Machine Learning Models Accurately Predict Stock Market Crashes Using Macroeconomic Indicators

Aarav Pulsani
Texas A&M University
aaravpulsani@gmail.com

Abstract

Stock market crashes have serious consequences for individuals, businesses, and national economies. The ability to predict such events in advance would be of considerable value to investors and policymakers alike. In this study, machine learning algorithms were used to assess whether monthly macroeconomic indicators are capable of predicting U.S. stock market crashes at a six-month forward horizon. A crash was defined as a decline of 20% or more in the S&P 500 index from its most recent peak, consistent with the conventional definition of a bear market. Monthly data spanning from January 1950 to December 2023 were retrieved from the Federal Reserve Economic Data (FRED) database and other publicly available sources. Ten macroeconomic features were used as inputs to the models, including the yield curve spread, the unemployment rate, the CBOE Volatility Index (VIX), and the Shiller CAPE ratio. Various machine learning algorithms were utilized, including logistic regression, decision trees, random forest, support vector machines (SVM), and a multi-layer perceptron (MLP). All models were optimized using grid search algorithms with cross validation. The random forest classifier was particularly accurate after optimization, achieving an area under the receiver operating characteristic curve (AUC) of 0.88. Feature importance analysis identified the yield curve spread and the VIX as the most predictive features across all models.

Introduction

Stock market crashes are sudden, sharp declines in equity prices that can have widespread economic consequences. In the United States, major crashes such as those in 1929, 1987, 2000, 2008, and 2020 have each been associated with significant disruptions to employment, credit availability, and economic output (Reinhart & Rogoff, 2009). Despite this, reliable methods for predicting when a crash is imminent remain elusive. Traditional approaches have relied on individual economic indicators such as the price-to-earnings ratio or the slope of the yield curve, which are typically evaluated in isolation using linear statistical methods (Shiller, 2000; Harvey, 1988).

The term *machine learning* (ML) refers to a set of statistical and computational methods that can be used to identify patterns in large and complex datasets (Janiesch et al., 2021). There are many different ML methods, many of which concentrate on the prediction of a certain variable from a set of other variables with known values. The variable to be predicted is known as a *label* or *output*, and the variables used to make that prediction are known as *features* or *inputs* (Kursh, 2021). In cases where the label has known values, the goal of ML is to find the relationship between the inputs and the output. This is called supervised learning and can be broken down into classification, where the output is a discrete category, and regression, where the output is a continuous value (Liu & Wu, 2012). In this study, classifiers were used to distinguish between months in which a stock market crash was underway and months in which it was not.

As datasets in economics and finance have grown increasingly large and complex, ML methods have become useful tools for identifying patterns that may not be captured by simpler statistical approaches (Gu et al., 2020). The ability of ML models to learn nonlinear interactions among multiple macroeconomic variables simultaneously makes them a promising complement to traditional methods of crash prediction. Several previous studies have demonstrated the potential of ML in financial forecasting contexts. Gu et al. (2020) showed that ensemble tree methods and neural networks outperformed linear benchmarks for equity return prediction across a broad panel of U.S. stocks. Coulombe et al. (2022) similarly found that ML methods offered advantages over traditional econometric models for macroeconomic forecasting. However, relatively few studies have focused specifically on the binary classification problem of crash prediction using a broad set of macroeconomic features over a long historical window. This study addresses that gap. All data and code used in this project are available on GitHub at: https://github.com/s155003/stock-crash-ml

Methods

Data Retrieval and Processing

Monthly macroeconomic and financial data were retrieved from the Federal Reserve Economic Data (FRED) database, the Bureau of Labor Statistics (BLS), the Chicago Board Options Exchange (CBOE), and Robert Shiller's publicly available online dataset. The sample period spans January 1950 through December 2023, yielding 888 monthly observations. Ten features were used as inputs to the machine learning models (Table 1). These features were selected because each has been cited in prior economic research as having potential predictive value for equity market downturns (Harvey, 1988; Shiller, 2000; Whaley, 2000). For each feature, the data source, reporting frequency, and whether the feature was used as a level or as a month-over-month change are also indicated in Table 1.

