Academic Journal of Natural Science
Journal Home: ajns.suaspress.org | CODEN: AJNSAE | NAAN: 40704
Vol. 1, No. 1, 2024

A Review of Feature Engineering Methods in Regression Problems

HE, Runhai [1*] LI, Boyi [2] LI, Fusheng [1] SONG, Qingqing [1]

[1] Belarusian State University, Belarus
[2] ITMO University, Russia

**** HE, Runhai is the corresponding author, E-mail: fpm.he@bsu.by***

Abstract: This paper comprehensively reviews the importance of feature engineering in regression problems and its existing methods. Different feature processing techniques, including data transformation, interactive feature generation, dimensionality reduction, and technical indicators, are analyzed, and some existing research results are combined to discuss how these techniques affect the predictive ability of the model. By systematically summarizing the advantages and disadvantages of various methods, this review provides researchers and practitioners with a comprehensive understanding of existing feature engineering techniques. In addition, this paper also explores the existing research gaps and possible future development directions, and proposes more application scenarios and research topics for related technologies.

Keywords: Data Preprocessing, Feature Selection, Feature Scaling, Deep Learning.

DOI: https://doi.org/10.5281/zenodo.13905622 **ARK:** https://n2t.net/ark:/40704/AJNS.v1n1a06

1 INTRODUCTION

Regression analysis is a core task in statistics and machine learning. It is widely used in finance, medicine, economics and other fields to predict continuous variables. As the diversity and complexity of data continue to increase, simple regression models often cannot provide sufficient prediction accuracy. At this time, feature engineering becomes a key technology to improve the performance of regression models. Feature engineering optimizes the input of the model by processing raw data, constructing new features, and selecting the most effective features, thereby improving the prediction effect.

Feature engineering refers to the process of extracting, selecting and transforming features from raw data to create more representative and predictive input variables. Feature engineering can directly affect the performance of regression models. Even under the same model and algorithm, suitable features can significantly improve the predictive ability and generalization of the model. In the context of big data and complex models, the application of feature engineering can reduce the risk of overfitting of the model and improve the interpretability of the model [1]. Especially in the fields of financial market prediction and medical diagnosis, feature engineering helps to discover implicit relationships that are crucial to prediction tasks.

This paper aims to provide a comprehensive review of feature engineering techniques in regression tasks, covering from traditional methods to the latest deep learning applications. This paper systematically explores core technologies such as feature selection, feature extraction, feature construction, and feature scaling, and focuses on the application of these technologies in specific fields such as finance. By combing through existing research results, it is hoped that a comprehensive overview of feature engineering technology will be provided for researchers and practitioners, while also pointing out possible future research directions.

2 OVERVIEW

The main goal of feature engineering is to minimize high-dimensionality, noise or redundant information, and to highlight the features with the most predictive value for regression tasks by optimizing data representation [1]. In regression tasks, good feature engineering can usually significantly improve the prediction accuracy and generalization ability of the model. Traditional feature engineering methods rely on statistical theory, such as feature scaling and transformation in multivariate linear regression. These methods help the model better fit the data structure by standardizing, normalizing or logarithmically transforming the data. In modern regression analysis, feature engineering begins to combine more automation technology and deep learning to deal with complex data structures (such as time series, images, and text).

Traditional feature scaling techniques are unsupervised. Its application of min-max normalization can significantly improve the performance of regression models such as multivariate linear regression by standardizing the data range.

However, this scaling process is not affected by the dependent variable. Niful Islam proposed a new feature scaling technique called DTization in his experiment. This technique uses decision trees and robust scalers for supervised feature scaling. The results show that the performance of this method is significantly improved compared with traditional feature scaling methods [2].

The application of deep learning in feature selection has opened up new directions for regression analysis. This type of method uses the nonlinear modeling capabilities of neural networks to automatically learn complex patterns and feature interactions in data. For example, autoencoders can discover the hidden structure of data by reconstructing the input, thereby identifying important features. In their study on representation learning, Bengio et al. (2013) proposed a method for learning hierarchical representations of data using deep networks. This method performed well in image recognition tasks and successfully extracted high-level features automatically from raw pixel data, greatly improving the accuracy of subsequent classification models. Experimental results show that compared with traditional hand-crafted features, this deep learning-based feature selection method reduces the error rate by more than 30% on the MNIST dataset, fully demonstrating its advantages in complex data processing [3].

