SCHOOL OF
OPERATIONS RESEARCH AND INFORMATION ENGINEERING
COLLEGE OF ENGINEERING
CORNELL FINANCIAL ENGINEERING MANHATTAN
CORNELL UNIVERSITY

Financial Engineering Project

Macro Investing: From Black Box to Crystal Box, Human to AI Spectrum

Presented to the Faculty of the Graduate School of Cornell University
in partial fulfillment of the
requirements for the Master of Engineering Degree
and the Financial Engineering Concentration

By:
Xiaolong Zhao, Ruipeng Deng, Minda Ren, Zhihan Chen, Sheng Zhang, Meng Yang

Faculty Advisor: Sasha Stoikov

Sponsor: Pictet Asset Management

December 2025

______________________________ ______________
Faculty Advisor's Signature Date

Macroeconomic Forecasting and Trading Framework of Equity and Bond Returns via Traditional Regressions and AI Models

Xiaolong Zhao, Ruipeng Deng, Minda Ren, Zhihan Chen, Sheng Zhang, Meng Yang

January 28, 2026

Abstract

The accuracy-interpretability trade-off is always acute in the financial modeling, where low-SNR and non-stationarity challenge both black-box and glass-box models. Traditional macro forecasting often relies on established models that prioritize transparency. However, recent advancements in artificial intelligence, particularly through Explainable Boosted Machines, have begun to address this challenge. The goal of this project is to develop forecasting and trading frameworks for returns of global equity and bond indices using 126 FRED macroeconomic features, integrating a transparent regression model with Explainable Boosting Machine (EBM) and Light Gradient Boosting Machine (LightGBM) to reconcile the interpretability with the predictive accuracy and to address the opacity inherent in AI-driven forecasting. Specifically, this project firstly fits an OLS model via a multi-criteria feature selection mechanism to capture price-related linear relationships; then trains EBM and LightGBM on the post-OLS residuals with non-price-derived features to learn interactions and nonlinear relationships; finally combined these models' complementary predictions to enhance model interpretability while harnessing the full power of machine learning.

Contents

1 Introduction

Macroeconomic investors increasingly rely on large panels of economic indicators and financial conditions to form expectations about the joint dynamics of global equity and bond returns. Recent regime shifts in inflation, monetary policy, and stock–bond co-movements have underscored the need for systematic forecasting tools that can translate high dimensional macro data into actionable trading signals. At the same time, the growing use of machine learning in asset management has raised concerns about model opacity, which can hinder risk management, communication with stakeholders, and regulatory acceptance. This tension between predictive accuracy and interpretability motivates the present study.

In this project, we design a macroeconomic forecasting and trading framework that aims to move macro investing from a purely black-box AI predictive modeling toward a transparent crystal-box implementation. Building on point in time data from the FRED MD database, we construct 126 transformed monthly indicators that span output, labor, housing, consumption, money and credit, interest and exchange rates, prices, and equity markets. The forecasting targets are monthly returns on the MSCI All Country World Index as a representative global equity portfolio and on the World Government Bond Index as a representative global sovereign bond portfolio over the period 1982 to 2024. This setting allows us to study how macro conditions map into broad asset class returns that are central to diversified portfolios.

Methodologically, we combine traditional regression techniques with advanced boosting machine models. As a benchmark, we estimate an ordinary least squares model that links each asset class return to a carefully selected subset of macro features obtained through multi-criteria ranking-based feature selection. We then build two augmented models that use the residuals from the OLS benchmark as targets for Explainable Boosting Machines and Light Gradient Boosting Machines, respectively. In both augmented models, the final forecast is the sum of the OLS prediction and the machine learning residual prediction, which preserves a clear economic baseline while allowing flexible nonlinear corrections. We evaluate these models using an 80 to 20 split between a training and validation sample from 1992 to early 2018 and an out of sample test period from 2018 to 2024 under both one month and two month lags between macro features and returns in order to address look-ahead bias in data releases.

Our empirical results show that integrating OLS with LightGBM can modestly improve out of sample fit and hit rates for both equity and bond returns at a one month lag, while backtests indicate that all machine learning augmented models produce higher Sharpe ratios and shorter drawdowns than a simple benchmark that invests directly in the target indices. At the same time, the OLS plus EBM specification delivers richer economic interpretability through global and local feature importance curves, which highlight the role of monetary policy indicators, consumption and housing activity, and selected labor and credit variables for both asset classes. When the lag between macro features and returns is extended to two months, apparent predictive content declines sharply, which illustrates the practical trade off between information timeliness and the avoidance of look-ahead bias.

The main contributions of this study are threefold. First, we provide a unified framework that reconciles transparent regression based macro models with flexible machine learning residual models, thereby offering investors a continuous spectrum between human readable and algorithmic forecasts. Second, we document how different macro features drive equity and bond return forecasts within this framework and relate these findings to existing evidence on macroeconomic drivers of stock–bond correlations and on machine learning applications in asset pricing and credit risk. Third, we translate statistical forecasts into simple systematic trading strategies that use the federal funds rate as the risk free benchmark and show that crystal box macro signals can enhance portfolio performance relative to passive exposure while retaining interpretability.

2 Literature Review

Recent work in financial engineering shows how machine learning methods and macroeconomic information can improve prediction of asset returns, credit events, and cross asset comovements. The following review focuses on three contributions that are directly related to this study.

2.1 Equity Return Prediction with LightGBM

Yang applies Light Gradient Boosting Machine to excess return prediction for Chinese A share stocks and compares it with cross sectional OLS regression (Yang). Using fifty firm level characteristics constructed from Wind and CSMAR data and a rolling out of sample design, the study reports a monthly out of sample R^2 of 2.13% for LightGBM versus 0.95% for OLS and shows that LightGBM based long only and long short portfolios achieve higher returns and smaller drawdowns than both OLS strategies and the CSI 300 index. Feature importance analysis indicates that liquidity and volatility characteristics such as abnormal turnover, trading volume, market capitalization, maximum daily return, and short horizon price deviation dominate the predictive structure, which underscores the central role of market microstructure in Chinese equity pricing (Yang).

2.2 Explainable Machine Learning for Credit Default Risk

Ma and coauthors develop credit default prediction models for Chinese listed real estate firms using annual financial ratios, MD&A text, stock bar investor comments, and distance to default from 2017 to 2021 (Ma et al.). They compare glass box models such as logistic regression and the Explainable Boosting Machine with black box models including random forest, support vector machine, and AdaBoost, and evaluate performance with accuracy, AUC, KS, and error rates. Across feature combinations, the Explainable Boosting Machine and AdaBoost achieve the strongest predictive accuracy, while SHAP, partial dependence, and individual conditional expectation plots reveal how MD&A tone, investor sentiment, and composite financial components jointly shape default risk, which highlights the value of explainable machine learning for high stakes credit screening in the Chinese real estate sector (Ma et al.).

2.3 Macroeconomic Drivers of the Stock–Bond Correlation

Baumann, Nazemi, and Fabozzi investigate the time varying correlation between stock and bond returns and its macroeconomic drivers using a machine learning approach (Baumann, Nazemi, and Fabozzi). Drawing on long horizon return series from the Stocks, Bonds, Bills, and Inflation database and a wide set of macro indicators from FRED, they document large shifts in the stock bond correlation, including during the recent episode of high inflation and rapid monetary tightening, and then use high dimensional macro panels and variable selection to identify variables that are most informative for predicting the correlation. The analysis emphasizes the influence of inflation, real activity, monetary conditions, housing, and related macro factors on the correlation and provides a ranked list of drivers that can inform portfolio construction when the traditional negative stock bond relationship becomes unstable (Baumann, Nazemi, and Fabozzi).

2.4 Research Implications

Together these studies demonstrate that tree based ensemble methods and related machine learning tools can deliver economically meaningful gains in predictive accuracy for equity returns, credit default events, and cross asset correlations, especially when combined with rich firm level and macroeconomic information. They also show that interpretability through feature importance and explainable artificial intelligence techniques is increasingly essential for applying such models in practice, which motivates the modeling choices adopted in the present study.

3 Data

3.1 Asset return targets

The empirical analysis uses two global asset class indices as forecasting targets. The equity target is the MSCI All Country World Index which covers large and mid capitalisation stocks across 23 developed markets and 24 emerging markets and captures about 85% of the investable global equity universe. The index is country weighted with the United States holding the largest share so that the same United States mega cap names dominate both MSCI ACWI and the S&P 500 and the two indices display highly correlated performance. The bond target is the World Government Bond Index which consists of fixed rate government bonds issued by developed markets in a market value weighted composition and again has the United States as the largest single country weight. Monthly total returns on these two indices from 1982 to 2024 form the dependent variables in the forecasting exercises.

3.2 Macroeconomic features

The explanatory variables are monthly macroeconomic indicators from the FRED MD database which serve as the covariate matrix X. The panel spans 1992 to 2024 and includes a broad set of series organised into eight groups that capture output and income labour market conditions consumption orders and inventories money and credit interest and exchange rates prices and stock market indicators. The indicators are constructed by United States statistical agencies and many series are seasonally adjusted at the source. This design provides a high dimensional yet structured macroeconomic information set that can be aligned with the monthly asset returns. In this project we use the December 2024 vintage, which provides 126 monthly macroeconomic indicators spanning eight categories (See Appendix for detailed features and descriptions):

- **Output and Income** (17 features): Industrial production indices, capacity utilization, and GDP components
- **Labor Market** (32 features): Employment levels, unemployment rates, average hours worked, and jobless claims
- **Housing** (10 features): Housing starts, building permits, and home sales
- **Consumption and Orders** (14 features): Retail sales, manufacturing orders, and inventories
- **Money and Credit** (14 features): Monetary aggregates, consumer credit, and loan volumes
- **Interest Rates and Spreads** (22 features): Treasury yields, corporate bond spreads, and the Fed funds rate

- **Prices** (21 features): Consumer and producer price indices, commodity prices, and inflation measures
- **Stock Market** (6 features): S&P 500 index, dividend yield, and price-earnings ratio

3.3 Transformations to stationarity

Many macroeconomic series appear in non stationary levels which can lead to misleading fits and unstable forecasts. Following the standard FRED MD convention each raw series x_t is transformed into a stationary series y_t using a transformation code Tcode $\in \{1, \ldots, 7\}$. The transformations stabilise both level and variance and are defined as follows for monthly data

- Tcode 1, no transformation
$$y_t = x_t$$
- Tcode 2, first difference
$$y_t = x_t - x_{t-1}$$
- Tcode 3, second difference
$$y_t = (x_t - x_{t-1}) - (x_{t-1} - x_{t-2}) = \Delta^2 x_t$$
- Tcode 4, logarithm
$$y_t = \log x_t$$
- Tcode 5, first difference of logarithm
$$y_t = \log x_t - \log x_{t-1} = \Delta \log x_t$$
- Tcode 6, second difference of logarithm
$$y_t = (\log x_t - \log x_{t-1}) - (\log x_{t-1} - \log x_{t-2}) = \Delta^2 \log x_t$$
- Tcode 7, growth rate in percent form
$$y_t = \frac{x_t}{x_{t-1}} - 1$$

These transformations produce approximately stationary versions of the macroeconomic indicators which are then used as inputs in the subsequent forecasting models. Detail instruction on the application of transformations to each macroeconomic variable can be seen in Appendix.

3.4 Further Processing

Step 1. Missing Value Processing
After processing transformation to each feature based on its corresponding Tcode to make the feature values more stationary, for some features with constructional missing value (due to quarterly reporting), simply forward fill the raw features data to deal with missing data before transformation. For some features with missing value at the start and the end of the timeline, just ignore them before transformation, further in alignment stage, it will be handled.

Step 2. Return Calculation and Alignment with features
We calculate equity monthly return (Y1) and bond monthly return (Y2) from the daily price data, then align the transformed feature values (X) with the monthly returns with the out-of-sample logic:

- 1-Month Lag: for example, feature value (X) reported on the first natural day of March is aligned with the return (Y) calculated as the changing rate between the price on the first business day of March and the price on the first business day of April.
- 2-Month Lag: for example, feature value (X) reported on the first natural day of March is aligned with the return (Y) calculated as the changing rate between the price on the first business day of April and the price on the first business day of May.

Step 3. Extreme data exclusion
The three months that highly affected by COVID are removed, from February 2020 to April 2020, to make the evaluations of fitted models more commonly robust.