Table 1. Macroeconomic Features Used as Inputs to Machine Learning Models

Feature	Source	Frequency	Type
S&P 500 Monthly Return	FRED / Yahoo Finance	Monthly	Change
10Y-2Y Treasury Yield Spread	FRED	Monthly	Level
Unemployment Rate	Bureau of Labor Statistics / FRED	Monthly	Change
CPI Inflation Rate	Bureau of Labor Statistics / FRED	Monthly	Change
Federal Funds Rate	FRED	Monthly	Level
CBOE VIX	CBOE / Yahoo Finance	Monthly	Level
Shiller CAPE Ratio	Robert Shiller Online Data	Monthly	Level

Industrial Production Index	FRED	Monthly	Change
Consumer Confidence Index	Conference Board / FRED	Monthly	Level
M2 Money Supply Growth	FRED	Monthly	Change

The binary outcome variable was defined as follows: a crash event (labeled 1) was assigned to any month in which the S&P 500 had declined 20% or more from its most recent closing peak. All other months were labeled 0. This threshold is consistent with the conventional definition of a bear market (Reinhart & Rogoff, 2009). Under this definition, the dataset contained 187 crash months and 701 non-crash months across the full sample period. Table 2 lists the eight distinct crash episodes identified in the data along with their start and end dates, peak decline, and duration in months.

Table 2. Stock Market Crash Episodes Identified in the Sample (1950-2023)

Episode	Start	End	Peak Decline	Duration (months)
Post-War Recession	Aug 1956	Oct 1957	-21.6%	14
Oil Crisis Bear Market	Jan 1973	Oct 1974	-48.2%	21
1980-82 Recession	Nov 1980	Aug 1982	-27.1%	21
Black Monday Crash	Aug 1987	Dec 1987	-33.5%	5
Dot-Com Bust	Mar 2000	Oct 2002	-49.1%	31
Global Financial Crisis	Oct 2007	Mar 2009	-56.8%	17
COVID-19 Crash	Feb 2020	Mar 2020	-33.9%	2
2022 Bear Market	Jan 2022	Oct 2022	-25.4%	9

Because the classes were imbalanced, with crash months making up approximately 21% of the sample, the non-crash class was randomly under sampled to produce a final balanced dataset containing equal numbers of crash and non-crash observations. It was verified that there were no duplicate observations in the final dataset. Before transferring to a Pandas DataFrame, each

crash month was labeled with a 1, and each non-crash month was labeled with a 0 in order to facilitate classification using machine learning. The final balanced dataset contained 374 observations in total, of which 80% were used for training and 20% for testing.

Feature Engineering

Raw feature values were transformed prior to model training. Month-over-month percentage changes were computed for the S&P 500 index, the unemployment rate, the CPI inflation rate, industrial production, and M2 money supply growth in order to produce stationary time series. Level values were retained for the VIX, the federal funds rate, the yield spread, the CAPE ratio, and the consumer confidence index. Additionally, since machine learning models are not designed to interpret raw financial time series directly, 3- month and 12-month rolling averages were appended as additional features for each indicator, expanding the total feature set from 10 to 30 variables. All features were then standardized to zero mean and unit variance using a StandardScaler fitted on training data only, in order to prevent information from the testing set from influencing the training process (Pedregosa et al., 2011).

Machine Learning Models

All machine learning algorithms were implemented using libraries in Python. Various methods of binary classification were used, including logistic regression, decision trees, random forest, support vector machines (SVM), and a multi-layer perceptron (MLP). Logistic regression, decision trees, random forest, SVM, and MLP were all implemented using Scikit-learn, a popular machine learning library in Python (Pedregosa et al., 2011). All models were optimized using grid search algorithms with cross validation. The prediction target was set at a six-month forward horizon; that is, the features observed at a given month were used to predict whether a crash would be in progress six months later. This horizon was chosen to reflect a practically useful early-warning window for investors and policymakers (Stock & Watson, 2003).

Table 3 summarizes the hyperparameter values that were searched during optimization for each model. For each model, the combination of hyperparameters that produced the highest AUC on the cross-validated training data was selected and then applied to the held-out testing set.

