『経済科学』第72巻第１・２号（2024年９月発行）
THE ECONOMIC SCIENCE, Vol.72, No.1・2, September 2024

DOI:10. 18999/ecos. 72. 1-2. 11

Economic Recession Prediction using Machine Learning: A MIDAS Approach*

DING Zirui**

This article analyzes the prediction of the recession in the United States by combining machine learning (ML) algorithms and a mixed data sampling approach (MIDAS) to construct the MIDAS-ML models. Some of the most representative machine learning techniques are implemented for modeling and forecasting U.S. recessions. The empirical analysis shows that the MIDAS-ML models outperform the benchmark MIDASLogit/Probit model in prediction abilities, as indicated by the evaluation of various statistical metrics.

Keywords: Recession, Machine Learning, Mixed Data Sampling

* 論文審査受付日：2024年1月17日。採用決定日：2024年４月30日（編集委員会）
Received for publication on January 17 2024. Revision accepted for publication on April 30, 2024.（*Editorial Committee*）

** 名古屋大学大学院経済学研究科博士後期課程
Graduate Student, Graduate School of Economics, Nagoya University

I. Introduction

Predicting recessions is a challenging task due to their diverse nature and causes. For example, the most recent recession in the United States occurred in 2020, triggered by the COVID-19 pandemic. The Great Recession, spanning from 2007 to 2009, resulted from the subprime mortgage crisis[1]. Furthermore, the economic downturn of 2001 was primarily due to the bursting of the ”dot-com bubble” and the “9/11” terrorist attacks. This study successfully uses different machine learning algorithms to predict the three recessions mentioned above. The MIxed DAta Sampling (MIDAS) approach is also used to maximize the utilization of high-frequency financial and economic data while minimizing information loss.

A common approach to predict recessions is to utilize the turning point dates determined by the Business Cycle Dating Committee of the National Bureau of Economic Research (NBER) in the United States. This paper also follows this approach. Researchers have used a variety of variables to model and forecast U.S. economic conditions. Among them, the slope of the U.S. yield curve has received the most attention. The early study of Estrella and Mishkin (1996) has already identified that the yield curve, especially the spread between 10-year and 3-month Treasury bills, is a valuable forecasting tool for recession prediction. Moreover, Estrella and Mishkin (1998) added stock indices, monetary aggregates, macro indicators, and leading indicator variables to the input variables and found that stock prices and some well-known macroeconomic indicators are helpful at one- to three-quarter horizons. The study by Estrella et al. (2003) used binary models that predict either recessions or inflationary pressures and found that the prediction of recessions is stable over their entire sample period in Germany and the United States. Moneta (2005) confirmed the importance of using spreads as predictors of recessions for the euro area and found that the yield spread between the ten-year government bond rate and the three-month interbank rate outperformed all other spreads in predicting recessions in the euro area. Drechsel and Scheufele (2012) focused on single and pooled leading indicator models to analyze the short-term forecasting performance of leading indicators for industrial production and found that pooling can substantially improve the reliability of leading indicator forecasts.

With the development of machine learning technology, more and more machine learning algorithms have been applied to the field of economics. Maehashi and Shintani (2020) used factor models and different machine learning algorithms to predict seven Japanese macroeconomic variables and found that machine learning models have better performance than autoregression models. In the field of recession prediction, many scholars (Coulombe et al., 2021; Gogas et al., 2015; Nyman and Ormerod, 2017; Goulet Coulombe et al., 2022; Vrontos et al., 2021) have employed machine learning techniques to accurately forecast economic recessions. These approaches have demonstrated notable performance in their predictive capabilities. As summarized by Gogas and Papadimitriou (2021), recent machine learning (ML) applications in economics, including business cycles and recession forecasting, seem to be very successful compared to traditional empirical models. Gogas et al. (2015) provided the first empirical study of the relationship between the yield curve and the real output of an economy using the Support Vector Machine classifier, which outperforms the Logit and Probit models regarding overall predictive accuracy. Berge (2015) evaluated the usefulness of various macroeconomic indicators in predicting recessions with different forecasting methods and revealed that the Bayesian model averaging and boosting algorithm demonstrate their effectiveness in producing probability-based recession forecasts. Döpke et al. (2017) used a machine learning approach known as Boosted Regression Trees to reexamine the usefulness of selected leading indicators for predicting recessions. Pierdzioch and Gupta (2019) also estimated Boosted Regression Trees on a sample of monthly data to shed light on the role of disaggregated uncertainty measures for forecasting U.S. recessions.

The study of Vrontos et al. (2021) is the first comprehensive, comparative study of machine learning techniques in the area of economic recession forecasting, which widely uses macroeconomic and financial indicators, focusing on the use of machine learning techniques to predict the probability of U.S. recessions, and showing that machine learning models outperform traditional econometric methods (Logit/Probit model) in recession predicting. However, some of these variables, like the yield curve, the stock index, and the initial claims, are sampled at a higher frequency (daily, weekly) than the recession data (monthly). Typically, high-frequency variables are time-aggregated in this situation, which can lead to some loss of information. To take advantage of high-frequency data, the MIDAS approach introduced by Ghysels et al. (2004), is able to use more information with higher flexibility. Ghysels et al. (2007) also mentioned that the MIDAS can capture the rich dynamics of high-frequency processes simply and concisely. Andreou et al. (2010) derived the asymptotic properties of the MIDAS nonlinear least squares estimator and compared it with traditional least squares estimators and showed that the traditional least squares estimator always has lower efficiency compared to the MIDAS least squares estimator. Moreover, MIDAS was not only limited to regression problems but was also used in classification contexts. Audrino et al. (2019) incorporated MIDAS into the conventional logit model to solve the binary classification problems using mixed frequency data and applied the MIDAS-Logit model to predict U.S. bank failures. Babii et al. (2022) offered a new perspective on the high-dimensional time series regression with data sampled at mixed frequencies to forecast U.S. GDP. Their model combined MIDAS and sparse-group LASSO to establish the model and outperformed other estimators and provides an example of using MIDAS in conjunction with machine learning in the field of economics. Galvão and Owyang (2022) proposed a MIDAS-Probit model and found that the weekly-sampled 10y-3m term spread outperformed the monthly-sampled to predict NBER recessions. Jiang et al. (2023) also used the MIDAS-Logit model in the area of bond ratings.

This paper extends the use of MIDAS to various machine-learning algorithms and further explores the study of Vrontos et al. (2021). Specifically, inspired by the study of Galvão and Owyang (2022), which used a MIDAS-Probit model with a weekly-sampled term spread and demonstrated superior performance compared to monthly-sampled data, this paper extends the application of MIDAS to the field of machine learning. Furthermore, following the recommendation of Xu et al. (2019), various intelligent learning methods are considered, and mixed-frequency analysis methods are applied to build richer, effective models for exploring the nonlinear patterns contained in mixed-frequency data. The choice of algorithms is mainly based on Vrontos et al. (2021), but some proven effective algorithms, such as the XGBoost, the Support Vector Machines, and the Neural Network, are also taken into consideration.

The main contribution of this study is improving the model predictability by applying the MIDAS approach to machine learning algorithms. As mentioned above, the study of Vrontos et al. (2021) using machine learning methods comprehensively for predicting the recession in three different horizons found that the machine learning models mostly beat the benchmark models (Logit/Probit model). Compared with it, our empirical result shows that applying the MIDAS approach together with machine learning methods has better performance than the study by Vrontos et al. (2021)[2] , with accuracy close to 95 percent and the AUC metrics[3] nearly 95 percent. The reason for the better empirical results can be explained as follows: Compared with the Vrontos et al. (2021) study, the addition of the (Unrestricted-) MIDAS (which is short for (U-)MIDAS) method allows data to be analyzed at a finer time resolution, using the information from high-frequency data while preserving the long-term trends present in low-frequency data, which is particularly helpful in improving prediction accuracy. As Xu et al. (2019) concluded, (U-)MIDAS technique integrates variables that are sampled at different frequencies, which allows the method to preserve the rich information present in high-frequency observations while avoiding likely challenges such as information loss, measurement inaccuracies, and timing issues associated with frequency conversion.

Compared with the Galvão and Owyang (2022) study, in which the MIDAS-Probit model is employed for 1-year ahead recession prediction, this study further improves predictive performance by introducing machine learning algorithms. The result of Galvão and Owyang (2022) shows that their best MIDAS-Probit model has an AUC of about 0.885[4)] , but most of the AUC of MIDAS-ML models in this paper is around 0.95, which is much higher.

The remainder parts of this article are divided into the following sections: Section 2 elaborates on the methodology employed in this study. Section 3 introduces the research data used in the study, along with the design of the empirical analysis. Section 4 presents the results of the empirical analysis. Finally, section 5 summarizes the research conclusions.

Ⅱ. Methodology

This section provides an overview of methodologies used in the models employed in predicting recessions in the United States. The (U-)MIDAS approach is utilized to treat variables, and the (U-)MIDAS-Probit and MIDAS-Logit models are used as benchmarks to compare performance with other U-MIDAS-ML[5)] models.

1. MIDAS

The MIDAS method is commonly used to forecast low-frequency data using high-frequency data. Following Xu et al. (2019), I also adopt (U-) MIDAS and combine it with different machine learning algorithms. The MIDAS approach consists of two steps. The first step is frequency alignment, and the second step is parameter restricting[6)]. Compared with the MIDAS, the U-MIDAS approach consists only of the first step of frequency alignment but omits the second step of parameter restricting. U-MIDAS can take full advantage of high-frequency data but the data dimension is not restricted.

First, the frequency alignment is applied on each daily and weekly predictor $d_{\tau}^{[i]}$, $w_{\tau'}^{[i]}$, to obtain $d_{t-l/m}^{[i]}$ and $w_{t-l/m}^{[i]}$ for $l = 0, 1, \cdots, m - 1$, where i denotes the $i - th$ input variable, τ and τ' denote the length of the daily and weekly data respectively, m denotes the mismatched frequency[7)], and t denotes the length of the low-frequency data.

Specifically, this step converts each high-frequency data variable from a matrix format of $(\tau \times 1)$ to a matrix format of $(t \times m)$. It is perhaps easier to introduce matrix notation to provide a clearer perspective. We define $R \equiv [D, W, M]$ as all raw independent variables, which consist of daily data D, weekly data W , and monthly data M, where $D \equiv [d^{[1]}, d^{[2]}, \cdots, d^{[n]}]$ as a $(\tau \times n)$ matrix that groups all the d vectors together, and $W \equiv [w^{[1]}, w^{[2]}, \cdots, w^{[n']}]$ as a $(\tau' \times n')$ matrix that groups all the w vectors together. Using $d^{[i]}$ as an example, the frequency alignment looks like this:

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ \vdots \\ \vdots \\ d_{\tau-1} \\ d_\tau \end{bmatrix} \xrightarrow{Re-Index} \begin{bmatrix} d_{\frac{1}{28}} \\ d_{\frac{2}{28}} \\ d_{\frac{3}{28}} \\ \vdots \\ \vdots \\ \vdots \\ d_{t-\frac{1}{28}} \\ d_t \end{bmatrix} \xrightarrow{Alighment} \begin{bmatrix} d_1 & d_{\frac{27}{28}} & \cdots & d_{\frac{1}{28}} \\ d_2 & d_{1\frac{27}{28}} & \cdots & d_{1\frac{1}{28}} \\ \vdots & & \ddots & \vdots \\ d_{t-1} & d_{(t-2)\frac{27}{28}} & \cdots & d_{(t-2)\frac{1}{28}} \\ d_t & d_{(t-1)\frac{27}{28}} & \cdots & d_{(t-2)\frac{1}{28}} \end{bmatrix} \tag{1}$$

In Equation 1, each daily data $d^{[i]}_1$, $d^{[i]}_2$, ..., $d^{[i]}_{\tau-1}$, $d^{[i]}_\tau$ is re-indexed into $d^{[i]}_{\frac{1}{m}}$, $d^{[i]}_{\frac{2}{m}}$, ..., $d^{[i]}_{t-\frac{1}{m}}$, $d^{[i]}_t$, and then be aligned into the matrix form with t rows and m columns. In this matrix, the first column is the value at the end of each month, and the other columns include information about daily lagged value. As mentioned in footnote 3, m equals 28 for aligning monthly and daily data. In this study, this transformation is performed to align the frequency to monthly for each vector w and d with a higher frequency of weekly or daily.

The aligned data can then take lag and be used as the input variables directly, that is the U-MIDAS method. After the frequency alignment process, the dimension of data and the number of parameters are boosted to a huge number[8]. Higher dimensions can provide a richer and more granulated feature representation, which helps to capture the complex patterns and relationships in the data more effectively, thereby improving the performance of the model. However, it also increases computational complexity, significantly adds to the computer's runtime, and may also increase the risk of overfitting. As a result, the second step of parameter constraint might be necessary. In this step, for each matrix $d^{[i]}$ mentioned above, a multiplication is performed, followed by a transformation into a single-column matrix achieved by the weighted multiplication,

$$x^{[i]} = \sum_{l=0}^{L^{[i]}} d^{[i]}_{t-\frac{l}{28}} \omega^{[i]} \ (\delta \,;\, l) \tag{2}$$

For weekly data, it is the same. In this case, each of the matrix $w^{[i]}$ generated in the frequency alignment step is also transformed to $x_{[i]}$

$$x^{[i]} = \sum_{l=0}^{L^{[i]}} \omega^{[i]}{}_{t-\frac{l}{4}} \omega^{[i]} (\delta \,;\, l) \tag{3}$$

In this study, the two parameters exponential Almon lag polynomial,

$$\omega^{[i]} (\delta \,;\, l) = \frac{\exp(\delta_1 l + \delta_2 l^2)}{\sum_{l=0}^{m-1} \exp(\delta_1 l + \delta_2 l^2)} \tag{4}$$

is adopted as weighted multiplication, where δ_1 and δ_2 are parameters, and l denotes the lag of high frequency data, l is from 0 to $m-1$.

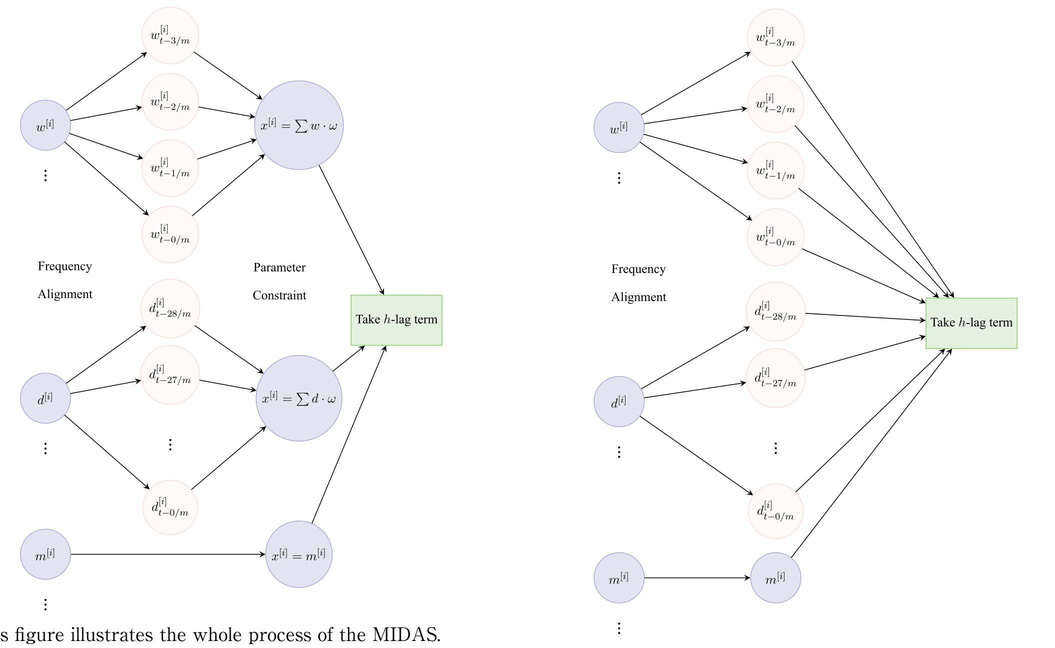
The dimension restriction process can also be shown in matrix form. Define $[x^{[i]}_1, x^{[i]}_2, \ldots, x^{[i]}_t]' \equiv x^{[i]}$ mentioned in Equation 2.

$$\begin{bmatrix} d_1 & d_{\frac{27}{28}} & \cdots & d_{\frac{1}{28}} \\ d_2 & d_{1\frac{27}{28}} & \cdots & d_{1\frac{1}{28}} \\ \vdots & & \ddots & \vdots \\ d_{t-1} & d_{(t-2)\frac{27}{28}} & \cdots & d_{(t-2)\frac{1}{28}} \\ d_t & d_{(t-1)\frac{27}{28}} & \cdots & d_{(t-2)\frac{1}{28}} \end{bmatrix} \times \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_{27} \\ \omega_{28} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{t-1} \\ x_t \end{bmatrix} \tag{5}$$

This study selects the optimal δ using a valid dataset, which will be mentioned below in the empirical design part. After this parameter restriction operation, each variable w, d is converted back into a single-column matrix x again. Then the h-lag term[9] will be taken and the $X \equiv [x^{[1]}_{t-h}, x^{[2]}_{t-h}, x^{[n]}_{t-h}]$ will be used as input variable to the machine learning. This is the whole process of the data pre-processing in this study.

