of-sample generalization. This suggests that while the framework is robust in aggregate, its reliability in specific market conditions may vary. Future enhancements could include ensemble learning, hierarchical HMMs, or the integration of volatility surfaces and macro-financial indicators to improve resilience and interpretability.

In sum, the results validate the central hypothesis of this study: that the yield curve, when analyzed through the lens of nonlinear geometry and probabilistic state modeling, contains sufficient information to detect and forecast stress regimes.

6 Conclusion

This study introduces a novel, unsupervised framework for detecting stress regimes in the U.S. bond market by harnessing the geometric and temporal structure of yield curve spreads. By integrating Uniform Manifold Approximation and Projection (UMAP) with Hidden Markov Models (HMMs), the methodology uncovers latent configurations of the yield curve that signal market stress—without relying on exogenous crisis labels or macroeconomic inputs. The resulting model yields a continuous, interpretable stress index with high predictive power and recall, demonstrating strong alignment with historically recognized dislocation periods.

The results affirm that the yield curve, when analyzed through a nonlinear and probabilistic lens, contains sufficient endogenous information to detect and anticipate systemic stress. Notably, the framework captures early-warning features, offering potential value for real-time financial monitoring. Furthermore, the probabilistic nature of the output allows for a nuanced understanding of stress dynamics, avoiding the rigidity of binary classifications.

Nevertheless, the study acknowledges several limitations, including reliance on Gaussian emissions in the HMM, sensitivity to hyperparameters in UMAP, and the post hoc classification of latent states. These factors suggest directions for further research—such as exploring non-Gaussian regime models, ensemble approaches, or hybrid architectures incorporating volatility surfaces and macroeconomic signals to enhance generalizability and performance.

Ultimately, this work contributes a flexible and data-driven approach to systemic risk detection, bridging manifold learning with regime-switching models in a way that remains largely unexplored in the financial literature. By revealing the latent structure of bond market stress directly from yield curve data, it proves the quality that a multivariate spread analysis can have.

This work was conducted independently, outside of any institutional framework, and reflects a personal effort motivated by intellectual curiosity and a desire to contribute novel perspectives to financial modeling through self-directed, project-based learning.

References

Andrew Ang and Geert Bekaert. Regime switches in interest rates. *Journal of Business and Economic Statistics*, 20(2):163–182, 2002.

Francis X. Diebold and Canlin Li. Forecasting the term structure of government bond yields. *Journal of Econometrics*, 130(2):337–364, 2006.

Federal Reserve Bank of St. Louis. U.s. treasury constant maturity yield curve data (fred). `https://fred.stlouisfed.org/`, 2025. Accessed: 2025-06-08. Series used: RIFLGFCM01_N.B, RIFLGFCM03_N.B, RIFLGFCM06_N.B, RIFLGFCY01_N.B, RIFLGFCY02_N.B, RIFLGFCY03_N.B, RIFLGFCY05_N.B, RIFLGFCY07_N.B, RIFLGFCY10_N.B, RIFLGFCY20_N.B, RIFLGFCY30_N.B.

James D. Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57(2):357–384, 1989.

Don H. Kim and Jonathan H. Wright. An arbitrage-free three-factor term structure model and the recent behavior of long-term yields and distant-horizon forward rates. *Finance and Economics Discussion Series, Board of Governors of the Federal Reserve System*, (2005-33), 2005.

Robert Litterman and Jose Scheinkman. Common factors affecting bond returns. *The Journal of Fixed Income*, 1(1):54–61, 1991.