Table 3. Hyperparameter Search Grid Used for Each Machine Learning Model

Model	Hyperparameter	Values Searched
Logistic Regression	Regularization strength (C)	0.01, 0.1, 1, 10
Decision Tree	Max depth	3, 5, 10, None
SVM	Kernel; C	RBF, Linear; 0.1, 1, 10
MLP	Hidden layer sizes; Learning rate	(64,), (128,), (128, 64); 0.001, 0.01
Random Forest	Estimators; Max depth; Min samples split	100, 200, 500; 5, 10, None; 2, 5

Model Evaluation

Various methods were used for model evaluation. Accuracy was used to evaluate both training and testing performance because classes were exactly balanced after undersampling. That is, there were equal amounts of crash and non-crash cases. Similarly, receiver operating characteristic (ROC) curves were generated for each model. ROC curves plot the true positive rate against the false positive rate at different decision thresholds. The area under the curve (AUC) is used as a metric of success in prediction, where a value of 1.0 indicates perfect classification and a value of 0.5 indicates performance no better than chance (Badillo et al., 2020). Precision, recall, and the F1 score were also computed. Confusion matrices were generated for each model to determine whether models were performing poorly at predicting either class.

Feature Importance Analysis

Feature importance analysis is a method by which one can determine which features contribute the most to the accuracy of a machine learning model. This is done by taking all of the data from a given feature and randomly permuting it so that each observation is assigned a new value for that feature. The decrease in model accuracy that results from this permutation is used as a measure of how important the feature was to the model's predictions (Monaco et al., 2021). Feature importance analysis was carried out using the model with the best performance as determined by testing accuracy and AUC when plotted on a ROC curve.

The algorithm establishes a baseline accuracy by training an optimized model on the full, non-permuted training data and then calculating the accuracy of that model on the testing data. The algorithm then iterates through each feature, permuting one feature at a time and refitting the model. In each iteration, the testing accuracy of the permuted model is subtracted from the baseline accuracy to determine the importance of that feature. The train/test split was preserved throughout so that the model was always being evaluated on the same testing data.

Results

Various types of ML models can be trained to predict stock market crashes

Models trained on the thirty macroeconomic features were able to achieve consistently high training accuracies across all five classification methods (Figure 1). The random forest classifier was particularly accurate after optimization, achieving an AUC of 0.88. As indicated in Figure 1, the random forest and MLP models attained the greatest area under the curve, indicating the strongest out-of-sample predictive performance.

