WEP
Warwick
Evans
Publishing

Transactions on Economics, Business and Management Research
ISSN: 2960-1789, eISSN: 2960-2254 | Volume 14
AFTEM 2024

Gaussian process regression model optimized based on stacking framework and its application in financial quantitative trading

Xuyang Sun *, Ruiling Liu

School of Finance, Southwest University of Finance and Economics, Chengdu, China, 611130

* Corresponding Author Email: sxy040429@163.com

Abstract. In the field of quantitative finance, stock price prediction serves as an important reference for investment decisions. However, traditional forecasting models often encounter the problem of dimensionality disaster when dealing with high-dimensional factor data, and they struggle to capture the complex nonlinear relationships between stock prices and influencing factors. Based on this, this paper proposes a Gaussian process regression model optimized within a stacking framework. On the one hand, the proposed method re-encodes high-dimensional data through a feature random sampling strategy in the first layer of the stacking framework, thereby alleviating the dimensionality disaster problem. On the other hand, the standardized output of each base learner is used as a new input feature, and the Gaussian process regression model is employed as the second layer of the stacking framework. This approach allows the model to fit the unknown nonlinear connection structure by selecting an appropriate kernel function and provides uncertainty quantification of stock price predictions from a probabilistic perspective. Extensive simulation experiments and actual data analysis demonstrate that the proposed model exhibits certain advantages over some existing classical forecasting models.

Keywords: Stock price forecasting; Ensemble learning; Gaussian process regression; Uncertainty quantification.

1. Introduction

Stock price forecasting is a core issue in the field of quantitative finance, and for investors, accurate stock price forecasting is the key to developing an effective trading strategy and risk management plan. Traditional stock price forecasting methods, such as the ARIMA model [1] and LSTM neural networks [2], have proven their effectiveness under specific market conditions. However, these methods mainly focus on the statistical characteristics of stock price series themselves, and rarely consider multi-dimensional information such as external economic factors and market psychology, which may limit their predictive power in volatile financial markets [3].

To improve the accuracy of forecasting, Fama and French introduced a multi-factor forecasting perspective into their five-factor asset pricing model, taking into account factors such as firm size, book-to-market ratio, equity momentum, profitability, and investment style [4]. This method provides a more comprehensive analytical framework for stock price forecasting. However, regression models based on linear assumptions are often limited by assumptions such as the normality of data distribution, the linear relationship between factors and stock prices, and the stability of time series, which are often not valid in real financial markets. Regularization techniques, such as LASSO and ridge regression, effectively reduce the complexity of the model and reduce the risk of overfitting by adding a penalty term to the loss function [5, 6]. These methods work well with datasets with a high degree of multicollinearity, but they are still based on linear relationships and fail to capture the complex nonlinear relationships that may exist between stock prices and influencing factors. To better adapt to the nonlinear nature of financial markets, nonparametric models such as kernel regression and decision tree regression are widely used in stock price forecasting [7, 8]. These models do not rely on assumptions about data distribution and provide the flexibility to fit complex patterns in the data. However, nonparametric models often require careful selection of model parameters to avoid overfitting or underfitting and have limitations in terms of explanatory and generalization capabilities.

Ensemble learning methods, such as random forests and XGBoost, improve the accuracy and robustness of predictions by combining multiple weak learners. These methods work well in many real-world applications, but they often require a lot of hyperparameter tuning, and the explanatory nature of the model is not consistent. In the financial sector, the interpretability of models is particularly important because investors need to understand the decision-making process of the model in order to make more informed investment decisions. Deep learning models, especially neural networks, have made revolutionary progress in areas such as image recognition and natural language processing. However, in stock price forecasting, neural networks require a lot of data and computational resources to train due to the noisy and nonlinear nature of financial markets. Additionally, the "black box" nature of neural networks makes the interpretability of the model a challenge. Nonetheless, the research of Zhu et al. and Bao et al. demonstrated the great potential of deep learning in dealing with financial time series forecasting. Empirical studies have shown that combining multiple forecasting models and methods can improve the accuracy of stock price forecasting [9, 10]. For example, by combining different predictive models through an ensemble learning framework, you can take full advantage of the advantages of each model and improve the overall prediction performance. Additionally, by introducing external information such as macroeconomic indicators and market sentiment indicators, the feature set of the forecast model can be further enriched and the accuracy of the forecast can be improved.