To offer a clear representation, the entire process is depicted in Figure 1. Specifically, in Figure 1 (a), the MIDAS model is illustrated, while in Figure 1 (b), the U-MIDAS model is depicted. The variables w, d, and m represent daily, weekly, and monthly data, respectively, and i denotes the i-th input variable.

Figure 1: Comparison of MIDAS and U-MIDAS Models



(a) Data Pre-process Progress of MIDAS Model

(b) Data Pre-process Progress of U-MIDAS Model

This figure illustrates the whole process of the MIDAS. Circles on the left side are different variables with different frequencies, which are reshaped into a matrix with the same length of monthly data, and finally transformed back to vectors again by multiplying the almon weight function. After that, these transformed vectors will take lag and become input variables for machine learning algorithms or benchmark models.

This figure illustrates the whole process of the U-MIDAS. Circles on the left side are different variables with different frequencies, which are reshaped into a matrix with the same length of monthly data and directly taken lag, to become input variables for machine learning algorithms or benchmark models.

2. Benchmark

(1) MIDAS Logit Model

The MIDAS Logit model is applied as a benchmark to evaluate the performance of machine learning methods, following the study of Vrontos et al. (2021), and Audrino et al. (2019). The estimation equation is:

$$\ln\left(\frac{p_t}{1-p_t}\right) = X_t\beta = \beta_0 + \sum_{i=1}^{N} \beta_i x_t^{[i]}, \tag{6}$$

where p_t is the probability of recession during the time t, X_t is the vector including all input variables, $x^{[1]}, x^{[2]}, \ldots, x^{[N]}$, generated by MIDAS approach mentioned in Section 2.1 in the time t, and β is the parameter vector to be estimated. The left-hand side is the log odds of a recession that happens at time t. To calculate the probability, the equation can be transformed to

$$p_t = \frac{\exp\left(\beta_0 + \sum_{i=1}^{N} \beta_i x^{[i]}{}_t\right)}{1 + \exp\left(\beta_0 + \sum_{i=1}^{N} \beta_i x^{[i]}{}_t\right)} \tag{7}$$

with the loss function of

$$L = \sum_{t=1}^{T} \left(-y_t \log\left(p_t(X_t)\right) - (1-y_t)\log(1-p_t(X_t))\right) \tag{8}$$

where y_t is a binary variable that simply indicates the occurrence (y_t=1) or absence (y_t=0) of a recession at the month t. In this part, the stochastic average gradient descent method is used to minimize the loss function.

(2) MIDAS Probit Model

The MIDAS Probit Model is another benchmark model, with the estimation equation defined as

$$p_t = \Phi(X_t\beta) \tag{9}$$

where Φ is the cumulative normal distribution function, β is a vector of coefficients, and X_t is a vector including the MIDAS transformed independent variables. The Probit model is also estimated by maximum likelihood, with the likelihood function defined as

$$L = \prod_{\{y_t=1\}} \Phi(X_t\beta) \prod_{\{y_t=0\}} [1 - \Phi(X_t\beta)]. \tag{10}$$

3. Machine Learning Methods

The study employs twelve machine learning algorithms, which are briefly described in Table 1. These algorithms can be categorized into four types based on their underlying principles. These algorithms utilize data pre-processed using the U-MIDAS approach as input variables to establish the U-MIDAS-ML[10)] models. The specific details of the algorithms will be provided in the Appendix.

Table 1: The list of machine learning algorithms

Type	Algorithm	Description
Penalized Logit Models	Lasso	Logistic regression with L1 regularization
	Ridge	Logistic regression with L2 regularization
	Elastic Net	Logistic regression with a combination of L1 and L2 regularization
Bayes Models	LDA	Linear Discriminant Analysis, a classifier with a linear decision boundary
	Gaussian NB	Naive Bayes algorithm assuming Gaussian distribution
	Bernoulli NB	Naive Bayes algorithm assuming Bernoulli distribution
Tree-based Models	Classification Tree	An algorithm using tree-based structures to partition samples
	Random Forest	Ensemble of decision trees using random feature and sample selection
	XGBoost	eXtreme Gradient Boosting, a Gradient Boosting algorithm applied on classification trees
Other Models	SVM	Support Vector Machines, an algorithm for finding optimal hyperplanes
	KNN	K-Nearest Neighbors algorithm
	Neural Network	Network with multiple layers and neurons

(1) Penalized Logit Models

The penalized logit models are similar to the standard Logit model in terms of the estimation function, but a regularization term is incorporated into the loss function to avoid overfitting and multicollinearity problem. We established three different models with different regularization terms: the Lasso model, by adding an L1 regularization; the Ridge model, by adding an L2 regularization; and the Elastic Net model, by adding both L1 and L2 regularizations. These three models need different hyperparameters, including the regularization strength for all three models and the ratio of two regularizations for the Elastic Net model. More details will be mentioned in the Appendix A.

(2) Naive Bayes Models

In this study, three different types of Naive Bayes models are used, including the Gaussian Naive Bayes model, the Bernoulli Naive Bayes model, and the Linear Discriminant Analysis (LDA) model. The difference between the first two algorithms is the computation of conditional probabilities, in other words, the different assumptions about the prior distribution. Furthermore, as mentioned in James et al. (2023), LDA is considered to be a special case of the Naive Bayes model, and thus, it is included within this category as well.

(3) Tree-Based models

Tree-based models are built upon the classification tree, a supervised machine learning technique that identifies patterns and generates a prediction model by learning basic decision rules. To enhance the outcomes, two approaches are employed: bagging and boosting. In this study, the Random Forest algorithm is chosen as a representative of the bagging technique, while the eXtreme Gradient Boosting (XGBoost) algorithm is utilized for boosting.

(4) Other models

We also apply some other algorithms, including the Support Vector Machine (SVM), the K-Nearest Neighbors (KNN), and the Artificial Neural Network (ANN).

4. Evaluation of Model Performance

The majority of performance evaluation metrics in classification tasks are based on the confusion matrix, which compares the number of correct and incorrect predictions to the true value, including four combinations of predicted and actual values of True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). The confusion matrix is shown in Table 2.

Table 2: The confusion matrix

	Not Recession	Recession
Predicted value: 0	TN	FN
Predicted value: 1	FP	TP

- True Positive (TP): It refers to the number of cases where the classifier correctly (Truly) predicts economic recession (Positive) among the actual recession cases.
- False Positive (FP): It refers to the number of cases where the classifier incorrectly (Falsely) predicts economic recession (Positive) among the actual non-recession cases.
- True Negative (TN): It refers to the number of cases where the classifier correctly (Truly) predicts non- recession (Negative) among the actual non-recession cases.
- False Negative (FN): It refers to the number of cases where the classifier incorrectly (Falsely) predicts non-recession (Negative) among the actual recession cases.

According to the confusion matrix, the following performance metrics of evaluation models can be calculated. *ROC and AUC.* The Area Under the Curve (AUC) is a measure of classifier accuracy and is commonly used in binary classification. When classifying data, the Receiver Operating Characteristic (ROC) curve is often used to evaluate classifier performance, and the area below the curve is the AUC metric. The ROC space is defined by the false positive rate (FPR) and true positive rate (TPR) horizontally and vertically, respectively. It illustrates the balance between true positive outcomes (benefits) and false positive results (costs).[11] The definition of TPR and FPR is as follows. Defining a set of threshold value *tv* which indicates a recession whenever $\hat{y} \geq tv$, and expansion whenever $\hat{y} \leq tv$. The *tv* is evenly spaced along the interval [0, 1]. More significant numbers of thresholds lead to a smoother ROC curve with more points. For example, a potential set with 11 thresholds would be 0, 0.1, 0.2, . . . , 0.9, 1. Then, the following conditional probabilities can be defined:

$$TPR(tv) = P[\hat{y}_t \geq tv \mid y_t = 1] \tag{11}$$

$$FPR(tv) = P[\hat{y}_t \geq tv \mid y_t = 0] \tag{12}$$

For each *tv*, we create a coordinate: (*TPR*, *FPR*). The ROC is generated by connecting these coordinates. The AUC is the integral of the ROC curve, representing the area under the curve.

$$\mathrm{AUC} = \int_0^1 \mathrm{ROC}(r)\,dr \tag{13}$$

Accuracy. The Accuracy represents the proportion of correctly classified samples out of the total number of samples. It is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

Precision. The Precision represents the proportion of correctly classified recession cases out of the total cases predicted as a recession by the classifier. It is defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

Sensitivity. Sensitivity represents the proportion of actual recession cases that are correctly predicted as a recession by the classifier. It is defined as follows:

$$Sensitivity = \frac{TP}{TP + FN} \tag{16}$$

Specificity. Specificity represents the proportion of actual non-recession cases correctly predicted as non- recession by the classifier. It is defined as follows:

$$Specificity = \frac{TN}{TN + FP} \tag{17}$$

Balanced Accuracy. Balanced Accuracy is usually used to address the problem of inaccurate accuracy due to class imbalance. It is the arithmetic mean of the Sensitivity (true positive rate) and Specificity (true negative rate).

$$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2} \tag{18}$$

F1 score. The F1 score is the harmonic average of Precision and Sensitivity commonly used to evaluate the overall performance of a classifier. It can be seen as a balance point between Precision and Sensitivity and is used to balance the accuracy and completeness of the classifier.

$$\mathrm{F1} = \frac{2{*}Precision\ {*}\ Sensitivity}{Rrecision + Sensitivity} \tag{19}$$

Kappa. The kappa coefficient is another measure of inter-rater reliability (the degree of agreement) between the predicted and actual results from a classifier. It induces the hypothetical probability of chance agreement to overcome the drawback of accuracy influenced by positive and negative example ratios. P_o represents the probability of inter-rater reliability between the classifier and actual results (in other words, the accuracy), while P_e represents the hypothetical probability of chance agreement.[12]

$$\kappa = \frac{P_o - P_e}{1 - P_e} = 1 - \frac{1 - P_o}{1 - P_e} \tag{20}$$

The right-hand side of Equation 20 can be seen as one minus a ratio of model error and baseline error, which implies that a lower model relative error corresponds to a higher κ value.

Pesaran-Timmermann test. The test proposed by Pesaran and Timmermann (1992) examines the accuracy of forecasts and focuses on the correct prediction of the direction. In this study, same as Vrontos et al. (2021), the null hypothesis is that the accuracy does not differ from the ratio that would be obtained in the case of no predictability, where the forecasts $\hat{y}$ and the realized values y are independent. In contrast, the alternative hypothesis is that there exists recession predictability.

The process of Pesaran-Timmermann test is, for time series y and $\hat{y}$ with n elements, firstly define the positive rate function:

$$p(y) = \frac{1}{n}\sum_{t=1}^{n} y_t, \quad p(\hat{y}) = \frac{1}{n}\sum_{t=1}^{n} \hat{y}_t \tag{21}$$

and

$$q(y) = \frac{p(y)(1 - p(y))}{n}, \; q(\hat{y}) = \frac{p(\hat{y})(1 - p(\hat{y}))}{n} \tag{22}$$

Then we calculate p, the same as P_e in calculating the κ statistic, and w, v for calculating the standard deviation.

$$p = p(y)\mathrm{p}(\hat{y}) + (1 - p(y))(1 - p(\hat{y})) \tag{23}$$

$$v = \frac{p(1 - p)}{n} \tag{24}$$

$$w = (2p(y) - 1)^2 q(\hat{y}) + (2p(\hat{y}) - 1)^2 q(y) + 4q(y)q(\hat{y}) \tag{25}$$

Finally, the test statistic is

$$\mathrm{PT} = \frac{Accurancy - p}{\sqrt{v - w}} \sim N(0, 1) \tag{26}$$

Ⅲ. Data and Empirical Design

1. Data

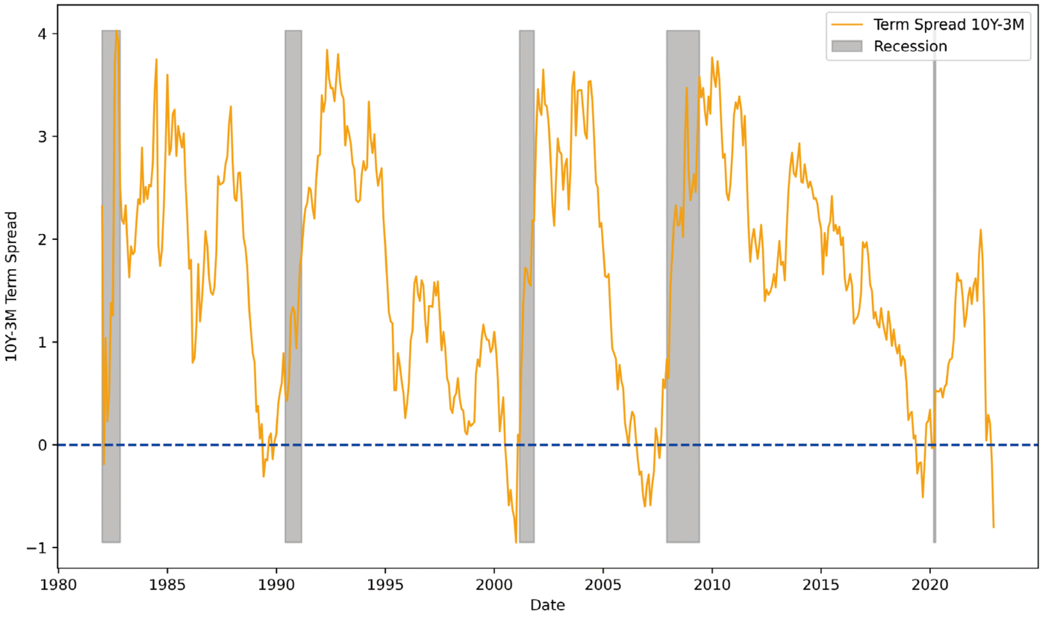
The Recession Indicator. The economic recession indicator is an observable variable from the National Bureau of Economic Research (NBER) Business Cycle Dating Committee's records concerning the timing of the U.S. business cycle. These records consist of alternating peak and trough dates in economic activity, denoting expansion and contraction. During recession periods, the recession indicator is set as 1, while during expansion periods, it is set as 0. Subsequently, various explanatory variables are employed to establish a binary classification model.