Gu et al. (2020) proposed an improved LSTM model for financial market price prediction by combining a set of technical indicators (moving average, relative strength index, and Bollinger bands, etc.) with high-frequency trading data. They used an autoregressive recurrent network and modeled the price prediction problem as a regression problem rather than a binary classification problem. The results show that this method significantly outperforms the previous LSTM model based on a simple recurrent network and exhibits excellent performance in price direction prediction. In contrast, the traditional GARCH model performs worse than this improved model, which shows that the combination of technical indicators and high-frequency data can effectively enhance the model's predictive ability [4].

These cases demonstrate the importance and diverse applications of feature engineering in regression analysis. From traditional feature scaling to deep learning and automated methods, various techniques can help improve the model's predictive performance.

3 METHODOLOGY

3.1 Feature Selection

3.1.1 Filter Methods

Filtering methods are a feature selection technique that is independent of learning algorithms and are known for their simplicity and computational efficiency. This method evaluates the importance of each feature based on a predefined statistical measure, such as correlation, mutual information, variance, or chi-square test. The core idea of filtering methods is to assign an importance score to each feature, then rank the features according to these scores and select the subset of features with the highest ranking. The main advantages of this method are its fast computational speed, strong scalability, and its particular suitability for high-dimensional datasets [5]. However, filtering methods also have their limitations, mainly because they may ignore the interactions between features because they usually evaluate each feature individually. Common filtering methods include Pearson correlation coefficient, Fisher score, information gain, and relief algorithm. Despite their simplicity, filtering methods remain an effective feature selection strategy in many practical applications, especially as a preprocessing step for more complex methods.

The commonly used correlation coefficient (Pearson correlation coefficient) calculation formula:

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}}$$

Where, X_i and Y_i are the feature and target variables respectively, and $\bar{X}$ and $\bar{Y}$ are their means.

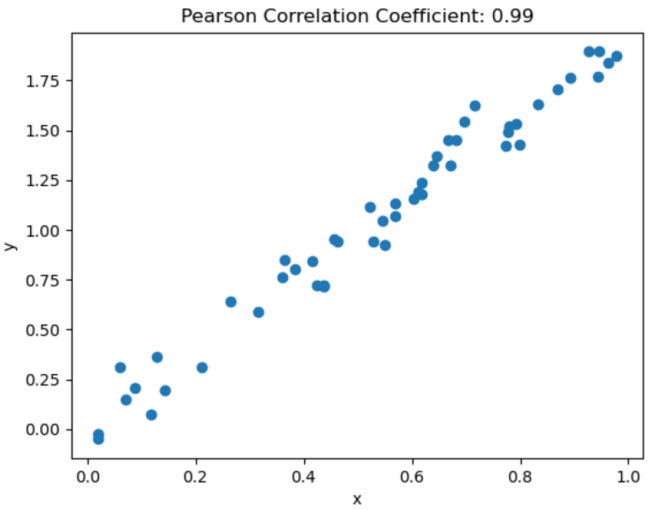

FIGURE 1. PEARSON CORRELATION EXAMPLE

The figure 1 shows an example of Pearson Correlation calculation, in this case, there is a very high correlation between x and y.

3.1.2 Wrapper Methods

Wrapper methods are a class of feature selection techniques that use a predefined learning algorithm to evaluate the quality of feature subsets. The core idea of this method is to "wrap" the feature selection process around the learning algorithm and search for the best feature combination by repeatedly training the model. Wrapper methods usually adopt heuristic search strategies such as forward selection, backward elimination, or recursive feature

elimination (RFE) to explore the feature space and find the subset with the best performance [6]. The main advantage of this method is that it can capture the interaction between features and select the feature subset that best suits a specific learning algorithm. However, wrapper methods are computationally expensive, especially when dealing with high-dimensional datasets, because it requires multiple training and evaluation of the model. In addition, wrapper methods may face the risk of overfitting, especially when the sample size is small. Despite this, in many applications, wrapper methods can still produce excellent feature subsets, especially when model performance is the primary consideration.