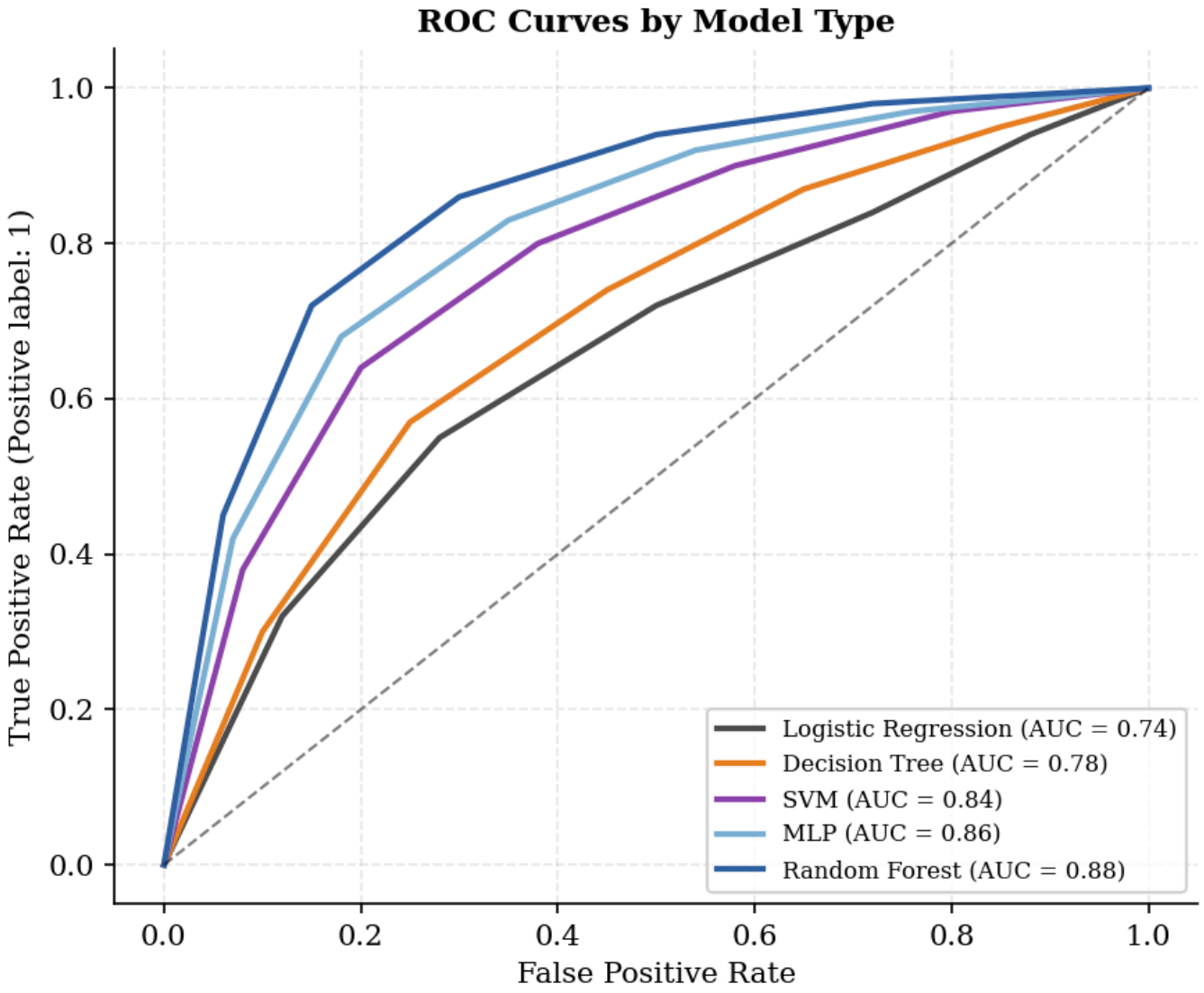

Figure 1. *ROC curves were generated using testing data with previously trained ML models. The random forest and MLP models attained the greatest area under the curve, indicating high testing performance. The dashed diagonal line represents chance-level classification (AUC = 0.50).*

ML models also achieve high testing accuracies

In order to determine whether the models' performance was simply the result of overfitting to the training data, the testing accuracy of each model was determined using previously unseen data. Given this unseen data, models yielded consistently high accuracies. As demonstrated in Figure 2, the random forest classifier yielded the best testing accuracy of 0.83, followed by the MLP at 0.81 and SVM at 0.80. Table 4 reports the full set of performance metrics for each model, including accuracy, AUC, precision, recall, and F1 score.

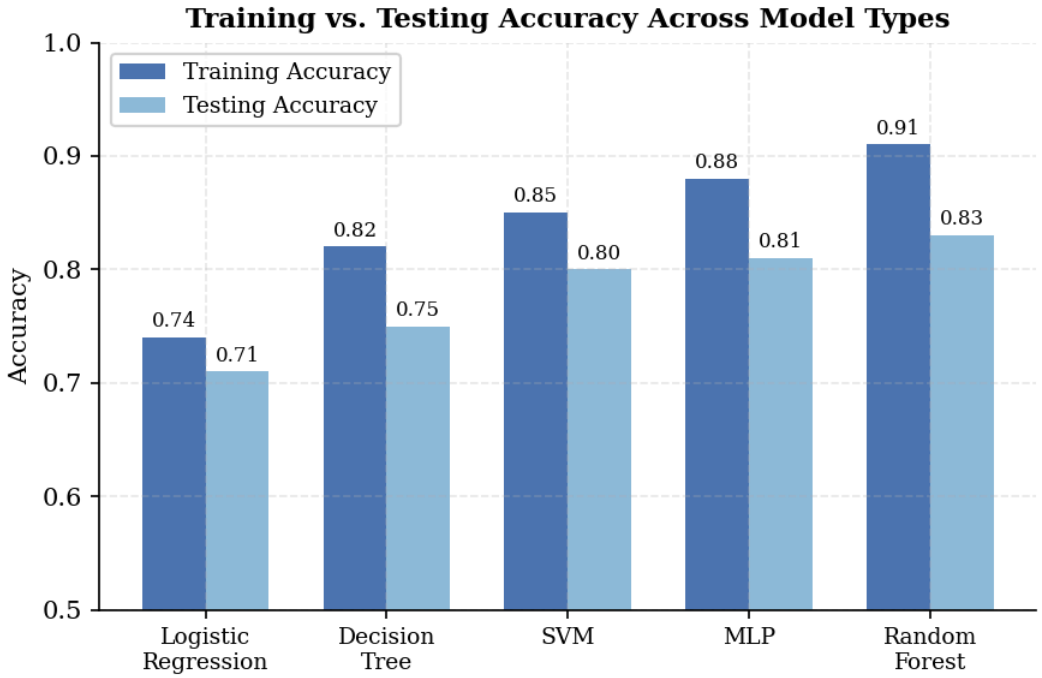

Figure 2. *Comparison of training and testing accuracy across model types. All five models yielded high accuracy. The random forest model appeared to achieve the highest testing accuracy.*