4 Models

4.1 Framework Overview

Step 1. Return modeling – traditional OLS:

- Use the training dataset to fit OLS model on equity returns and bond returns (y) by the pool of 126 FRED features , but not all 126 features will be actually selected into the model fitting, only about 15 features (the number is variable based on the choice) will be selected from the pool of 126 features, through sophisticated multi-criteria feature selection by applying Rankings and Unions (which will be discussed below), these selected features are denoted as denote as $X^{all,selected}$, and denote the fitted model as f^{OLS} **(Model A)**.
- Then for each month t in the training dataset, apply the fitted OLS model f^{OLS} back to the training dataset to output:
  - linear prediction: $\widehat{y_t^{OLS}} = f^{OLS}(X_t^{all,selected})$
  - residual: $e_t = y_t - \widehat{y_t^{OLS}}$
- Within the whole training range, get a time series of e_t, called $\overrightarrow{e_{train}}$

Step 2. Residual modeling (boosting machines focus on macro non-linearity):

- Train EBM and LightGBM on **residuals** (not on y), using **macro-only** FRED features $X^{macro-only}$:
  - 116 macro-only FRED features: 126 features exclude 10 price-like features: S&P 500, S&P div yield, S&P PE ratio, VIXCLSx, TWEXAFEGSMTHx, EXSZUSx, EXJPUSx, EXUSUKx, EXCAUSx, OILPRICEx
  - Because after Model A has captured the linear component of returns using all features (including price-like ones), Residual models should focus on searching for non-linear macro structure in the leftover signal. Feeding price-like variables again would re-introduce the strongest market signals, drown out macro effects, and cover the contributions of other features.
- **Model B (OLS + EBM-residual)**:

Use the training dataset to fit EBM model on $\overrightarrow{e_{train}}$ by $X_{train}^{macro-only}$, denote the fitted model as g^{EBM}

- **Model C (OLS + LightGBM-resid)**:

  Use the training dataset to fit LightGBM model on $\overrightarrow{e_{train}}$ by $X_{train}^{macro-only}$, denote the fitted model as g^{LGBM}

Step 3. Combine predictions for residual modeling:

- For each month m in the test dataset, apply the f^{OLS}, g^{EBM}, g^{LGBM} to output:
  - $\widehat{y_m^A} = f^{OLS}(X_m^{all,selected})$
  - $\widehat{e_m^{EBM}} = g^{EBM}(X_m^{macro-only})$
  - $\widehat{y_m^B} = \widehat{y_m^A} + \widehat{e_m^{EBM}}$
  - $\widehat{e_m^{LGBM}} = g^{LGBM}(X_m^{macro-only})$
  - $\widehat{y_m^C} = \widehat{y_m^A} + \widehat{e_m^{LGBM}}$

Step 4. Return modeling – boosting machines:

- **Model D:** Use the training dataset to fit EBM model directly on equity returns and bond returns (y) by all 126 features (X^{126}), denote as f^{EBM}
- **Model E:** Use the training dataset to fit LightGBM model directly on equity returns and bond returns (y) by all 126 features, denote as f^{LGBM}
- For each month m in the test dataset, apply the f^{EBM}, f^{LGBM} to output:
  - $\widehat{y_m^D} = f^{EBM}(X_m^{126})$
  - $\widehat{y_m^E} = f^{LGBM}(X_m^{126})$

We will compare the five models through Performance matrices, Confusion matrices, and backtesting results in the Section 5. Also, there exist some notes of the above four steps:

- All fitting/tuning happens on training/validation only; models are frozen before scoring the test set to avoid any test leakage.
- We have three sets of X:
  - $X^{all,selected}$ contains around 15 features by Rankings and Unions from the pool of 126 features, used for model A
  - $X^{macro-only}$ contains 116 macro-only features, used for Model B, Model C
  - X^{126} contains all 126 features, used for Model D, Model E
- On test dataset, we predict residuals only via the trained models; we do not recompute "actual residuals" using y on test dataset.

4.2 Train-Test Splits

To ensure the comparability across all models in the project, including these 3 separate models, we apply a common train-test split structure, where the train dataset (including cross-validation) will take 80% of the whole time series of data, from March 1992 to February 2018. And the other 20% time series of data will be the test dataset with a fixed range from April 2018 to November 2024, there exists a one month gap between train and test dataset to avoid data leakage. The train-test split is same across all models in this project, but within the training range, the specific train-cross validation split will be different based on each model's different parameter tuning mechanism.

4.3 Traditional Regression Model

4.3.1 Feature Analysis

Feature analysis by one-to-one 5-year rolling OLS
For each transformed feature (x) in the pool of 126 FRED features, fit one-x-to-one-y OLS regression with the equity monthly return (y1) and bond monthly return (y2) separately. Specifically, we split the timeline using rolling window (window size = 5 years, rolling step = 1 year), then only use the data within each 5-year window to make OLS regression analysis for each feature separately. We care about three criteria rankings of each feature by the OLS analysis. Specifically:

Rank 1. OLS Beta Sign Stability:
Record the sign of beta of each feature in each time window's regression, rank features by each feature's sign stability of beta over time (across rolling window blocks), where:

$$sign\ stability = 1 - \frac{(\#\ of\ sign\ flips\ across\ adjacent\ blocks)}{(\#\ of\ effective\ adjacent\ blcok\ pairs)}$$

Ignore pairs where any sign is 0 or NA.

Rank 2. OLS $\mathbf{R^2}$ Stability:
Record the R^2 of each feature in each time window's regression, compute each feature's median(R^2) and CV(R^2) = std(R^2) / mean(R^2) across all time blocks. Firstly, set a boundary of median(R^2) to split all features into two parts, then separately rank features within each part by the feature's CV(R^2) , since we don't want to give the feature with too small median R^2 a high ranking even if it has lower CV.

Rank 3. Full Timeline OLS $\mathbf{R^2}$ Value:
Use the whole timeline dataset (no split of time line) to train OLS regression for each feature to get the sign of beta and R^2 of each feature, then, rank features by each feature's full-sample R^2 value.

Feature analysis by global yearly Lasso
Use all 126 transformed features to do global yearly Lasso (L1) regression (global means not one-to-one like above OLS analysis). For each Lasso, only Top 30 features will be selected into the model, which means instead of applying GridSearchCV to pick best alpha then decide a yearly-variable selected number, we fix the selected number as 30 across all yearly Lasso models (then automatically adjust penalty alpha based on the fixed 30 selected number) to keep the comparability in below rankings. We care about four criteria rankings of each feature by the Lasso analysis. Specifically:

Rank 4. Lasso Selection Rate:
Rank the selection rate of each feature as the proportion of years in which each feature is selected across all yearly Lasso models

Rank 5. Lasso Beta Sign Stability:
Record the sign of beta of each feature only within the years when this feature is selected, skip the years when this feature is not selected, rank the sign stability of beta of each feature over time, where

$$sign\ stability = 1 - \frac{(\#\ of\ sign\ flips\ across\ adjacent\ \ years\ when\ this\ feature\ is\ selected)}{(\#\ of\ effective\ selected\ years'\ adjacent\ pairs)}$$

Rank 6. Lasso ΔR^2 Stability:
Define ΔR^2 of each feature as "drop-one-feature $\mathbf{R^2}$ changing of the Lasso model". Apply drop-one only to features selected by Lasso in that year (for other features, set their $\Delta R^2 = 0$ for that year). Thus, each feature has a sequence ofΔR^2 values across years (containing both zeros and non-zero values, and zeros must also be considered). With these 126 sequences, compute median(ΔR^2) and CV(ΔR^2) = std(ΔR^2) / mean(ΔR^2) of each feature from its sequence. Then firstly set a boundary of median(ΔR^2) to split all features into two parts, then separately rank features within each part by the feature's CV(ΔR^2) , to avoid giving the feature with too small median ΔR^2 a high ranking even if it has lower CV.

Rank 7. Full Timeline Lasso ΔR^2 Value:
Use the whole timeline dataset to do Lasso regression, instead of yearly Lasso. Apply drop-one only to features selected by whole timeline Lasso (for other unselected features, set their $\Delta R^2 =$ 0). Thus, each feature has only one ΔR^2 value (instead of a sequence). Rank each feature's full ΔR^2 value from global whole timeline sample Lasso.

4.3.2 Feature Selection

After feature analysis, the next step is to do the feature selection using different combinations of results from rankings. Define a selection tool called union score of each feature as the sum of ranking position of this feature in each considered ranking:

$$Union\ Score = \sum_i Position\ of\ this\ feature\ in\ Rank\ i$$

We design three different Union mechanisms to do feature selection (3 Unions for equity models, 3 Unions for bond models):

Union_OLS_bygroup: The first Union combines the results from Rank 1 and Rank 3, and then for each of 8 Feature Group, selects the features with the TOP 10% lowest union scores, next, integrates the winner features of each group together to get a final Union. There exist 15 selected features in this Union. The reason of choosing 15 as the number of features used is that through robustness checking, it is the number that best balances the performance and simplicity of fitted OLS, definitely, the number of features can be easily varied based on preference.

Union_OLS_global: The second Union still combines the results from Rank 1 and Rank 3, but never selects features separately by group, just globally ranks the union score across all 126 features, to select the features with the TOP 15 lowest union scores.

Union_Lasso_global: The third union combines the results from Rank 4, 5, and 7, similarly, globally ranks the union score across all 126 features, to select the features with the TOP 15 lowest union scores.

4.3.3 OLS Fitting

After feature selections, the third step is to use the features in three Unions to fit 3 benchmark OLS models, then pick the best one as the **Model A**. There exist two parameters to be tuning within the cross validation dataset during fitting, one is the winsorization level — the clipping strength applied to the extreme values of features, X, we will use cross validation to choose the best quantile threshold or MAD multiplier as our winning winsorization parameter.

The other parameter to be tuned is the Cook's-distance trimming fraction, which decides the dropping fraction of the extreme observations in the training set. After cross validation, we get a winning group of these two parameters separately for each of the 3 OLS models, then we use the whole training dataset and the winning parameters to fit the 3 models before comparison during the test window. These two parameters are both just used to avoid possible unexpected effects of some extreme outliers to make the OLS model more robust and commonly applicable.

4.3.4 Best OLS Model as Model A

We apply three performance evaluation criteria for the models in this project, based on these three criteria, we can pick the best OLS model for equity and bond separately:

- **Test_R2:** the first one is R^{2} on the test set, which represents the proportion of variance in the equity or bond return explained by the model, so the model with higher test R^{2} is better;
- **Test_RMSE:** the second one is RMSE on the test set, which represents the square root of the mean squared prediction error of the model, so the model with lower test RMSE is better;
- **Test_Hit:** the third one is hit rate, which represents the percent of test observations with the same sign between actual return and predicted return, so the model with higher test hit rate is better. Hit rate is important since we will construct the backtesting of our trading strategies based on the sign of predicted return. Test_Hit_positive records the hit accuracy of positive returns, Test_Hit_negative records the hit accuracy of negative returns.

Table 1: Performance Matrices of OLS models

Union	test_r2	test_rmse	test_hit	test_positive_hit	test_negative_hit
OLS by-group — Equity	0.246015	0.0370468	0.779221	0.807018	0.7
OLS by-group — Bond	0.199836	0.0132431	0.623377	0.57377	0.8125
OLS global — Equity	0.246249	0.0370411	0.766234	0.814815	0.652174
OLS global — Bond	0.246599	0.0128503	0.675325	0.622642	0.791667
Lasso global — Equity	0.284514	0.0360886	0.779221	0.818182	0.681818
Lasso global — Bond	0.173281	0.0134611	0.597403	0.559322	0.722222

According to Table 1, For the equity return models, the best one is the OLS model fitted by the features in the Union_Lasso_global, with the highest test R2, the lowest RMSE, and the highest hit rate among three equity models. Meanwhile, for the bond return model, the best one is the OLS model fitted by the features in another Union, the Union_OLS_global, also with the highest test R2, the lowest RMSE, and the highest hit rate among three bond models. Therefore, these two OLS models are selected as our **Model A** for equity and bond.

4.3.5 Features and Parameters Used

Features:

Table 2: The significance and coefficient of each feature in Model A for Equity

feature	description	group	p_value	coef
S&P 500	S&P's Common Stock Price Index: Composite	8	1.21874e-36	0.907322
EXSZUSX	Switzerland / U.S. Foreign Exchange Rate	6	0.00592243	-0.228628
HWI	Help-Wanted Index for United States	2	0.0144579	2.83724e-05
UMCSENTX	Consumer Sentiment Index	4	0.0358041	-0.00117675
BAA	Moody's Seasoned Baa Corporate Bond Yield	6	0.11959	0.0321099
NONBORRES	Reserves Of Depository Institutions	5	0.255164	-0.0272074
CP3MX	3-Month AA Financial Commercial Paper Rate	6	0.291017	0.0271851
CES0600000007	Avg Weekly Hours: Goods-Producing	2	0.292305	-0.00316782
TB3MS	3-Month Treasury Bill	6	0.323452	-0.0241848
AAA	Moody's Seasoned Aaa Corporate Bond Yield	6	0.324417	-0.0205657
UEMP15T26	Civilians Unemployed for 15-26 Weeks	2	0.338673	0.0338658
UEMPMEAN	Average Duration of Unemployment (Weeks)	2	0.348382	-0.00294267
BAAFFM	Moody's Baa Corporate Bond Minus FEDFUNDS	6	0.68829	0.00300715
AAAFFM	Moody's Aaa Corporate Bond Minus FEDFUNDS	6	0.718499	-0.00289464
HOUSTW	Housing Starts, West	3	0.814679	0.000893541

According to Table 2, we can view the 15 features with their corresponding coefficients that are included in the Model A for Equity, they are ranked by the significance (p-value). It is obvious that in the Model A for Equity, most of features come from Group 6 called Interest and exchange rates. Next, the features with p-value lower than 10 percent will be considered as top features, which come from Group 6, Group 8 called Stock market, Group 2 called Labor market, and Group 4 called Consumption, orders, and inventories.