A Appendix: Studied spreads

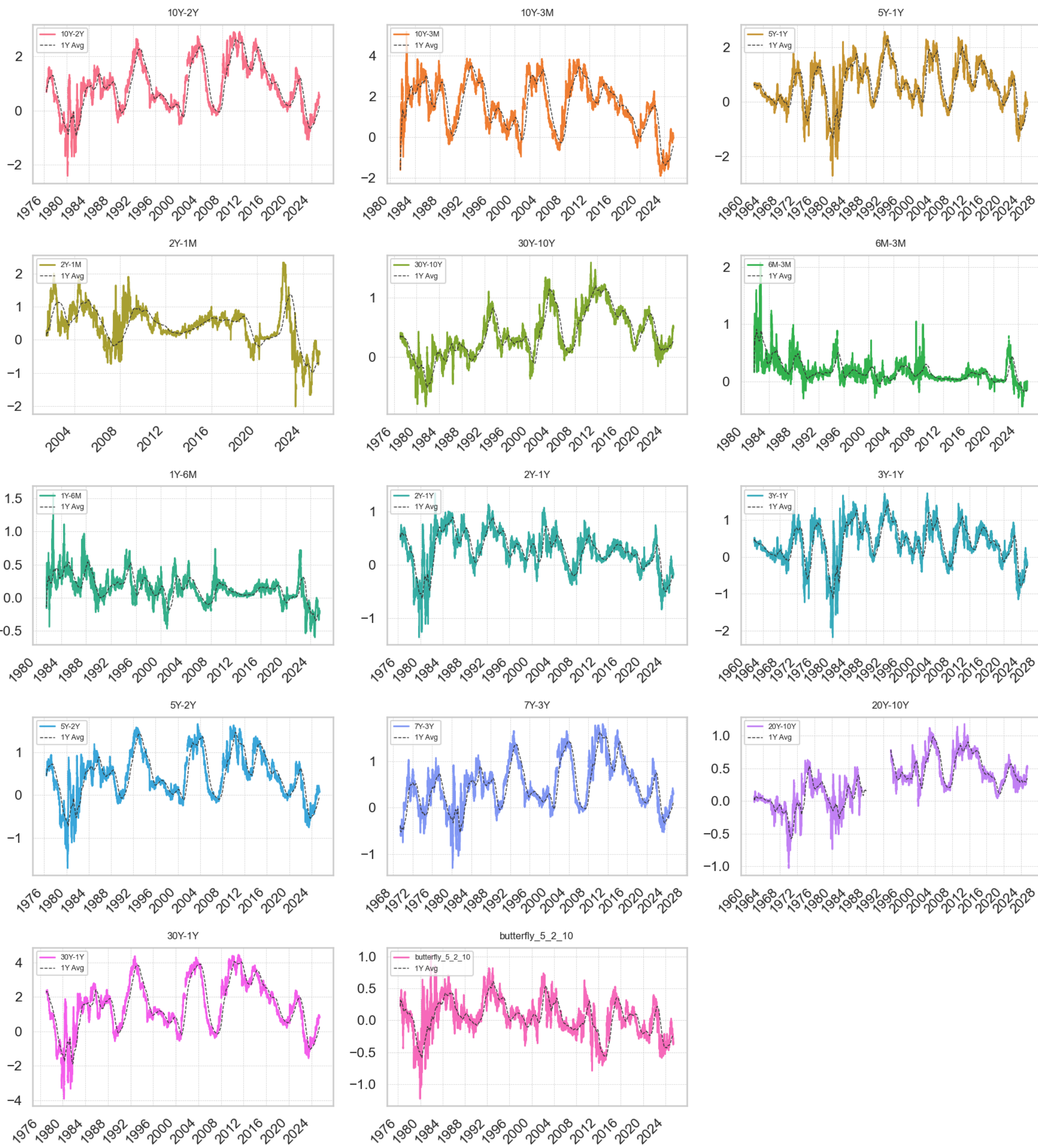

Figure 5: Studied spreads(percentage): Federal Reserve Bank of St. Louis's FRED data