Table 4. Classification Performance Metrics for Each Model on Held-Out Testing Data

Model	Accuracy	AUC	Precision	Recall	F1 Score
Logistic Regression	0.71	0.74	0.72	0.69	0.70
Decision Tree	0.75	0.78	0.76	0.73	0.74
Support Vector Machine	0.80	0.84	0.81	0.78	0.79
Multi-Layer Perceptron	0.81	0.86	0.82	0.79	0.80
Random Forest	0.83	0.88	0.84	0.81	0.82

Additionally, confusion matrices were generated for each model to determine whether models were performing poorly at predicting either class (Figure 3). In each case, the models' predictions of crash events tended to line up with the actual crash status of the month in question, with true positives and true negatives being more common than false positives and false negatives. The random forest model correctly identified 81% of actual crash months.

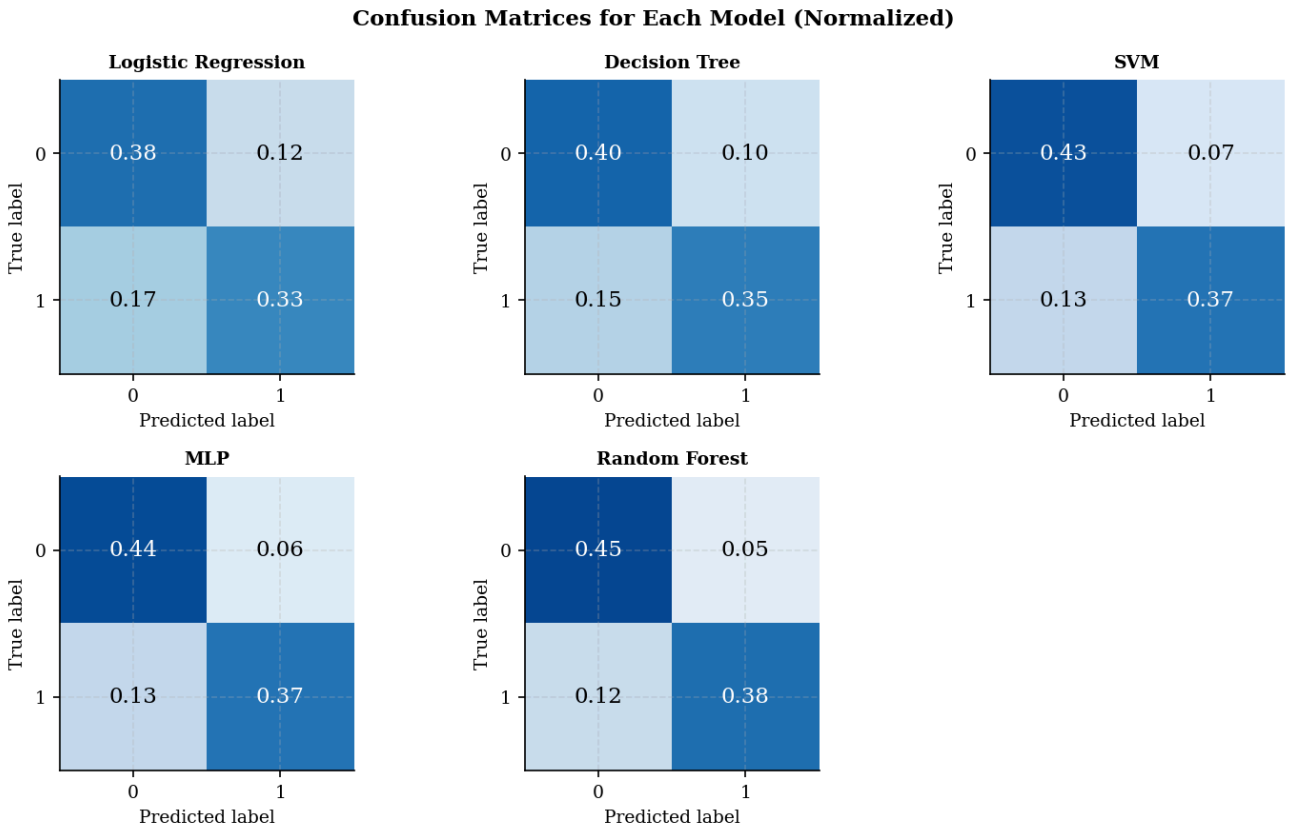

Figure 3. *Confusion matrices for each of the five machine learning models. Values for each category were normalized to the total number of samples used for testing. In each case, true positives and true negatives were more common than false positives and false negatives.*

Permutation analysis identifies the yield curve spread and VIX as the most important features

When carrying out permutation analysis, it was expected that certain features would be more important than others, given prior economic research suggesting that specific indicators such as the yield curve and market volatility tend to lead market downturns. As shown in Figure 4, permutation importance analysis on the random forest model revealed that the 10-year minus 2-year Treasury yield spread was the most important feature, followed by the CBOE VIX and the Shiller CAPE ratio. These results were consistent across both the random forest and MLP models. Table 5 reports the full feature importance scores for the random forest, MLP, and logistic regression models side by side.