Table 3: The significance and coefficient of each feature in Model A for Bond

feature	description	group	p_value	coef
IPB51222S	IP: Residential Utilities	1	0.0159855	0.0287616
GS10	10-Year Treasury Rate	6	0.0227887	-0.0227313
S&P 500	S&P's Common Stock Price Index: Composite	8	0.0436381	-0.0824559
CP3MX	3-Month AA Financial Commercial Paper Rate	6	0.0526529	-0.00603828
CLAIMSX	Initial Claims	2	0.114713	0.0197047
CUSR0000SAC	CPI: Commodities	7	0.136561	-0.48531
S&P DIV YIELD	S&P's Composite Common Stock: Dividend Yield	8	0.198893	-0.0261187
GS1	1-Year Treasury Rate	6	0.199714	-0.0137532
DNDGRG3M086SBEA	Personal Cons. Exp: Nondurable goods	7	0.200347	0.323395
TB3MS	3-Month Treasury Bill	6	0.212179	0.00703499
AAA	Moody's Seasoned Aaa Corporate Bond Yield	6	0.216202	-0.00773869
GS5	5-Year Treasury Rate	6	0.399899	0.0089715
TB6MS	6-Month Treasury Bill	6	0.530077	0.00724673
CPIULFSL	CPI: All Items Less Food	7	0.793269	-0.0962558
BAA	Moody's Seasoned Baa Corporate Bond Yield	6	0.970273	0.000197028

According to Table 3, we can view the 15 features with their corresponding coefficients that are included in the Model A for Bond. Most of features still come from Group 6, where the top-significance features come from Group 6, Group 8, and Group 1 called Output & income.

Based on the two feature tables of Model A, we can say that the features in Group 6 and 8 are most important predictors for both equity and bond. Meanwhile, if ignoring the significance, the selected features come from all eight groups, which means the diversification of our traditional feature selection mechanism is also acceptable. That's also the reason of keeping the by-group Union selection mechanism.

Parameters:

For the Equity Model A, its winning winsorization parameter is 0.03 quantile threshold, its winning Cook's-distance trimming fraction parameter is 0.0167.

For the Bond Model A, its winning winsorization parameter is also 0.03 quantile threshold, its winning Cook's-distance trimming fraction parameter is 0.

4.4 AI Models - Boosting Machines

4.4.1 Explainable Boosting Machine (EBM)

Theory Explainable Boosting Machine (EBM) is a glass-box model in the generalized additive model (GAM) family. It models the conditional expectation of the response as an additive combination of main effects and a limited number of pairwise interactions:

$$g(\mathbb{E}[y \mid x]) = \beta_0 + \sum_i f_i(x_i) + \sum_{(i,j)\in\mathcal{I}} f_{ij}(x_i, x_j), \tag{1}$$

where $g(\cdot)$ is a link function (e.g., identity for regression, logit for classification), β_0 is a global bias term, $f_i(\cdot)$ are univariate shape functions for each feature, and $f_{ij}(\cdot,\cdot)$ are interaction terms for a small set of feature pairs $\mathcal{I}$.

This additive structure keeps the model fully decomposable: any prediction can be written as a baseline plus a sum of interpretable contributions. The $f_i(\cdot)$ are learned as flexible, non-linear functions, allowing EBM to capture thresholds, plateaus, and other non-linear effects that linear regression cannot, while preserving the ability to inspect exactly how each feature influences the prediction.

Training Procedure EBM is trained using a boosting-style coordinate descent procedure that updates one component at a time. At a high level, the algorithm cycles through features and, for each feature, performs a small corrective step:

- Given the current model, compute residuals (negative gradients of the loss) with respect to the predictions.
- For a single feature x_i, fit a very shallow decision tree (typically depth-1 or depth-2) to predict these residuals using only x_i.
- Update the corresponding shape function $f_i(\cdot)$ by adding a small multiple of the tree's output, controlled by a *very small learning rate.*

The algorithm loops over all features and repeats this process across many boosting iterations. When interaction terms are enabled, the same logic is applied to selected pairs (x_i, x_j) to update $f_{ij}(\cdot,\cdot)$, but the number of such interaction terms is kept deliberately small.

Two design choices are critical:

1. **Small learning rate.** Each update uses a very small step size, so no single iteration can dominate the model. This has two consequences: (i) the model does not overweight any individual feature early in training, and (ii) the effective solution is largely invariant to the order in which features are visited, because many tiny updates are averaged over repeated passes through all features.

2. **Shallow trees per feature.** Each per-feature learner is a shallow tree, which enforces smooth, low-variance shape functions and keeps the complexity of each $f_i(\cdot)$ tightly controlled. The goal is not to grow a deep forest, but to accumulate many small, simple corrections that add up to a stable one-dimensional effect.

The resulting model is therefore closer to a regularized additive surface than to a typical boosted forest: prediction power comes from many small, coordinated adjustments across features, rather than from a few highly expressive trees.

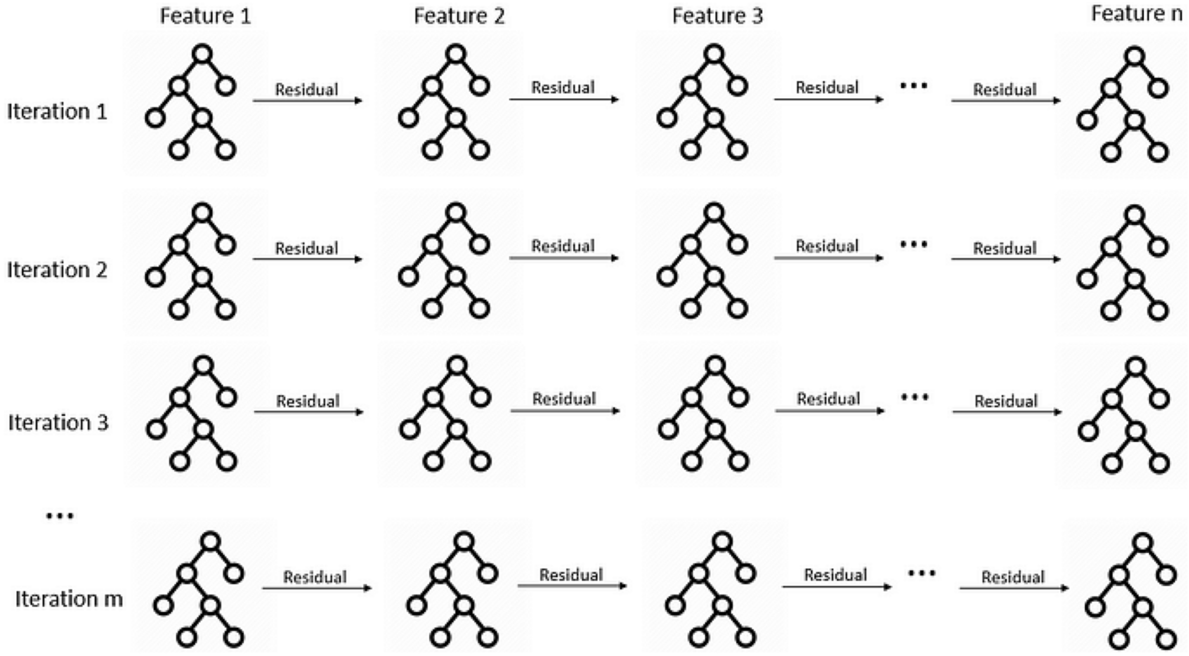


Figure 1: EBM training workflow: in each boosting iteration, shallow trees are fit feature by feature on residuals, with small learning-rate updates to the corresponding shape functions.

Cross-Validation To obtain an honest estimate of out-of-sample performance in a time-series setting, the EBM is evaluated using a time-aware cross-validation scheme inspired by purged k-fold with embargo. The full time series is first split into an 80% training window and a 20% hold-out test window. All model selection and tuning is performed exclusively within the training window.

Within that training window, a 5-fold cross validation is constructed over contiguous time blocks. For each fold:

1. A contiguous segment of the training window is designated as the *validation block*.

2. An *embargo region* is defined immediately after the validation block, with length set to approximately 1% of the total training period (when sufficient data is available).

3. The *training set* for that fold consists of *all remaining observations* in the training window, i.e., all data **before** the validation block and all data **after** the embargo region. The model is thus trained on data from both earlier and later periods, but never on the validation block itself or its embargo buffer.

This design has two goals. First, by holding out the entire validation block and its subsequent embargo region, the fold reduces temporal leakage arising from short-horizon dependence, overlapping feature windows, or regime shifts that span the validation boundary. Second, by allowing the training set to include data both before and after the validation block, the procedure preserves sample efficiency, using as much data as possible while still respecting temporal structure.

The folds slide forward through the training window, so that different time segments serve as validation blocks across the 5 folds. Model hyperparameters (e.g., number of boosting iterations, maximum tree depth, learning rate, and number of interactions) are selected by averaging the validation performance across folds. After tuning, the final model is refit on the entire training window and evaluated once on the untouched 20% test window.

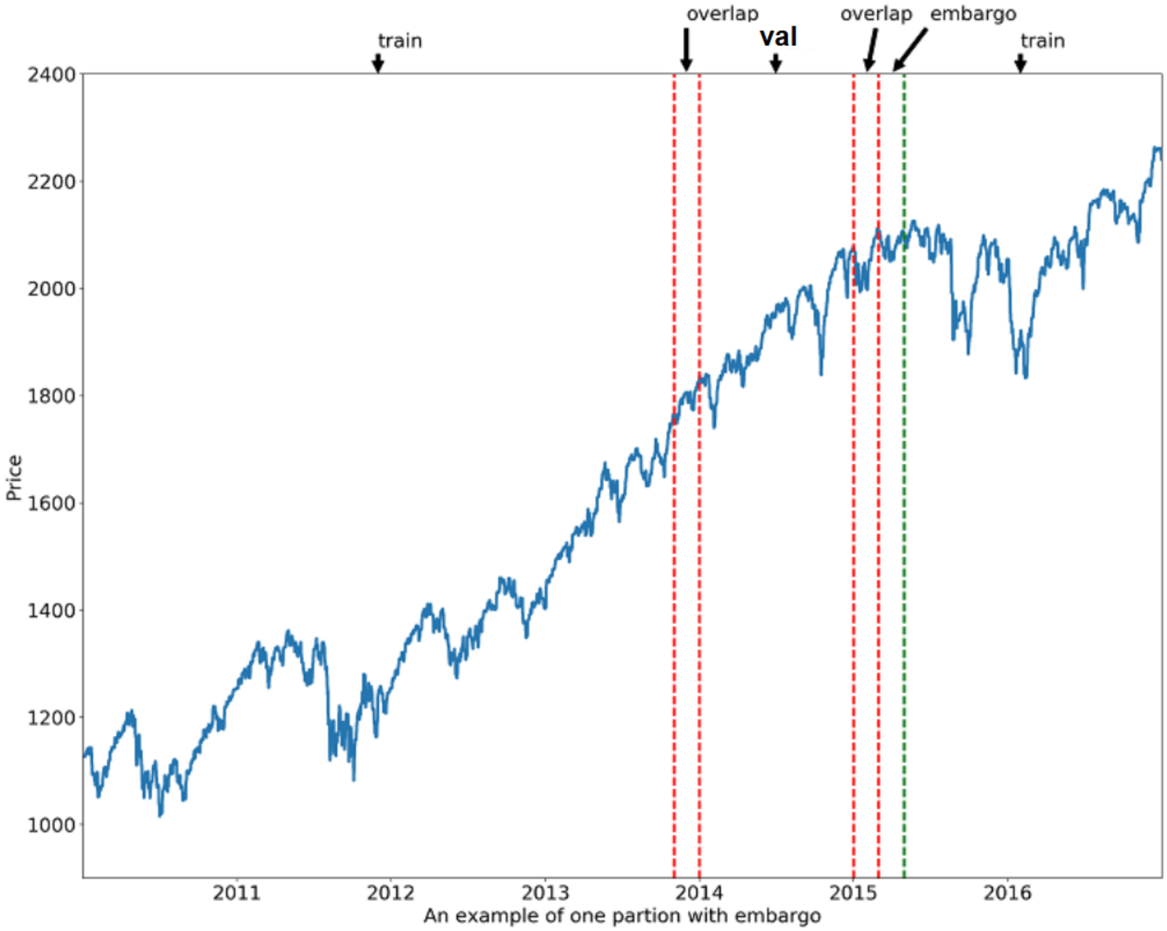


Figure 2: Time-series cross-validation scheme (purged *k*-fold with embargo). For each fold, the validation block (blue) is held out; an embargo region (gray) after validation is removed from training; all remaining data before and after are used for training (green).

This setup ensures that performance estimates for the EBM reflect realistic forward-looking behavior and are not inflated by subtle temporal leakage.