Predictor Variables. Researchers have assessed various leading indicators of the U.S. business cycle, particularly those associated with recession periods. In this study, following the analysis of Vrontos et al. (2021), a total of 52 leading indicators obtained from the Federal Reserve Bank of St. Louis' FRED database are used. Detailed information is provided in Table B.6 in the Appendix.

As shown in Table B.6, variables with different frequencies are used in this study, including daily[13], weekly and monthly data. Besides, some quarterly variables, such as real gross domestic product (GDP), real gross domestic income (GDI), and corporate profits, are transformed into monthly frequencies using natural cubic spline interpolation. This also follows the study of Vrontos et al. (2021). Among the 52 variables, the most important one is the first variable in Table A.1, that is, the term spread of 10-year Tbill and 3-month Tbill, and this variable is used to build the benchmark model in the empirical analysis. Besides, it should be noted that there exists collinearity among the 52 variables. The correlation matrix is shown in Table B.7.

Figure 2 intuitively illustrates the relationship between the 10y-3m spread and recession. The line indicates the term spread with a 12-month lag, and the shaded part shows when the recession happened. It can be seen that when the 10y-3m term spread is smaller than zero, the recession may happen 12 months later. It is consistent with the study of Bauer and Mertens (2018), *"A simple rule of thumb that predicts a recession within two years when the term spread is negative had correctly signaled all nine recessions since 1955 and had only one false positive, in the mid-1960s, when an inversion was followed by an economic slowdown but not an official recession. The delay between the term spread turning negative, and the beginning of a recession has ranged between 6 and 24 months."*

Figure 2: The relationship between 10y-3m term spread and recession



In this figure, the line indicates the term spread with a 12-month lag, and the shaded part shows when the recession happened.

This study analyses three different forecasting windows, short-term, medium-term, and long-term, due to the fluctuation in the effectiveness of various forecasting variables when predicting economic activity over different forecast horizons. These three prediction horizons are defined as follows:

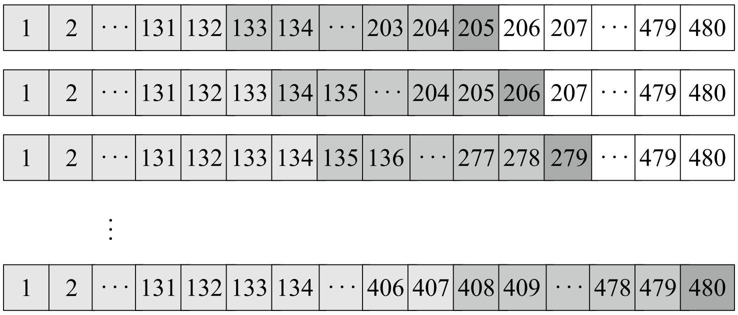
- Short-term: Predictors lagged by 3, 6, and 12 months are used for a 3-month ahead prediction.
- Mid-Term: Predictors lagged by 6 and 12 months are used for a 6-month ahead prediction.
- Long-term: Predictors lagged by 12 months are used for a 12-month ahead prediction.

2. Empirical Design

The analyzed period spans from January 1983 to December 2022 (480 months) and is divided into training, validating, and testing sets. Following Vrontos et al. (2021), an expanding-rolling splitting schema is used. The training data set is initially set as the first 132 months (11 years, from January 1983 to December 1994), following an expanding schema; the validation period is a rolling window of 72 months. The testing data set, which has the same start point as Vrontos et al. (2021), is from January 2000 to December 2022 (264 months). The training data is used to establish the model, while the valid data is used for finding optimal hyperparameters. The testing data is employed to evaluate the forecasting performance of each model.

To provide a more straightforward explanation, a diagram is drawn below. In Figure 3, each row represents a data split, in which white rectangle denotes data not used, the light grey represents the training set, the grey represents the validating set, and the dark grey represents the testing set that only contains one piece of data at a time. By comparing different rows, it can be easily seen that the training dataset is expanding while the validating dataset is rolling. For example, in the first row, the first 132 data are used to train the model, and the later 72 data with an index from 133 to 204 are used for validation and tuning hyperparameters. In the second row, the test dataset is expanded to 133, while the validation dataset keeps the same length of 72 but rolls to the right, with an index from 134 to 205. By repeating this expanding-rolling process for 275 iterations, the predicted values with an index from 205 to 480 can be obtained respectively.

Figure 3: Schematic diagram of the expanding-rolling data splitting schema



This figure illustrates the expanding-rolling data splitting schema. The numbering of the rectangles represents the time, *t*. Each rectangle contains data of y_t and X_t, where X_t includes the lagged value of variables processed after the MIDAS process. Each line represents a cycle of a prediction.

The benefit of this method is that it consistently preserves the complete historical information within the training set using a recursive approach. Additionally, the window size gradually expands to include the most recent observation. In line with the method of Gu et al. (2020), there is no employment of cross-validation to uphold the chronological order of the data for prediction.

As mentioned in the methodology part, the overall predictive performance of the analysis heavily depends on the selection of hyperparameters. For example, if the penalty term in lasso regression or support vector machines is too strong, it will ignore some features and lead to underfitting. Conversely, if it is too weak, it will identify noise as features and lead to overfitting. Therefore, the best out-of-sample prediction can only be achieved by choosing the optimal hyperparameters. To find out the best hyperparameters, in the training process, various models with different sets of hyperparameters are trained using the training dataset. These models are then evaluated using a separate validation dataset to measure their accuracy. This evaluation aims to identify the model with the optimal performance on the validation dataset. This selected model is considered the final model, which is expected to generalize well to testing data.

Ⅳ. Empirical Result

1. Model performance in short-term

For empirical analysis, the Python programming environment (Van Rossum et al., 1995), the Statsmodels library (Seabold and Perktold, 2010), and the Scikit-learn library (Pedregosa et al., 2011) are used. In this section, the empirical results are presented. Table 3 shows the out-of-sample performance of different models for the short-term horizon (3 months ahead).

Table 3: Performance Evaluation - Short term

		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
		Accuracy	AUC	Precision	Sensitivity	Specificity	Balanced	F1	Kappa	P-T	p-value
	Logit and Probit models										
1.	U-MIDAS-Logit-YC	0.884	0.736	0.467	0.467	0.935	0.701	0.467	0.402	6.684	0.000
2.	U-MIDAS-Probit-YC	0.880	0.722	0.448	0.433	0.935	0.684	0.441	0.374	6.222	0.000
3.	MIDAS-Logit-YC	0.870	0.772	0.417	0.500	0.915	0.707	0.455	0.381	6.378	0.000
4.	MIDAS-Probit-YC	0.866	0.774	0.410	0.533	0.907	0.720	0.464	0.389	6.541	0.000
	Penalized logit models										
5.	U-MIDAS-Lasso	0.938	0.965	0.724	0.700	0.967	0.834	0.712	0.677	11.276	0.000
6.	U-MIDAS-Ridge	0.924	0.939	0.610	0.833	0.935	0.884	0.704	0.662	11.191	0.000
7.	U-MIDAS-Elastic Net	0.949	0.964	0.735	0.833	0.963	0.898	0.781	0.753	12.559	0.000
	Naive Bayes Models										
8.	U-MIDAS-LDA	0.949	0.925	0.735	0.833	0.963	0.898	0.781	0.753	12.559	0.000
9.	U-MIDAS-Gaussian NB	0.888	0.880	0.485	0.533	0.931	0.732	0.508	0.445	7.412	0.000
10.	U-MIDAS-Bernoulli NB	0.721	0.683	0.159	0.367	0.764	0.565	0.222	0.083	1.566	0.059
	Tree-based Models										
11.	U-MIDAS-Tree	0.899	0.853	0.529	0.600	0.935	0.767	0.562	0.505	8.432	0.000
12.	U-MIDAS-Random Forest	0.913	0.944	0.600	0.600	0.951	0.776	0.600	0.551	9.174	0.000
13.	U-MIDAS-XGBoost	0.964	0.972	0.917	0.733	0.992	0.863	0.815	0.795	13.333	0.000
	Other Models										
14.	U-MIDAS-SVM	0.942	0.952	0.733	0.733	0.967	0.850	0.733	0.701	11.664	0.000
15.	U-MIDAS-KNN	0.946	0.940	0.692	0.900	0.951	0.926	0.783	0.752	12.659	0.000
16.	U-MIDAS-Neural Network	0.964	0.966	0.833	0.833	0.980	0.907	0.833	0.813	13.531	0.000

This table reports the performance evaluation measures of the Short-term horizon (3-months-ahead) forecasts obtained by (U)-MIDAS-Logit/Probit models and several U-MIDAS-ML models for the out-of-sample period (Jan. 2000 – Dec. 2022).

To begin with, the U-MIDAS-Logit-YC in the first row is the U-MIDAS-Logit model mentioned in Equation 1 with only term spread (10y-3m) of yield curve daily data as the explanatory variable. Similarly, the U-MIDAS-Probit-YC model in the second row is also used as another benchmark. Moreover, the MIDAS-Logit/Probit-YC models are also used as another two benchmarks. In the first two rows, the accuracy of these two U-MIDAS benchmark models is about 0.88, as shown in column 1. Additionally, both of these models have achieved an AUC of over 72% (shown in column 2), indicating their overall performance. However, the Precision (around 0.467) in column 3 indicates that out of the cases predicted as positive, only 46.7% are true positives. The Sensitivity in column 4 tells us that the model can correctly identify 47% of the actual positive cases. The Sensitivity and the Precision are seemingly less than ideal, but consistent with the result of Vrontos et al. (2021)[14] Meanwhile, Specificity in column 5 indicates that the model can correctly identify 93.5% of the actual negative cases. The balanced accuracy of these two benchmark models shown in column 6 of about 0.7 considers both Sensitivity and Specificity, providing a more robust accuracy measure suitable for imbalanced datasets. The F1 score of about 0.476 in column 7 also combines Precision and Sensitivity, providing a single metric that balances both aspects. The kappa value[15], approximately 0.4 in column 8, quantifies the level of agreement between the model's predictions and the actual classes while also considering the chance agreement. Finally, the

P-T statistic in column 10 and the p-value of the hypothesis test in column 11 show that the null hypothesis is significantly rejected.

Besides, the MIDAS-Logit-YC and MIDAS-Probit-YC models in the third and fourth rows are constructed using the MIDAS process instead of U-MIDAS. Compared with U-MIDAS, the results are similar, but with slightly lower accuracies and slightly higher AUCs. So it is hard to say which method, MIDAS or U-MIDAS is better. Due to the U-MIDAS model being able to make full use of high-frequency data without any information loss, as well as saving the runtime of the computer program, we use the U-MIDAS method as a data pre-preprocess for machine learning models.

When it comes to machine learning methods, almost all of them except for the U-MIDAS-Bernoulli NB model show higher predictability than the benchmarks. First, for the penalized logit models in the 5th to 7th row, all three models beat the benchmark models in all metrics. Each of these three models has an Accuracy over 92%, better than 88% of benchmark Accuracy. The AUCs of these three models are also higher than 93%, much better than 73% of benchmarks. As mentioned above, the Precision and Sensitivity of the benchmark models are only over 0.4, while the two metrics of the U-MIDAS-Lasso model and the U-MIDAS-Elastic Net model are around 0.7, which is a significant improvement over the benchmark. Other metrics also beat the benchmarks similarly with higher values. The U-MIDAS-Ridge model is not as good as the other two penalized models on Precision, with only around 61 percent, but it also beats the benchmarks.

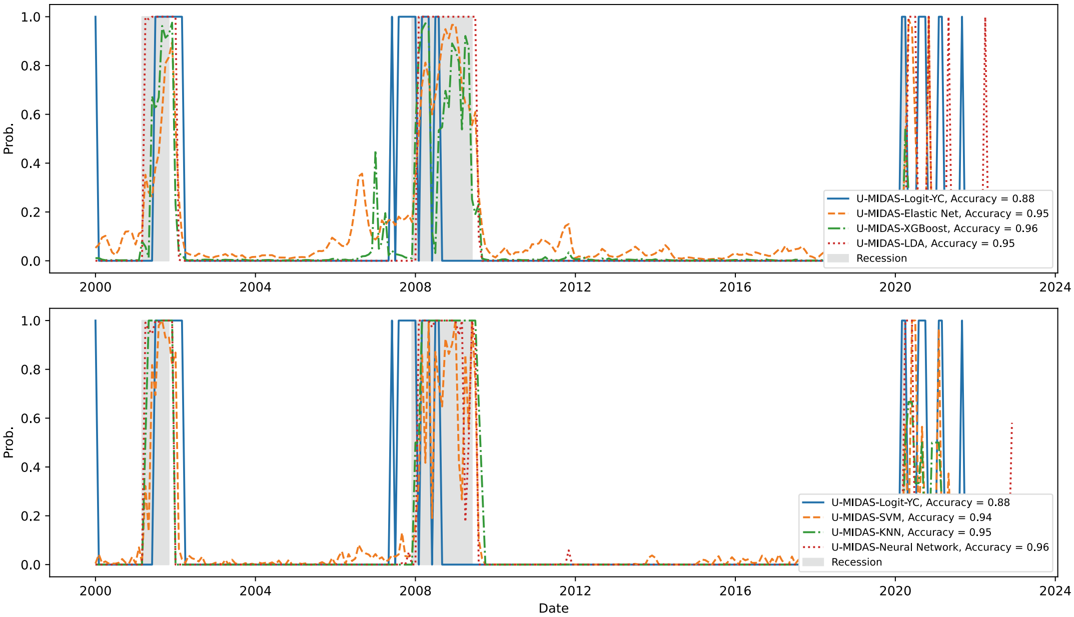
The 8th-10th rows show the result of U-MIDAS-Naive Bayes Models. The first two models show higher performance on all metrics than the benchmarks. However, the U-MIDAS-Bernoulli NB model does not beat the benchmarks. As for tree-based models shown in the 11th-13th rows, the U-MIDAS-XGBoost algorithm performs best, with the highest AUC. Then U-MIDAS-Random Forest also improved the performances of U-MIDAS-Tree. They also beat the benchmarks on all metrics. And finally, in the last three rows, all these models beat the benchmarks on all metrics again.

Overall, considering all these metrics in Table 3, almost all machine learning models consistently provide accurate predictions in identifying recessions and non-recession periods. They show predictability in capturing the actual occurrence of recessions while minimizing false positives. Among them, the U-MIDAS-XGBoost has the highest Accuracy of 0.964 as well as the highest AUC of 0.972.

To visualize their predictive capabilities, Figure 4 illustrates the probability of a recession occurring as predicted by each model. In this figure, the solid line represents one of the benchmarks, the U-MIDAS-Logit-YC model, while other various dashed lines represent several well-performing machine learning algorithms mentioned above. But in order to keep the figure from being too cluttered, only some of the models are shown here in two subfigures. Besides, the shaded regions correspond to three historical recession periods: the “dot-com” bubble recession, the Great Recession, and the COVID-19 recession. Upon observing the figure, it becomes evident that their strong performance across various evaluation metrics and their ability to capture both historical and real-time recession periods make them promising models for predicting future economic downturns. It can be seen that during the recession period, all these methods show high probability.