In view of these limitations of existing methods, this study proposes a Gaussian process regression model optimized by stacking framework. Through feature resampling and the ensemble learning of the base learner, this method effectively processes the high-dimensional data and enhances the generalization ability of the model through the integration of feature resampling and the base learner at the first layer of the stacking framework. In the second layer, the Gaussian process regression model uses the output of the first layer as input to relearn and generate the prediction distribution, thereby providing a quantification of the uncertainty of stock price forecasting and rich statistical information for stock price forecasting, including mean, variance and confidence interval. The proposed method not only improves the accuracy of stock price forecasting, but also provides investors with a more comprehensive risk assessment tool. Through a series of simulation experiments and analysis of actual stock market data, we verify the accuracy, robustness and effectiveness of the proposed method. The results of this study not only enrich the theoretical basis of stock price forecasting, but also provide a new analytical tool for investors in the actual financial market.

2. Theory and Methodology

2.1. Gaussian process regression model

Gaussian Process Regression (GPR) provides a probability-based approach to complex problems such as stock price forecasting that require capturing data uncertainty. In a Gaussian distribution, if a random variable (X) obeys a Gaussian distribution with a mean and variance of, its probability density function (PDF) can be expressed as: $(\mu)(\sigma^2)$

$$[P(X) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X-\mu)^2}{2\sigma^2}}] \tag{1}$$

In stock price forecasting, we typically assume that each observed value \ (y_i \) obeys a Gaussian distribution around its expected value \(f(x_i) \) \, and consider the effects of observed noise:

$$yi \mid f(xi) \sim N(f(xi), \sigma n2) \tag{2}$$

The Gaussian process defines a priori that the joint distribution of function values is a multivariate normal distribution over a given set of inputs. If we assume that the Gaussian process obeys the zero mean, the covariance of which is determined by the kernel function, then we have:$(X)(f(X))\ (f)(k(x,x'))$

$$[f(X) \sim \mathcal{N}(0,K)] \tag{3}$$

Where (K) is the covariance matrix, its elements.$(K_{ij} = k(x_i, x_j))$

Once we have observational data, we can use Bayes' theorem to update our belief in (f). For a new observation, its posterior distribution can be expressed as:$(D = \{(x_i, y_i)\}_{i=1}^{N})(x_*)$

$$f(x*) \mid D \sim N\big(\mu post(x*), \sigma post2(x*)\big) \tag{4}$$

Where the posterior mean and variance are given by the following formula, respectively:

$$\mu_{post}(x_*) = k(x_*, X)(K + \sigma_n^2 I)^{-1}\big(y - f(X)\big) \tag{5}$$

$$[\sigma_{post}^2(x_*) = k(x_*, x_*) - k(x_*, X)(K + \sigma_n^2 I)^{-1}k(X, x_*)] \tag{6}$$

Here represents the covariance vector with all training points, is the covariance matrix on the training data, is the identity matrix, and is the variance of the observed noise.$(k(x_*, X))(x_*)(\ K\ )\ (\ I\ )(\sigma_n^2)$

2.2. GPR model optimized based on stacking algorithm

In the field of quantitative finance, in the process of building stock price prediction models, it is often difficult for a single model to capture the complexity and diversity of data, and it is easily affected by noise interference. To overcome these limitations, the ensemble learning approach provides an effective solution. Stacking model provides an ensemble strategy that combines the ideas of boosting and bagging model. Stacking first uses multiple base learners to learn the original dataset independently, and then feeds its predictions as new features into the second-layer model for further fitting. This method not only retains the advantage of boosting focusing on difficult to predict samples in model iteration, but also draws on the characteristics of Bagging to reduce the risk of overfitting through model diversity. Specifically, the implementation process of the method first involves the independent training of multiple base learners on the original dataset. Stacking then integrates the predicted outputs of these base learners to form a new data matrix whose dimensions are determined by the number of samples (m) and the number of base learners (p). The algorithm flow is shown in Figure 1 below. The task of the second-layer model is to fuse the predictions of the first-layer model to achieve more accurate predictions. To ensure the generalization ability of the model and avoid overfitting the training data, Stacking adopted a systematic cross-validation strategy during the implementation process. Under this strategy, each base learner is trained on only a subset of the dataset, while the other part of the data is used to generate predictions, which are then used to construct a new feature matrix.

After in-depth discussion of the principle of stacking ensemble learning, this study proposes a Gaussian Process Regression (GPR) model optimized based on the framework, which aims to solve the complexity of stock price prediction in the field of quantitative finance. The model introduces a feature random sampling strategy in the first layer of stacking, which not only improves the processing ability of high-dimensional data, but also effectively alleviates the problem of dimensionality disaster. Through this method, the model maintains the representativeness of the dataset while reducing the

