A Feature Engineering Approach for Tree-based Machine Learning Sales Forecast, Optimized by a Genetic Algorithm Based Sales Feature Framework

2022 5th International Conference on Artificial Intelligence and Big Data (ICAIBD) | 978-1-6654-9913-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICAIBD55127.2022.9820532

Jiezhen Li
Industry Technology Department, Hitachi China Research Laboratory
Hitachi (China), Ltd. Guangzhou Branch
Guangzhou, China
jzhli@hitachi.cn

***Abstract*—In the new retail era, the sales pace and consumers' consuming habits are disrupted dramatically by e-commerce and mobile consuming market developments and become various, volatile, and complicated. In sales forecasting, tree-based machine learning algorithms are popular and impressive. Feature engineering, on the other hand, is the most significant impediment to blending business and technological knowledge. An innovative way to solving the feature engineering problem of employing tree-based algorithms in sales forecast is proposed in this work. And a genetic algorithm is presented to enlarge the feature base and to explore potential values from features. To ensure that features with important business characteristics are picked, a novel feature framework for sales forecast is offered. Also given is a genetic algorithm for expanding the feature base and exploring potential values from features. The performance of tree-based models from AutoGluon will next be evaluated using a public dataset, Kaggle Rossmann sales data. The results suggest that the proposed strategy can improve the accuracy and stability of decision tree algorithms in sales forecasting significantly.**

Keywords—sales forecast, feature engineering, genetic algorithm, tree-based algorithm, machine learning

I. Introduction

Sales forecast is one of the most concerning issues in the business world. Because the majority of management uncertainty stem from the market, sales forecasting is a direct assessment measure of a company's management performance. Traditional sales forecasting approaches, such as expert inquiry and linear statistical methodology, are finding it challenging to keep up with increasingly complicated needs as the market becomes more volatile.

With the rapid development of artificial intelligence, machine learning becomes widely used in sales forecast field and deployed in the industry. In particular, tree-based algorithms are proved to be effective and practicable in sales forecast field with diverse data. However, sales data are usually deficient, fuzzy, and unbalanced, which largely increases the difficulty of feature engineering and impedes the application of tree-based models. Moreover, a handful of sales management theories can describe such dynamic market demands in quantitative ways. It causes feature engineering turns into a nonstandard process for sales forecast requiring highly specialized backgrounds.

Feature engineering is a vital challenge for applying tree-based algorithms in the industry, because of strong business relevance, deficient data of low quality, and lack of specified theory instruction.

In this paper, a systematic and practical method of feature engineering will be presented with an innovative feature generation framework optimized with a Genetic Algorithm (GA). Then, the effect on tree-based algorithms will be tested with Kaggle Rossmann Store public dataset. Four tree-based models from AutoGluon will be used for comparison.

II. Related Works

Sales forecast relates to extensive research. Concluding professional opinions to instruct prediction is an effective approach with reliable experts [1]. Statistical measures are more commonly used taking advantage of time series data and patterns, such as Auto-Regression model (AR), Moving Average model (MA), and Autoregressive Integrated Moving Average model (ARIMA) [2][3]. With deficient data, time series forecasting methods can be practical in relatively stable markets. For a dynamic market, it is indicated that nonlinear models are more capable of sales forecast than traditional linear models [4]. Machine learning algorithms have advantages over linear models in complex sales situations. In [5][6][7], Support Vector Regression (SVR) is researched for sales forecast. Then, a typical tree-based model XGBoost is introduced by Chen T. et al. [8], optimized from Gradient Boosting Decision Tree (GBDT). Also, LightGBM is formed to improve the efficiency of GBDT [9]. Except for boosting method, bagging is also a common tree-based algorithm like Random Forest (RF) model, applying Bootstrap Aggregation technique and multiple decision trees [10].

Tree-based machine learning models show outstanding capability in analyzing sales data providing considerably high performance with understandable frameworks [11]. While tree-based algorithms are advancing towards maturity, feature engineering plays a decisive role in sales forecasting, including feature selection and feature generation. Features with irrelevant information cause more computational cost and low forecasting accuracy [12]. Correlation analysis is researched to conduct feature selection for tree-based models [13], such as pearson coefficient and spearman coefficient. More feature selection methods are under researching, including Principal Component Analysis (PCA), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) algorithms [14][15]. Genetic Algorithm (GA) is another effective feature selection method simulating and analyzing features as natural evolution [16]. Applying tree-based genetic algorithms to feature engineering is also researched in detail [17].