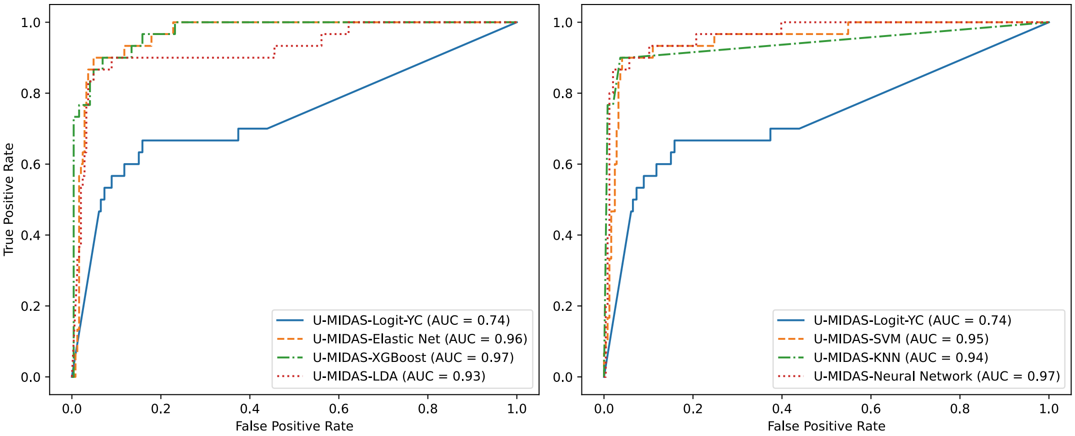
Figure 5 illustrates the ROC curves of each model, showing the trade-off between the True Positive Rate（Sensitivity）and the False Positive Rate （1-Specificity） at various classification thresholds. A higher ROC curve closer to the top-left corner indicates better performance, as it signifies a higher true positive rate and a lower false positive rate. It can be seen that even though the ROC curves of the U-MIDAS-ML models are much on the top-left side of the benchmark and leave a larger area under the curve, meaning that U-MIDAS-ML models have better average performance over the entire distribution of positive and negative samples. In other words, they have a low misjudgment rate and a high correct discrimination rate even under different classification thresholds.

Figure 4: Out-of-Sample Prediction Result in the short-term



This figure illustrates the model prediction result of the short-term horizon (3-months-ahead) forecasts obtained by MIDAS- Logit/Probit models and several MIDAS-ML models for the out-of-sample period (Jan. 2000 – Dec. 2022). Different lines denote the probability given by the models, and the shaded area represents the period during which the recession actually occurred. In order to keep the figure from being too cluttered, only some of the models are shown here in two subfigures.

Figure 5: ROC curves in the short-term



This figure shows the ROC curves for several models. The solid line illustrates the benchmark model and other dash or dot lines illustrate different MIDAS-ML models. The closer the curve is to the upper left corner, the better the model performs. For details of ROC, please see Section 2.4.

2. Model performance in mid-term

Table 4 represents the predictive results of various models for the mid-term economic recession. Compared to the short-term recession predictions, the mid-term recession results are similar. The first four rows in the table indicate that the（U-）MIDAS-Probit/Logit model has an accuracy of more than 85 percent and an AUC of over 75, and exhibits excellent significance.

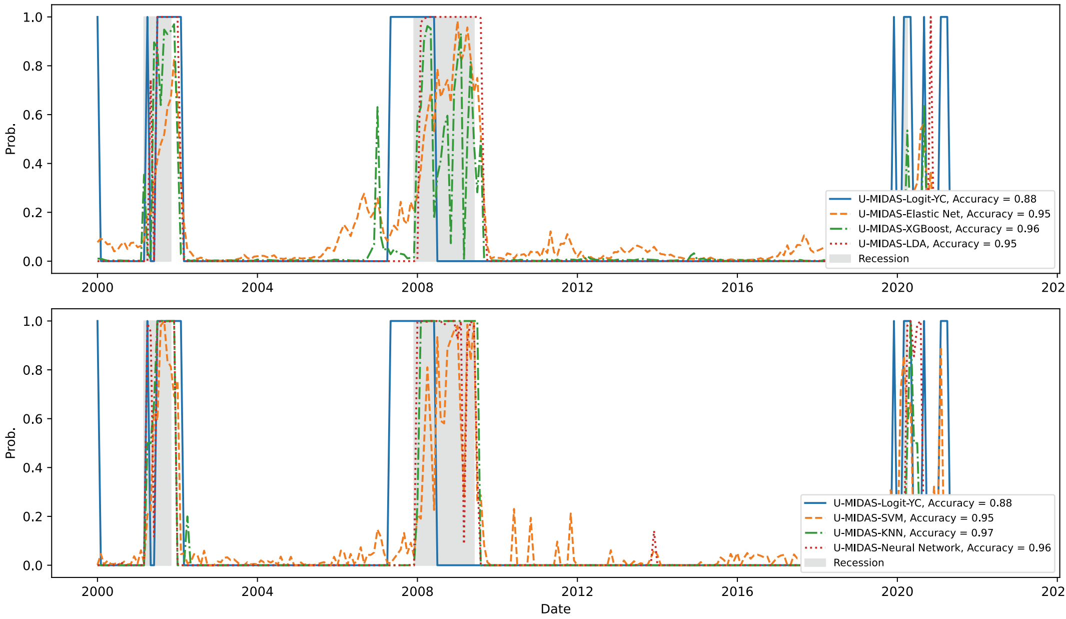
However, compared to the other machine learning models, almost every MIDAS-ML model, except for the U-MIDAS-Bernoulli NB model, beats nearly all four benchmarks on every metric. It is noticeable that the U-MIDAS-KNN model shows the highest accuracy of about 96.7 percent. The U-MIDAS-SVM and U- MIDAS-Lasso model both have the highest AUC, around 95.4 percent. The predicted probability and ROC curve are shown in Figure 6. The ROC curves are shown in Figure 7.

Table 4: Performance Evaluation - Mid term

		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
		Accuracy	AUC	Precision	Sensitivity	Specificity	Balanced	F1	Kappa	P-T	p-value
	Logit and Probit models										
1.	U-MIDAS-Logit-YC	0.884	0.771	0.469	0.500	0.931	0.715	0.484	0.419	6.972	0.000
2.	U-MIDAS-Probit-YC	0.888	0.760	0.484	0.500	0.935	0.717	0.492	0.429	7.136	0.000
3.	MIDAS-Logit-YC	0.859	0.764	0.378	0.467	0.907	0.687	0.418	0.338	5.674	0.000
4.	MIDAS-Probit-YC	0.855	0.768	0.361	0.433	0.907	0.670	0.394	0.312	5.227	0.000
	Penalized logit models										
5.	U-MIDAS-Lasso	0.935	0.954	0.714	0.667	0.967	0.817	0.690	0.653	10.881	0.000
6.	U-MIDAS-Ridge	0.949	0.926	0.722	0.867	0.959	0.913	0.788	0.759	12.706	0.000
7.	U-MIDAS-Elastic Net	0.946	0.962	0.759	0.733	0.972	0.852	0.746	0.715	11.908	0.000
	Naive Bayes Models										
8.	U-MIDAS-LDA	0.953	0.907	0.793	0.767	0.976	0.871	0.780	0.753	12.540	0.000
9.	U-MIDAS-Gaussian NB	0.877	0.867	0.438	0.467	0.927	0.697	0.452	0.382	6.367	0.000
10.	U-MIDAS-Bernoulli NB	0.732	0.724	0.203	0.500	0.760	0.630	0.288	0.158	3.042	0.001
	Tree-based Models										
11.	U-MIDAS-Tree	0.899	0.812	0.526	0.667	0.927	0.797	0.588	0.531	8.923	0.000
12.	U-MIDAS-Random Forest	0.906	0.940	0.562	0.600	0.943	0.772	0.581	0.528	8.788	0.000
13.	U-MIDAS-XGBoost	0.957	0.940	0.821	0.767	0.980	0.873	0.793	0.769	12.806	0.000
	Other Models										
14.	U-MIDAS-SVM	0.953	0.954	0.815	0.733	0.980	0.857	0.772	0.746	12.433	0.000
15.	U-MIDAS-KNN	0.967	0.941	0.839	0.867	0.980	0.923	0.852	0.834	13.885	0.000
16.	U-MIDAS-Neural Network	0.960	0.933	0.806	0.833	0.976	0.904	0.820	0.797	13.272	0.000

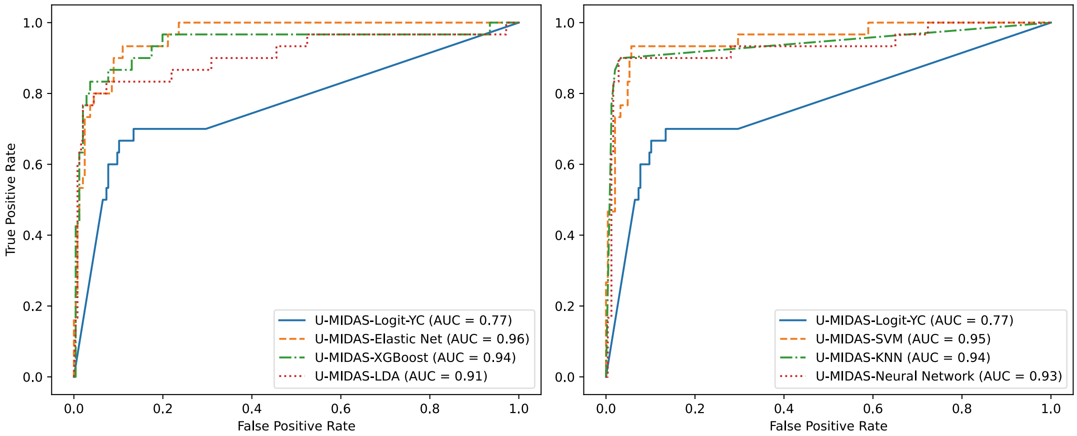
This table reports the performance evaluation measures of the Mid-term horizon（6-months-ahead）forecasts obtained by（U）-MIDAS- Logit/Probit models and several U-MIDAS-ML models for the out-of-sample period（Jan. 2000 – Dec. 2022）.

Figure 6: Out-of-Sample Prediction Result in the mid-term



This figure illustrates the model prediction result of the mid-term horizon (6-months-ahead) forecasts obtained by MIDAS-Logit/Probit models and several MIDAS-ML models for the out-of-sample period (Jan. 2000 – Dec. 2022). Different lines denote the probability given by the models, and the shaded area represents the period during which the recession actually occurred.

Figure 7: ROC curves in the mid-term



This figure shows the ROC curves for several models. The solid line illustrates the benchmark model and other dash or dot lines illustrate different MIDAS-ML models.

3. Model performance in long-term

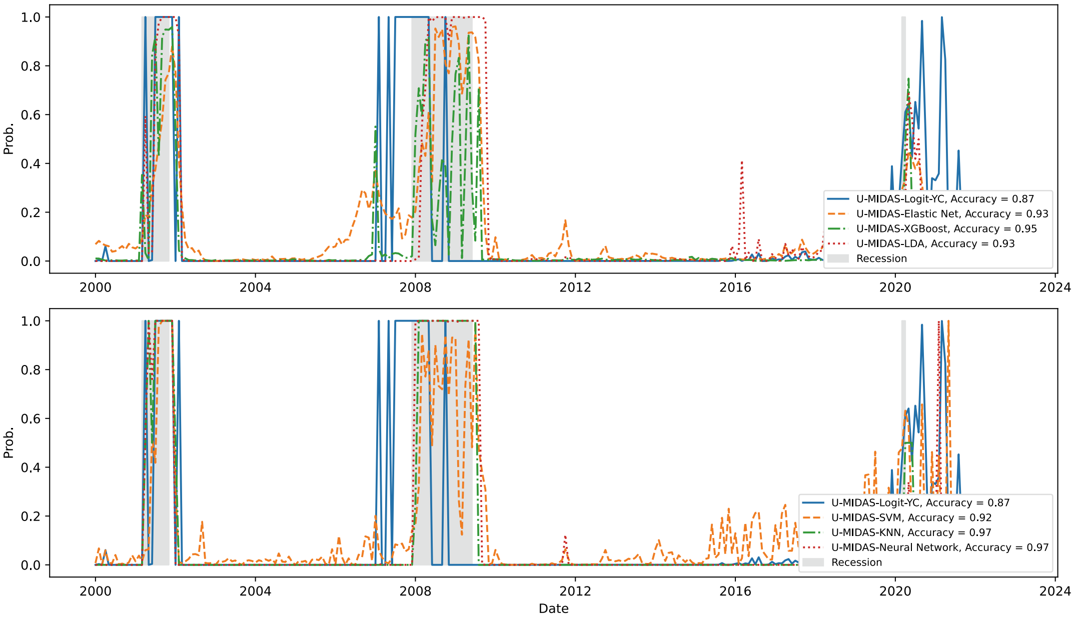
As for the performance of long-term prediction, as shown in Table 4, compared with the other two horizons, all metrics of almost all models, including four benchmarks, are slightly lower than those of the short-term and mid-term. But the conclusion in the long-term is consistent with the mid-term as well as the short-term, that is, the vast majority of U-MIDAS-ML models (except for the U-MIDAS-Bernoulli NB model) show better prediction performance than the four benchmarks on almost all of the metrics evaluated. The prediction results and ROC curves are also shown in Figure 8 and 9.

Table 5: Performance Evaluation - Long term

		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
		Accuracy	AUC	Precision	Sensitivity	Specificity	Balanced	F1	Kappa	P-T	p-value
	Logit and Probit models										
1.	U-MIDAS-Logit-YC	0.866	0.631	0.405	0.500	0.911	0.705	0.448	0.372	6.242	0.000
2.	U-MIDAS-Probit-YC	0.866	0.667	0.410	0.533	0.907	0.720	0.464	0.389	6.541	0.000
3.	MIDAS-Logit-YC	0.873	0.823	0.432	0.533	0.915	0.724	0.478	0.406	6.811	0.000
4.	MIDAS-Probit-YC	0.870	0.820	0.417	0.500	0.915	0.707	0.455	0.381	6.378	0.000
	Penalized logit models										
5.	U-MIDAS-Lasso	0.924	0.946	0.696	0.533	0.972	0.752	0.604	0.562	9.463	0.000
6.	U-MIDAS-Ridge	0.938	0.929	0.686	0.800	0.955	0.878	0.738	0.704	11.758	0.000
7.	U-MIDAS-Elastic Net	0.931	0.962	0.667	0.733	0.955	0.844	0.698	0.660	10.995	0.000
	Naive Bayes Models										
8.	U-MIDAS-LDA	0.928	0.902	0.647	0.733	0.951	0.842	0.688	0.647	10.790	0.000
9.	U-MIDAS-Gaussian NB	0.877	0.848	0.444	0.533	0.919	0.726	0.485	0.416	6.953	0.000
10.	U-MIDAS-Bernoulli NB	0.761	0.773	0.263	0.667	0.772	0.720	0.377	0.262	5.091	0.000
	Tree-based Models										
11.	U-MIDAS-Tree	0.873	0.821	0.447	0.700	0.894	0.797	0.545	0.476	8.191	0.000
12.	U-MIDAS-Random Forest	0.906	0.944	0.559	0.633	0.939	0.786	0.594	0.541	9.022	0.000
13.	U-MIDAS-XGBoost	0.946	0.952	0.800	0.667	0.980	0.823	0.727	0.697	11.666	0.000
	Other Models										
14.	U-MIDAS-SVM	0.924	0.933	0.629	0.733	0.947	0.840	0.677	0.634	10.594	0.000
15.	U-MIDAS-KNN	0.967	0.944	0.839	0.867	0.980	0.923	0.852	0.834	13.885	0.000
16.	U-MIDAS-Neural Network	0.967	0.966	0.839	0.867	0.980	0.923	0.852	0.834	13.885	0.000

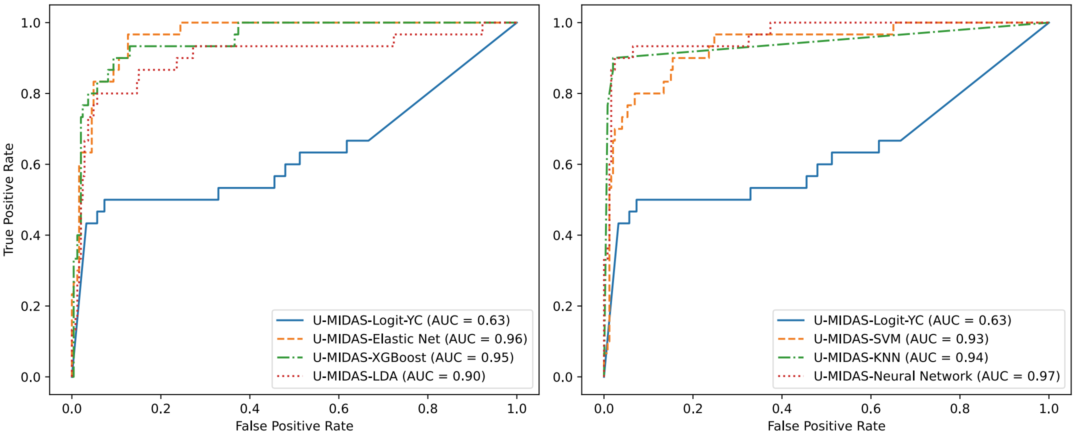
This table reports the performance evaluation measures of the Long-term horizon (12-months-ahead) forecasts obtained by (U)-MIDAS- Logit/Probit models and several U-MIDAS-ML models for the out-of-sample period (Jan. 2000 – Dec. 2022).