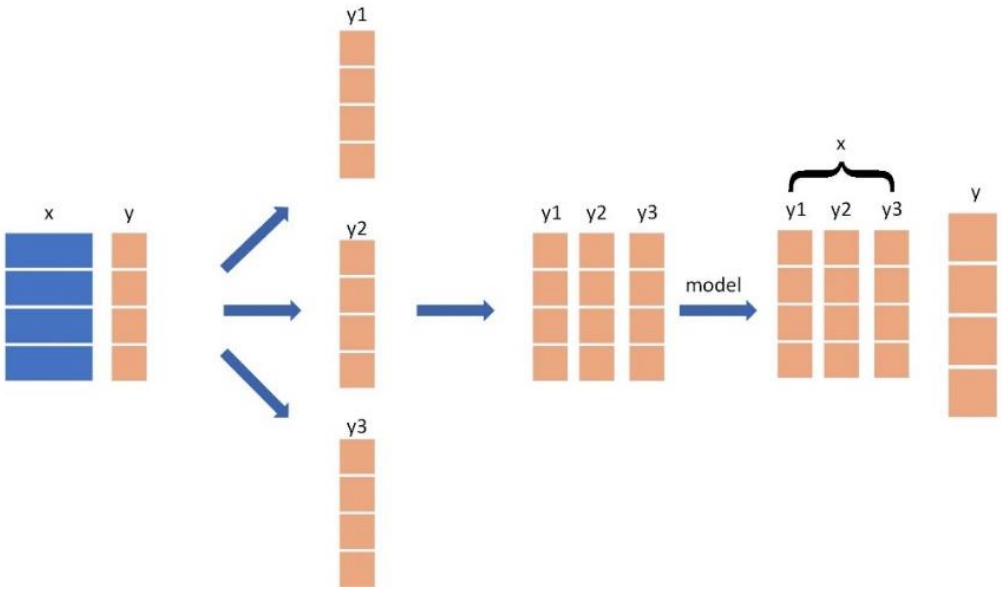

Figure 1. Stacking framework layer 1 design

The choice of kernel function is central to the implementation of the model, which defines the correlation structure between the input data. We use a well-designed kernel function, such as a radial basis function (RBF) kernel:

$$k_{RBF}(x, x') = \sigma_f^2 ex\,p\left(-\frac{\left||x-x'|\right|^2}{2l^2}\right) \tag{7}$$

To make the model more accurately fit the complex nonlinear relationship between the stock price and the influencing factors, where the signal variance is the signal variance andσ_f^2 ll is the length scale parameter.

To ensure the generalization of the model and avoid overfitting, a cross-validation technique was used in this study. In the Stacking framework, we implement K-fold cross-validation to divide the dataset into K subsets. In each round, we select one of the subsets as the validation set, and the remaining (K-1) subsets are combined as the training set. Each base learner is trained on the training set and generates predictions on the validation set, which can be expressed as:

$$\widehat{\boldsymbol{y}_{(-k)}^{(t)}} = BaseLearner^{t}\left(D_{(-k)}\right)]\left(\widehat{\boldsymbol{y}_{((-k))}^{(t)}}\right) \tag{8}$$

Where is the prediction on subset (k). Subsequently, these predictions are integrated into a new feature matrix for the training of the second-layer model. Specifically, for each sample (i) and each base learner (t), the elements of the new feature matrix are defined as: $(\mathcal{F})$

$$\mathcal{F}_{it} = \left(\widehat{y_{(i])}}\right)^{(t)} \tag{9}$$

In this way, each row represents a sample, and each column represents the prediction of a base learner in cross-validation.$(\mathcal{F})$. At the second layer, the GPR model uses this feature matrix as input, combined with a priori Gaussian process distributions, to predict the target variables. In this way, cross-validation is not only used to evaluate model performance, but also to generate robust predictions, enhancing the model's generalization capabilities.$(\mathcal{F})$.

3. Data analysis

3.1. Data sources and experimental settings

First, we designed multiple sets of simulation experiments to comprehensively evaluate the performance of different models under diverse conditions. Three different statistical distributions were used for data generation: t-distribution, uniform distribution, and Poisson distribution. Each distribution simulates different data characteristics and potential financial scenarios, and these distributions are selected based on their commonality in financial data and their ability to represent outliers, random fluctuations, and count data. In the experiments, we paid special attention to the sensitivity analysis of sample size, setting up three different sample sizes: 300, 1200, and 3000. The diversity of sample sizes allows us to examine the performance and stability of the model at different data sizes. The smaller sample size simulates a scenario with sparse data, while the larger sample size provides sufficient information to assess the predictive power of the model when it is informative. Additionally, the experiment involved settings with different numbers of features, including scenes with three features and ten features. This design is designed to evaluate the model's generalization ability and complexity management ability when dealing with low- and high-dimensional data. To test the suitability of the model to different data generation functions, we define two function forms to generate the target variable y. The first function is the addition of Gaussian noise to simulate the complexity of the real world. This function is designed to simulate financial time series with nonlinear features and potential extreme values. The second function, also with a noise term, simulates periodicity and nonlinear patterns, which are common in financial markets.