However, standard feature engineering methods are rarely researched, which has large demand in the industry. The following paper will demonstrate a practical and systematic feature engineering approach for sales forecast.

III. Methodology

A combination of a newly created sales feature framework and a genetic algorithm is used to design the proposed feature engineering approach. Using tree-based methods, it may greatly simplify and standardize feature engineering for sales forecasting. Fig. 1 depicts the overall procedure. The basic procedure can be broken down into three stages:

1) To conduct feature engineering with the 8C sales feature framework.
2) To conduct feature engineering with the genetic algorithm.
3) To import features processed by the above two steps to tree-based machine learning models to predict sales.

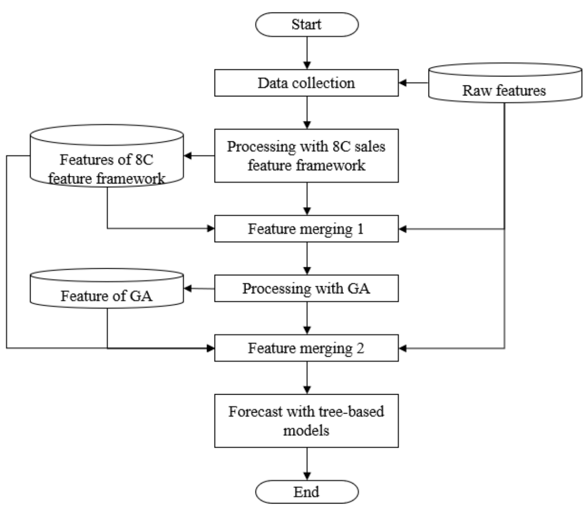

Fig. 1. The overall process of the proposed approach.

A. 8C Feature Framework

In most sales forecast cases, data filtering and data integration majorly rely on subjective experience. The various expertise constrains feature data quality, which affects the performance of forecast performance significantly while feature engineering is a critical part of machine learning models. Additionally, tree-based machine learning models have weaknesses in dealing with sales data containing time series characteristics, which raises more difficulties for feature engineering.

To cope with the nonstandard feature engineering process, we propose an 8C feature framework to guide data filtering and data integration in sales forecast cases. The 8C feature framework consists of the current effect, the carryover effect, the collaboration effect, the cost effect, the connatural effect, the competition effect, the customer effect, and the correlative effect. These effects are the pivotal factors influencing sales performance, based on numerous former researches on sales analysis and operation experience. Definitions of the 8C refer to TABLE I.

TABLE I. Definitions of 8C Sales Feature Framework

Terms	Definition
Current Effect	Factors influence the target sales performance directly.
Carryover Effect	Factors influence the target sales indirectly and usually are current effect factors of last period.
Collaboration Effect	Factors influence the target sales performance with combined sales strategies.
Cost Effect	Factors relevant to sales cost, promotion cost or advertisement cost, etc.
Connatural Effect	Factors from the object itself.
Competition Effect	Factors from competitors which are similar to the current effect of competitors.
Customer Effect	Factors of customer characteristics and preferences.
Correlative Effect	Factors from the external related environment.

In light of the 8C feature framework, feature engineering of sales forecast can be conducted in following steps:

1) To select all sales relevant data from business operation data as basic features.
2) To filter features according to the 8C feature framework.
3) To generate new features with basic features through combination or conversion according to the 8C feature framework.

After feature engineering with 8C feature framework, data relevant to sales performance is selected and generated based on business insights, including features reflecting time series characteristics. However, two existing problems, which are data deficiency and experience constraints, limit sales forecast performance in practice. The data deficient problem is very common because it is virtually impossible to collect all required data from business records directly, depending on the level of digitization of companies. And experience constraints are caused by conventional thinking and difficulties of quantitative analysis of business in various industries and scenes. To cope with these problems, we introduce a genetic algorithm to generate hidden and effective features. Features generated by the 8C feature framework will be imported to GA for further processed.

B. Genetic Algorithm

In most cases of sales forecast, feature combination with certain patterns can improve the forecast performance reasonably. Nonetheless, combining all features with certain patterns and selecting relevant features are near impossible with large amounts of data in practice, which causes high computing power consumption and low computing efficiency. The genetic algorithm is an competent technique to generate and select features with large amounts of data. In this paper, a genetic algorithm is applied to automatically generate new features in specified ways and select appropriate new features according to relevance to sales performance.