Figure 8: Out-of-Sample Prediction Result in the long-term



This figure illustrates the model prediction result of the long-term horizon (12-months-ahead) forecasts obtained by MIDAS-Logit/Probit models and several MIDAS-ML models for the out-of-sample period (Jan. 2000 – Dec. 2022). Different lines denote the probability given by the models, and the shaded area represents the period during which the recession actually occurred.

Figure 9: ROC curves in the long-term



This figure shows the ROC curves for several models. The solid line illustrates the benchmark model and other dash or dot lines illustrate different MIDAS-ML models.

Ⅴ. Conclusion

The conclusions of this study are as follows: First and most importantly, we apply the MIDAS and U-MIDAS methodology to the Logit and Probit benchmark models with ten-year and three-month spreads and use more high-frequency information in the training and testing of the models. Regardless of the prediction horizon, the model's prediction accuracy is over 86 percent. We also also use the U-MIDAS method in machine learning methods. This is a complement to the study of Vrontos et al. (2021), which fills the gap by applying the method in benchmark models and provides new ideas and methods for prediction research in related fields.

Second, this paper demonstrates that the yield curve, particularly the ten-year and three-month spreads, is predictive of recessions in all three horizons: long-, medium-, and short-term. These findings are consistent with other existing literature, like (Choi et al., 2023). Therefore, using the yield curve as a predictor variable in the benchmark model is a reasonable choice.

Moreover, adding other variables to the benchmark model can improve the performance of the model. However, due to the high degree of multicollinearity in economic variables, the "kitchen sink approach" of adding variables without selection can lead to model misspecification. Therefore, the introduction of machine learning algorithms, especially regularization algorithms with the ability to screen variables, can solve the multicollinearity problem and improve the model performance.

Fourth, this paper demonstrates that almost all machine learning models outperform traditional economics models in predicting recession in all terms. This result is also consistent with the study of Vrontos et al. (2021). The analysis is carried out using an expanding-rolling window approach, which incorporates various model performance measures including hypothesis testing to ensure statistical significance and mitigate the effects of random chance.

In conclusion, this paper combines mixed-frequency data with machine learning algorithms to provide a new approach to economic forecasting and finds better performance than traditional models. This provides new ideas for future research directions in economic forecasting.

However, this study currently has several limitations. For example, the recession analysis in this study is limited to the United States and has yet to be researched in other countries. In addition, this study only contributes to the prediction accuracy by using the yield curve as the primary variable supplemented by other economic variables, without investigating the relative importance of these variables. Furthermore, due to the general principle that more extensive data sets yield better results in machine learning, only 52 variables might be considered insufficient.

For future work, we have the following suggestions: First, machine learning algorithms require a large amount of data and can handle high-dimensional data. As a result, we can explore more factors that may influence recessions in the United States by including more macroeconomic variables to explore their relationship with predictive models. For example, in the Davig and Hall (2019) research, 135 variables in the FRED-MD dataset (McCracken and Ng, 2016) are used. Moreover, we could even consider incorporating non-traditional unstructured and qualitative input variables, such as using large language generation models to perform sentiment analysis for economic forecasting, like Consoli et al. (2022), Liu et al. (2021).

Second, the machine learning algorithms used in this study are still simple and can be further optimized and improved to increase the Accuracy and Precision of the predictive models. For example, the use of more complex models such as Long Short Term Memory (LSTM) neural networks, which have been shown to have advantages in handling time series data[16]. In addition, the LSTM model can capture both long-term trends and short-term fluctuations in time series data, making them a potential candidate for combination with the MIDAS approach.

Third, validating the universality of

recession prediction patterns by using data from more countries would be beneficial. Besides, it is also worth considering using methods such as Shapley Additive exPlanations (SHAP) value or feature importance score using XGBoost or Random forest to explore the importance of predictive variables.

Notes

1) There are also other views on the causes of the Great Recession. For more details, please refer to Jagan-nathan et al. (2013).

2) The testing period of this paper and Vrontos et al. (2021) is not quite the same. The testing dataset of this paper ranges from January 2000 to December 2022, while the study of Vrontos et al. (2021) ranges from January 2000 to July 2019. In comparison, this paper includes the economic recession in 2020 caused by COVID-19, which makes the prediction more difficult. In addition, an attempt was made to remove the prediction results from July 2019, and it was found that the model performance metrics were improved.

3) More details about model performance metrics are explained later in Section 2.4

4) In that study, the metric ROC_S is used, which equals $2AUC-1$. The original result is $ROC\_S=$ 0.770, to compare with my result, I solved that AUC = 0.885

5) In this study, I only show the result of U-MIDAS-ML models but not MIDAS-ML models because the result of U-MIDAS-ML and MIDAS-ML is quite similar

6) This step is to constrain the number of parameters, in other words, to reduce the dimension of the variables.

7) For daily and monthly data, m is set to 28, since there are at least 28 days in a month; Similarly, for weekly and monthly data, m = 4.

8) For daily data, the number of variables increased to 28 times by adding 27 daily lagged variables to the monthly aligned data. And for weekly data, the dimension increased to 4 times.

9) Here h denotes the prediction horizon. In this study, three horizons are used: Short, Mid, and Long term. For the Short-term analysis, X is lagged by 3, 6, and 12 months. For the Mid-term analysis, X is lagged by 6 and 12 months. For Long-term analysis, X is lagged by 12 months only.

10) The MIDAS method was also utilized in this study. The results obtained with MIDAS were found to be quite similar to those obtained with U-MIDAS. Therefore, only the results of U-MIDAS-ML models are presented in the results section. It is important to note that the MIDAS method requires extensive computational time due to the large number of hyperparameters that need to be explored. These hyperparameters include those within each machine learning algorithm as well as the weights of the MIDAS parameter restricting step. Additionally, the iterative nature of the rolling-window analysis further contributes to the runtime. Consequently, we recommend using U-MIDAS-ML models.

11) More details can be found in (Liu and Moench, 2016)

12) Specifically, $P_e = p(y)p(\hat{y}) + (1-p(y))(1-p(\hat{y}))$.

13) Certain variables, such as Credit Spread (Code 21 in Table B.6), are available daily. Nevertheless, the monthly frequency is used instead because there was a lack of daily data in earlier years, which led to an incomplete dataset.

14) In the short-term study of Vrontos et al. (2021), the Precision and Sensitivity of Logit-YC model is only 0.231 and 0.400 respectively.

15) It can be regarded as a modified accuracy that eliminates chance consistency. For further explanation, please refer to the description of equation 20 in Section 2.4.

16) See Siami-Namini et al. (2018)

References

Andreou Elena, Ghysels Eric, and Kourtellos Andros (2010), "Regression models with mixed sampling frequencies." *Journal of Econometrics*, 158(2), pp.246-261.

Audrino Francesco, Kostrov Alexander, and Ortega Juan-Pablo (2019), "Predicting us bank failures with midas logit models." *Journal of Financial and Quantitative Analysis*, 54(6), pp.2575-2603.

Babii Andrii, Ghysels Eric, and Striaukas Jonas (2022), "Machine learning time series regressions with an application to nowcasting." *Journal of Business & Economic Statistics*, 40(3), pp.1094-1106.

Bauer Michael D and Mertens Thomas M (2018),"Economic forecasts with the yield curve." *FRBSF Economic Letter*, 7, 8-07.

Berge Travis J (2015), "Predicting recessions with leading indicators: Model averaging and selection over the business cycle." *Journal of Forecasting*, 34 (6), pp.455-471.

Breiman Leo (2001), "Random forests." *Machine learning*, 45, pp.5-32.

Chen Tianqi and Guestrin Carlos (2016), "Xgboost: A scalable tree boosting system." In "Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining," pp.785-794.

Choi Jaehyuk, Ge Desheng, Kang Kyu Ho, and Sohn Sungbin (2023), "Yield spread selection in predicting recession probabilities." *Journal of Forecasting.*

Consoli Sergio, Barbaglia Luca, and Manzan Sebastiano (2022), "Fine-grained, aspect-based sentiment analysis on economic and financial lexicon." *Knowledge-Based Systems*, 247, 108781.

Cortes Corinna and Vapnik Vladimir (1995), "Support-vector networks." *Machine learning*, 20, pp.273-297.

Coulombe Philippe Goulet, Marcellino Massimiliano, and Stevanovic̀ Dalibor (2021), "Can machine learning catch the covid-19 recession?" *National Institute Economic Review*, 256, pp.71-109.

Davig Troy and Hall Aaron Smalter (2019), "Recession forecasting using bayesian classification." *International Journal of Forecasting*, 35(3), pp.848-867.

Döpke Jörg, Fritsche Ulrich, and Pierdzioch Christian (2017), "Predicting recessions with boosted regression trees." *International Journal of Forecasting*, 33(4), pp.745-759.

Drechsel Katja and Scheufele Rolf (2012), "The performance of short-term forecasts of the german economy before and during the 2008/2009 recession." *International Journal of Forecasting*, 28(2), pp.428-445.

Estrella Arturo and Mishkin Frederic S (1996), "The yield curve as a predictor of us recessions." *Current issues in economics and finance*, 2(7).

Estrella Arturo and Mishkin Frederic S (1998), "Predicting us recessions: Financial variables as leading indicators." *Review of Economics and Statistics*, 80(1), pp.45-61.

Estrella Arturo, Rodrigues Anthony P, and Schich Sebastian (2003), "How stable is the predictive power of the yield curve? evidence from germany and the united states." *Review of Economics and Statistics*, 85(3), pp.629-644.

Galvão Ana Beatriz and Owyang Michael (2022), "Forecasting low-frequency macroeconomic events with high-frequency data." *Journal of Applied Econometrics*, 37(7), pp.1314-1333.

Ghysels Eric, Santa-Clara Pedro, and Valkanov Rossen (2004), "The midas touch: Mixed data sampling regression models." *UCLA: Finance.*

Ghysels Eric, Sinko Arthur, and Valkanov Rossen (2007), "Midas regressions: Further results and new directions." *Econometric reviews*, 26(1), pp.53-90.

Gogas Periklis and Papadimitriou Theophilos (2021), "Machine learning in economics and finance." *Computational Economics*, 57, pp.1-4.

Gogas Periklis, Papadimitriou Theophilos, Matthaiou Maria, and Chrysanthidou Efthymia (2015), "Yield curve and recession forecasting in a machine learning framework." *Computational Economics*, 45, pp.635-645.

Coulombe Philippe Goulet, Leroux Maxime, Stevanovic Dalibor, and Surprenant Stéphane (2022), "How is machine learning useful for macroeconomic forecasting?" Journal of Applied Econometrics, 37(5), pp.920-964.

Gu Shihao, Kelly Bryan, and Xiu Dacheng (2020), "Empirical asset pricing via machine learning." *The Review of Financial Studies*, 33(5), pp.2223-2273.

Hoerl Arthur E and Kennard Robert W (1970), "Ridge regression: Biased estimation for nonorthogonal problems." *Technometrics*, 12(1), pp.55-67.

Jagannathan Ravi, Kapoor Mudit, and Schaumburg Ernst (2013), "Causes of the great recession of 2007-2009: The financial crisis was the symptom not the disease!" *Journal of Financial Intermediation*, 22(1), pp.4-29.

James Gareth, Witten Daniela, Hastie Trevor, Tibshirani Robert, and Taylor Jonathan (2023), *An introduction to statistical learning: With applications in python.* Springer Nature.

Jiang Cuixia, Nie Yubing, and Xu Qifa (2023), "A midas multinomial logit model with applications for bond ratings." *Global Finance Journal*, 57, 100867.

Liu Weiling and Moench Emanuel (2016), "What predicts us recessions?" *International Journal of Forecasting*, 32(4), pp.1138-1150.

Liu Zhuang, Huang Degen, Huang Kaiyu, Li Zhuang, and Zhao Jun (2021), "Finbert: A pre-trained financial language representation model for financial text mining." In "Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence," pp.4513-4519.

Maehashi Kohei and Shintani Mototsugu (2020), "Macroeconomic forecasting using factor models and machine learning: an application to japan." *Journal of the Japanese and International Economies*, 58, 101104.

McCracken Michael W and Ng Serena (2016), "Fred-md: A monthly database for macroeconomic research." *Journal of Business & Economic Statistics*, 34(4), pp.574-589.

Moneta Fabio (2005), "Does the yield spread predict recessions in the euro area?" *International Finance*, 8 (2), pp.263-301.

Nyman Rickard and Ormerod Paul (2017), "Predicting economic recessions using machine learning algorithms." *arXiv preprint arXiv:1701.01428.*

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., and Duchesnay E. (2011), "Scikit-learn: Machine learning in Python." *Journal of Machine Learning Research*, 12, pp.2825-2830.

Pesaran M Hashem and Timmermann Allan (1992), "A simple nonparametric test of predictive performance." *Journal of Business & Economic Statistics*, 10(4), pp.461-465.

Pierdzioch Christian and Gupta Rangan (2019), "Uncertainty and forecasts of us recessions." *Studies in Nonlinear Dynamics & Econometrics*, 24(4), 20180083.

Seabold Skipper and Perktold Josef (2010), "Statsmodels: Econometric and statistical modeling with Python." In "Proceedings of the 9th Python in Science Conference,"

Siami-Namini Sima, Tavakoli Neda, and Namin Akbar Siami (2018), "A comparison of arima and lstm in forecasting time series." In "2018 17th IEEE international conference on machine learning and applications (ICMLA)," pp.1394-1401. IEEE.

Tibshirani Robert (1996), "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1), pp.267-288.

Rossum Guido Van, Drake Fred L, et al. (1995), *Python reference manual*, volume 111. Centrum voor Wiskunde en Informatica Amsterdam.

Vrontos Spyridon D, Galakis John, and Vrontos Ioannis D (2021), "Modeling and predicting us recessions using machine learning techniques." *International Journal of Forecasting*, 37(2), pp.647-671.

Xu Qifa, Zhuo Xingxuan, Jiang Cuixia, and Liu Yezheng (2019), "An artificial neural network for mixed frequency data." *Expert Systems with Applications*, 118, pp.127-139.

Zou Hui and Hastie Trevor (2005), "Regularization and variable selection via the elastic net." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2), pp.301-320.