Interpretation and Visualization Because EBM maintains separate components for each feature and interaction, interpretation reduces to inspecting these learned shape functions:

- **Global importance.** Overall feature importance scores summarize which $f_i(\cdot)$ contribute most to the loss reduction and to prediction variance. This ranking is used to identify the dominant drivers in the model.
- **Shape functions.** For each continuous feature x_i, the learned curve $f_i(x_i)$ shows how the feature's contribution varies across its range. These curves are typically plotted together with the empirical distribution (e.g., a histogram or kernel density) of x_i to distinguish well-supported regions from sparsely populated tails.
- **Interaction terms.** When interaction components $f_{ij}(x_i, x_j)$ are enabled, they are visualized using 2D heatmaps or contour plots, which reveal non-additive effects between selected pairs of variables.
- **Local explanations.** For a given observation x, the prediction decomposes into

  $$g(\hat{y}) = \beta_0 + \sum_i f_i(x_i) + \sum_{(i,j)\in\mathcal{I}} f_{ij}(x_i, x_j),$$

  providing an exact additive breakdown of how each feature pushes the prediction up or down relative to the baseline.

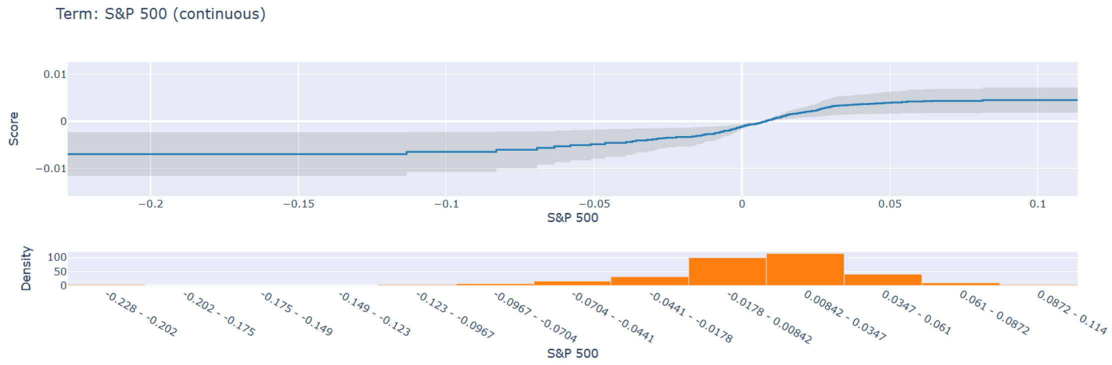


Figure 3: Example EBM main-effect shape function for an S&P 500 feature. The blue line shows the learned contribution $f_i(x_i)$; the orange bars show the empirical density of the feature, emphasizing regions with strong data support.

This combination of a carefully regularized training procedure (small learning rate, shallow trees) and additive structure yields a model that is both competitive in accuracy and straightforward to audit using a small set of plots.

(Hyper)Parameters Used

Table 4: Best hyperparameters for EBM Bond and Equity models

Hyperparameter	What it Controls	Bond Model	Equity Model	Test Range
interactions	Number of interactions	7	7	$[5, 10]$
max_bins	Number of bins per feature	16	16	$\{16, 32, 64, 128\}$
learning_rate	Boosting learning rate	1.33e-4	2.64e-4	$[10^{-4}, 5 \times 10^{-2}]$
min_samples_leaf	Minimum samples in a leaf	38	174	$[5, 200]$
max_leaves	Maximum leaves per tree	4	2	$[2, 5]$
outer_bags	Bagging iterations	9	21	$[8, 30]$
validation_size	Data used for validation	0.34	0.29	$[0.15, 0.35]$
max_rounds	Maximum boosting rounds	4863	1082	$[1000, 5000]$
early_stopping_rounds	Rounds before stopping	43	49	$[10, 50]$
early_stopping_tolerance	Minimum improvement	1.0e-4	1.0e-4	fixed
random_state	Random seed	42	42	fixed
n_jobs	Number of parallel threads	-1	-1	fixed

The table summarizes the optimal hyperparameters found for the EBM Bond model and the EBM Equity model, along with the search ranges explored during hyperparameter tuning. Overall, both models share a very similar structural configuration, but several key differences highlight how the models adapt to the characteristics of bond vs. equity returns.

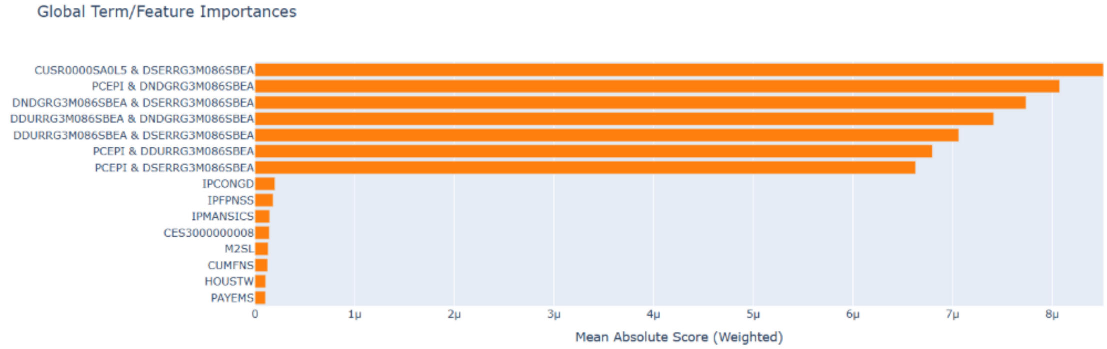
If we look at model complexity, both models allow the same maximum depth of feature interactions, meaning neither asset class required higher-order interaction terms to improve performance.

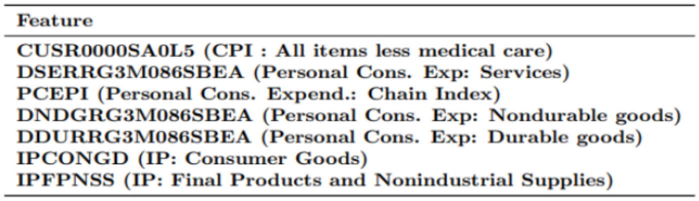
For bonds, the model uses more leaves per tree and smaller minimum samples per leaf to fit subtle nonlinear patterns in the residuals. In contrast, the equity model forces very large leaf sizes and fewer leaves, which makes the function much smoother and much less sensitive to local noise. That's is to say, bond model is a tree with finer but more leaves, and equity model is a tree with larger but fewer leaves.

If we look at the regularization, the equity model uses more outer bags and stops after fewer boosting rounds. It suggests that the model is conservative and only learns the strongest, most persistent signals. The bond model, on the other hand, needs fewer bags but many more boosting rounds. The model can safely use extra flexibility and more iterations to capture finer structure.

Overall, the contrast between these two columns is consistent with our economic intuition:equity returns are noisy and require a highly regularized, low-variance model, while bond returns are smoother and can be modeled with higher flexibility.

Explainability

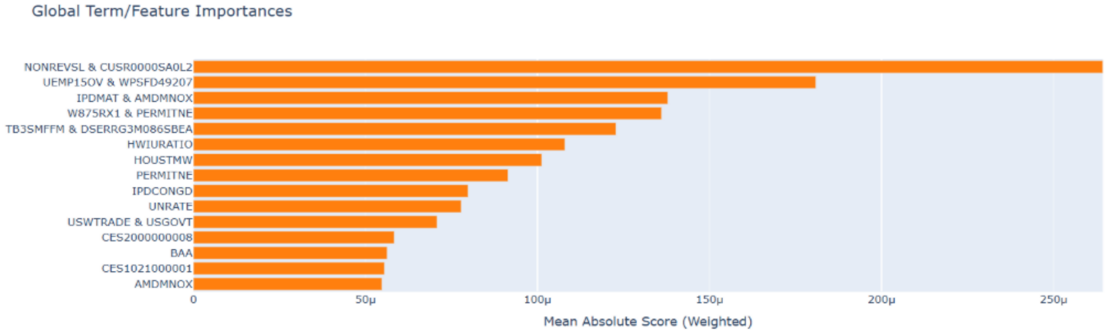


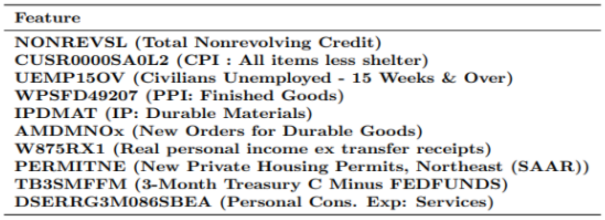


Feature
CUSR0000SA0L5 (CPI : All items less medical care)
DSERRG3M086SBEA (Personal Cons. Exp: Services)
PCEPI (Personal Cons. Expend.: Chain Index)
DNDGRG3M086SBEA (Personal Cons. Exp: Nondurable goods)
DDURRG3M086SBEA (Personal Cons. Exp: Durable goods)
IPCONGD (IP: Consumer Goods)
IPFPNSS (IP: Final Products and Nonindustrial Supplies)

Figure 4: EBM Equity Feature Importance

Here are some top features which contribute most to the predictability of the model. We can see that the interaction functions contribute most to the prediction and almost all the features in interaction functions are from consumer spending. In economic intuition, consumption did serve as a good indication of equity prosperity and the model further demonstrates it. For deeper research on equity and macroeconomic indicators, it's a good direction to start from consumer spending.





Feature
NONREVSL (Total Nonrevolving Credit)
CUSR0000SA0L2 (CPI : All items less shelter)
UEMP15OV (Civilians Unemployed - 15 Weeks & Over)
WPSFD49207 (PPI: Finished Goods)
IPDMAT (IP: Durable Materials)
AMDMNOx (New Orders for Durable Goods)
W875RX1 (Real personal income ex transfer receipts)
PERMITNE (New Private Housing Permits, Northeast (SAAR))
TB3SMFFM (3-Month Treasury C Minus FEDFUNDS)
DSERRG3M086SBEA (Personal Cons. Exp: Services)

Figure 5: EBM Bond Feature Importance

For bond models, it's more complex. Consumer credit and household spending are the dominant drivers. Labor market stress signals also matter significantly. In addition to these factors, we have Industrial and Production Indicators, Monetary and Financial Market Variables.

4.4.2 LightGBM

Theory

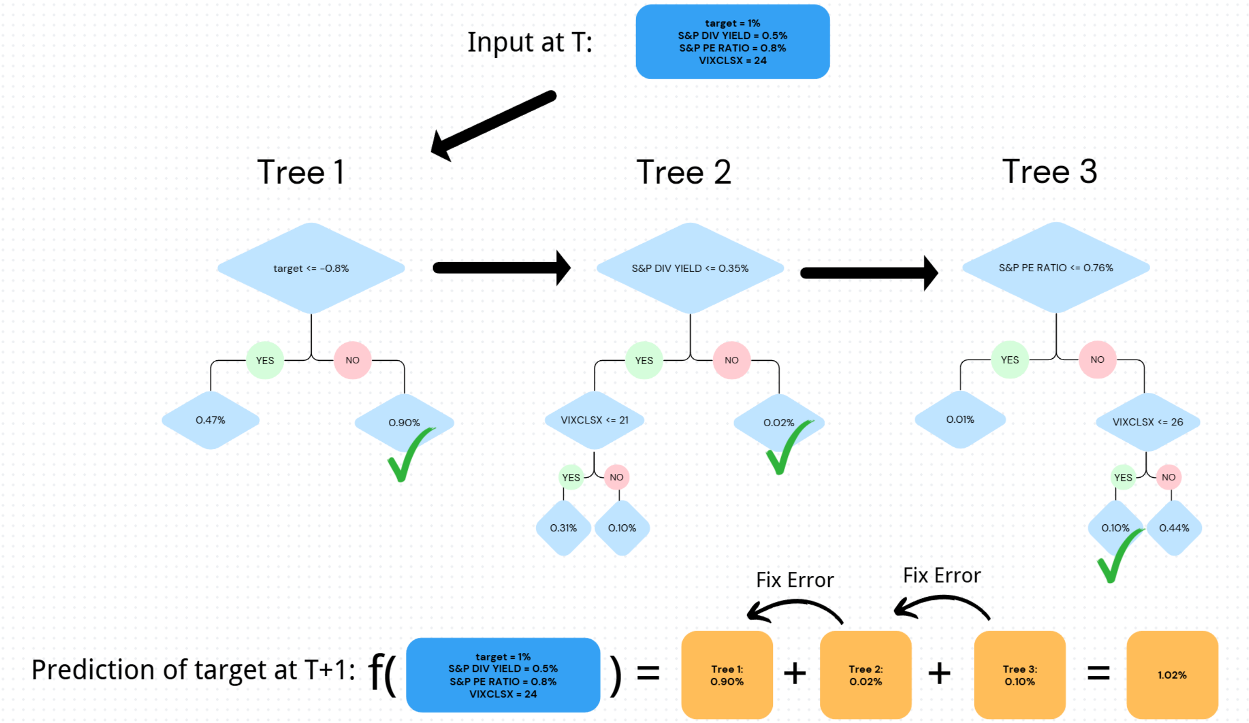


Figure 6: Illustration of LightGBM inference under gradient boosting: each successive tree is fit to the residuals of the current ensemble, and the final prediction is obtained by aggregating per-tree contributions (scaled by the learning rate).