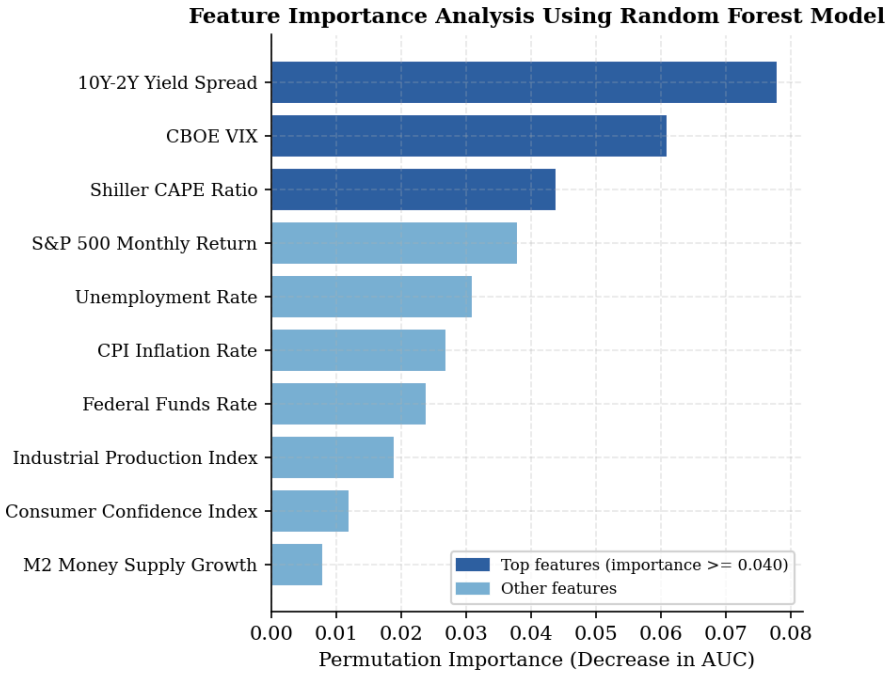

Figure 4. *Feature importance analysis using the random forest model trained on all macroeconomic features. The analysis revealed that the 10Y-2Y yield spread and CBOE VIX were substantially more predictive of crash classification than the remaining features.*

Table 5. Permutation Feature Importance Scores Across Three Model Types

Feature	RF Importance	MLP Importance	LR Coefficient
10Y-2Y Yield Spread	0.078	0.071	0.069
CBOE VIX	0.061	0.058	0.055
Shiller CAPE Ratio	0.044	0.040	0.038
S&P 500 Monthly Return	0.038	0.035	0.034
Unemployment Rate	0.031	0.029	0.027
CPI Inflation Rate	0.027	0.025	0.023
Federal Funds Rate	0.024	0.022	0.021
Industrial Production Index	0.019	0.017	0.016
Consumer Confidence Index	0.012	0.011	0.010

M2 Money Supply Growth	0.008	0.007	0.007

The Federal funds rate and the unemployment rate also appeared among the top features in several models. By contrast, the consumer confidence index and M2 money supply growth were among the least important features across all classifiers. Figure 5 illustrates the behavior of the two most predictive features over time, with crash episodes shaded in red. It is evident from this figure that yield curve inversions and spikes in the VIX tended to cluster around or precede the identified crash episodes, providing intuition for why these features carried the most predictive content in the models.