Appendix A. Details of machine learning algorithms

1. Penalized U-MIDAS Logit Model

The penalized U-MIDAS Logit regression is similar to the standard U-MIDAS Logit regression in terms of the estimation function, but it incorporates a regularization term in the loss function to minimize overfitting. Specifically, in the process of the parameter estimation of β in Penalized Logit Regression, a regularization term $\gamma(\beta)$ is added to the loss function:

$$\min_{\beta} \sum_{t=1}^{T} (-y_t \log(p(X_t)) - (1-y_t)\log(1-p(X_t))) + \lambda\gamma(\beta) \qquad (A.1)$$

where the λ denotes the strength of regularization term $\gamma(\beta)$, which is explained below. To get consistent with the notation of other machine learning algorithms, the equation is transformed here by setting $C = \lambda^{-1}$:

$$\min_{\beta} C\sum_{t=1}^{T} (-y_t \log(p(X_t)) - (1-y_t)\log(1-p(X_t))) + \gamma(\beta) \qquad (A.2)$$

In equation A.2, C represents the inverse of regularization strength. Specifically, smaller C values imply higher regularization, reducing the impact of predictors with limited explanatory power. The hyperparameter- choosing approach will be mentioned in Section 3 below.

The U-MIDAS-LASSO logit model. Lasso regression is an appropriate model for datasets with a high number of features, as its algorithmic design allows for automatic feature selection and management of highly correlated features. The Lasso approach is proposed by Tibshirani (1996). The method is by adding a $L1$ term to the loss function. Specifically, the $\gamma(\beta)$ in Equation A.2 is

$$\gamma(\beta) = \|\beta\|_1 = \sum_{i=1}^{N} |\beta_i|$$

The U-MIDAS-Ridge logit model. The Ridge regression proposed by Hoerl and Kennard (1970), differs from Lasso regression as it only shrinks the coefficients of insignificant features to near zero rather than compressing them to zero. The Ridge regression is considered more stable when multiple significant predictor variables are presented. However, it is not a suitable model for datasets that contain many features but only a few truly relevant features. Similarly, the loss function is also Equation A.2, with $\gamma(\beta) = \frac{1}{2}\|\beta\|_2^2 = \frac{1}{2}\sum_{i=1}^{N}\beta_i^2$

The U-MIDAS-Elastic-Net logit model. Elastic Net, a mixed linear model, is a combination of two penalty techniques, L1 and L2, designed to manage the limitations and improve the advantages of each method. This hybrid technique was proposed by Zou and Hastie (2005). The Elastic Net can be applied to datasets with heteroscedasticity and is effective in the elimination of multicollinearity. The penalty term of the Elastic Net is

$$\gamma(\beta) = \rho\|\beta\|_1 + \frac{1-\rho}{2}\|\beta\|_2^2 = \rho\sum_{i=1}^{N}|\beta_i| + \frac{1-\rho}{2}\sum_{i=1}^{N}\beta_i^2,$$

where ρ is another hyperparameter controlling the strength of L1 regularization versus L2 regularization.

2. The Bayes Models

The Naive Bayes Models. The Naive Bayes is considered a simple but effective supervised model with the “naive” assumption of conditional independence between every pair of features (Davig and Hall (2019)), which leverages Bayes’ theorem to calculate the conditional probability of the observed data belonging to a particular class of observations. The Bayesian theorem for classes can be expressed as:

$$P(y \mid X) = \frac{P(X \mid y)}{P(X)} \qquad \text{(A.3)}$$

where X is the vector including all input variables $[x^{[1]}, x^{[2]}, \ldots, x^{[n]}]$. Using the naive conditional independence assumption that

$$P(x^{[i]} \mid y, x^{[1]}, x^{[2]}, \ldots, x^{[i-1]}, x^{[i+1]}, \ldots, x^{[n]}) = P(x^{[i]} \mid y) \quad \text{(A.4)}$$

For all i, the relationship is

$$P(y \mid x^{[1]}, x^{[2]}, \ldots, x^{[n]}) = \frac{P(y)\prod_{i=1}^{n}P(x^{[i]} \mid y)}{P(x^{[1]}, x^{[2]}, \ldots, x^{[n]})} \qquad \text{(A.5)}$$

The decision rule is:

$$\hat{y} = \arg\max_{y} P(y)\prod_{i=1}^{n}P(x^{[i]} \mid y) \qquad \text{(A.6)}$$

U-MIDAS Linear Discriminant analysis. Linear discriminant analysis (LDA) is a supervised learning technique with the basic idea that, given a set of training samples, try to project the samples onto a line. Meanwhile, the projection points of similar samples are as close as possible, and the centers of the projection points of dissimilar samples are as far away as possible.

LDA can be derived from simple probabilistic models that represent the class conditional distribution of the data $P(X_t \mid y_t = k)$ for each class k, which denotes recession ($k=1$) or not ($k=0$). Predictions can then be obtained by using Bayes’ rule for each training sample $x \in R^d$:

$$P(y_t = k \mid x_t) = \frac{P(x_t \mid y_t = k)P(y_t = k)}{P(x_t)} \qquad \text{(A.7)}$$

and then select the class k which maximizes this posterior probability. More specifically, for LDA, $P(x_t \mid y_t)$ is modelled as a multivariate Gaussian distribution with density:

$$P(x_t \mid y = k) = \frac{1}{(2\pi)^{d/2}|\sum_k|^{1/2}} \exp\left(-\frac{1}{2}(x_t - \mu_k)^t \sum_k^{-1}(x_t - \mu_k)\right), \quad \text{(A.8)}$$

where d denotes the number of features, and μ denotes the mean value of x. According to the model above, the log of the posterior is:

$$\log P(y_t = k \mid x_t) = \\ -\frac{1}{2}(x_t - \mu_k)^t \sum\nolimits^{-1} (x_t - \mu_k) + \log P(y_t = k) + Cst. \tag{A.9}$$

The term $(x_t - \mu_k)^t \Sigma^{-1}(x^t - \mu_k)$ corresponds to the Mahalanobis Distance between the sample x_t and the mean μ_k.[17]

3. Tree based methods

Classification Tree. The Classification Tree is another supervised machine learning method to make classifications. The process includes two main steps: node splitting and threshold determination. Specifically, the first step of node splitting generally happens when the attribute represented by a node does not give a judgment, thus splitting this node into two child nodes. Then, the second step of threshold determination is to choose an appropriate threshold value to minimize the training error. By repeating these two steps, a tree is grown.

To improve generalizability and robustness over a single classification tree, two kinds of ensemble methods are widely used by combining the predictions of several trees. The first is averaging methods, and the second is boosting methods. On the one hand, in averaging methods, the driving principle is to build several estimators independently and then average their predictions. On average, the combined estimator is usually better than any single base estimator because its variance is reduced. On the other hand, in boosting methods, base estimators are built sequentially, and one tries to reduce the bias of the combined estimator. The motivation is to combine several weak models to produce a powerful ensemble.

Random Forest. The random forest approach by Breiman (2001) is a machine learning method used to make predictions by generating numerous regression trees and averaging the outcome of each tree to find the final result. To construct the random forest, a bootstrap is performed on the dataset, and from these, subsets of features are randomly chosen to grow various trees. These two sources of randomness can reduce the variance and overcome the overfitting problem. The final result is given by averaging the probabilistic prediction of each tree.

XGBoost. The XGBoost is an algorithm proposed by Chen and Guestrin (2016). The basic idea is to combine classifiers that are iteratively constructed through the resampling of the training data by assigning increased weight to misclassified observations. Thus, a new classifier is produced that can boost the performance of previous problematic cases. This process is repeated to combine several classifiers into a final classifier by applying a weighted majority vote.

4. Other Models

Support Vector Machine. The Support Vector Machine (SVM) is a machine learning algorithm proposed by Cortes and Vapnik (1995), widely used to solve classification and regression problems. The basic concept of an SVM is to select a few specific data points, known as support vectors (SV), from the data set, which are used to create a hyperplane that effectively separates the two classes of observations. This hyperplane is chosen as the solution that maximizes the distance between the support vectors. In the case of linear separability, the SVM can find the hyperplane directly in the original feature space. And even if the data is not linearly differentiable, SVM uses kernel function techniques to map the data into a higher dimensional feature space, thus making the data linearly differentiable. For the n-dimensional feature vector $X \equiv [x^{[1]}, x^{[2]}, \ldots, x^{[n]}]$, a linear boundary (hyperplane) can be defined as $w'X + b = 0$, where $w =$ is the normal vector determining the direction of the hyperplane, and b is the bias term determining the distance between the hyperplane and the origin. The hyperplane can classify the training samples correctly, that is, for a sample (X_t, y_t) if $y_t = +1$[18], then $w'X_t + b > 0$; else if

$y_t = -1$, then $w^T X_t + b < 0$. The Support Vector Machine solves the following primal problem:

$$\begin{gathered} \min_{w,b,\zeta} \frac{1}{2} w'w + C \sum\nolimits_{t=1}^{n} \zeta_t \\ \text{subject to } y_t(w'\phi(X_t) + b) \geq 1 - \zeta_t, \\ \zeta_t \geq 0,\ t = 1, \ldots, n \end{gathered} \tag{A.10}$$

The principle is to maximize the margin (by minimizing $\|w\|^2$) while incurring a penalty when a sample is misclassified or within the margin boundary. And, some misclassified samples are allowed to be at a distance ζ_i from their correct margin boundary. And the term C controls the strength of this penalty. The function ϕ maps X into a higher dimensional space. And the dual problem to Eq.(5) is:

$$\begin{gathered} \min_{\alpha} \frac{1}{2} \alpha' Q \alpha - e'\alpha \\ \text{subject to } y'\alpha = 0 \\ 0 \leq \alpha_t \leq C,\ t = 1, \ldots, n \end{gathered} \tag{A.11}$$

where e is the vector of all ones, and Q is a $n \times n$ positive semidefinite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, where $K(X_i, X_j) = \phi(X_i)'\phi(X_j)$ is the kernel. The terms αi are called the dual coefficients, which are upper-bounded by C. Here is a list of kernel functions used in this study.[19)]

Linear: $K_1(X_i, X_j) = X_i'X_j$

Radial basis function (RBF):

$$K_2(X_i, X_j) = e^{-\nu X_i - X_j^2} \tag{A.12}$$

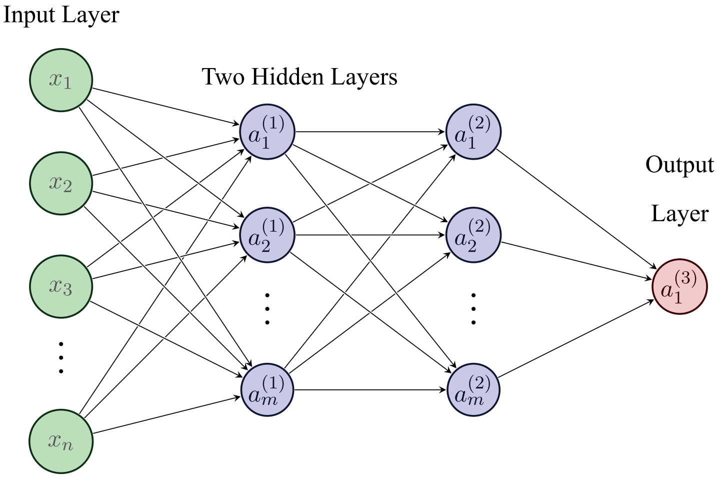
Polynomial: $K_3(X_i, X_j) = (\nu X'_i X_j + \gamma)^d$

Sigmoid: $K_4(X_i, X_j) = \tanh(\nu X_i'X_j + \gamma)$

K-Nearest Neighbors. K-Nearest Neighbors (KNN) is a widely used algorithm for classification, which is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the class with the most representatives within the nearest neighbors of the point. In other words, the new observation is assigned to the most common class among its K-nearest neighbors. There are two critical elements in the KNN algorithm: the measurement of distance and the choice of K. This study uses the Euclidean distance metric for computing distances in a multidimensional predictor space with quantitative variables. As for a selection of K, generally, a more extensive K suppresses the effects of noise but makes the classification boundaries less distinct.

Neural Network. The Neural Network is a supervised learning algorithm that learns a function $f(\cdot): R^N \rightarrow R^1$ by training on a dataset, where N is the number of dimensions for input and 1 is the number of dimensions for output. For solving this recession classification question, the function passes through the logit function $g(z) = 1/(1+e^{-z})$ to output a number from 0 and 1, standing for the probability of recession. However, unlike logit regression, there are also one or multiple non-linear hidden layers between the input and output layers. Figure A.10 provides an example of a neural network with two hidden layers, in which x denotes input variables, and a_m indicates the $m - th$ artificial neurons in a layer. The choice of hyperparameters including the number of hidden layers and number of nodes in each layer will be mentioned in Section 3.

Figure A.10: An example figure of Neural



This figure illustrates a Neural Network with two hidden layers. The circles on the left side denote n input variables, and the circles in the middle are artificial neurons in 2 hidden layers. The circle on the right is the output neural.

Appendix B. Variables and correlation

In this section, the variables used in this study, and their correlation matrix is listed.

Table B.6: The list of predictive variables

	Predictive variable	**Unit**	**Transform**	**Freq.**
1	Term spread - 10-year Treasury yield - 3-month Tbill rate	%	-	D
2	Term spread - 10-year Treasury yield - 2-year Treasury yield	%	-	D
3	Term spread - 5-year Treasury yield - 3-month Tbill rate	%	-	D
4	10-year Treasury yield	%	-	D
5	3-month Treasury Bill rate	%	-	D
6	Initial Claims	Number	-	W
7	Initial Claims (4-week moving average)	Number	-	W
8	Rate of unemployment	%	-	M
9	Long-term Non-Accelerating Inflation Rate of Unemployment	%	-	Q
10	Unemployment Gap - Unemployment rate - Long-term NAIRU	%		Q
11	Average W Hours Total Private	%	YoY % Diff	M
12	Average Hourly Earnings Total Private	%	YoY % Diff	M
13	Non-Farm Payrolls - Total Private	Thousands	-	M
14	Non-Farm Payrolls - Total Private	%	YoY % Diff	M
15	Real Money Supply M1	%	MoM % Diff	M
16	Real Money Supply M1	%	YoY % Diff	M
17	Real Money Supply M2	%	MoM % Diff	M
18	Real Money Supply M2	%	YoY % Diff	M
19	Moody's BAA Yield	%	-	M
20	Moody's BAA Yield	%	MoM % Dif	M
21	Credit Spread - Moody's BAA Yield over 10-year Treasury Yield	%	-	M
22	Effective Federal Funds rate	%	-	M
23	S&P 500 Index M return	%	MoM % Diff	D
24	Heavy truck sales	Millions	-	M
25	Heavy truck sales	Millions	MoM % Diff	M
26	Motor Vehicle Retail Sales: Domestic Autos	Millions	-	M
27	Motor Vehicle Retail Sales: Domestic Autos	Millions	MoM % Diff	M
28	Industrial Production Growth	%	YoY % Diff	M
29	Real GDP Growth	%	MoM % Diff	Q
30	Real GDP Growth	%	YoY % Diff	Q
31	Real GDI Growth	%	MoM % Diff	Q
32	Real GDI Growth	%	YoY % Diff	Q
33	Output Gap (Real GDP - Real Potential GDP)	%	-	Q
34	Housing starts	%	YoY % Diff	M
35	Housing permits	%	YoY % Diff	M
36	Ratio of residential investment to GDP	%	-	Q
37	Change in ratio of residential investment to GDP (yoy, %)	%	YoY % Diff	Q
38	Change in ratio of short-term household liabilities to Disposable Personal Income (yoy, %)	%	YoY % Diff	Q
39	Change in ratio of average (GDP+GDI) to Potential GDP	%	-	Q
40	Corporate Profits	%	YoY % Diff	Q
41	Corporate Profits After Tax	%	YoY % Diff	Q
42	Chicago Fed National Financial Conditions Index	Index	Normalized	W
43	Change in Chicago Fed National Financial Conditions Index	Index	MoM % Diff	W
44	Chicago Fed National Financial Conditions Leverage SubIndex	Index	Normalized	W
45	Change in University of Michigan consumer sentiment	%	YoY % Diff	M
46	Real Personal Income excluding Transfers	%	YoY % Diff	M
47	Real Manufacturing and Trade Industries Sales	%	YoY % Diff	M
48	Capacity Utilization Rate	%	-	M
49	Producer Price Index by Commodity for Final Demand: Finished Goods	%	YoY % Diff	M
50	Producer Price Index by Commodity for Intermediate Demand by Commodity Type: Unprocessed Goods for Intermediate Demand	%	YoY % Diff	M
51	Producer Price Index by Commodity Metals and metal products: Primary nonferrous metals	%	YoY % Diff	M
52	Personal Consumption Expenditures	%	YoY % Diff	M

The "YoY % Diff" and "MoM % Diff" on the "Transform" column denote the Year-on-Year percentage difference and the Month-on-Month percentage difference respectively. The "D", "W", "M" and "Q" on the "Freq." column denotes daily, weekly, monthly and annually respectively.