LightGBM is a gradient-boosted decision tree (GBDT) framework that constructs an additive predictive model by sequentially fitting a collection of shallow decision trees. Instead of estimating a single complex function, the boosting procedure iteratively minimizes a differentiable loss: at each iteration, a new tree is trained to approximate the negative gradient of the objective function with respect to the current ensemble predictions, thereby correcting residual errors left by previous trees.

To efficiently identify split points, LightGBM adopts a histogram-based split finding algorithm. Continuous features are discretized into a finite number of bins, and candidate splits are evaluated using aggregated gradient and Hessian statistics to maximize the reduction in the objective function. Tree growth follows a leaf-wise strategy, whereby the leaf that yields the largest gain is split at each step. This approach allows the model to capture complex nonlinear interactions under a fixed leaf budget while maintaining computational efficiency.

As illustrated in Fig. 6, the first tree produces an initial prediction based on features observed at time T (e.g., dividend yield, P/E ratio, and volatility). Subsequent trees are fit to the residuals of the current ensemble and act as incremental corrections. The final forecast at $T+1$ is obtained by summing the contributions from all trees, scaled by the learning rate, with additional regularization and subsampling mechanisms (e.g., depth and leaf constraints, row and column sampling) employed to mitigate overfitting.

Hyperparameters Used

Table 5: Optimized LightGBM hyperparameters for residual-prediction Model C across bond and equity datasets. Hyperparameters are selected via Optuna with a Tree-structured Parzen Estimator (TPE) sampler over 500 trials to maximize cross-validated R^2

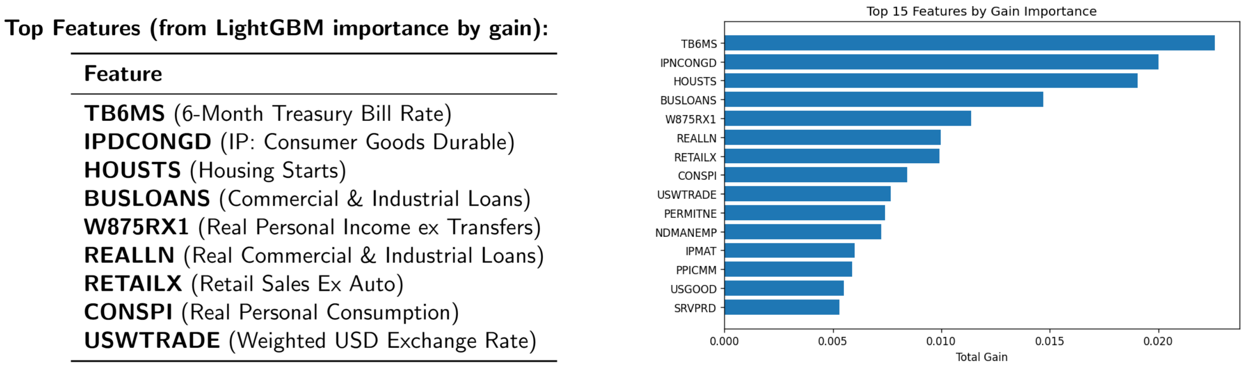
Hyperparameter	What it Controls	Bond Model C	Equity Model C
num_leaves	Maximum tree leaves	31	135
max_depth	Maximum tree depth	6	12
learning_rate	Boosting learning rate	0.0776	0.0823
min_child_samples	Minimum data in leaf	10	10
subsample	Row sampling ratio	0.687	0.687
colsample_bytree	Column sampling ratio	0.839	0.952
reg_alpha	L1 regularization	0.142	0.020
reg_lambda	L2 regularization	0.650	0.585
n_estimators	Number of boosting rounds	3	3

Table 5 reports the key hyperparameters used in the residual-prediction LightGBM models (Model C) and highlights systematic differences between the bond and equity specifications. The table summarizes the principal controls governing model complexity and regularization, including tree depth, number of leaves, learning rate, subsampling ratios, and ℓ_1/ℓ_2 penalties. All hyperparameters are tuned using Optuna's Tree-structured Parzen Estimator (TPE) sampler over 500 trials, with the objective of maximizing cross-validated R^2.

The resulting configurations reveal clear structural contrasts between the two asset classes. The equity model is characterized by deeper trees and a substantially larger number of leaves, indicating richer nonlinear interactions in equity returns. In contrast, the bond model favors shallower trees and stronger ℓ_1 regularization, yielding a more parsimonious and sparse representation. These differences are consistent with the view that bond returns are driven by a smaller set of dominant risk factors, whereas equity returns exhibit more complex nonlinear dynamics.

Overall, the optimized hyperparameters not only improve predictive performance but also provide economically interpretable evidence of fundamental differences in the return-generating processes across asset classes.

Explainability: Equity Model



Top Features (from LightGBM importance by gain):

Feature
TB6MS (6-Month Treasury Bill Rate)
IPDCONGD (IP: Consumer Goods Durable)
HOUSTS (Housing Starts)
BUSLOANS (Commercial & Industrial Loans)
W875RX1 (Real Personal Income ex Transfers)
REALLN (Real Commercial & Industrial Loans)
RETAILX (Retail Sales Ex Auto)
CONSPI (Real Personal Consumption)
USWTRADE (Weighted USD Exchange Rate)

(a) Top macroeconomic features ranked by gain-based importance.

(b) Top 15 features by gain importance from the LightGBM model.

Figure 7: Feature importance from the LightGBM residual-prediction equity model. Gain measures the cumulative reduction in the loss function attributable to splits on each feature across all trees.

Figures 7a and 7b report feature importance measures derived from the LightGBM residual-prediction model, based on gain—the cumulative reduction in the objective function resulting from splits on a given feature across all trees. This metric highlights variables that contribute most strongly to improving predictive accuracy within the nonlinear ensemble.

Short-term interest rates, proxied by the 6-month Treasury bill rate (TB6MS), emerge as the most influential feature. This finding underscores the central role of monetary policy conditions in shaping equity returns, consistent with their impact on discount rates, risk premia, and broader financial conditions. Consumer- and demand-oriented indicators—such as durable goods production, real consumption, and housing starts—also rank prominently, reflecting the importance of growth expectations and cyclical dynamics for equity valuation.

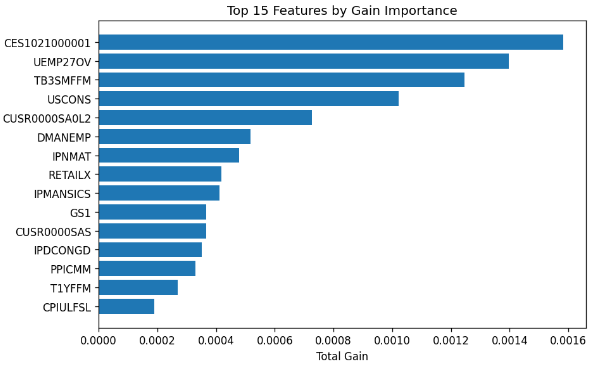
Measures of credit availability and income, including commercial and industrial loans and real personal income, further contribute to explanatory power, highlighting the link between financing conditions, corporate profitability, and equity performance. Finally, the weighted U.S. dollar exchange rate appears among the top features, indicating that external competitiveness and currency valuation play a nontrivial role in determining equity returns.

Explainability: Bond Model

Top Features (from LightGBM importance by gain):

Feature
CES1021000001 (Construction Employment)
UEMP27OV (Unemployment 27+ Weeks)
TB3SMFFM (3-Mo T-Bill minus Fed Funds)
USCONS (Construction Spending)
CUSR0000SA0L2 (CPI Less Shelter)
DMANEMP (Durable Goods Employment)
IPNMAT (IP: Nondurable Materials)
RETAILX (Retail Sales Ex Auto)
IPMANSICS (IP: Manufacturing)

(a) Top macroeconomic features ranked by gain-based importance.



(b) Top 15 features by gain importance from the LightGBM model.

Figure 8: Feature importance from the LightGBM residual-prediction bond model. Gain measures the cumulative reduction in the loss function attributable to splits on each feature across all trees.

Figures 8a and 8b report gain-based feature importance from the **bond** LightGBM residual-prediction model. As in the equity case, gain captures the cumulative reduction in the objective function attributable to splits on each variable, highlighting features that most effectively improve predictive accuracy for bond returns.

Employment- and construction-related variables, including construction employment and construction spending, dominate the importance ranking. These variables proxy real economic activity and labor market conditions, which are closely linked to expectations about future growth, inflationary pressures, and monetary policy responses—key determinants of bond yields and excess bond returns.

Inflation-related measures, such as CPI excluding shelter, and short-term rate spreads, such as the 3-month Treasury bill minus the federal funds rate, also rank prominently. Their importance reflects the sensitivity of bond returns to changes in inflation dynamics and shifts in the expected path of monetary policy, consistent with term structure and expectations-based models of bond pricing.

Manufacturing activity and retail indicators further contribute to explanatory power, complementing traditional interest-rate-based factors by capturing cyclical fluctuations in real activity that influence term premia.

5 Model Assessment

5.1 Performance Metrics

All performance metrics are generated by applying the models to the test dataset.

5.1.1 Individual Models Before Integration

Table 6: Performance Matrices of Three Individual Models for Equity (1-Month Lag)

Model Type	Test R^2	Test RMSE	Test Hit Rate
Model A (OLS)	28.45%	0.0361	77.92%
Model D (EBM)	22.79%	0.0374	75.64%
Model E (LightGBM)	26.42%	0.0366	74.03%

Table 7: Performance Matrices of Three Individual Models for Bond (1-Month Lag)

Model Type	Test R^2	Test RMSE	Test Hit Rate
Model A (OLS)	24.66%	0.0129	67.53%
Model D (EBM)	19.75%	0.0132	64.10%
Model E (LightGBM)	16.76%	0.0135	64.93%

According to Table 6 and Table 7, for both Equity and Bond, the OLS benchmark model is the undoubted winner with the highest test R^2, the lowest test RMSE, and the highest hit rate. The two more boosting machine models do not perform better than the traditional regression model if we separately apply them in predictions.

There are only three common Top features existing in all three Equity models, which are S&P 500 Index, Switzerland / U.S. Foreign Exchange Rate, and Help-Wanted Index for United States. And there are only two common Top features existing in all three Bond models, which are 10-Year Treasury Rate, IP: Residential Utilities. These prove that there exist significant differences in the selected features between traditional regression model and AI model.

Thus, we realize that if only applying these three models separately, we cannot fully harness AI models' strong power of capturing nonlinear effects and learning interactions. Therefore, we have integrated the traditional regression model with two boosting machine models. Let Human collaborate with AI.

5.1.2 Integrated Models

This section compares the performance matrices of **Model A**, **Model B**, and **Model C** under two lag specifications. Model A is a baseline OLS forecast; Model B augments the OLS forecast with an EBM model fitted on OLS residuals; and Model C augments the OLS forecast with a LightGBM model fitted on OLS residuals. Performance is evaluated using three standard test metrics: Test R^2, Test RMSE, and Test Hit Rate.

Table 8: Performance Matrices of Integrated Models for Equity with 1-Month Lag

Model Type	Test R^2	Test RMSE	Test Hit Rate
Model A (OLS)	28.45%	0.0361	77.92%
Model B (EBM + OLS)	28.45%	0.0361	77.92%
Model C (LightGBM + OLS)	28.84%	0.0360	79.22%

According to Table 8, under the one-month lag specification, the Equity results indicate that Model A and Model B are effectively indistinguishable. Both deliver the same Test R^2 of 28.45%, Test RMSE of 0.0361, and Test Hit Rate of 77.92%. In this setting, the EBM residual correction does not yield a measurable incremental gain over the OLS baseline for Equity returns. By contrast, Model C achieves a modest but consistent improvement across all three metrics, with a higher R^2 = 28.84%, a lower RMSE = 0.0360, and a higher Hit Rate = 79.22%.

Table 9: Performance Matrices of Integrated Models for Bond with 1-Month Lag

Model Type	Test R^2	Test RMSE	Test Hit Rate
Model A (OLS)	24.66%	0.01285	67.53%
Model B (EBM + OLS)	23.87%	0.01292	68.83%
Model C (LightGBM + OLS)	26.41%	0.01270	70.13%

According to Table 9, for Bond results under the same one-month lag, Model C again delivers the strongest overall performance with a higher Test R^2 = 26.41%, a lower RMSE = 0.01270, and a higher Hit Rate = 70.13%. The performance gaps between Model C and other two models are even larger for Bond returns than for Equity returns.

Taken together across equity and bond panels, under the one-month lag specification, the **Predictive Accuracy Ordering is: Model C $>$ Model A $\geq$ Model B**.