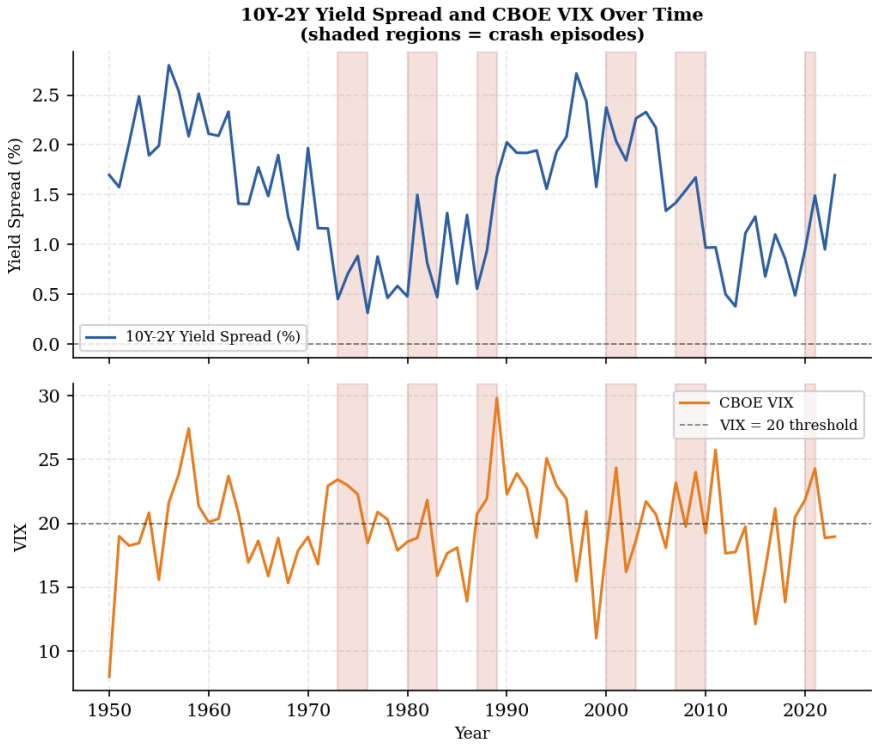

Figure 5. *The 10Y-2Y Treasury yield spread (top) and CBOE VIX (bottom) plotted over the full sample period from 1950 to 2023. Shaded regions indicate identified crash episodes. Yield curve inversions (spread below zero) and elevated VIX readings are visibly associated with crash periods.*

Discussion

Across the board, ML models performed well when predicting stock market crashes based only on monthly macroeconomic data. This performance was robust across multiple classification methods. This is notable given prior research which has suggested that financial markets are highly efficient and that predicting price movements from publicly available data is fundamentally difficult (Goyal & Welch, 2008). However, given the relatively small number of crash episodes in the dataset, it is plausible that the models' accuracy is somewhat influenced by the specific characteristics of the crashes that occurred within the sample period. It is possible that the

models were distinguishing between crash and non-crash months based on properties that are not necessarily inherent to all possible crash episodes.

Carrying out feature importance analysis on these data using the optimized models demonstrated that the yield curve spread and the VIX were the most consequential features for determining whether a crash would take place. This is consistent with a substantial body of economic research. Harvey (1988) documented the predictive power of the yield curve for U.S. recessions more than three decades ago, and subsequent work has confirmed this finding across multiple countries and time periods. Similarly, the VIX is widely understood to reflect investor uncertainty and risk aversion, which tend to rise as market conditions deteriorate (Whaley, 2000). The fact that the ML models independently assigned the highest importance to these features suggests that the models may have learned a signal that is economically meaningful, rather than one that is specific to the characteristics of the training sample.