Table B.7: The correlation matrix of the variables

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24	X25	X26	X27	X28	X29	X30	X31	X32	X33	X34	X35	X36	X37	X38	X39	X40	X41	X42	X43	X44	X45	X46	X47	X48	X49	X50	X51	X52
X1	1.00	0.86	0.91	0.13	-0.24	0.04	0.04	0.54	0.17	0.34	-0.44	0.04	-0.28	-0.15	-0.03	-0.06	-0.07	-0.03	0.19	0.00	0.28	-0.24	-0.29	-0.54	0.14	0.08	0.04	-0.03	0.04	-0.34	0.07	-0.40	-0.51	0.17	0.17	-0.22	-0.05	-0.07	-0.56	-0.09	-0.11	0.02	0.01	-0.07	0.20	-0.28	0.00	-0.33	-0.14	-0.07	-0.11	-0.05
X2	0.86	1.00	0.58	-0.24	-0.55	0.07	0.07	0.51	-0.15	0.12	-0.55	0.09	0.02	-0.28	-0.00	0.02	0.03	0.10	-0.16	-0.05	0.43	-0.54	-0.14	-0.44	0.11	-0.14	0.03	-0.12	-0.01	-0.48	0.01	-0.53	-0.36	0.14	0.14	-0.36	-0.12	-0.18	-0.50	-0.06	-0.06	-0.16	0.01	-0.06	0.15	-0.34	-0.06	-0.49	-0.15	-0.01	-0.14	-0.24
X3	0.91	0.58	1.00	0.47	0.13	0.01	0.01	0.47	0.45	0.48	-0.24	-0.03	-0.53	-0.03	-0.05	-0.14	-0.14	-0.14	0.50	0.03	0.08	0.12	-0.40	-0.57	0.13	0.29	0.04	0.05	0.09	-0.15	0.11	-0.22	-0.57	0.15	0.15	-0.06	-0.01	0.05	-0.51	-0.12	-0.16	0.17	0.01	-0.09	0.22	-0.16	0.05	-0.13	-0.12	-0.11	-0.08	0.14
X4	0.13	-0.24	0.47	1.00	0.93	-0.05	-0.05	0.25	0.95	0.78	0.20	-0.18	-0.92	0.20	-0.09	-0.26	-0.15	-0.29	0.97	-0.03	-0.29	0.93	-0.66	-0.51	0.01	0.71	0.05	0.20	0.13	0.33	0.12	0.31	-0.59	-0.00	0.01	0.21	-0.12	0.30	-0.24	-0.14	-0.14	0.41	0.04	-0.07	0.16	0.19	0.13	0.35	-0.03	-0.12	-0.02	0.43
X5	-0.24	-0.55	0.13	0.93	1.00	-0.06	-0.06	0.05	0.87	0.64	0.36	-0.19	-0.80	0.25	-0.07	-0.23	-0.12	-0.27	0.89	-0.03	-0.39	1.00	-0.54	-0.30	-0.04	0.67	0.03	0.21	0.12	0.45	0.09	0.45	-0.39	-0.06	-0.05	0.29	-0.10	0.32	-0.03	-0.11	-0.10	0.40	0.04	-0.04	0.08	0.29	0.13	0.46	0.02	-0.09	0.02	0.44
X6	0.04	0.07	0.01	-0.05	-0.06	1.00	0.97	0.51	-0.02	0.01	0.27	0.03	-0.08	-0.62	0.40	0.42	0.64	0.45	0.01	-0.07	0.27	-0.06	0.07	-0.22	-0.09	-0.22	-0.08	-0.44	-0.36	-0.35	-0.39	-0.34	-0.38	-0.06	-0.03	-0.09	-0.16	-0.13	-0.44	-0.12	-0.13	0.20	-0.13	0.12	-0.18	-0.33	-0.35	-0.42	-0.21	-0.17	-0.11	-0.44
X7	0.04	0.07	0.01	-0.05	-0.06	0.97	1.00	0.53	-0.02	0.00	0.28	0.06	-0.09	-0.64	0.49	0.45	0.62	0.46	0.01	-0.13	0.26	-0.06	0.07	-0.22	-0.06	-0.22	-0.03	-0.44	-0.28	-0.33	-0.32	-0.32	-0.36	-0.06	-0.03	-0.08	-0.15	-0.12	-0.42	-0.11	-0.12	0.18	-0.17	0.11	-0.18	-0.33	-0.34	-0.42	-0.20	-0.17	-0.10	-0.43
X8	0.54	0.51	0.47	0.25	0.05	0.51	0.53	1.00	0.30	0.59	-0.10	0.08	-0.48	-0.50	0.21	0.22	0.26	0.29	0.35	-0.19	0.41	0.05	-0.32	-0.74	0.07	-0.01	0.10	-0.28	0.01	-0.54	-0.01	-0.51	-0.79	0.20	0.24	-0.45	-0.40	-0.33	-0.87	-0.17	-0.21	0.30	-0.05	0.06	0.20	-0.34	-0.19	-0.57	-0.21	-0.20	-0.20	-0.23
X9	0.17	-0.15	0.45	0.95	0.87	-0.02	-0.02	0.30	1.00	0.86	0.05	-0.16	-0.97	0.16	-0.07	-0.23	-0.10	-0.23	0.94	-0.07	-0.22	0.87	-0.76	-0.58	0.01	0.80	0.06	0.18	0.15	0.38	0.15	0.36	-0.67	0.04	0.06	0.23	-0.10	0.35	-0.28	-0.14	-0.14	0.33	0.04	-0.10	0.18	0.18	0.15	0.36	-0.15	-0.18	-0.07	0.38
X10	0.34	0.12	0.48	0.78	0.64	0.01	0.00	0.59	0.86	1.00	-0.16	-0.10	-0.90	0.07	-0.07	-0.25	-0.06	-0.18	0.82	-0.10	0.00	0.65	-0.78	-0.72	0.04	0.65	0.08	0.17	0.11	0.14	0.13	0.16	-0.81	0.09	0.11	-0.13	-0.31	0.08	-0.48	-0.12	-0.14	0.34	0.04	-0.02	0.26	0.10	0.12	0.12	-0.13	-0.17	-0.11	0.21
X11	-0.44	-0.55	-0.24	0.20	0.36	0.27	0.28	-0.10	0.05	-0.16	1.00	-0.17	-0.02	-0.14	0.13	0.19	0.10	0.11	0.20	0.12	-0.08	0.35	0.35	0.07	-0.10	-0.26	-0.03	-0.18	-0.10	-0.09	-0.15	-0.04	0.17	-0.18	-0.17	0.16	-0.18	0.08	0.11	-0.10	-0.10	0.40	-0.04	0.14	-0.38	-0.10	-0.28	0.02	0.40	0.19	0.02	0.17
X12	0.04	0.09	-0.03	-0.18	-0.19	0.03	0.06	0.08	-0.16	-0.10	-0.17	1.00	0.10	0.09	0.07	0.43	0.02	0.20	-0.25	0.03	-0.28	-0.20	0.21	0.18	0.21	-0.08	0.04	0.46	0.33	-0.01	0.35	0.04	0.17	0.37	0.35	0.05	0.21	-0.22	0.13	0.28	0.23	-0.37	0.01	-0.24	0.24	0.38	0.61	0.15	0.01	0.21	0.31	0.17
X13	-0.28	0.02	-0.53	-0.92	-0.80	-0.08	-0.09	-0.48	-0.97	-0.90	-0.02	0.10	1.00	-0.07	0.02	0.14	0.05	0.16	-0.92	0.10	0.16	-0.80	0.74	0.68	-0.04	-0.73	-0.07	-0.13	-0.16	-0.24	-0.15	-0.22	0.78	-0.10	-0.12	-0.11	0.15	-0.23	0.44	0.13	0.15	-0.33	-0.02	0.10	-0.22	-0.11	-0.13	-0.25	0.17	0.20	0.10	-0.31
X14	-0.15	-0.28	-0.03	0.20	0.25	-0.62	-0.64	-0.50	0.16	0.07	-0.14	0.09	-0.07	1.00	-0.28	-0.49	-0.36	-0.49	0.09	0.15	-0.52	0.24	0.00	0.32	0.03	0.36	-0.04	0.77	0.15	0.59	0.17	0.61	0.33	0.14	0.10	0.24	0.45	0.09	0.52	0.31	0.23	-0.25	0.12	-0.20	0.19	0.68	0.63	0.69	0.32	0.30	0.28	0.71
X15	-0.03	-0.00	-0.05	-0.09	-0.07	0.40	0.49	0.21	-0.07	-0.07	0.13	0.07	0.02	-0.28	1.00	0.29	0.42	0.22	-0.07	-0.06	0.10	-0.07	0.09	-0.04	-0.12	-0.13	0.14	-0.22	-0.22	-0.13	-0.27	-0.13	-0.14	-0.04	-0.01	-0.03	0.00	-0.08	-0.17	-0.04	-0.06	0.03	-0.39	0.04	-0.10	-0.13	-0.16	-0.18	-0.10	-0.09	-0.03	-0.23
X16	-0.06	0.02	-0.14	-0.26	-0.23	0.42	0.45	0.22	-0.23	-0.25	0.19	0.43	0.14	-0.49	0.29	1.00	0.23	0.73	-0.25	-0.05	0.09	-0.23	0.43	0.09	0.09	-0.33	0.12	-0.26	0.24	-0.20	0.23	-0.18	0.06	0.09	0.13	0.01	0.10	-0.20	-0.09	0.05	-0.01	-0.06	-0.11	0.03	-0.17	-0.11	0.03	-0.23	-0.14	-0.02	0.04	-0.18
X17	-0.07	0.03	-0.14	-0.15	-0.12	0.64	0.62	0.26	-0.10	-0.06	0.10	0.02	0.05	-0.36	0.42	0.23	1.00	0.46	-0.07	-0.04	0.39	-0.11	0.06	-0.03	-0.12	-0.15	-0.12	-0.33	-0.38	-0.15	-0.40	-0.13	-0.20	0.03	0.05	-0.07	-0.00	-0.21	-0.24	-0.21	-0.24	0.14	-0.14	0.10	-0.10	-0.05	-0.24	-0.36	-0.30	-0.23	-0.12	-0.36
X18	-0.03	0.10	-0.14	-0.29	-0.27	0.45	0.46	0.29	-0.23	-0.18	0.11	0.20	0.16	-0.49	0.22	0.73	0.46	1.00	-0.21	-0.08	0.40	-0.26	0.33	0.02	0.03	-0.27	0.06	-0.40	0.14	-0.14	0.12	-0.13	-0.02	0.18	0.22	-0.03	0.13	-0.30	-0.16	-0.18	-0.25	0.04	-0.06	0.02	-0.02	-0.10	-0.10	-0.50	-0.40	-0.25	-0.18	-0.27
X19	0.19	-0.16	0.50	0.97	0.89	0.01	0.01	0.35	0.94	0.82	0.20	-0.25	-0.92	0.09	-0.07	-0.25	-0.07	-0.21	1.00	-0.03	-0.08	0.89	-0.69	-0.60	-0.02	0.66	0.05	0.07	0.07	0.23	0.06	0.21	-0.68	-0.03	-0.01	0.12	-0.21	0.26	-0.37	-0.25	-0.25	0.54	0.04	0.02	0.12	0.09	0.00	0.19	-0.07	-0.18	-0.12	0.34
X20	0.00	-0.05	0.03	-0.03	-0.03	-0.07	-0.13	-0.19	-0.07	-0.10	0.12	0.03	0.10	0.15	-0.06	-0.05	-0.04	-0.08	-0.03	1.00	-0.05	-0.04	0.12	0.11	-0.04	-0.04	-0.05	0.11	-0.13	0.04	-0.11	0.04	0.14	0.01	0.01	0.07	0.07	0.04	0.14	0.04	0.02	0.02	-0.00	0.06	-0.12	0.02	0.05	0.13	0.22	0.17	0.07	0.11
X21	0.28	0.43	0.08	-0.29	-0.39	0.27	0.26	0.41	-0.22	0.00	-0.08	-0.28	0.16	-0.52	0.10	0.09	0.39	0.40	-0.08	-0.05	1.00	-0.37	0.01	-0.32	-0.10	-0.34	-0.00	-0.63	-0.30	-0.52	-0.32	-0.53	-0.30	-0.09	-0.08	-0.42	-0.36	-0.25	-0.51	-0.45	-0.46	0.46	-0.02	0.41	-0.20	-0.49	-0.59	-0.77	-0.22	-0.24	-0.42	-0.48
X22	-0.24	-0.54	0.12	0.93	1.00	-0.06	-0.06	0.05	0.87	0.65	0.35	-0.20	-0.80	0.24	-0.07	-0.23	-0.11	-0.26	0.89	-0.04	-0.37	1.00	-0.55	-0.31	-0.04	0.67	0.03	0.20	0.11	0.44	0.09	0.45	-0.40	-0.08	-0.07	0.28	-0.12	0.33	-0.04	-0.12	-0.11	0.43	0.04	-0.02	0.07	0.29	0.12	0.45	0.03	-0.09	0.02	0.44
X23	-0.29	-0.14	-0.40	-0.66	-0.54	0.07	0.07	-0.32	-0.76	-0.78	0.35	0.21	0.74	0.00	0.09	0.43	0.06	0.33	-0.69	0.12	0.01	-0.55	1.00	0.56	0.00	-0.76	-0.05	-0.09	-0.03	-0.23	-0.04	-0.21	0.70	0.09	0.07	-0.07	0.26	-0.33	0.30	0.09	0.02	-0.21	-0.03	0.03	-0.23	-0.03	-0.03	-0.18	0.35	0.31	0.14	0.03
X24	-0.54	-0.44	-0.57	-0.51	-0.30	-0.22	-0.22	-0.74	-0.58	-0.72	0.07	0.18	0.68	0.32	-0.04	0.09	-0.03	0.02	-0.60	0.11	-0.32	-0.31	0.56	1.00	0.08	-0.25	-0.05	0.22	0.06	0.36	0.07	0.40	0.85	-0.01	-0.03	0.38	0.54	0.06	0.80	0.41	0.37	-0.42	-0.00	-0.03	-0.13	0.35	0.25	0.30	0.13	0.17	0.33	0.08
X25	0.14	0.11	0.13	0.01	-0.04	-0.09	-0.06	0.07	0.01	0.04	-0.10	0.21	-0.04	0.03	-0.12	0.09	-0.12	0.03	-0.02	-0.04	-0.10	-0.04	0.00	0.08	1.00	0.05	0.18	0.15	0.27	0.01	0.28	0.01	0.01	0.13	0.16	0.03	0.10	-0.05	0.01	0.09	0.07	-0.14	-0.03	-0.13	0.08	0.09	0.22	0.05	-0.00	0.02	0.06	0.10