Beyond predictive metrics, the models are also compared along two model-structure-related dimensions. The first dimension is Model Transparency, which is defined as the extent to which the model can be directly read and understood from its functional form and parameters. Under this definition, Model A is most transparent because it is a single linear regression with a moderate number of coefficients. The OLS fitting is constructed from the pool of 126 candidate FRED features but retains only a selected subset (15 predictors) after multi-criteria feature selection, which supports the direct coefficient-based inspection. Model B is less transparent than pure OLS because it adds a non-linear residual layer (EBM) on top of the linear baseline, even though the residual component remains relatively structured. Model C is least transparent because its residual layer is a LightGBM ensemble with many splits and interactions, making it difficult to interpret directly from model parameters.

Accordingly, the **Transparency Ordering is: Model A $\geq$ Model B $\gg$ Model C**.

The second dimension is Economic Interpretability, which is defined as a distinct concept: it refers to how naturally the model structure maps into an economically meaningful narrative. Model B ranks highest on this dimension because it preserves the OLS component as a clean set of linear exposures while using EBM to model the remaining residual variation via smooth, additive shape functions (and only a limited number of interactions when explicitly included). This interpretation is strengthened by the residual-modeling design that the residual models are trained on macro-only features, specifically, 116 macro variables after excluding ten "price-like" predictors, so that the residual learner is directed toward non-linear macro structure rather than re-using dominant price signals. Model A remains economically interpretable at the linear level, but it cannot represent non-linear macro effects by construction. Model C, despite strong predictive accuracy in the one-month lag setting, is hardest to interpret economically because the LightGBM residual layer can encode irregular, high-order statistical interactions that are difficult to summarize into a compact economic mechanism.

Thus, the **Economic Interpretability Ordering is: Model B $>$ Model A $\gg$ Model C**

Table 10: Performance Matrices of Integrated Models for Equity with 2-Month Lag

Model Type	Test R^2	Test RMSE	Test Hit Rate
Model A (OLS)	0.68%	0.0427	59.74%
Model B (EBM + OLS)	0.69%	0.0427	59.74%
Model C (LightGBM + OLS)	-1.72%	0.0432	59.74%

According to Table 10, under the two-month lag specification, predictive performance deteriorates materially relative to the one-month case. For Equity returns, Model A and Model B remain nearly identical, with Test R^2 of 0.68% and 0.69%, identical RMSE = 0.0427, and identical Hit Rate = 59.74%. Model C under-performs other two models in two-month lag setting, with an even negative Test R^2 = -1.72% and a higher RMSE = 0.0432.

Table 11: Performance Matrices of Integrated Models for Bond with 2-Month Lag

Model Type	Test R^2	Test RMSE	Test Hit Rate
Model A (OLS)	-9.91%	0.01568	50.65%
Model B (EBM + OLS)	-11.04%	0.01576	48.05%
Model C (LightGBM + OLS)	-12.56%	0.01587	48.05%

According to Table 11, for Bond returns, all three models exhibit negative test R^2, where Model A performs best (least badly).

Overall, in two-month lag setting, the residual-boosting does not improve predictive accuracy beyond the OLS baseline and can be detrimental, particularly for the LightGBM residual model. The around-50% hit rates suggest that, when macroeconomic variables lead returns by two months, all three models contain little to no out-of-sample directional information: their sign predictions are essentially no better than a coin flip, implying that the macro features' ability to forecast return direction largely vanishes at this horizon and is dominated by noise or unstable relationships. The close-to-zero and even negative R^2 values indicate that these models' out-of-sample squared prediction errors exceed those of a naive benchmark that always predicts the test-sample mean; in other words, the models fail not only to improve directional accuracy but also to outperform the simplest mean-based baseline in terms of magnitude (mean-squared-error) forecasting.

The reason for introducing the two-month lag specification is to avoid possible look-ahead bias. Think about an example, if we want to predict the price change from December 1st, 2025 to January 1st, 2026, or said January return, for avoiding look ahead bias, we should only use the information of macro features up to December 1st, but we find a problem in the FRED database, where the date index of each feature cannot be ensured to be the same as the real published date, some features recorded with the time index of December 1st may be actually published on some days later than December 1st, which bring the risk of look-ahead bias. For avoiding this risk, the safest and simplest way is to add one more lag between the features and returns, however, the significant deterioration of performance of models suggests that this method is too conservative, it will throw out some information that is actually legal to be incorporated in the model. The bad results under two-month lag specification do not mean that our fitted models are useless in forecasting. Instead, according to the results under one-month lag specification, we believe that our integration methods improve the predictive accuracy to a certain extent, but we also need to further explore a more suitable method to eliminate the risk of look-ahead bias in the future, to make our performance evaluations of these models more realistic and trustworthy.

5.2 Equity Models Confusion Matrix

The following figures present the confusion matrix for predicting equity index returns from all three models. We see that all three equities models perform very similarly in terms of binary classification using confusion matrix in either cases. We observe that Model C (Linear + LightGBM) performs slightly better than other two models in the 1 Month Lag case. In the case of equity, we believe that EBM and LightGBM do not learn much useful information on top of the ones learned by linear model.

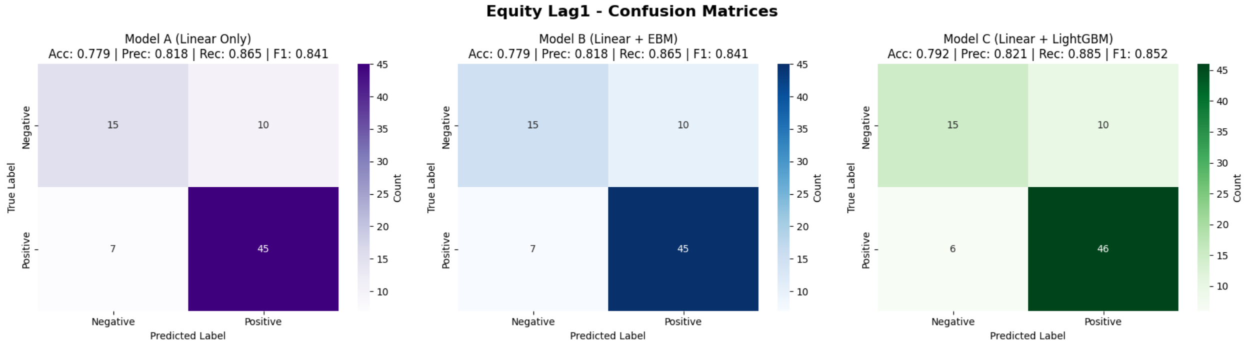


Figure 9: Equity Models Confusion Matrix with 1 Month Lag between X and Y

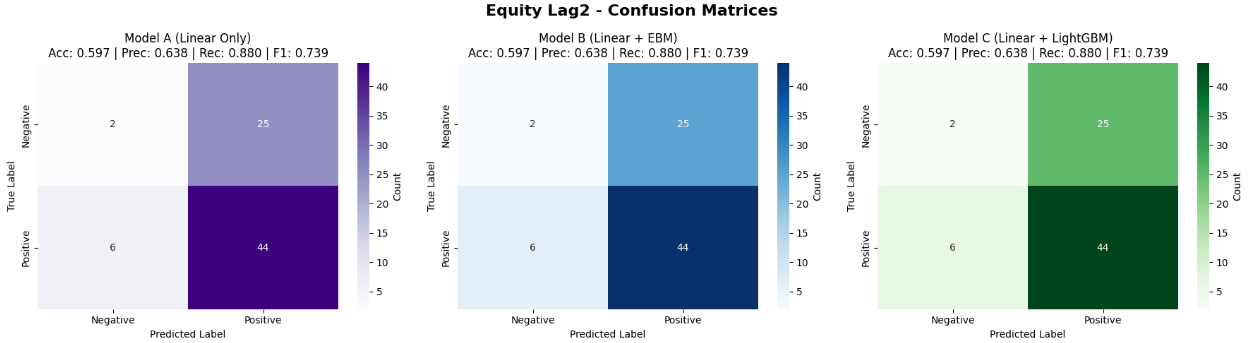


Figure 10: Equity Models Confusion Matrix with 2 Months Lag between X and Y

5.3 Bond Models Confusion Matrix

The following figures present the confusion matrix for predicting bond index returns from all three models. In the case of one month lag between x and y variables, we observe that Model C (Linear + LightGBM) performs the best and Model A (Linear Model) performs the worst out of all three models. This might suggest that both LightGBM and EBM learn new information about relationship between macro variables and equity index return that is not learned by the linear model. In the case of two months lag between x and y variables, both Model B and Model C perform worse than Model A, suggesting that in this case EBM and LightGBM don't learn useful information in addition to linear predictions.

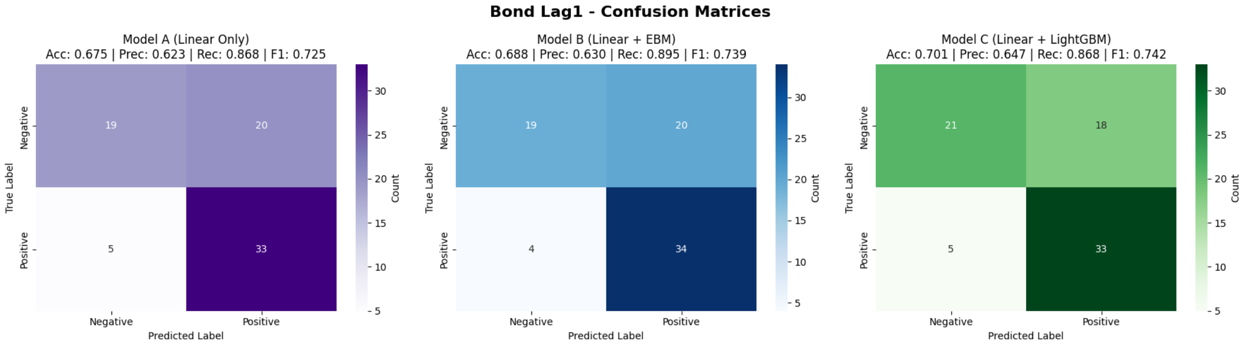


Figure 11: Bond Models Confusion Matrix with 1 Month Lag between X and Y

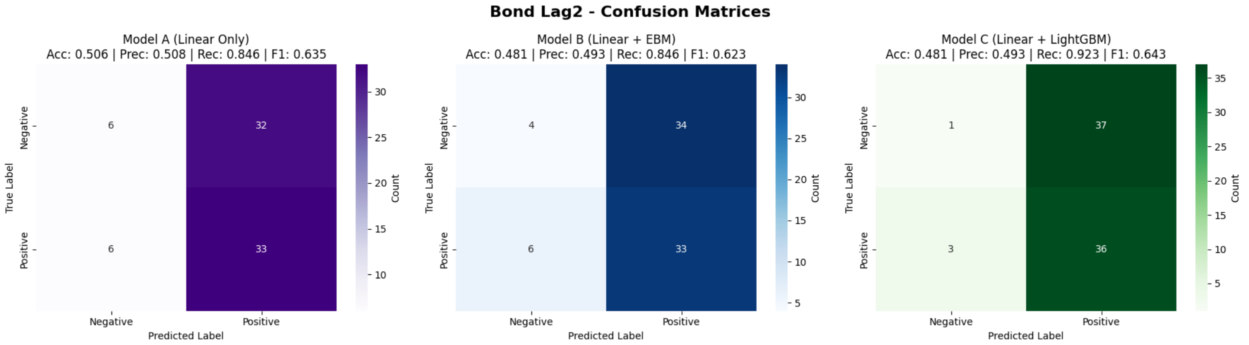


Figure 12: Bond Models Confusion Matrix with 2 Months Lag between X and Y

6 Backtesting

6.1 Equity Models Backtest Performance Analysis

For equity predictions at 1-month lag, all three models demonstrate exceptional but very similar outperformance, with Model C achieving the highest performance with 543% cumulative return versus 129% for the buy-and-hold benchmark—a 4.2x improvement. Model C maintains superior risk-adjusted metrics with a Sharpe ratio of 2.45, Sortino ratio of 4.74, and reduced maximum drawdown of −18.41% (versus −24.12% for buy-and-hold). The maximum drawdown period is also compressed from 23 months to 11 months, demonstrating faster recovery.

However, performance degrades significantly at 2-month prediction horizons. For equities, Model C's cumulative return drops from 543% to 98%, though it still marginally outperforms the benchmark (97.75% vs. 93.29%). Notably, Models A and B underperform with higher drawdowns (−29.65%), while Model C maintains benchmark-level risk.

The portfolio evolution charts illustrate Model C's outperformance among three models in terms of return and risk metrics. The divergence between 1-month and 2-month performance highlights the potential inflation brought by look-ahead bias due to macro statistics reported after first day of the month as well as decays in alpha brought by legal macro features.