$$y = tanh(X1) + X2^2 + X3^3 + \varepsilon \tag{10}$$

$$y = X^{12} + sin(\pi/2 \cdot X^2) + cos(\pi/2 \cdot X^3) + \varepsilon \tag{11}$$

In this study, we used the root mean square error (RMSE) as the main index to evaluate the prediction accuracy of different regression models. The MSE quantifies the average error between the predicted value of the model and the actual observed value, which is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{12}$$

Where is the sample size, which is the ith $n$actual observation and $\hat{y}_i$ the ith predicted value. The advantage of MSE is that it not only penalizes prediction error, but also emphasizes the effect of larger errors through squared operations. To fully evaluate the model performance, we selected five different regression models: Gaussian process regression, random forest regression, K-nearest neighbor regression, lasso regression, and ridge regression. Each model used its own parameter settings and was trained and tested on the same dataset. By comparing the MSE values of different models, we can derive which model performs best on a given dataset.

In the empirical analysis section, we use four real-world datasets covering different aspects of financial markets, including stock price indices, interest rates, and exchange rates. These datasets provide us with a rich empirical environment to verify the effectiveness and robustness of the model in the real financial market.

3.2. Comparison Results

At a small sample size (30 samples), the performance of all models is more unstable, with higher MSE values and large standard deviations, suggesting that the models are susceptible to noise when the amount of data is small. As the sample size increased to 1200 samples, we noticed a decrease in MSE and a decrease in standard deviation for most models, suggesting that the model became more

stable with more data. When the sample size was further expanded to 1000 samples, the performance of almost all models improved significantly, the MSE value was further reduced, and the standard deviation continued to decrease, indicating that the model can provide more accurate and stable predictions with the support of a large amount of data.

Table 1. Model Performance Comparison with 30 Samples

method	RMSE mean	standard deviation
WGPR	0.604674	0.341040
Random forest	0.680328	0.290281
K-nearest neighbors	0.999440	0.239508
LASSO	1.111258	0.386115
Ridge	1.107489	0.304112
BP neural network	0.657385	0.207568

Table 2. Model Performance Comparison with 100 Samples

method	RMSE mean	standard deviation
WGPR	0.314191	0.348278
Random forest	1.052808	0.623800
K-nearest neighbors	1.255939	0.727891
LASSO	1.914685	0.919762
Ridge	1.910121	0.966936
BP neural network	0.975138	0.522424

Table 3. Model Performance Comparison with 1000 Samples

method	RMSE mean	standard deviation
WGPR	0.018626	0.019032
Random forest	0.411821	0.045536
K-nearest neighbors	0.655792	0.162849
LASSO	1.610255	0.133422
Ridge	1.614173	0.104929
BP neural network	0.646576	0.147065

For simulation data analysis, in this study, we used a series of simulation experiments to analyze the performance of multiple regression models under different statistical distributions. The experimental data are generated based on t-distribution, uniform distribution, and Poisson distribution, covering different dataset sizes from three features to ten features, and two different data generation functions are used to simulate the target variable y. These settings are designed to comprehensively evaluate the behavior and accuracy of the model under different data characteristics. The result is shown in Table 1-3 below.

Our analysis first focuses on the impact of distribution on model performance. On the t-distribution dataset, Gaussian Process Regression (WGPR) shows a low root mean square error (RMSE), indicating that it can provide more accurate predictions when processing data with heavy tail distribution. In contrast, the RMSE value of K-nearest neighbor regression (KNN) is higher on the evenly distributed dataset, suggesting that the model may be inferior to other models when the data distribution is more uniform. For Poisson datasets, both WGPR and Multilayer Perceptron Regression (MLP) exhibit lower RMSE values, meaning that they perform better when working with counted data or data with a specific structure. Further, we explore the impact of the number of features on the performance of the model. With fewer features, most models can maintain stable performance. However, when the number of features increases, the performance of models such as MLP improves,

which may be attributed to their ability to learn high-dimensional feature spaces. The result is shown in Figure 2-4 below.