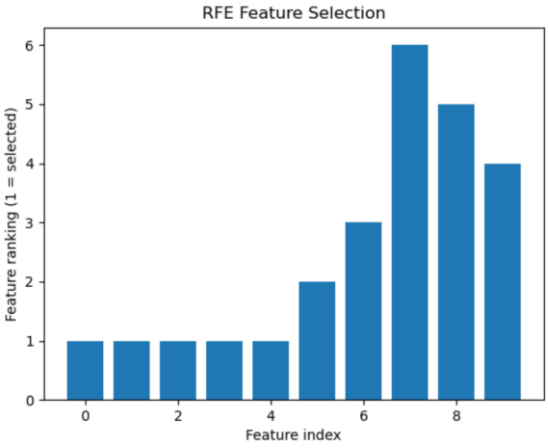

FIGURE 2. RFE FEATURE SELECTION EXAMPLE

The figure 2 shows an example of Recursive Feature Elimination Feature Selection. In this case, feature 6 has the highest ranking, indicating that it's the most important, while features 0 through 4 have lower rankings, suggesting less significance in the model.

3.1.3 Embedded Methods

Embedded methods integrate the feature selection process directly into model training, attempting to optimize both model performance and feature subsets in a unified framework. The core idea of this type of method is to perform feature selection within the structure of the learning algorithm, usually as part of the model building process. Embedded methods can capture the interactions between features and are more efficient than wrapper methods because they complete feature selection in a single model training process. Common embedded methods include Lasso regression with L1 regularization, elastic nets, and tree-based methods (such as feature importance of random forests). These methods achieve feature selection by introducing penalty terms in the objective function or directly considering feature importance in the model structure. The advantage of embedded methods is that they can achieve a good balance between model performance and the number of features, while the computational efficiency is usually between filtering methods and wrapper methods. However, a potential disadvantage of embedded methods is that they may be limited by specific learning algorithms [7].

The regularized form of Lasso regression:

$$\min\left(\frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}|w_j|\right)$$

Where, λ is a regularization parameter used to control the importance of features. The tree model selects key features by the feature importance score when splitting nodes.

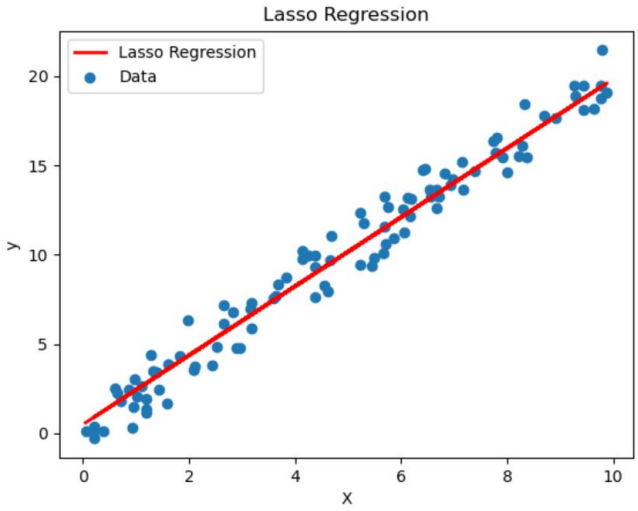

FIGURE 3. LASSO REGRESSION EXAMPLE

The Figure 3 shows the result of Lasso Regression example, where the red line represents the fitted regression line, and the blue points represent the data. The figure demonstrates that Lasso successfully fits a linear model to the data.

3.1.4 Feature selection combined with deep learning

The application of deep learning in feature selection has opened up new directions for regression analysis. Such methods use the nonlinear modeling capabilities of neural networks to automatically learn complex patterns and feature interactions in data. For example, autoencoders can discover the hidden structure of data by reconstructing the input, thereby identifying important features. The representation learning method proposed by Bengio et al. (2013) shows how to use deep networks to learn hierarchical representations of data and then select the most discriminative features. In practice, these methods can often capture complex feature relationships that are difficult to discover with traditional techniques, thereby improving the prediction accuracy of regression models [3].