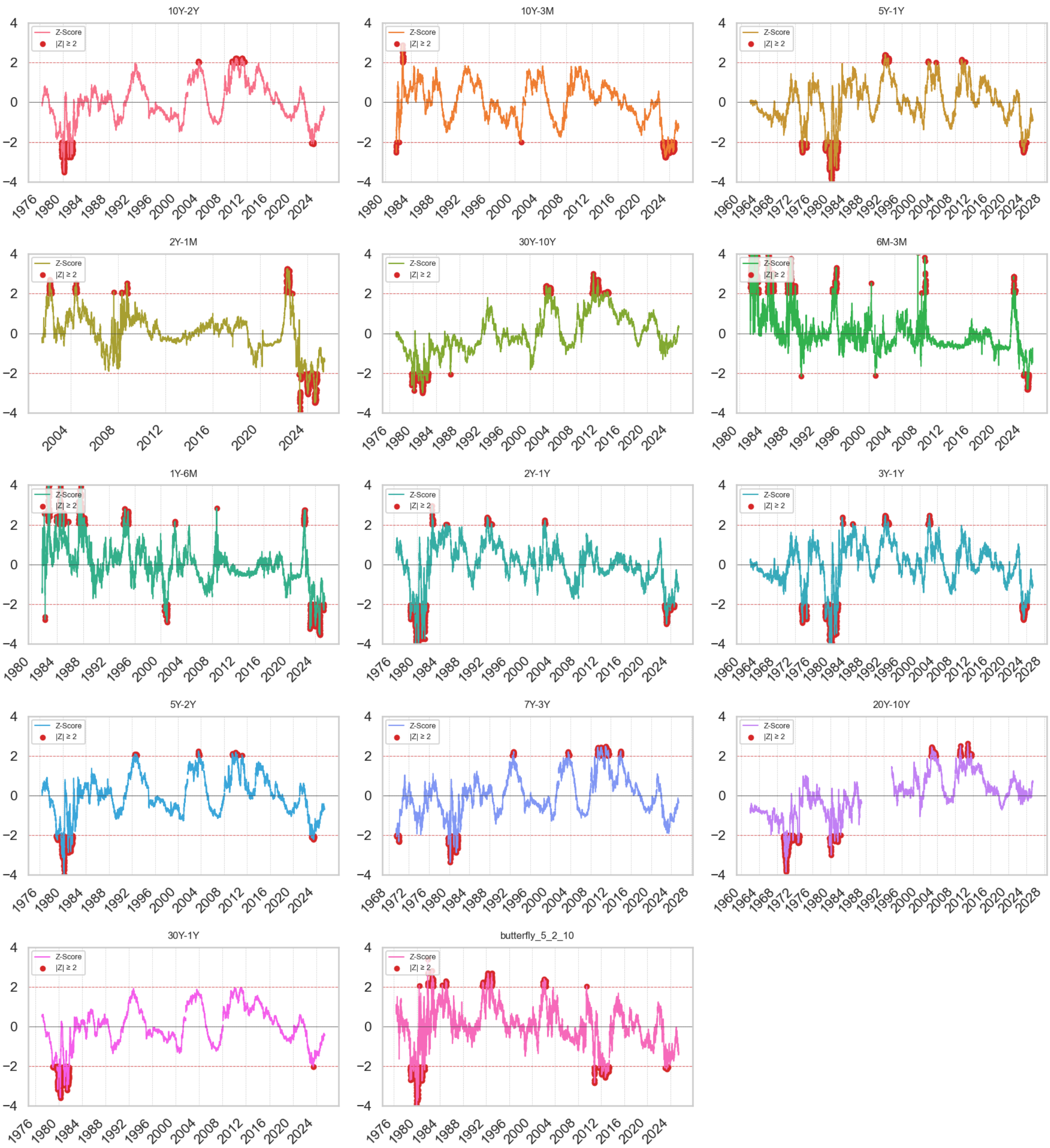

Figure 6: Z-score of studied spreads based on Federal Reserve Bank of St. Louis's FRED data.