The feature combination logic is optimized and set as shown in TABLE II, concluded from business applications.

TABLE II. FEATURE COMBINATION LOGICS OF GA METHOD

Logics	Explanation
$x_i + x_j$	Equals feature i plus feature j.
$x_i - x_j$	Equals feature i minus feature j.
$x_i \times x_j$	Equals feature i multiplied by feature j.
$\begin{cases} x_i \div x_j, & x_j \neq 0 \\ 1, & x_j = 0 \end{cases}$	Equals feature i divided by feature j when feature i is not 0, otherwise equals 1.
$\begin{cases} \ln\lvert x_i \rvert, & x_i \neq 0 \\ 0, & x_i = 0 \end{cases}$	Equals taking the natural logarithm of the absolute value of feature i when feature i is not zero, otherwise equals 0.
$\sqrt{\lvert x_i \rvert}$	Equals the square root of the absolute value of feature i.
$\begin{cases} 1/x_i, & x_i \neq 0 \\ 0, & x_i = 0 \end{cases}$	Equals the reciprocal value of feature i when feature i is not 0, otherwise equals 0.
$min\{x_i, x_j\}$	Equals the maximum value between feature i and feature j.
$max\{x_i, x_j\}$	Equals the minimum value between feature i and feature j.
$x_i \times x_j \times x_k$	Equals multiplying feature i, feature j, and feature k.
$x_i \times x_j \times x_k \times x_l$	Equals multiplying feature i, feature j, feature k, and feature l.
$x_i \times x_j \times x_k \times x_l \times x_m$	Equals multiplying feature i, feature j, feature k, feature l, and feature m.
$\left((x_i{}^2) \times x_j\right)$	Equals feature i squared multiplied by feature j.

Fig. 2 demonstrates the flow diagram to illustrate the process of feature engineering with GA. It starts with population initialization. The reproduction stage, using the combination logics mentioned above, is the major stage for feature generation, including selection, crossover, and mutation. Then, an evaluation process will be used to choose valuable new features by applying spearman coefficient. For reasons of generating features with as many types as possible, it is usually set to conduct the generation process multiple times to meet the feature amount requirement instead of only one time. With this GA method, potential features beyond traditional business understanding will be discovered. Also, new relevant features will improve the performance of tree-based models by increasing the proportion of relevant features if the original feature base is not constructed properly with comprehensive business and data science knowledge. It can largely improve the sales forecast accuracy in real cases. Besides, this GA method can optimize the 8C feature framework by utilizing new features and complementing missing features.

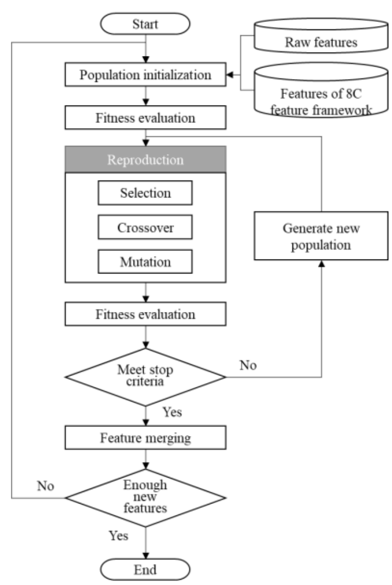

Fig. 2. Flow diagram of feature engineering with GA.

C. Tree-based Machine Learning Model

In response to this regression problem, we adopt tree-based models from AutoGluon, to maintain the same analysis environment and to purely exhibit the effect of the feature engineering method. AutoGluon is a convincing automated machine learning tool developed by Amazon [18], which can avoid model setting influences and provide stable and authentic results for common tree-based machine learning models.

IV. Experiments and Results

A. DataSet

To test our model, we use the Rossmann Store Sales dataset, which is a public Kaggle competition dataset comprising store, promotion, and competitor data. In order to test the forecast performance with our model, we select data from January 1st in 2013 to April 30th in 2015 as training data and select data from May 1st to July 31st in 2015 as testing data. To evaluate the forecasting ability of our model, we will conduct rotated predictions for each day in testing data. The dataset description is shown in TABLE III.