3.2 FEATURE EXTRACTION

3.2.1 Principal Component Analysis (PCA)

Principal component analysis is a dimensionality

reduction technique that is often used to deal with multicollinearity problems in regression analysis. PCA converts the original features into a set of linearly uncorrelated new features (principal components) by finding the main direction of variation in the data. In the context of regression, PCA can identify the feature combination that best explains the variation of the dependent variable. This method is particularly suitable for processing high-dimensional data and can effectively reduce the number of features while retaining most of the information. The core idea of PCA is to maximize the projected variance. Its main steps include data centering, calculating the covariance matrix, solving eigenvalues and eigenvectors, and selecting principal components [8]. Although PCA can improve the stability and explanatory power of the model, it may lead to a decrease in the interpretability of the features. In regression analysis, principal component scores can be used as new predictor variables to build a more robust regression model.

The core formula of PCA:

$$PC_i = a_{i1} * X_1 + a_{i2} * X_2 + \cdots + a_{in} * X_n$$

Where PC_i is the i-th principal component, a_{ij} is the element of the eigenvector, and X_j is the original feature.

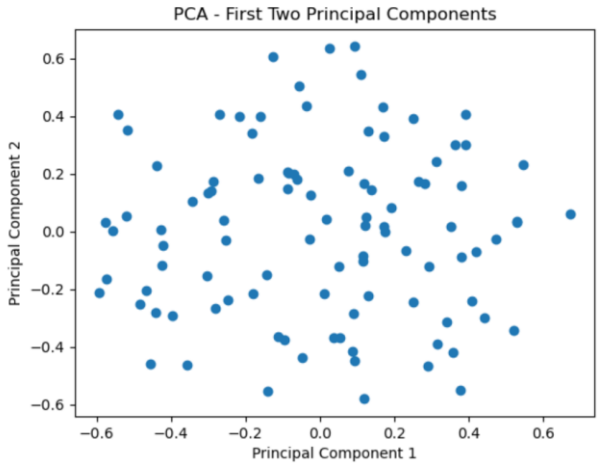

FIGURE 4. PCA OF FIRST 2 COMPONENTS EXAMPLE

The Figure 4 illustrates Principal Component Analysis, showing a scatter plot of the first two principal components. The plot highlights the variance captured by the two components, displaying the spread of the data in terms of these new features.

3.2.2 Independent Component Analysis (ICA)

Independent component analysis is a technique used in regression analysis to handle signal separation and feature extraction. Unlike PCA, the goal of ICA is to find statistically independent latent variables or source signals, rather than just uncorrelated components. In regression problems, ICA can identify potential independent predictors that may have a more direct impact on the dependent variable. ICA assumes that the observed data is a linear mixture of unknown source signals and separates these signals by maximizing the non-Gaussianity of the source signals. When applying ICA in regression analysis, the separated independent components can be used as new predictor variables, which may reveal complex relationships that are difficult to capture with traditional methods. However, the results of ICA can be difficult to interpret and are sensitive to initial conditions [9].

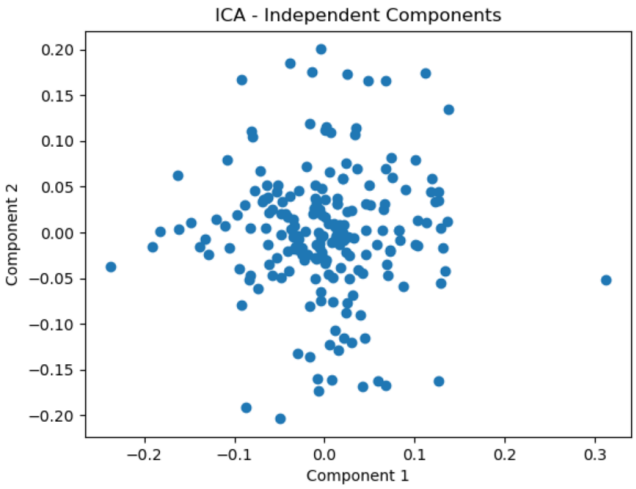

FIGURE 5. ICA OF 2 COMPONENTS EXAMPLE

The Figure 5 shows data projected onto two independent components using Independent Component Analysis (ICA). The data points are scattered without any apparent clusters, suggesting independence between components.