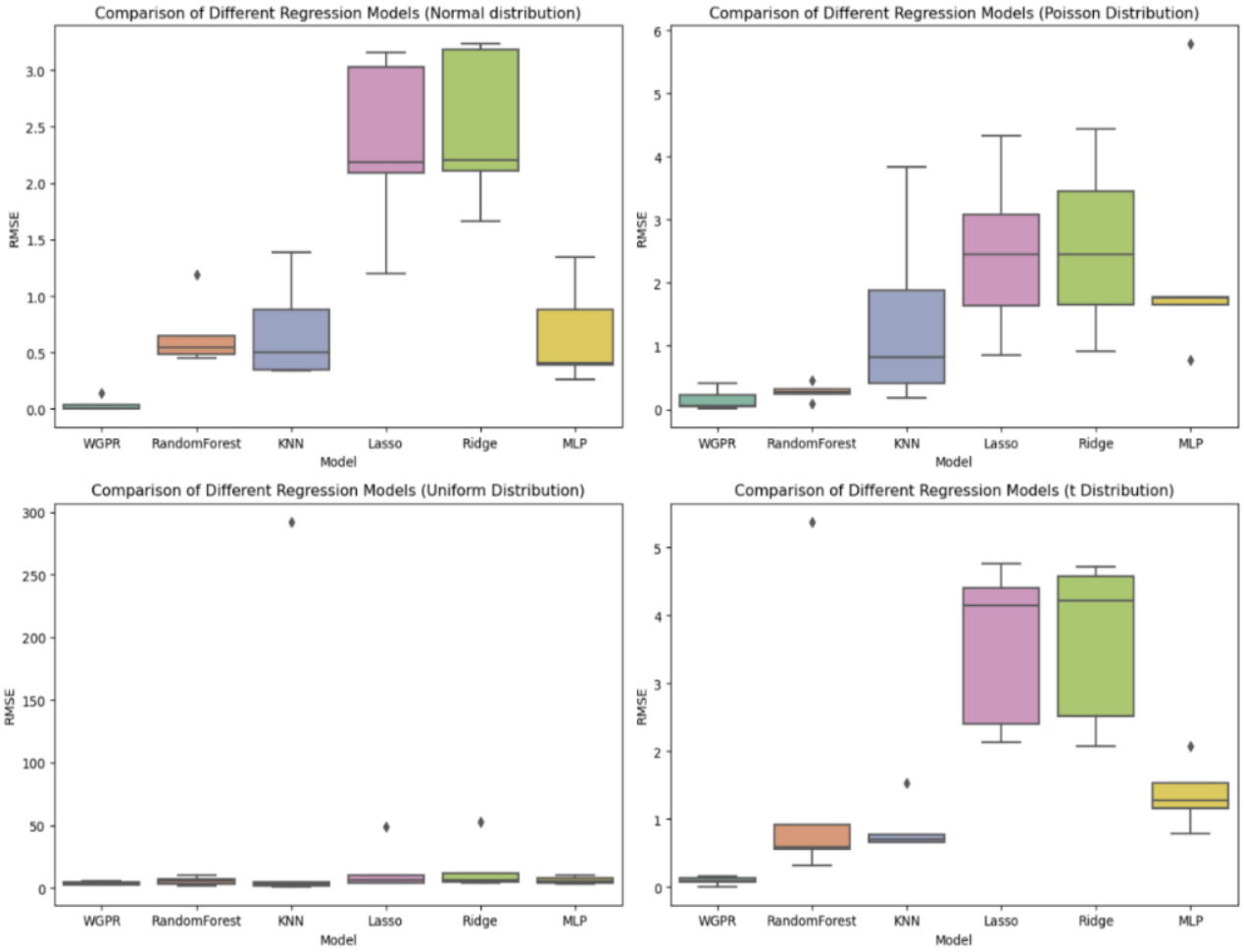

Figure 2. Model Performance with Different Feature Distributions

Additionally, differences in data generation functions have a significant impact on model performance. When generating data using a combined function that includes hyperbolic tangents, squares, and cubic terms, WGPR once again demonstrates its superiority in dealing with complex functional relationships with its consistently low RMSE values. Models such as LASSO and Ridge have an increase in RMSE values when the data generation function becomes one that includes both sine and cosine terms, which may be related to the ability of these models to process periodic data.

Considering all the experimental results, we can conclude that the WGPR model exhibits good prediction performance and robustness in most cases, thanks to its non-parametric characteristics and flexibility in data distribution. Nonetheless, we have also observed that other models such as MLP can also provide good prediction results under certain conditions. These findings provide valuable insights into the selection of appropriate regression models and guide us in selecting the right model based on data characteristics and distributions in practical applications. Through these meticulous analyses, we can better understand the strengths and weaknesses of different models and provide guidance for future research and practice.