TABLE III. Description of Rossmann Store Dataset

(Kaggle Rossmann Store)	Training data	Testing data
Sample	310527	102580
Period	2014/5/1~2015/4/30	2015/5/1~2015/7/31
Store	1115	1115
Number of raw features	16	16

B. Feature Engineering

Lacking enough data is a common situation in practice due to insufficiency or difficulty of data gathering. The 8C feature framework can help most feature engineering situations with insufficient features for sales forecast. First, we apply the 8C feature framework to generate or transform sales relevant features with raw features, because this dataset is with just a few features directly relevant to sales performance. After using the 8C feature framework, a new set of features is obtained and shown in TABLE IV. In terms of this case, raw features cover five aspects in the 8C feature framework, which are the current effect, the carryover effect, the connatural effect, the competition effect, and the correlative effect. We conduct feature engineering with the 8C feature framework to complement the collaboration effect and the customer effect, while features of the cost effect are missing in raw data. Also, additional features are increased and improved in original areas.

TABLE IV. New Feature Set after Applying 8C Sales Feature Framework

Raw feature	Features generated by 8C sales feature framework
Current Effect	
- Day of week - Date - Whether store is open - Whether special promotion is conducted - Whether continuous promotion is running - Continuous promotion type	- Whether continuous promotion is renewing
Carryover Effect	
- Continuous promotion since week/year	- History daily highest/lowest/median sales volume - Sales volume in recent 7 days and the median number - Continuous promotion lasting time for that store
Collaboration Effect	
NULL	- Whether special promotion and continuous promotion is running together
Cost Effect	
NULL	NULL
Connatural Effect	
- Store ID - Store type - Store assortment	NULL
Competition Effect	
- Competitor distance - Competitor open month/year	NULL
Customer Effect	
NULL	- History daily highest/lowest/median customers - The number of customers in recent 7 days and the median number
Correlative Effect	
- State holiday - School holiday	- Year/month/day - Workdays or weekends

After generating features with the 8C feature framework, we increase features from 16 to 43, including 2 raw features being transformed into new features in better patterns.

Then, the genetic algorithm is applied to enlarge the feature base and to explore potential feature characteristics beyond traditional business understanding. Our proposed genetic feature engineering method generates 40 new features, which is approximately equal to the number of features we already have, through four rounds of combining and selecting features with certain patterns. Among these 40 new features, 3 new features are generated in the same form and are deleted to be 37 new features generated by GA. The appearance of duplicate new features in different generation processes represents these new features showing strong relations with target sales. Then, these 37 new features will be merged into raw features and features generated by the 8C feature framework as decision tree-based models' entire data input, becoming 80 features in aggregate. The feature composition is presented in Table V and Fig. 3.

TABLE V. THE NUMBER OF FEATURES IN EACH STAGE OF FEATURE ENGINEERING

	Raw	8C feature framework	GA
The number of features	14	29	37

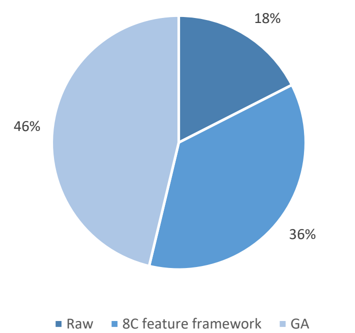

Fig. 3. Composition of feature input.

C. Model Setup

For the purpose of evaluating the effect of the proposed feature engineering method, we introduce frequently-used decision tree-based models in AutoGluon to conduct predictions considering popular boosting and bagging tree-based models, including XGBoost, LightGBM, CatBoost, and Random Forest. Models will use the training data to train and use the testing data to exhibit the performance difference between raw features and engineered features.

D. Model Evaluation

We adopt Mean Absolute Percentage Error (MAPE) and Root Mean Square Percentage Error (RMSPE) to analyze models' forecast accuracy on testing data.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\widehat{y_i}-y_i}{y_i}\right|, \quad RMSPE = \frac{1}{n}\sqrt{\sum_{i=1}^{n}\left(\frac{\widehat{y_i}-y_i}{y_i}\right)^2}$$

E. Experiment Result

Four decision tree-based models of AutoGluon are executed with both datasets of raw features and new features engineered with the proposed method. The performance comparison is illustrated in TABLE VI. The performance improvement is displayed in Fig. 4, which shows complete improvements to results with raw features.