3.2.3 Linear Discriminant Analysis (LDA)

Although linear discriminant analysis is mainly used for classification, it also has applications in regression analysis, especially when dealing with categorical predictor variables. The core idea of LDA is to find a linear combination that makes data of different categories have the greatest separability after projection. In the context of regression, LDA can be used to create new features that maximize the differences between categories while minimizing the differences within categories [10]. This is particularly useful for dealing with multi-category predictor variables or exploring the relationship between dependent variables and categorical independent variables. LDA assumes that the data follows a multivariate normal distribution and that each category has the same covariance matrix. In regression analysis, the results of LDA projection can be used as new predictor variables, which may improve the predictive ability and explanatory power of the model.

The core formula of LDA:

$$W = \sum_{w}^{-1} (\mu 1 - \mu 2)$$

Where w is the weight vector, $\sum w$ is the within-class covariance matrix, μ1 and μ2 are the mean vectors of the two classes

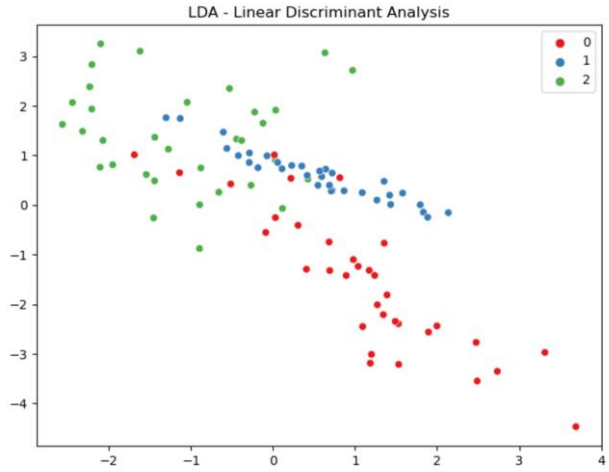

FIGURE 6. LDA EXAMPLE

The Figure 6 shows a classification result where data points are projected onto a linear discriminant axis, effectively separating them into three distinct classes (red, blue, green), showing clear class separability.

3.2.4 Feature extraction combined with deep learning

Deep learning has demonstrated outstanding capabilities in the field of feature extraction, opening up new avenues for regression analysis. Convolutional neural networks (CNNs) and autoencoders are two widely used deep learning architectures, each with its own characteristics in feature extraction.

CNN performs well when processing structured data such as images or time series. In regression analysis, CNN can automatically learn hierarchical feature representations, from low-level features (such as edges, textures) to high-level features (such as complex patterns). The convolutional and pooling layers of CNN can effectively capture local correlations and spatial invariance, which are essential for many regression tasks. For example, in house price prediction, CNN can extract key features from house pictures, such as architectural style, surrounding environment, etc., which may have a significant impact on prices.

Autoencoders focus on learning compressed representations of data. In regression problems, autoencoders can be used for dimensionality reduction and feature learning. It learns latent features by trying to reconstruct the input data, which often capture the essential structure of the data. For high-dimensional regression problems, autoencoders can learn a low-dimensional, information-rich feature space that may be more suitable for regression analysis than the original features. For example, in financial forecasting, autoencoders can extract key factors from a large number of economic indicators that may have a significant impact on asset price trends.

3.3 FEATURE CONSTRUCTION

3.3.1 Categorical Encoding

Encoding categorical variables is a key step in dealing with non-numerical features in regression analysis. Commonly used methods include one-hot encoding and label encoding. One-hot encoding creates a new binary feature for each category and is suitable for situations where there is no order between categories; label encoding maps categories to integers and is suitable for ordered categories. The target encoding method proposed by Micci-Barreca (2001) can effectively handle high-cardinality categorical variables by replacing categories with the conditional mean of the target variable. These encoding methods have their own advantages and disadvantages, and the selection should consider data characteristics and model requirements [11].