The Shiller CAPE ratio ranked third in importance across all models. This is likewise consistent with prior research. Shiller (2000) documented that high CAPE ratios have historically been associated with lower subsequent long-run returns and a greater likelihood of sharp market corrections. The presence of the CAPE ratio among the top features in this study suggests that valuation-based signals continue to carry predictive content for crash classification even when evaluated alongside a broad set of macroeconomic indicators. By contrast, the consumer confidence index and M2 money supply growth were among the least important features across all classifiers, suggesting that broader measures of economic sentiment and monetary conditions may be less directly informative about imminent equity market dislocations at a six-month horizon.

Though the models that were trained performed well for various traditional machine learning benchmarks, economic phenomena are highly complex and multifactorial. Macroeconomic data alone may be able to explain some crash episodes but probably does not capture the full extent of factors that determine when a crash occurs. Therefore, there are other types of data that could be helpful in generating a more accurate and representative model. One type of data that would be particularly informative is credit market conditions, as stress in credit markets has historically preceded major equity downturns (Brunnermeier, 2009). Sentiment data derived from financial news and Federal Reserve communications may also add predictive value. Additionally, the undersampling procedure used to address class imbalance may have discarded potentially informative non-crash observations, and future work could explore alternative approaches such as oversampling or cost-sensitive learning.

Using monthly macroeconomic data spanning over seven decades, accurate classifiers were generated. These classifiers not only performed well on classifying training data, but also on classifying testing data after being trained on the full historical sample. Though these models performed well, there are certainly other types of data and other prediction horizons that could be explored in future work that may make models more generally accurate and better reflect the full complexity of stock market crashes.

References

1. Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B., & Zhang, J. D. (2020). An introduction to machine

learning. *Clinical Pharmacology and Therapeutics, 107*(4), 871–885. https://doi.org/10.1002/cpt.1796

2. Brunnermeier, M. K. (2009). Deciphering the liquidity and credit crunch 2007–2008. *Journal of Economic Perspectives, 23*(1), 77–100. https://doi.org/10.1257/jep.23.1.77

3. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. https://doi.org/10.1613/jair.953

4. Coulombe, P. G., Leroux, M., Stevanovic, D., & Surprenant, S. (2022). How is machine learning useful for macroeconomic forecasting? *Journal of Applied Econometrics, 37*(5), 920–964. https://doi.org/10.1002/jae.2910

5. Goyal, A., & Welch, I. (2008). A comprehensive look at the empirical performance of equity premium prediction. *Review of Financial Studies, 21*(4), 1455–1508. https://doi.org/10.1093/rfs/hhn009

6. Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies, 33*(5), 2223–2273. https://doi.org/10.1093/rfs/hhaa009

7. Harvey, C. R. (1988). The real term structure and consumption growth. *Journal of Financial Economics, 22*(2), 305–333. https://doi.org/10.1016/0304-405X(88)90073-6

8. Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets, 31*(3), 685–695. https://doi.org/10.1007/s12525-021- 00475-2

9. Kotsiantis, S. (2007). Supervised machine learning: A review of classification techniques. *Informatica (Lithuanian Academy of Sciences), 31*(3), 249–268.

10. Kursh, S., & Schnure, A. (2021). An introduction to the “how to” for AI and machine learning. *Business Education Innovation Journal, 13*(2), 14–23.

11. Liu, Q., & Wu, Y. (2012). Supervised learning. In N. M. Seel (Ed.), *Encyclopedia of the sciences of learning* (pp. 3243–3245). Springer. https://doi.org/10.1007/978-1-4419-1428-6_451

12. Monaco, A., Pantaleo, E., Amoroso, N., Lacalamita, A., Lo Giudice, C., Fonzino, A., Fosso, B., Picardi, E., Tangaro, S., Pesole, G., & Bellotti, R. (2021). A primer on machine learning techniques for genomic applications. *Computational and Structural Biotechnology Journal, 19*, 4345–4359. https://doi.org/10.1016/j.csbj.2021.07.021

13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

14. Reinhart, C. M., & Rogoff, K. S. (2009). *This time is different: Eight centuries of financial folly*. Princeton University Press.

15. Shiller, R. J. (2000). *Irrational exuberance*. Princeton University Press.

16. Stock, J. H., & Watson, M. W. (2003). Forecasting output and inflation: The role of asset prices. *Journal of Economic Literature, 41*(3), 788–829.

https://doi.org/10.1257/jel.41.3.788

17. Whaley, R. E. (2000). The investor fear gauge. *Journal of Portfolio Management, 26*(3), 12–17. https://doi.org/10.3905/jpm.2000.319728