TABLE VI. MODEL OVERALL PERFORMANCE COMPARISON

Models	Daily sales forecast with raw features		Daily sales forecast with engineered features	
	MAPE	RMSPE	MAPE	RMSPE
XGBoost	21.1%	28.7%	**13.8%**	**19.1%**
LightGBM	21.5%	28.8%	**10.8%**	**14.7%**
CatBoost	30.5%	37.9%	**13.5%**	**18.3%**
Random Forest	19.9%	29.1%	**10.2%**	**14.4%**

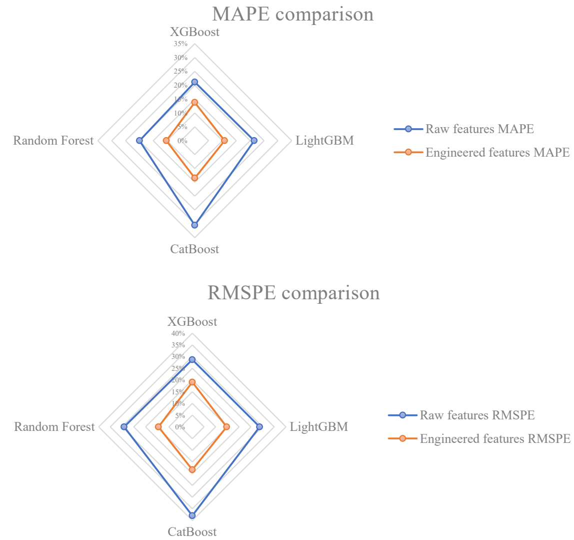

Fig. 4. Overall comparison with MAPE and RMSPE for four models.

To demonstrate the stability of our proposed method, we will compare the results from LightGBM and Random Forest, the best-performing models in the aforementioned experiment, in three months. Furthermore, these two models are typical in tree-based algorithms fairly representing boosting and bagging methods respectively. It can be concluded from the comparisons exhibited in TABLE VII that our proposed method is more stable over time, especially for the results in the third month. This improvement owes to the concept of the 8C sales feature framework strengthened by GA, providing more sufficient and multi-dimensional data for tree-based models to learn the essence of sales rather than relying on time-series features. In this case, it can reduce computational ability consumption by reducing update frequency. Furthermore, a more stable model is beneficial to extract business experience from data via model interpretation and feature importance analysis.

TABLE VII. COMPARISON OF DAILY SALES IN THREE MONTHS

Period	Daily sales forecast with raw features		Daily sales forecast with engineered features	
	MAPE	RMSPE	MAPE	RMSPE
LightGBM				
1st month	21.7%	28.1%	**11.1%**	**14.9%**
2nd month	20.4%	29.7%	**9.6%**	**12.9%**

3rd month	22.1%	28.7%	**11.5%**	**15.8%**
Random Forest				
1st month	22.7%	30.9%	**10.6%**	**15.7%**
2nd month	16.9%	27.5%	**8.7%**	**12.0%**
3rd month	20.0%	28.7%	**10.9%**	**14.9%**

In general, sales forecast is mostly used for replenishment. Short supply, overstocking, or product expiration are common outcomes of mismatches between actual sales volume and predicted results. The influences on replenishment and inventory are accumulative, so we conduct comparisons to illustrate the influence for weeks. We compare discrepancies in aggregated weekly sales with LightGBM and Random Forest results to assess cumulative influences. TABLE VIII compares how the suggested feature engineering technique can successfully offset cumulative influences by decreasing predicted deviations to actual sales volume.

TABLE VIII. COMPARISON OF AGGREGATED WEEKLY SALES

Period	Weekly sales forecast with raw features		Weekly sales forecast with engineered features	
	MAPE	RMSPE	MAPE	RMSPE
LightGBM				
1st week	20.5%	22.9%	**6.9%**	**9.3%**
2nd week	9.4%	12.1%	**8.7%**	**9.9%**
3rd week	28.8%	31.2%	**6.8%**	**8.2%**
4th week	9.7%	12.6%	**4.2%**	**5.2%**
Random Forest				
1st week	18.1%	22.4%	**4.7%**	**7.0%**
2nd week	15.7%	16.5%	**8.1%**	**9.0%**
3rd week	26.0%	30.2%	**5.8%**	**6.7%**
4th week	13.8%	14.9%	**6.6%**	**7.6%**

Additionally, the most appropriate number of genetic generated features is approximately equal to the number of features including raw features and features generated by the 8C feature framework according to practice experience. To evaluate the proper number of features generated by the genetic algorithm, we test three feature sets with LightGBM and Random Forest as well. Three datasets have roughly 0.5, 1, and 1.5 times the number of features combining raw features with 8C feature framework features, containing 20, 40, and 60 new generating features by GA, respectively. The result illustrated in Fig. 5 shows the same conclusion.