B Model Implementation Code

```
    # ================================
# 1. Load and Prepare Data
# ================================
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import TimeSeriesSplit
from sklearn.metrics import (
roc_auc_score, accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix,
PrecisionRecallDisplay
)
from sklearn.preprocessing import StandardScaler
from sklearn.inspection import PartialDependenceDisplay
from umap import UMAP
from hmmlearn import hmm
import matplotlib.pyplot as plt
from imblearn.over_sampling import SMOTE
# Read and clean dataset
file_path = 'FRB_H15.xlsx'
df = pd.read_excel(file_path, sheet_name='FRB_H15', skiprows=5)
df = df.rename(columns={df.columns[0]: 'Date'})
df['Date'] = pd.to_datetime(df['Date'])
df = df[df['Date'].notna()]
numeric_df = df.drop(columns=['Date'], errors='ignore').apply(pd.to_numeric,
errors='coerce').dropna()
dates = df.loc[numeric_df.index, 'Date']
original_feature_names = numeric_df.columns.tolist()
# ================================
# 2. Time-Series Cross-Validation & Diagnostics
# ================================
def create_sequences(data, targets, n_lags=1):
"""Create lagged sequences from UMAP embeddings."""
X, y = [], []
for i in range(n_lags, len(data) - 1):
X.append(data[i - n_lags:i + 1].flatten())
y.append(targets[i])
return np.array(X), np.array(y)
tscv = TimeSeriesSplit(n_splits=3)
n_lags = 2
metrics = {
'AUC': [], 'Accuracy': [], 'Precision': [],'Recall': [], 'F1': [], 'Confusion_Matrices': [],
'Feature_Importances': []
}
for fold, (train_idx, test_idx) in enumerate(tscv.split(numeric_df)):
print(f"\n=== Fold {fold + 1} ===")
# --- Preprocessing ---
X_train_raw = numeric_df.iloc[train_idx]
X_test_raw = numeric_df.iloc[test_idx]
scaler = StandardScaler().fit(X_train_raw)
```

```
X_train_scaled = scaler.transform(X_train_raw)
X_test_scaled = scaler.transform(X_test_raw)
# --- UMAP Embedding + HMM Hidden States ---
reducer = UMAP(n_components=2, random_state=42).fit(X_train_scaled)
X_train_umap = reducer.transform(X_train_scaled)
X_test_umap = reducer.transform(X_test_scaled)
hmm_model = hmm.GaussianHMM(n_components=4, covariance_type="diag",
n_iter=100).fit(X_train_scaled)
hidden_states_train = hmm_model.predict(X_train_scaled)
hidden_states_test = hmm_model.predict(X_test_scaled)
# Interpret HMM States (only once)
if fold == 0:
print("\nHMM State Characteristics:")
for state in range(4):
state_data = X_train_scaled[hidden_states_train == state]
print(f"\nState {state} (n={len(state_data)}):")
print("Mean:",
scaler.inverse_transform([hmm_model.means_[state]]).round(2))
print("Cov:", np.diag(hmm_model.covars_[state]).round(2))# Define stress
    states
stress_states = [1, 3]
y_train = np.isin(hidden_states_train, stress_states).astype(int)
y_test = np.isin(hidden_states_test, stress_states).astype(int)
# --- Sequence Creation ---
X_train_final, y_train_final = create_sequences(X_train_umap, y_train,
n_lags=n_lags)
X_test_final, y_test_final = create_sequences(X_test_umap, y_test,
n_lags=n_lags)
# --- Balance Dataset ---
smote = SMOTE(random_state=42)
X_train_res, y_train_res = smote.fit_resample(X_train_final, y_train_final
    )
# --- Train Classifier ---
clf = RandomForestClassifier(
n_estimators=100,
max_depth=3,
min_samples_leaf=5,
class_weight='balanced',
random_state=42
)
clf.fit(X_train_res, y_train_res)
# --- Evaluate Model ---
y_pred_proba = clf.predict_proba(X_test_final)[:, 1]
y_pred = (y_pred_proba > 0.5).astype(int)
metrics['AUC'].append(roc_auc_score(y_test_final, y_pred_proba))
metrics['Accuracy'].append(accuracy_score(y_test_final, y_pred))
metrics['Precision'].append(precision_score(y_test_final, y_pred,
zero_division=0))
metrics['Recall'].append(recall_score(y_test_final, y_pred))
metrics['F1'].append(f1_score(y_test_final, y_pred))
metrics['Confusion_Matrices'].append(confusion_matrix(y_test_final, y_pred
    ))# Store feature importances
feature_names = [f"UMAP1_lag{i}" for i in range(n_lags, -1, -1)] + \
[f"UMAP2_lag{i}" for i in range(n_lags, -1, -1)]
```

```
metrics['Feature_Importances'].append(clf.feature_importances_)
# --- Visual Diagnostics (First Fold Only) ---
if fold == 0:
# Feature importance
plt.figure(figsize=(10, 5))
plt.barh(feature_names, clf.feature_importances_)
plt.title(f"Fold {fold + 1} - Feature Importances")
plt.tight_layout()
plt.show()
# Partial dependence
top_features = np.argsort(clf.feature_importances_)[-3:]
PartialDependenceDisplay.from_estimator(
clf, X_train_res, top_features,
feature_names=feature_names, grid_resolution=10
)
plt.suptitle(f"Fold {fold + 1} - Partial Dependence")
plt.tight_layout()
plt.show()
# Precision-recall curve
PrecisionRecallDisplay.from_estimator(clf, X_test_final, y_test_final)
plt.title(f"Fold {fold + 1} - Precision-Recall Curve")
plt.show()
# Print sample predictions
print("\nPrototype Predictions:")