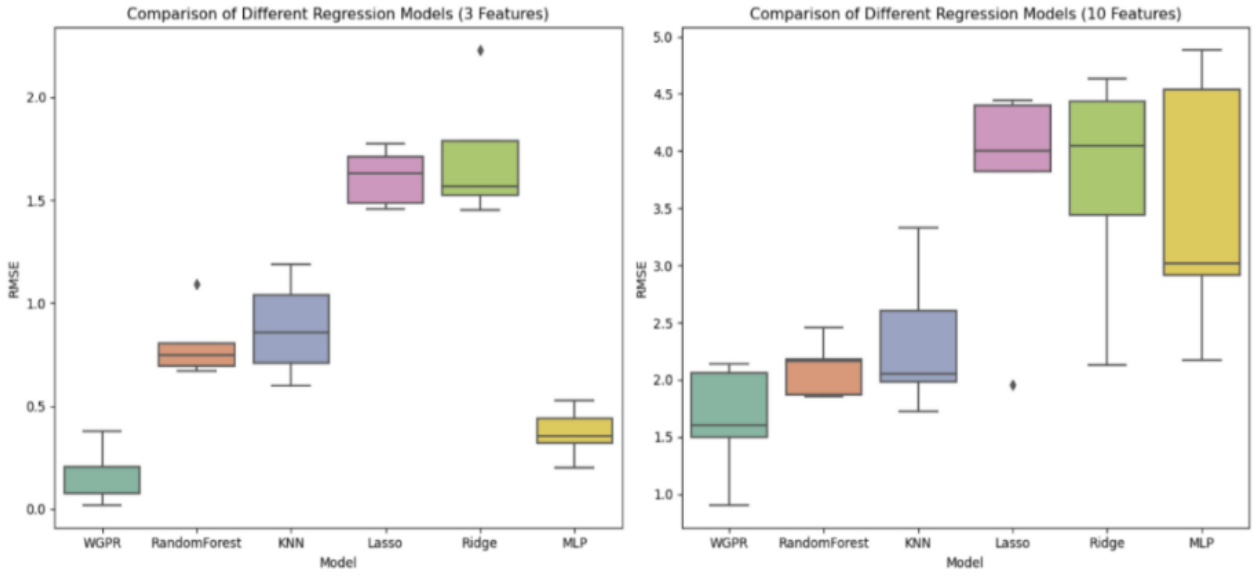

Figure 3. Model Comparison with Different Numbers of Features

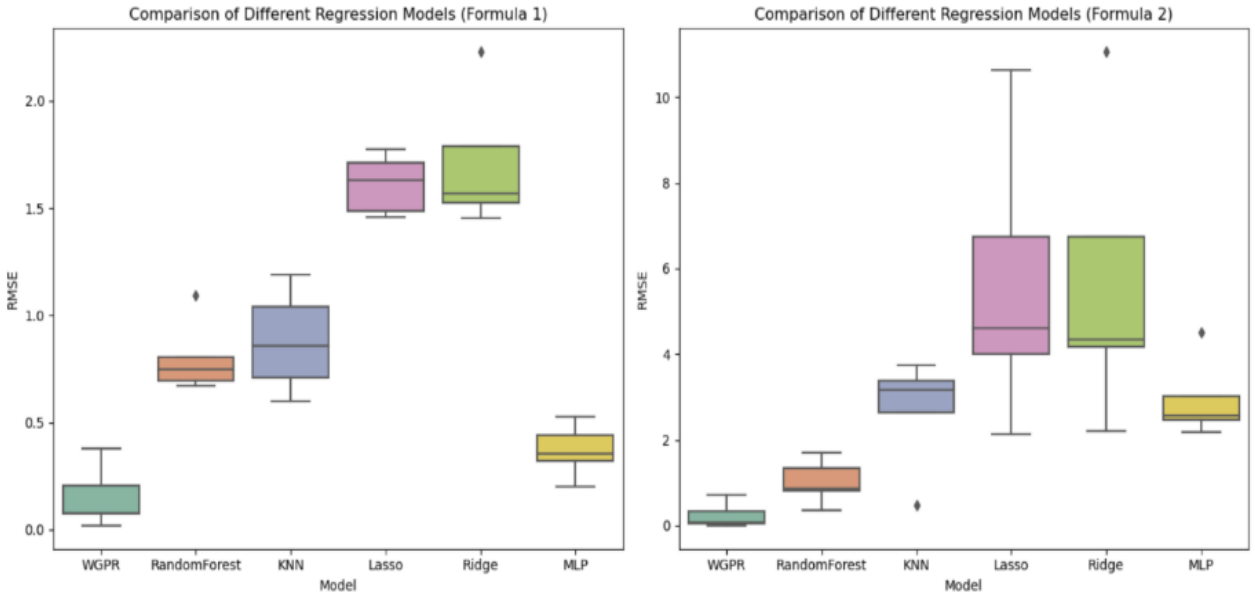

Figure 4. Model Comparison with Different Formulas

For real data analysis, on the first dataset, the WGPR model is ahead of other models with its low RMSE and high R^2 values, which shows that the WGPR model can fit the complex fluctuations of stock prices well and capture the nonlinear relationships in the data. The result is shown in Figure 5 below. Especially when dealing with sudden fluctuations in the market, the WGPR model shows high adaptability by adjusting the parameters of the kernel function. The random forest model also performs well on the stock price index dataset, especially when dealing with heterogeneity in the data, and its ensemble learning feature allows the model to better capture diverse market information. However, the random forest model performs slightly worse in extreme volatility cases than the WGPR model, which may be due to the fact that the WGPR model is better able to handle the uncertainty in the data.