3.3.2 Missing Value Handling

Missing value processing is crucial in regression analysis. Improper processing may lead to model bias. Common methods include simple imputation (such as mean and median imputation) and advanced techniques (such as multiple imputation and K nearest neighbor imputation). Little and Rubin (2002) systematically studied the missing data mechanism and proposed strategies for processing MCAR, MAR and MNAR data [12]. In practice, the MICE (Multiple Imputation Chained Equations) method is popular because of its flexibility and adaptability to complex missing patterns. Choosing an appropriate missing value processing method requires a trade-off between computational efficiency, data structure preservation and model performance.

The formula of MICE method:

$$P(X_{miss} \mid X_{obs}, \theta) = P(X_1 \mid X_{-1}, \theta_1) * \ldots * P(X_p \mid X_{-p}, \theta_p)$$

Where X_{miss} is the missing data, X_{obs} is the observed data, and θ is the model parameter.

3.3.3 Interaction Features

Interaction features capture the nonlinear relationship between variables in regression analysis and enhance the expressiveness of the model. Common methods include polynomial features, tree-based methods (such as feature importance of random forests), and automatic feature engineering tools. The MARS (Multivariate Adaptive Regression Spline) model proposed by Friedman (1991) can automatically discover important feature interactions [13]. In practical applications, combining domain knowledge and data-driven methods to construct interaction features can often significantly improve model performance. However, excessive use of interaction features may lead to overfitting, and a careful balance needs to be struck between model complexity and generalization ability.

The basic form of the MARS model:

$$f(X) = \beta_0 + \sum_m \beta_m h_m(X)$$

Where, β_0 is the intercept, β_m are the coefficients, $h_m(X)$ and are the basis functions. This formula essentially builds a piecewise linear model to fit the data. Very flexible, very powerful.

3.3.4 Technical indicators in financial data

In financial analysis, technical indicators are widely used to predict stock prices, foreign exchange and other assets. The following will focus on several common technical indicators and their calculation methods.

Moving Averages: Calculate the mean value in n time windows

$$MA_t = \frac{1}{n}\sum_{i=0}^{n-1} P_{t-i}$$

Where MA_t is the moving average at time t, n is the time window size, and P_{t-i} is the price at time t-i

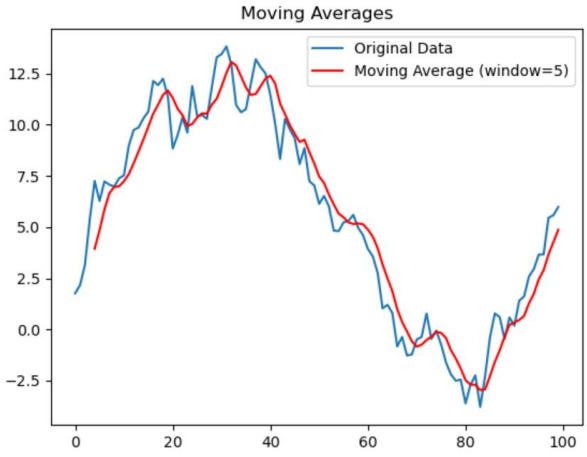

FIGURE 7. MOVING AVERAGE EXAMPLE

The figure 7 presents a time series with a moving average smoothing (window = 5). The red line represents the smoothed trend, while the blue line shows the original fluctuating data.

Relative Strength Index (RSI): Measuring whether an asset is overbought or oversold

$$RSI = 100 - \frac{100}{1 + RS}$$

Where, RS is the ratio of the recent average increase to the average decrease.

RSI values range from 0 to 100:

• Generally, RSI > 70 is considered an overbought signal

• RSI < 30 is considered an oversold signal

MACD indicator: measures trend by the difference between fast- and slow-moving averages

$$MACD = EMA_{12} - EMA_{26}$$

Where, EMA_12 and EMA_26 are the 12-day and 26-day exponential moving averages respectively.