X26	0.08	-0.14	0.29	0.71	0.67	-0.22	-0.22	-0.01	0.80	0.65	-0.26	-0.08	-0.73	0.36	-0.13	-0.33	-0.15	-0.27	0.66	-0.04	-0.34	0.67	-0.76	-0.25	0.05	1.00	0.20	0.36	0.22	0.66	0.23	0.64	-0.42	0.07	0.08	0.29	0.20	0.30	0.05	-0.03	-0.04	-0.02	0.04	-0.23	0.26	0.38	0.37	0.56	-0.31	-0.22	-0.04	0.31
X27	0.04	0.03	0.04	0.05	0.03	-0.08	-0.03	0.10	0.06	0.08	-0.03	0.04	-0.07	-0.04	0.14	0.12	-0.12	0.06	0.05	-0.05	-0.00	0.03	-0.05	-0.05	0.18	0.20	1.00	-0.01	0.22	0.01	0.21	0.01	-0.05	0.04	0.06	-0.01	-0.01	-0.00	-0.04	0.03	0.02	-0.02	-0.09	-0.04	0.04	-0.01	0.06	-0.01	-0.02	-0.04	-0.04	0.07
X28	-0.03	-0.12	0.05	0.20	0.21	-0.44	-0.44	-0.28	0.18	0.17	-0.18	0.46	-0.13	0.77	-0.22	-0.26	-0.33	-0.40	0.07	0.11	-0.63	0.20	-0.09	0.22	0.15	0.36	-0.01	1.00	0.32	0.45	0.37	0.51	0.23	0.25	0.22	0.26	0.33	0.09	0.46	0.38	0.34	-0.39	0.09	-0.33	0.32	0.70	0.87	0.67	0.28	0.42	0.43	0.64
X29	0.04	-0.01	0.09	0.13	0.12	-0.36	-0.28	0.01	0.15	0.11	-0.10	0.33	-0.16	0.15	-0.22	0.24	-0.38	0.14	0.07	-0.13	-0.30	0.11	-0.03	0.06	0.27	0.22	0.22	0.32	1.00	0.28	0.89	0.27	0.16	0.30	0.32	0.20	0.25	0.04	0.23	0.20	0.17	-0.26	0.09	-0.30	0.24	0.25	0.46	0.22	-0.05	0.06	0.09	0.32
X30	-0.34	-0.48	-0.15	0.33	0.45	-0.35	-0.33	-0.54	0.38	0.14	-0.09	-0.01	-0.24	0.59	-0.13	-0.20	-0.15	-0.14	0.23	0.04	-0.52	0.44	-0.23	0.36	0.01	0.66	0.01	0.45	0.28	1.00	0.27	0.96	0.24	0.01	0.01	0.58	0.56	0.39	0.65	0.21	0.17	-0.27	0.05	-0.22	0.06	0.57	0.46	0.73	-0.10	-0.03	0.15	0.42
X31	0.07	0.01	0.11	0.12	0.09	-0.39	-0.32	-0.01	0.15	0.13	-0.15	0.35	-0.15	0.17	-0.27	0.23	-0.40	0.12	0.06	-0.11	-0.32	0.09	-0.04	0.07	0.28	0.23	0.21	0.37	0.89	0.27	1.00	0.29	0.14	0.28	0.29	0.19	0.25	0.02	0.25	0.21	0.18	-0.27	0.10	-0.27	0.26	0.28	0.48	0.24	-0.05	0.09	0.13	0.30
X32	-0.40	-0.53	-0.22	0.31	0.45	-0.34	-0.32	-0.51	0.36	0.16	-0.04	0.04	-0.22	0.61	-0.13	-0.18	-0.13	-0.13	0.21	0.04	-0.53	0.45	-0.21	0.40	0.01	0.64	0.01	0.51	0.27	0.96	0.29	1.00	0.24	0.07	0.08	0.53	0.54	0.30	0.67	0.27	0.21	-0.27	0.05	-0.25	0.07	0.64	0.51	0.74	-0.07	0.00	0.18	0.43
X33	-0.51	-0.36	-0.57	-0.59	-0.39	-0.38	-0.36	-0.79	-0.67	-0.81	0.17	0.17	0.78	0.33	-0.14	0.06	-0.20	-0.02	-0.68	0.14	-0.30	-0.40	0.70	0.85	0.01	-0.42	-0.05	0.23	0.16	0.24	0.14	0.24	1.00	-0.07	-0.10	0.28	0.35	0.07	0.83	0.29	0.29	-0.39	0.04	-0.01	-0.19	0.27	0.21	0.25	0.34	0.36	0.30	0.17
X34	0.17	0.14	0.15	-0.00	-0.06	-0.06	-0.06	0.20	0.04	0.09	-0.18	0.37	-0.10	0.14	-0.04	0.09	0.03	0.18	-0.03	0.01	-0.09	-0.08	0.09	-0.01	0.13	0.07	0.04	0.25	0.30	0.01	0.28	0.07	-0.07	1.00	0.91	0.03	0.36	-0.32	-0.13	0.12	0.05	-0.32	0.10	-0.50	0.42	0.20	0.43	-0.05	-0.05	0.05	0.00	0.33
X35	0.17	0.14	0.15	0.01	-0.05	-0.03	-0.03	0.24	0.06	0.11	-0.17	0.35	-0.12	0.10	-0.01	0.13	0.05	0.22	-0.01	0.01	-0.08	-0.07	0.07	-0.03	0.16	0.08	0.06	0.22	0.32	0.01	0.29	0.08	-0.10	0.91	1.00	0.03	0.37	-0.34	-0.16	0.13	0.05	-0.32	0.10	-0.53	0.46	0.17	0.42	-0.07	-0.08	0.04	-0.03	0.31
X36	-0.22	-0.36	-0.06	0.21	0.29	-0.09	-0.08	-0.45	0.23	-0.13	0.16	0.05	-0.11	0.24	-0.03	0.01	-0.07	-0.03	0.12	0.07	-0.42	0.28	-0.07	0.38	0.03	0.29	-0.01	0.26	0.20	0.58	0.19	0.53	0.28	0.03	0.03	1.00	0.52	0.79	0.53	0.46	0.46	-0.24	0.01	-0.28	-0.05	0.27	0.31	0.42	0.08	0.20	0.39	0.40
X37	-0.05	-0.12	-0.01	-0.12	-0.10	-0.16	-0.15	-0.40	-0.10	-0.31	-0.18	0.21	0.15	0.45	0.00	0.10	-0.00	0.13	-0.21	0.07	-0.36	-0.12	0.26	0.54	0.10	0.20	-0.01	0.33	0.25	0.56	0.25	0.54	0.35	0.36	0.37	0.52	1.00	0.05	0.42	0.39	0.26	-0.55	0.03	-0.46	0.14	0.41	0.46	0.34	-0.12	0.03	0.19	0.29
X38	-0.07	-0.18	0.05	0.30	0.32	-0.13	-0.12	-0.33	0.35	0.08	0.08	-0.22	-0.23	0.09	-0.08	-0.20	-0.21	-0.30	0.26	0.04	-0.25	0.33	-0.33	0.06	-0.05	0.30	-0.00	0.09	0.04	0.39	0.02	0.30	0.07	-0.32	-0.34	0.79	0.05	1.00	0.33	0.32	0.39	-0.00	-0.01	0.01	-0.21	-0.02	0.01	0.35	0.15	0.18	0.31	0.24
X39	-0.56	-0.50	-0.51	-0.24	-0.03	-0.44	-0.42	-0.87	-0.28	-0.48	0.11	0.13	0.44	0.52	-0.17	-0.09	-0.24	-0.16	-0.37	0.14	-0.51	-0.04	0.30	0.80	0.01	0.05	-0.04	0.46	0.23	0.65	0.25	0.67	0.83	-0.13	-0.16	0.53	0.42	0.33	1.00	0.36	0.38	-0.40	0.04	-0.10	-0.12	0.52	0.41	0.61	0.23	0.30	0.35	0.32
X40	-0.09	-0.06	-0.12	-0.14	-0.11	-0.12	-0.11	-0.17	-0.14	-0.12	-0.10	0.28	0.13	0.31	-0.04	0.05	-0.21	-0.18	-0.25	0.04	-0.45	-0.12	0.09	0.41	0.09	-0.03	0.03	0.38	0.20	0.21	0.21	0.27	0.29	0.12	0.13	0.46	0.39	0.32	0.36	1.00	0.96	-0.42	0.03	-0.17	-0.01	0.22	0.37	0.35	0.33	0.33	0.48	0.25
X41	-0.11	-0.06	-0.16	-0.14	-0.10	-0.13	-0.12	-0.21	-0.14	-0.14	-0.10	0.23	0.15	0.23	-0.06	-0.01	-0.24	-0.25	-0.25	0.02	-0.46	-0.11	0.02	0.37	0.07	-0.04	0.02	0.34	0.17	0.17	0.18	0.21	0.29	0.05	0.05	0.46	0.26	0.39	0.38	0.96	1.00	-0.40	0.03	-0.18	-0.03	0.15	0.31	0.34	0.34	0.34	0.46	0.18
X42	0.02	-0.16	0.17	0.41	0.40	0.20	0.18	0.30	0.33	0.34	0.40	-0.37	-0.33	-0.25	0.03	-0.06	0.14	0.04	0.54	0.02	0.46	0.43	-0.21	-0.42	-0.14	-0.02	-0.02	-0.39	-0.26	-0.27	-0.27	-0.27	-0.39	-0.32	-0.32	-0.24	-0.55	-0.00	-0.40	-0.42	-0.40	1.00	-0.00	0.64	-0.25	-0.23	-0.49	-0.28	0.10	-0.11	-0.19	-0.05
X43	0.01	0.01	0.01	0.04	0.04	-0.13	-0.17	-0.05	0.04	0.04	-0.04	0.01	-0.02	0.12	-0.39	-0.11	-0.14	-0.06	0.04	-0.00	-0.02	0.04	-0.03	-0.00	-0.03	0.04	-0.09	0.09	0.09	0.05	0.10	0.05	0.04	0.10	0.10	0.01	0.03	-0.01	0.04	0.03	0.03	-0.00	1.00	-0.03	0.05	0.08	0.09	0.04	0.07	0.06	0.01	0.11
X44	-0.07	-0.06	-0.09	-0.07	-0.04	0.12	0.11	0.06	-0.10	-0.02	0.14	-0.24	0.10	-0.20	0.04	0.03	0.10	0.02	0.02	0.06	0.41	-0.02	0.03	-0.03	-0.13	-0.23	-0.04	-0.33	-0.30	-0.22	-0.27	-0.25	-0.01	-0.50	-0.53	-0.28	-0.46	0.01	-0.10	-0.17	-0.18	0.64	-0.03	1.00	-0.35	-0.21	-0.42	-0.17	0.12	-0.03	-0.05	-0.25
X45	0.20	0.15	0.22	0.16	0.08	-0.18	-0.18	0.20	0.18	0.26	-0.38	0.24	-0.22	0.19	-0.10	-0.17	-0.10	-0.02	0.12	-0.12	-0.20	0.07	-0.23	-0.13	0.08	0.26	0.04	0.32	0.24	0.06	0.26	0.07	-0.19	0.42	0.46	-0.05	0.14	-0.21	-0.12	-0.01	-0.03	-0.25	0.05	-0.35	1.00	0.28	0.40	0.03	-0.33	-0.20	0.05	0.18
X46	-0.28	-0.34	-0.16	0.19	0.29	-0.33	-0.33	-0.34	0.18	0.10	-0.10	0.38	-0.11	0.68	-0.13	-0.11	-0.05	-0.10	0.09	0.02	-0.49	0.29	-0.03	0.35	0.09	0.38	-0.01	0.70	0.25	0.57	0.28	0.64	0.27	0.20	0.17	0.27	0.41	-0.02	0.52	0.22	0.15	-0.23	0.08	-0.21	0.28	1.00	0.70	0.58	0.03	0.15	0.32	0.54
X47	0.00	-0.06	0.05	0.13	0.13	-0.35	-0.34	-0.19	0.15	0.12	-0.28	0.61	-0.13	0.63	-0.16	0.03	-0.24	-0.10	0.00	0.05	-0.59	0.12	-0.03	0.25	0.22	0.37	0.06	0.87	0.46	0.46	0.48	0.51	0.21	0.43	0.42	0.31	0.46	0.01	0.41	0.37	0.31	-0.49	0.09	-0.42	0.40	0.70	1.00	0.56	0.07	0.29	0.41	0.62
X48	-0.33	-0.49	-0.13	0.35	0.46	-0.42	-0.42	-0.57	0.36	0.12	0.02	0.15	-0.25	0.69	-0.18	-0.23	-0.36	-0.50	0.19	0.13	-0.77	0.45	-0.18	0.30	0.05	0.56	-0.01	0.67	0.22	0.73	0.24	0.74	0.25	-0.05	-0.07	0.42	0.34	0.35	0.61	0.35	0.34	-0.28	0.04	-0.17	0.03	0.58	0.56	1.00	0.17	0.17	0.30	0.50
X49	-0.14	-0.15	-0.12	-0.03	0.02	-0.21	-0.20	-0.21	-0.15	-0.13	0.40	0.01	0.17	0.32	-0.10	-0.14	-0.30	-0.40	-0.07	0.22	-0.22	0.03	0.35	0.13	-0.00	-0.31	-0.02	0.28	-0.05	-0.10	-0.05	-0.07	0.34	-0.05	-0.08	0.08	-0.12	0.15	0.23	0.33	0.34	0.10	0.07	0.12	-0.33	0.03	0.07	0.17	1.00	0.83	0.38	0.50
X50	-0.07	-0.01	-0.11	-0.12	-0.09	-0.17	-0.17	-0.20	-0.18	-0.17	0.19	0.21	0.20	0.30	-0.09	-0.02	-0.23	-0.25	-0.18	0.17	-0.24	-0.09	0.31	0.17	0.02	-0.22	-0.04	0.42	0.06	-0.03	0.09	0.00	0.36	0.05	0.04	0.20	0.03	0.18	0.30	0.33	0.34	-0.11	0.06	-0.03	-0.20	0.15	0.29	0.17	0.83	1.00	0.49	0.49
X51	-0.11	-0.14	-0.08	-0.02	0.02	-0.11	-0.10	-0.20	-0.07	-0.11	0.02	0.31	0.10	0.28	-0.03	0.04	-0.12	-0.18	-0.12	0.07	-0.42	0.02	0.14	0.33	0.06	-0.04	-0.04	0.43	0.09	0.15	0.13	0.18	0.30	0.00	-0.03	0.39	0.19	0.31	0.35	0.48	0.46	-0.19	0.01	-0.05	0.05	0.32	0.41	0.30	0.38	0.49	1.00	0.34
X52	-0.05	-0.24	0.14	0.43	0.44	-0.44	-0.43	-0.23	0.38	0.21	0.17	0.17	-0.31	0.71	-0.23	-0.18	-0.36	-0.27	0.34	0.11	-0.48	0.44	0.03	0.08	0.10	0.31	0.07	0.64	0.32	0.42	0.30	0.43	0.17	0.33	0.31	0.40	0.29	0.24	0.32	0.25	0.18	-0.05	0.11	-0.25	0.18	0.54	0.62	0.50	0.50	0.49	0.34	1.00

the correlation between all 52 variables is listed in this table.