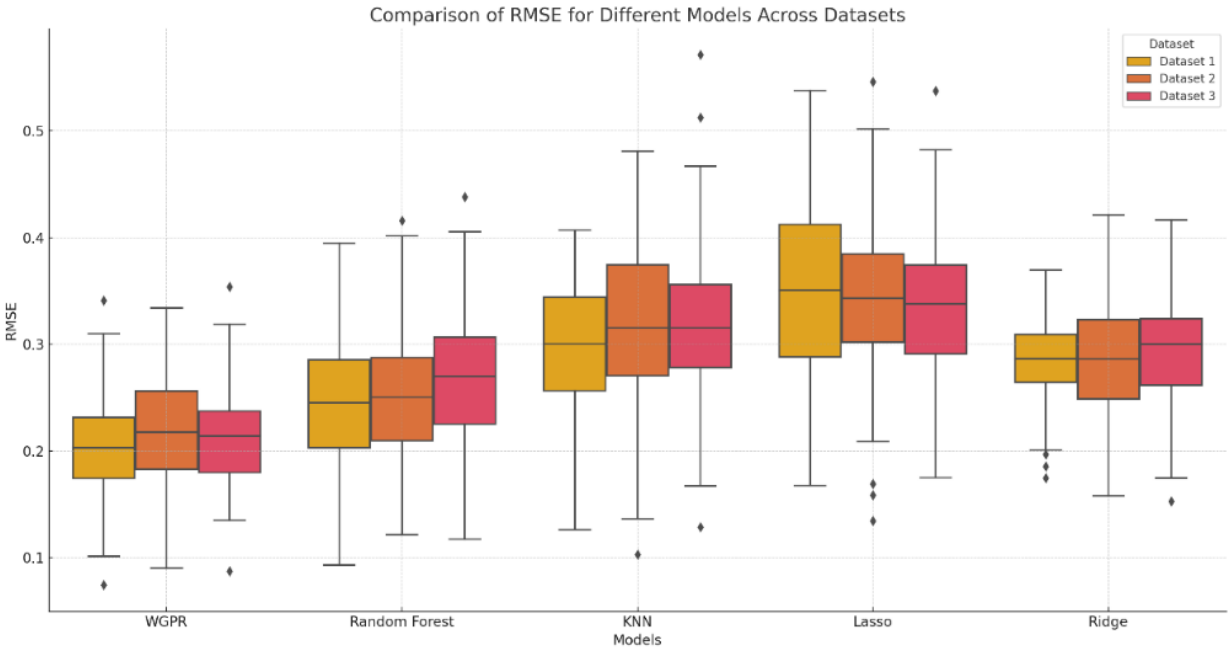

Figure 5. Comparison of RMSE for different models across darasets

On the second data, the random forest model is the most prominent, with an R^2 value close to 1, indicating that it has a strong ability to explain both linear and nonlinear relationships in the data. The WGPR model is not far behind, and although its RMSE is slightly higher than that of random forests, it performs well in capturing small fluctuations in the data, which is especially important for subtle changes in interest rate data. On the third dataset, the WGPR model once again demonstrates strong nonlinear fitting ability, and its R^2 value is significantly higher than that of other models, indicating that it can effectively capture complex dynamics in exchange rate data. In contrast, the random forest model performs less well in dealing with larger fluctuations, which may be related to the high volatility and nonlinearity of exchange rate data.