Note that MACD is often used in conjunction with a signal line (usually the 9-day EMA of the MACD) to generate buy and sell signals:

MACD line crossing above the signal line may indicate a buy signal

MACD line crossing below the signal line may indicate a sell signal

In practical applications, these technical indicators are usually not used in isolation, but in combination with each other:

1. Trend confirmation: For example, the moving average can be used to confirm the overall trend, and then the RSI or MACD can be used to find specific entry or exit points.

2. Divergence identification: When the price trend is inconsistent with the technical indicator trend, it may indicate a trend reversal.

3. Filtering signals: The signal of one indicator can be used to filter the signal of another indicator to reduce false signals.

Gu et al. (2020) innovatively integrated these technical indicators into an autoregressive recurrent neural network (AR-RNN) model. The method mainly includes indicator selection, feature engineering and model integration. The addition of these technical indicators significantly improves the prediction performance of the model. Compared with the benchmark model that only uses raw price data, the study found that the prediction effects of different combinations of technical indicators on different financial assets and time scales are different, emphasizing the importance of indicator selection [4].

4 FEATURE SCALING

4.1 STANDARDIZATION

Standardization is a common preprocessing technique in regression analysis, which converts features into a distribution with a mean of 0 and a standard deviation of 1. This process helps to eliminate the influence of feature dimensions and make features of different scales comparable. Gelman (2008) showed that standardization can improve the numerical stability and convergence speed of certain regression models (such as ridge regression) [14]. In practice, standardization is particularly beneficial for optimization algorithms based on gradient descent, which can accelerate the model training process. However, for rule-based models

such as decision trees, the effect of standardization is relatively small.

The formula of standardization:

$$X' = \frac{X - \mu}{\sigma}$$

Where μ is the characteristic mean and σ is the standard deviation

4.2 Normalization

Normalization is a technique for scaling features to a fixed range (usually [0, 1]) and is widely used in regression analysis. Compared with standardization, normalization retains the distribution characteristics of the original data and is suitable for models that are sensitive to the feature range. Han et al. (2011) pointed out that normalization performs well when processing data with a large number of outliers [15]. In practical applications, normalization is often used in image processing and neural network input layers to effectively prevent the gradient vanishing problem. Choosing normalization or standardization requires a trade-off based on data distribution and model characteristics.

Min-Max Normalization Formula:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

4.3 Analysis of The Applicability of Feature Scaling in Different Regression Models

The applicability of feature scaling methods (such as standardization and normalization) in different regression models varies. For linear regression, logistic regression, and neural networks, feature scaling can usually significantly improve model performance and training efficiency. Kuhn and Johnson (2013) showed that feature scaling is almost necessary for distance-based models (such as KNN and SVM) [16]. However, for tree-based models such as decision trees and random forests, the impact of feature scaling is relatively small. In practice, it is necessary to choose an appropriate scaling strategy based on the specific problem and model, and sometimes it is even necessary to combine multiple methods to achieve the best results.

Feature importance evaluation formula (taking random forest as an example):

$$\text{Importance}(X_i) = \sum \frac{(\text{Impurity\_decrease(node)} \mid \text{split on } X_i)}{n\_trees}$$

5 DISCUSSION

Traditional feature selection methods, such as PCA and linear regression, are simple and easy to use, but may have limitations when dealing with high-dimensional data. In contrast, model-based methods, such as random forests and XGBoost, although they can handle complex nonlinear relationships, have high computational costs and require a lot of parameter tuning. Data transformation techniques, such as standardization and normalization, are excellent in improving model stability, but their actual effect depends on the characteristics of the data distribution. Finally, although automated feature engineering tools (such as automatic machine learning platforms) have improved the efficiency of feature processing, they still lack in interpretability and transparency.

6 FUTURE WORK

Feature engineering remains a crucial area in machine learning and statistical modeling, especially in regression problems. How to further develop and improve feature processing techniques will directly affect the performance of the model. In future research, it is recommended to focus on the following directions: First, how to more effectively handle high-dimensional and sparse features, especially in large-scale data sets. Second, the combination of automated feature engineering and deep learning is expected to further improve the efficiency of feature generation in complex tasks. Finally, with the rise of explanatory machine learning, how to improve the interpretability of the feature engineering process while enhancing the predictive power of the model will also be an important research topic in the future.