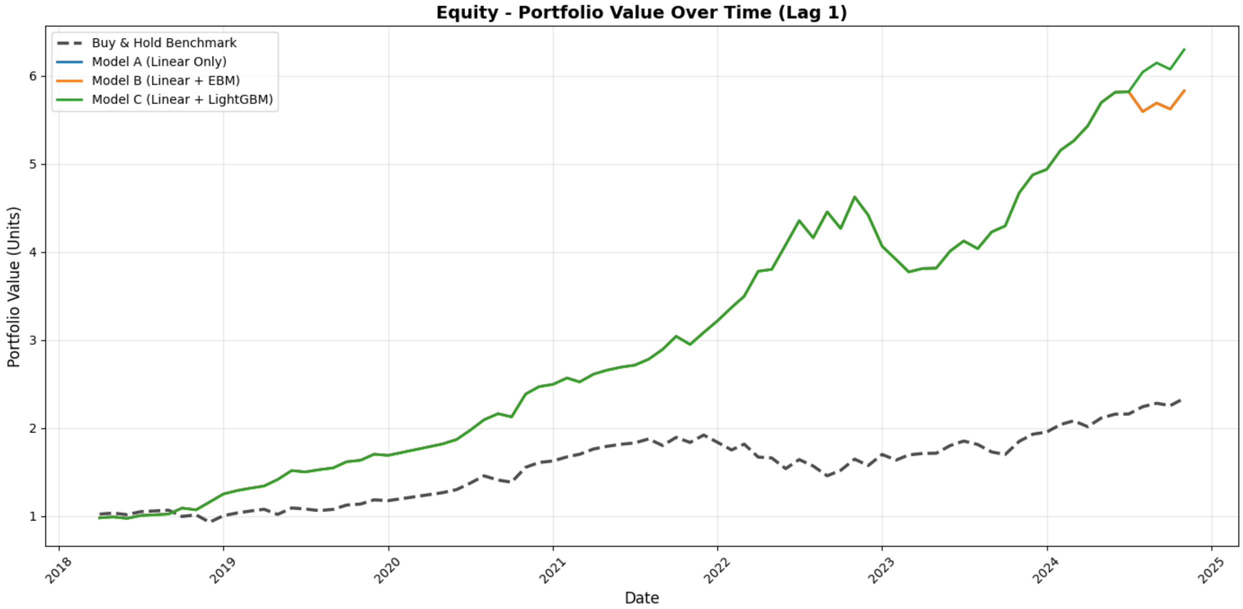


Figure 13: Equity Models (1 Month Lag) Backtest Chart

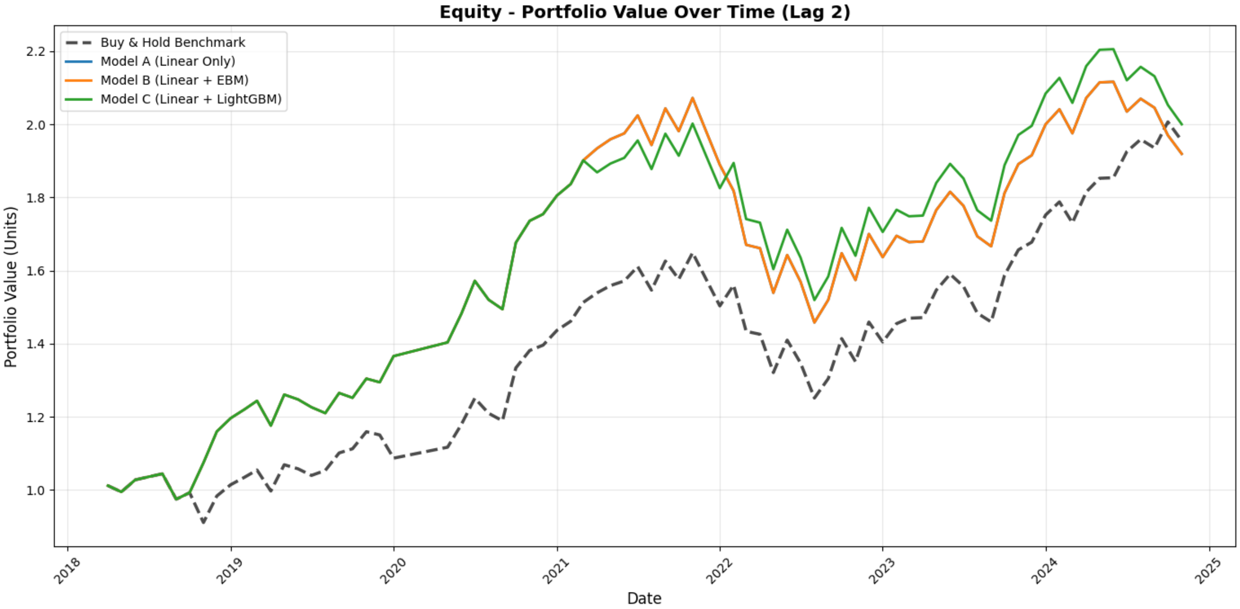


Figure 14: Equity Models (2 Month Lag) Backtest Chart

Table 12: Backtest Performance: Equity Returns (1 Month Lag)

Metric	**Buy & Hold**	**Model A**	**Model B**	**Model C**
Cumulative Return	128.77%	495.47%	495.47%	**543.27%**
Annualized Return	13.77%	32.06%	32.06%	**33.65%**
Annualized Volatility	14.97%	12.99%	12.99%	**12.75%**
Sharpe Ratio	0.76	2.28	2.28	**2.45**
Sortino Ratio	1.40	4.61	4.61	**4.74**
Calmar Ratio	0.57	1.74	1.74	**1.83**
Maximum Drawdown	-24.12%	-18.41%	-18.41%	**-18.41%**
Max DD Period	2022-01 to 2023-11	2022-12 to 2023-10	2022-12 to 2023-10	2022-12 to 2023-10

Table 13: Backtest Performance: Equity Returns (2 Month Lag)

Metric	**Buy & Hold**	**Model A**	**Model B**	**Model C**
Cumulative Return	93.29%	89.76%	89.76%	**97.75%**
Annualized Return	10.82%	10.50%	10.50%	**11.21%**
Annualized Volatility	15.04%	15.06%	15.06%	**15.02%**
Sharpe Ratio	0.56	0.54	0.54	**0.59**
Sortino Ratio	1.06	1.13	1.13	**1.22**
Calmar Ratio	0.45	0.35	0.35	**0.46**
Maximum Drawdown	-24.12%	-29.65%	-29.65%	**-24.12%**
Max DD Period	2021-12 to 2023-10	2021-12 to 2024-04	2021-12 to 2024-04	2021-12 to 2023-12

6.2 Bond Models Backtest Performance Analysis

The bond models exhibit strong performance at 1-month horizons but also experience substantial degradation at 2-month lags similar to equity models. At 1-month lag, Model C (Linear + LightGBM) demonstrates best performance with 75.58% cumulative returns versus 6.83% for the buy-and-hold benchmar while maintaining lower volatility (4.49% vs. 5.19%) and reduced maximum drawdown (−2.84% vs. −14.86%). The Sharpe ratio of 1.51 indicates strong risk-adjusted returns, substantially exceeding the benchmark's negative Sharpe ratio of −0.26, with Fed Funds Rate of each month being the the risk free rate. All ML models compress the maximum drawdown period from 47 months to just 4 months and effectively evaded the bond market crash in 2022, demonstrating faster recovery from adverse periods.

However, the 2-month prediction horizon reveals a huge drop in model efficacy. Model C's performance collapses dramatically, returning only 3.75% with a negative Sharpe ratio of −0.35, underperforming even the benchmark's 7.82% return. Interestingly, Model B (Linear + EBM) emerges as the most robust at 2-month lag, delivering 9.48% returns with the lowest drawdown (−12.34%) and the best Sharpe ratio (−0.19) among all strategies. This suggests that EBM's interpretable additive structure provides more stable predictions when forecasting farther into the future, whereas LightGBM's complex interactions may capture noise rather than signal at extended horizons. Such drastic drop in all model performance from 1 month lag to 2 month lag scenario again illustrates the inflation of backtest performance caused by look-ahead bias and also legal macro features at 1 month lag.

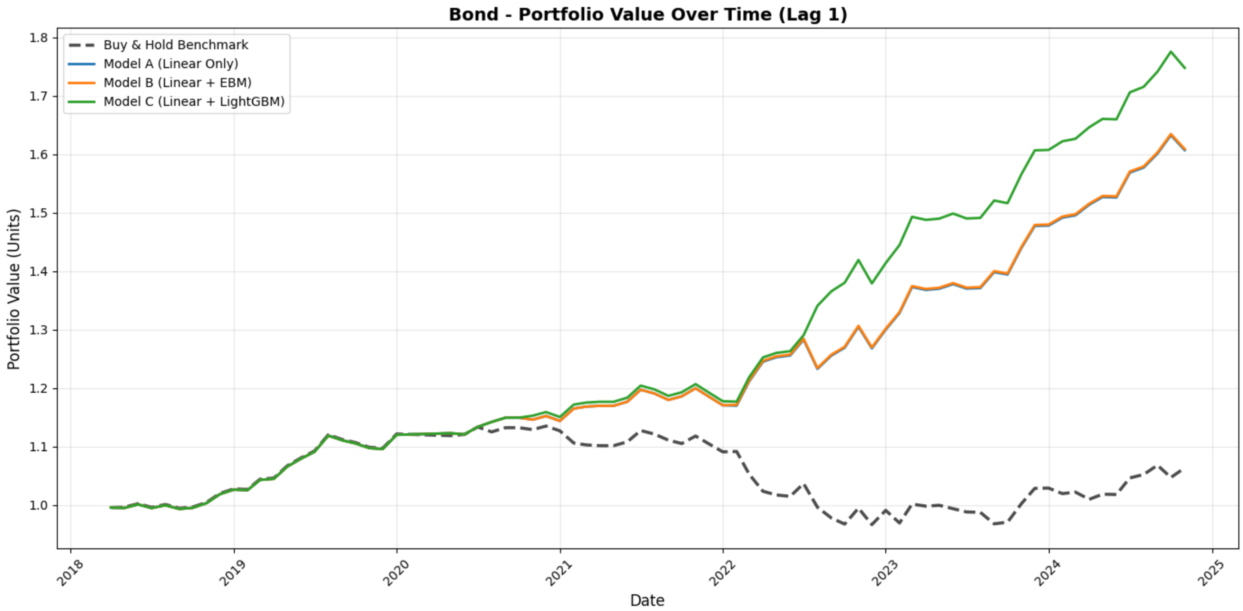


Figure 15: Bond Models (1 Month Lag) Backtest Chart

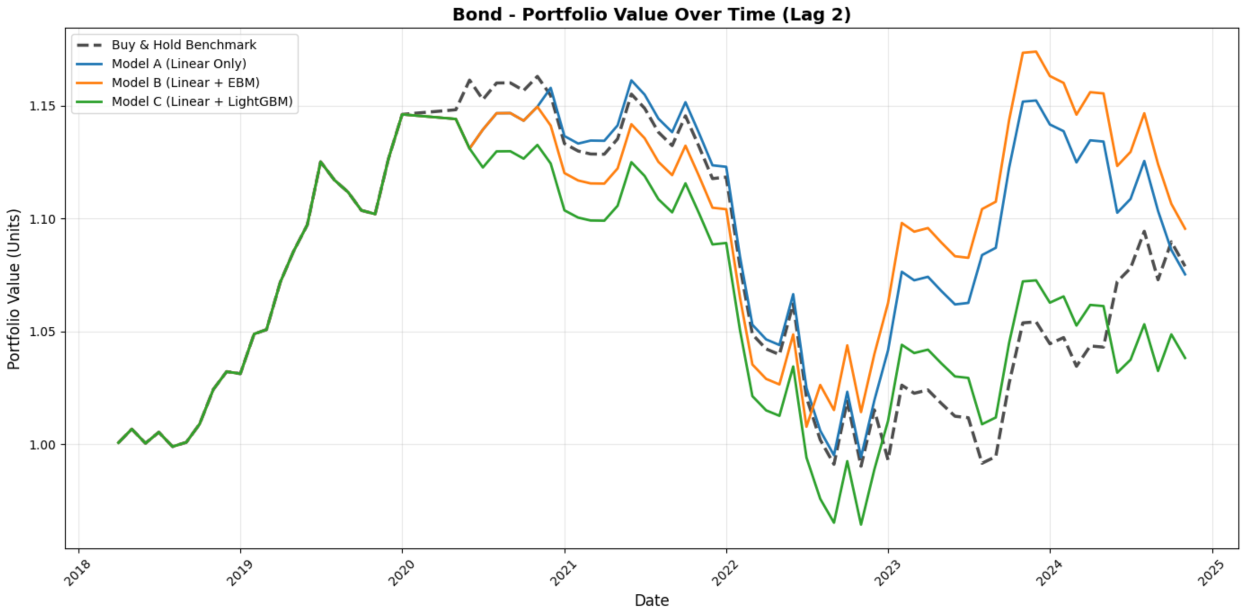


Figure 16: Bond Models (2 Month Lag) Backtest Chart

Table 14: Backtest Performance: Bond Returns (1 Month Lag)

Metric	**Buy & Hold**	**Model A**	**Model B**	**Model C**
Cumulative Return	6.83%	61.45%	61.64%	**75.58%**
Annualized Return	1.03%	7.75%	7.77%	**9.17%**
Annualized Volatility	5.19%	4.70%	4.70%	**4.49%**
Sharpe Ratio	-0.26	1.14	1.14	**1.51**
Sortino Ratio	-0.40	1.69	1.68	**3.01**
Calmar Ratio	0.07	1.99	1.99	**3.23**
Maximum Drawdown	-14.86%	-3.90%	-3.90%	**-2.84%**
Max DD Period	2021-01 to 2024-11	2018-07 to 2018-10	2018-07 to 2018-10	2018-07 to 2018-10

Table 15: Backtest Performance: Bond Returns (2 Month Lag)

Metric	**Buy & Hold**	**Model A**	**Model B**	**Model C**
Cumulative Return	7.82%	7.46%	**9.48%**	3.75%
Annualized Return	1.18%	1.13%	**1.42%**	0.58%
Annualized Volatility	5.25%	5.25%	**5.24%**	5.26%
Sharpe Ratio	-0.23	-0.24	**-0.19**	-0.35
Sortino Ratio	-0.36	-0.37	**-0.29**	-0.53
Calmar Ratio	0.08	0.08	**0.12**	0.04
Maximum Drawdown	-14.86%	-14.39%	**-12.34%**	-15.87%
Max DD Period	2020-12 to 2024-11	2021-07 to 2024-11	2020-12 to 2023-10	2020-05 to 2024-11

7 Conclusion

This paper develops an interpretable macroeconomic forecasting framework that bridges traditional regression models and advanced boosting machine learning techniques, a practical path from black-box AI predictive modeling toward a ”crystal-box” approach. By integrating an economically interpretable OLS model with residual modeling by Explainable Boosting Machines and Light Gradient Boosting Machines, we demonstrate that it is possible to improve predictive performance and trading outcomes, after more effectively controlling for look-ahead bias.