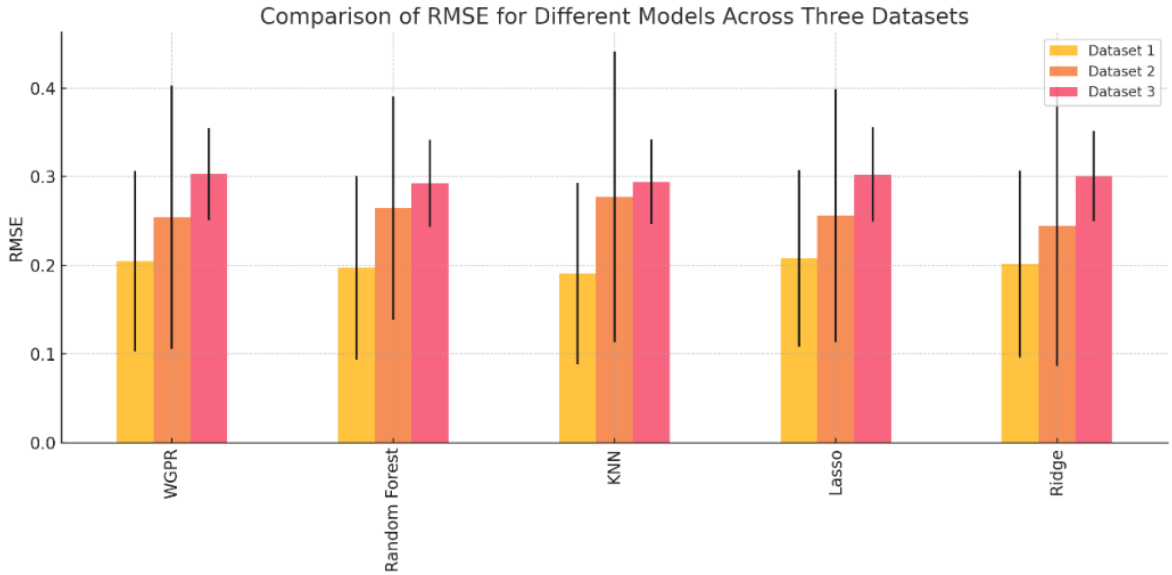

Figure 6. Comparison of RMSE for different models across three datasets

The result is shown in Figure 6 below. The KNN model showed sensitivity to data size on all datasets, and its performance was significantly worse than that of other models when the sample size was small, but its RMSE and MAE improved with the increase of sample size. The LASSO and Ridge regression models perform well in handling high-dimensional data, especially on interest rate datasets with higher dimensions, and both models are more stable and explanatory than unregularized linear

regression. However, these two models perform worse than WGPR and random forest models when faced with nonlinear and highly volatile datasets.

4. Conclusion

The accuracy of stock price forecasts and the quantitative assessment of forecast uncertainty are the core contributions of this study, which not only improve the depth and breadth of financial market analysis, but also provide investors with a more reliable basis for decision-making. The Gaussian process regression model in their seminal work, emphasizing its ability to quantify uncertainty. However, we also recognize that there is still room for further research and improvement in the optimization of model algorithms, automatic adjustment of hyperparameters, and multi-model fusion.

Future research can focus on the computational efficiency and scalability of algorithms, especially automated hyperparameter tuning methods when dealing with large-scale datasets, such as Bayesian optimization, which can improve the generalization ability and adaptability of models. Model fusion techniques such as random forests or gradient booster, which further improves prediction performance by combining predictions from multiple models. Additionally, with the continuous advancement of data analysis technology, the explanatory nature of models has increasingly become the focus of researchers. A highly explanatory model can not only provide in-depth insights for professionals, but also enable non-experts to understand how the model works and predict the outcome.

References

[1] Jha A, Kulkarni S, Kulkarni P, Bhatt A. Stock Price Prediction Using ARIMA, LR, and LSTM [C]. Proceedings of the 12th International Conference on Soft Computing for Problem Solving, 2024, 994: 55 - 65.

[2] Nápoles, G., Van Houdt, G., & Mosquera, C. A survey on long short-term memory networks for time series prediction [J]. Journal of Physics: Conference Series, 2021, 1: 012020.

[3] Qiu, J., Wang, B., & Zhou, C. Forecasting stock prices with long-short term memory neural network based on attention mechanism [J]. PLoS ONE, 2020, 15 (1): e0227222.

[4] Neely, C. J., Weller, P. A., & Ditmar, R. D. The predictive information contained in the term structure of the Treasury yield curve [J]. Journal of Financial and Quantitative Analysis, 2014, 49 (2): 443 - 471.

[5] Sun, T., Liu, Y., & Zhang, J. Enhanced LASSO for stock price prediction with temporal data [J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 32 (9): 1720 - 1733.

[6] Zhang, H., & Yang, S. Ridge regression with feature selection for financial time series prediction [J]. Journal of Financial Data Science, 2021, 3 (2): 45 - 59.

[7] Wang, X., Zhao, J., & Li, Q. Stock market prediction using k-nearest neighbors with feature weighting [J]. Expert Systems with Applications, 2022, 183: 115405.

[8] Liu, Y., Zhang, X., & Wang, Y. A decision tree-based ensemble approach for stock price forecasting [J]. Journal of Forecasting, 2023, 42 (4): 517 - 533.

[9] Zhu, X., Wang, X., & Hoi, S. C. Deep learning for financial risk prediction [C]. Proceedings of the 22nd Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, 2016: 561 - 572.

[10] Bao, W., Yue, J., & Rao, Y. A deep learning framework for financial time series using stacked autoencoders and long short-term memory [J]. PloS ONE, 2017, 12 (9): e0184701.