7 CONCLUSION

Feature engineering plays a vital role in regression analysis. This paper systematically reviews various methods, covering the latest progress of traditional techniques and automated tools. Traditional feature selection and transformation methods are effective in processing simple data, but they show limitations when facing high-dimensional and complex nonlinear data. Model-based techniques, such as Lasso regression and random forest, have improved adaptability to complex data, but their "black box" characteristics affect interpretability, and parameter adjustment is complex and computationally expensive. Automated feature engineering has improved efficiency through the AutoML platform, but its transparency and flexibility still need to be further improved. Future research should focus on feature selection methods that efficiently process large-scale data, combined with automated feature generation of deep learning, and improve interpretability. In addition, the migration ability of feature engineering in cross-domain applications is also worth exploring. Although the existing technology has made significant progress, there is still a broad research space in feature processing, model

optimization and interpretability, which will provide new breakthroughs in solving regression problems.

ACKNOWLEDGMENTS

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

FUNDING

Not applicable.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable.

INFORMED CONSENT STATEMENT

Not applicable.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

PUBLISHER'S NOTE

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

AUTHOR CONTRIBUTIONS

Not applicable.

ABOUT THE AUTHORS

HE, Runhai

Research fields: Data analysis and Software Engineering.

LI, Boyi

Faculty of Digital Transformation, ITMO University, St. Petersburg, Russia.

LI, Fusheng

Faculty of Applied Mathematics and Computer Science, Belarusian State University, Minsk, Belarus.

SONG, Qingqing

Faculty of Applied Mathematics and Computer Science; Belarusian State University; 4 Nezavisimosti Avenue, Minsk 220030, Belarus; e-mails: fpm.sunC@bsu.by

REFERENCES

[1] Dong, G., & Liu, H. (Eds.). (2018). Feature engineering for machine learning and data analytics. CRC press.

[2] Islam, N. (2024). DTization: A new method for supervised feature scaling. arXiv preprint arXiv:2404.17937.

[3] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1798-1828. https://doi.org/10.1109/TPAMI.2013.50.

[4] Gu, Y., Yan, D., Yan, S., & Jiang, Z. (2020). Price forecast with high-frequency finance data: An autoregressive recurrent neural network model with technical indicators. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20) (pp. 2485–2492). Association for Computing Machinery. https://doi.org/10.1145/3340531.3412738.

[5] Duch, W. (2006). Filter methods. In I. Guyon, M. Nikravesh, S. Gunn, & L. A. Zadeh (Eds.), Feature extraction (Studies in Fuzziness and Soft Computing, Vol. 207, pp. 89-117). Springer. https://doi.org/10.1007/978-3-540-35488-8_4

[6] Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. Artificial intelligence, 97(1-2), 273-324. https://doi.org/10.1016/S0004-3702(97)00043-X.

[7] Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. bioinformatics, 23(19), 2507-2517. https://doi.org/10.1093/bioinformatics/btm344

[8] Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

[9] Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. Neural networks, 13(4-5), 411-430. https://doi.org/10.1016/S0893-6080(00)00026-5

[10] Izenman, A. J. (2013). Linear discriminant analysis. In Modern multivariate statistical techniques (pp. 237-280). Springer, New York, NY.

[11] Micci-Barreca, D. (2001). A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. ACM SIGKDD Explorations Newsletter, 3(1), 27-32. https://doi.org/10.1145/507533.507538

[12] Little, R. J., & Rubin, D. B. (2002). Statistical analysis with missing data (2nd ed.). John Wiley & Sons.

[13] Friedman, J. H. (1991). Multivariate adaptive regression splines. The Annals of Statistics, 19(1), 1-67. https://doi.org/10.1214/aos/1176347963

[14] Gelman, A. (2008). Scaling regression inputs by dividing by two standard deviations. Statistics in Medicine, 27(15), 2865-2873. https://doi.org/10.1002/sim.3107

[15] Han, J., Pei, J., & Kamber, M. (2011). Data mining: Concepts and techniques (3rd ed.). Morgan Kaufmann.

[16] Kuhn, M., & Johnson, K. (2013). Applied predictive modeling. Springer.