For predictions with 1 month lag between features and indices returns:

- **Predictive Accuracy:** OLS + LightGBM $>$ OLS $\geq$ OLS + EBM
- **Model Transparency:** OLS $\geq$ OLS + EBM $\gg$ OLS + LightGBM
- **Economic Interpretability:** OLS + EBM $>$ OLS $\gg$ OLS + LightGBM

For predictions with 2 month lag between features and indices returns:

- **Predictive Accuracy:** OLS $\approx$ OLS + EBM $>$ OLS + LightGBM

8 Executive Summary (Key 3 Messages)

- OLS learns good amount of information to predict returns on equity and bond indices, and performance can be enhanced by using machine learning models to learn residuals of OLS prediction, after controling for look-ahead bias and retaining legal macro features. The overall strategy would outperform standalone models and buy-and-hold benchmark in most settings.
- Macroeconomic signals related to monetary policy, consumption, housing, and labor markets consistently emerge as key drivers of both equity and bond returns.
- Predictive gains and trading performance decay rapidly as the forecast horizon extends, highlighting the importance of information timeliness and careful handling of look-ahead bias in macro-based strategies.

9 Limitations and Future Work

Several limitations remain critical to this project. The analysis relies on U.S.-centric macroeconomic indicators, which may not fully capture global dynamics embedded in international asset indices. Model performance is also sensitive to data release timing, transformation choices, and lag assumptions. Future research could extend the framework to multi-country macro panels, higher-frequency data, proprietary data with controlled look-ahead bias and regime-dependent modeling. Incorporating real-time data vintages, alternative explainability methods, and portfolio-level allocation rules would further enhance robustness and practical relevance.

References

[1] Baumann, Friedrich, Abdolreza Nazemi, and Frank J. Fabozzi. *Macroeconomic Drivers of Stocks and Bonds*. CFA Institute Research Foundation, 2025.

[2] Ma, Yuanyuan, et al. “Credit Default Prediction of Chinese Real Estate Listed Companies Based on Explainable Machine Learning.” *Finance Research Letters*, vol. 58, 2023, article 104305.

[3] Yang, Jie. “Application of LightGBM in the Chinese Stock Market.” *Proceedings of the 4th International Conference on Big Data Information and Computer Network BDICN 2025*. 2025. Preprint, doi:10.20944/preprints202501.0303.v1.

Appendix: FRED-MD Variable Definitions and Transformations

The column tcode denotes the following data transformation for a series x_t: (1) no transformation; (2) Δx_t; (3) $\Delta^2 x_t$; (4) $\log(x_t)$; (5) $\Delta \log(x_t)$; (6) $\Delta^2 \log(x_t)$; (7) $\Delta(x_t/x_{t-1} - 1.0)$. Variables marked with an asterisk (*) indicate adjustments relative to raw FRED data. :contentReference[oaicite:1]index=1

Table 16: Group 1: Output and income

id	tcode	fred	description
1	1	RPI	Real Personal Income
2	2	W875RX1	Real personal income ex transfer receipts
3	6	INDPRO	IP Index
4	7	IPFPNSS	IP: Final Products and Nonindustrial Supplies
5	8	IPFINAL	IP: Final Products (Market Group)
6	9	IPCONGD	IP: Consumer Goods
7	10	IPDCONGD	IP: Durable Consumer Goods
8	11	IPNCONGD	IP: Nondurable Consumer Goods
9	12	IPBUSEQ	IP: Business Equipment
10	13	IPMAT	IP: Materials
11	14	IPDMAT	IP: Durable Materials
12	15	IPNMAT	IP: Nondurable Materials
13	16	IPMANSICS	IP: Manufacturing (SIC)
14	17	IPB51222s	IP: Residential Utilities
15	18	IPFUELS	IP: Fuels
16	20	CUMFNS	Capacity Utilization: Manufacturing

Table 17: Group 2: Labor market

id	tcode	fred	description
21	2	HWI	Help-Wanted Index for United States
22	2	HWIURATIO	Ratio of Help Wanted/No. Unemployed
23	5	CLF16OV	Civilian Labor Force
24	5	CE16OV	Civilian Employment
25	2	UNRATE	Civilian Unemployment Rate
26	2	UEMPMEAN	Average Duration of Unemployment (Weeks)
27	5	UEMPLT5	Civilians Unemployed - Less Than 5 Weeks
28	5	UEMP5TO14	Civilians Unemployed for 5–14 Weeks
29	5	UEMP15OV	Civilians Unemployed - 15 Weeks & Over
30	5	UEMP15T26	Civilians Unemployed for 15–26 Weeks
31	5	UEMP27OV	Civilians Unemployed for 27 Weeks and Over
32	5	CLAIMSx	Initial Claims
33	5	PAYEMS	All Employees: Total nonfarm
34	5	USGOOD	All Employees: Goods-Producing Industries
35	5	CES1021000001	All Employees: Mining and Logging: Mining
36	5	USCONS	All Employees: Construction
37	5	MANEMP	All Employees: Manufacturing
38	5	DMANEMP	All Employees: Durable goods
39	5	NDMANEMP	All Employees: Nondurable goods
40	5	SRVPRD	All Employees: Service-Providing Industries
41	5	USTPU	All Employees: Trade, Transportation & Utilities
42	5	USWTRADE	All Employees: Wholesale Trade
43	5	USTRADE	All Employees: Retail Trade
44	5	USFIRE	All Employees: Financial Activities
45	5	USGOVT	All Employees: Government
46	1	CES0600000007	Avg Weekly Hours : Goods-Producing
47	2	AWOTMAN	Avg Weekly Overtime Hours : Manufacturing
48	1	AWHMAN	Avg Weekly Hours : Manufacturing
127	6	CES0600000008	Avg Hourly Earnings : Goods-Producing
128	6	CES2000000008	Avg Hourly Earnings : Construction
129	6	CES3000000008	Avg Hourly Earnings : Manufacturing

Table 18: Group 3: Housing

id	tcode	fred	description
50	4	HOUST	Housing Starts: Total New Privately Owned
51	4	HOUSTNE	Housing Starts, Northeast
52	4	HOUSTMW	Housing Starts, Midwest
53	4	HOUSTS	Housing Starts, South
54	4	HOUSTW	Housing Starts, West
55	4	PERMIT	New Private Housing Permits (SAAR)
56	4	PERMITNE	New Private Housing Permits, Northeast (SAAR)
57	4	PERMITMW	New Private Housing Permits, Midwest (SAAR)
58	4	PERMITS	New Private Housing Permits, South (SAAR)
59	4	PERMITW	New Private Housing Permits, West (SAAR)

Table 19: Group 4: Consumption, orders, and inventories

id	tcode	fred	description
3	5	DPCERA3M086SBEA	Real personal consumption expenditures
4	5	CMRMTSPLx	Real Manu. and Trade Industries Sales
5	5	RETAILx	Retail and Food Services Sales
64	5	ACOGNO	New Orders for Consumer Goods
65	5	AMDMNOx	New Orders for Durable Goods
66	5	ANDENOx	New Orders for Nondefense Capital Goods
67	5	AMDMUOx	Unfilled Orders for Durable Goods
68	5	BUSINVx	Total Business Inventories
69	2	ISRATIOx	Total Business: Inventories to Sales Ratio
130	2	UMCSENTx	Consumer Sentiment Index

Table 20: Group 5: Money and credit

id	tcode	fred	description
70	6	M1SL	M1 Money Stock
71	6	M2SL	M2 Money Stock
72	5	M2REAL	Real M2 Money Stock
73	6	BOGMBASE	Monetary Base
74	6	TOTRESNS	Total Reserves of Depository Institutions
75	7	NONBORRES	Reserves Of Depository Institutions
76	6	BUSLOANS	Commercial and Industrial Loans
77	6	REALLN	Real Estate Loans at All Commercial Banks
78	6	NONREVSL	Total Nonrevolving Credit
79	2	CONSPI	Nonrevolving consumer credit to Personal Income
132	6	DTCOLNVHFNM	Consumer Motor Vehicle Loans Outstanding
133	6	DTCTHFNM	Total Consumer Loans and Leases Outstanding
134	6	INVEST	Securities in Bank Credit at All Commercial Banks

Table 21: Group 6: Interest and exchange rates

id	tcode	fred	description
84	2	FEDFUNDS	Effective Federal Funds Rate
85	2	CP3Mx	3-Month AA Financial Commercial Paper Rate
86	2	TB3MS	3-Month Treasury Bill
87	2	TB6MS	6-Month Treasury Bill
88	2	GS1	1-Year Treasury Rate
89	2	GS5	5-Year Treasury Rate
90	2	GS10	10-Year Treasury Rate
91	2	AAA	Moody's Seasoned Aaa Corporate Bond Yield
92	2	BAA	Moody's Seasoned Baa Corporate Bond Yield
93	1	COMPAPFFx	3-Month Commercial Paper Minus FEDFUNDS
94	1	TB3SMFFM	3-Month Treasury C Minus FEDFUNDS
95	1	TB6SMFFM	6-Month Treasury C Minus FEDFUNDS
96	1	T1YFFM	1-Year Treasury C Minus FEDFUNDS
97	1	T5YFFM	5-Year Treasury C Minus FEDFUNDS
98	1	T10YFFM	10-Year Treasury C Minus FEDFUNDS
99	1	AAAFFM	Moody's Aaa Corporate Bond Minus FEDFUNDS
100	1	BAAFFM	Moody's Baa Corporate Bond Minus FEDFUNDS
101	5	TWEXAFEGSMTHx	Trade Weighted U.S. Dollar Index
102	5	EXSZUSx	Switzerland / U.S. Foreign Exchange Rate
103	5	EXJPUSx	Japan / U.S. Foreign Exchange Rate
104	5	EXUSUKx	U.S. / U.K. Foreign Exchange Rate
105	5	EXCAUSx	Canada / U.S. Foreign Exchange Rate

Table 22: Group 7: Prices

id	tcode	fred	description
106	6	WPSFD49207	PPI: Finished Goods
107	6	WPSFD49502	PPI: Finished Consumer Goods
108	6	WPSID61	PPI: Intermediate Materials
109	6	WPSID62	PPI: Crude Materials
110	6*	OILPRICEx	Crude Oil, spliced WTI and Cushing
111	6	PPICMM	PPI: Metals and metal products
113	6	CPIAUCSL	CPI : All Items
114	6	CPIAPPSL	CPI : Apparel
115	6	CPITRNSL	CPI : Transportation
116	6	CPIMEDSL	CPI : Medical Care
117	6	CUSR0000SAC	CPI : Commodities
118	6	CUSR0000SAD	CPI : Durables
119	6	CUSR0000SAS	CPI : Services
120	6	CPIULFSL	CPI : All Items Less Food
121	6	CUSR0000SA0L2	CPI : All items less shelter
122	6	CUSR0000SA0L5	CPI : All items less medical care
123	6	PCEPI	Personal Cons. Expend.: Chain Index
124	6	DDURRG3M086SBEA	Personal Cons. Exp: Durable goods
125	6	DNDGRG3M086SBEA	Personal Cons. Exp: Nondurable goods
126	6	DSERRG3M086SBEA	Personal Cons. Exp: Services

Table 23: Group 8: Stock market

id	tcode	fred	description
80*	5	S&P 500	S&P's Common Stock Price Index: Composite
82*	2	S&P div yield	S&P's Composite Common Stock: Dividend Yield
83*	5	S&P PE ratio	S&P's Composite Common Stock: Price-Earnings Ratio
135*	1	VIXCLSx	VIX