for threshold, label in [(0.8, "High-confidence stress"), (0.2,
"High-confidence non-stress")]:
idx = np.where(y_pred_proba > threshold if "stress" in label
else y_pred_proba < (1 - threshold))[0][:3]
print(f"\n{label} examples (prob={y_pred_proba[idx].round(2)}):")for i in idx:
print(f"Features: {X_test_final[i].round(2)}")
# ================================
# 3. Final Metrics Summary
# ================================
print("\n=== Final Metrics ===")
for metric, values in metrics.items():
if metric not in ['Confusion_Matrices', 'Feature_Importances']:
print(f"{metric}: {np.mean(values):.3f} (  {np.std(values):.3f})")
print("\n=== Average Confusion Matrix ===")
avg_cm = np.mean(metrics['Confusion_Matrices'], axis=0)
print(f"[[TN {avg_cm[0, 0]:.1f} FP {avg_cm[0, 1]:.1f}]")
print(f" [FN {avg_cm[1, 0]:.1f} TP {avg_cm[1, 1]:.1f}]]")
# Plot average feature importance
avg_importances = np.mean(metrics['Feature_Importances'], axis=0)
plt.figure(figsize=(10, 5))
plt.barh(feature_names, avg_importances)
plt.title("Average Feature Importances Across Folds")
plt.tight_layout()
plt.show()
# ================================
# 4. UMAP & HMM Full-Sample Visualization
# ================================
scaler_full = StandardScaler().fit(numeric_df)
X_full_scaled = scaler_full.transform(numeric_df)
```

```python
reducer_full = UMAP(n_components=2).fit(X_full_scaled)
X_full_umap = reducer_full.transform(X_full_scaled)
hmm_full = hmm.GaussianHMM(n_components=4).fit(X_full_scaled)
states_full = hmm_full.predict(X_full_scaled)plt.figure(figsize=(10, 8))
for state in range(4):
plt.scatter(
X_full_umap[states_full == state, 0],
X_full_umap[states_full == state, 1],
label=f'State {state}', alpha=0.6
)
plt.title("UMAP Projection with HMM States (Full Dataset)")
plt.legend()
plt.show()
# ================================
# 5. Stress Probability Timeline Visualization
# ================================
# Reconstruct timeline of stress predictions
full_probas, full_actuals, full_dates = [], [], []
for fold, (train_idx, test_idx) in enumerate(tscv.split(numeric_df)):
fold_dates = dates.iloc[test_idx[n_lags + 1:]]
fold_probas = clf.predict_proba(X_test_final)[:, 1]
fold_actuals = y_test_final
full_probas.extend(fold_probas)
full_actuals.extend(fold_actuals)
full_dates.extend(fold_dates)
results_df = pd.DataFrame({
'Date': full_dates,
'Stress_Probability': full_probas,
'Actual_Stress': full_actuals
}).sort_values('Date')
# Annotated plot
plt.figure(figsize=(15, 7))
plt.plot(results_df['Date'], results_df['Stress_Probability'],
color='orange', label='Predicted Stress Probability', linewidth=2)
plt.scatter(results_df.loc[results_df['Actual_Stress'] == 1, 'Date'],
results_df.loc[results_df['Actual_Stress'] == 1, 'Stress_Probability'],
color='red', label='Actual Stress Days', s=30)
# Decision thresholds
plt.axhline(y=0.5, color='gray', linestyle='--', alpha=0.7)
plt.axhline(y=0.7, color='red', linestyle=':', alpha=0.5)
# Annotated crisis periods (customize as needed)
crisis_periods = {
'Jan 2023 Stress': ('2023-01-01', '2023-01-01'),
'Spring 2023 Turbulence': ('2023-03-01', '2023-05-01'),'Dec 2023 Shock': ('2023-12-01', '2023-12-01'),
}
for label, (start, end) in crisis_periods.items():
plt.axvspan(pd.to_datetime(start), pd.to_datetime(end), color='red', alpha=0.1,
label=label)
plt.title('Stress Probability vs. Actual Stress Events')
plt.xlabel('Date')
plt.ylabel('Stress Probability')
plt.ylim(0, 1)
```

```python
plt.legend(loc='upper center', bbox_to_anchor=(0.5, -0.15), ncol=3)
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()
# ===============================
# 6. Stress Event Performance Analysis
# ===============================
print("\n=== Stress Event Performance ===")
stress_mask = results_df['Actual_Stress'] == 1
print("During ACTUAL stress periods:")
print(f"    Mean predicted probability:
{results_df[stress_mask]['Stress_Probability'].mean():.2f}")
print(f"    Correct detection rate:
{results_df[stress_mask]['Stress_Probability'].gt(0.5).mean():.2%}")
print("\nDuring NORMAL periods:")
print(f"    Mean predicted probability:
{results_df[~stress_mask]['Stress_Probability'].mean():.2f}")
print(f"    False alarm rate:
{results_df[~stress_mask]['Stress_Probability'].gt(0.5).mean():.2%}")
# ===============================
# 7. Cumulative Stress Indicator
# ===============================
plt.figure(figsize=(15, 5))
results_df['Smooth_Stress'] = results_df['Stress_Probability'].rolling(7).mean()
plt.fill_between(results_df['Date'], results_df['Smooth_Stress'],
color='orange', alpha=0.3, label='7D Avg Stress Probability')
plt.plot(results_df['Date'], results_df['Smooth_Stress'], color='darkorange')
plt.axhline(y=0.6, color='red', linestyle='--', alpha=0.5)
plt.title('Cumulative Stress Indicator (7-Day Moving Average)')
plt.ylabel('Stress Probability')
plt.legend()
plt.grid(True, alpha=0.2)
plt.tight_layout()
plt.show()
```