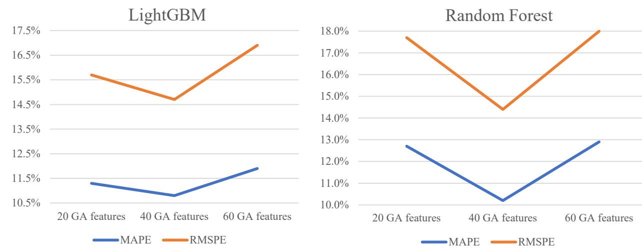

Fig. 5. Performance comparison of different scales of new features generated by GA.

Finally, we examine the effect of the proposed feature engineering method in this research using LightGBM and Random Forest to investigate feature importance. We use the feature importance analysis module in AutoGluon to calculate the importance value of each feature. In Fig. 6, the feature importance ranking distributions are exhibited, and new features generated using the suggested approach obviously play a decisive role. The dominant feature of LightGBM and Random Forest according to feature importance is the same feature generated by GA with the history median sales volume and the sales volume of the day before. It also proves the effective results of feature engineering with the proposed method and the logic is shown in Fig. 7. A preliminary reason analysis of this feature is that daily sales of Rossmann Store are iterative, , with the relatively steady history median sales volume serving as the base value and the changing sales volume of the day before serving as the reference value. The combination of these two features can provide a potential logic to recognize the trend of target sales or even the value range based on market and store characteristics.

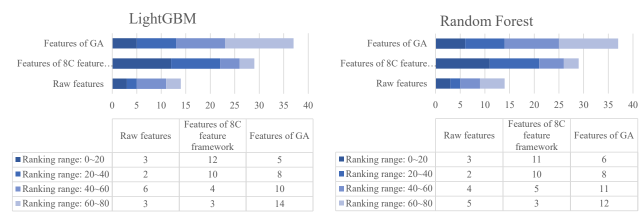

Fig. 6. Feature importance ranking distribution.

$$\boldsymbol{The\ top\ one\ feature} = HistoryMedianSales + \max(HistoryMedianSales, OneDayBeforeSales)$$

Fig. 7. The combination logic of the dominant feature according to feature importance.

We can deduce the following from the outcomes of the experiment:

1) In this experiment, the proposed technique significantly improves all testing models simply by feature engineering based on raw characteristics. The Random Forest model, in particular, outperforms the other three models, achieving 10.2 percent for MAPE and 14.4 percent for RMSPE. The CatBoost model, as the most remarkbly improved model, improves MAPE

by 17 percent and RMSPE by 19.6 percent when compared to raw features.

2) With the suggested method's ability to lower model update frequency and training data size, tree-based models may become more stable for various sales conditions, which is useful to application in practice and feature analysis for business strategy.

3) Potential features beyond typical business understanding are uncovered using the proposed feature engineering method, some of which contribute more than raw features in tree-based models.

V. Discussion

As the results of the experiments reveal, the proposed strategy can improve tree-based model performance for sales forecasting in a stable and meaningful way. The quality of data from businesses is rarely as good or as complete as that from the Rossmann Store dataset. Furthermore, realistic sales forecasting tends to concentrate on the product dimension, which is more challenging to handle. In fact, we evaluated the performance using an internal dataset, which is a typical dataset for fresh commodities from the retail business, and it yielded even better results, with MAPE improvements of up to 30 percent.

Additionally, the proposed method flexibly combines the 8C feature framework, a newly raised feature generation framework, and the GA to conduct feature engineering for sales forecast. In terms of the 8C feature framework, the carryover impact is the most essential category, while other categories may be supplemented by the GA approach, which is also why the 8C feature framework and the GA are combined as a systematic and standard way for feature engineering. Therefore, this method can largely reduce feature engineering complexity for sales forecast in real business.

VI. Conclusion

This paper presents a systematic, standard, and effective feature engineering method for applying tree-based machine learning tools to sales forecast in practice, based on a concluded feature framework and a genetic algorithm. Our methodology decreases business knowledge requirements and provides a practical strategy for dealing with real-world scenarios involving sparse and indirect data. Furthermore, the proposed approach avoids the majority of specialized data processing and increases the industry's universality of tree-based machine learning technologies.

The future works should be the interpretation of features obtained by the GA approach that are unintelligible to traditional business knowledge. These new mysterious aspects are likely to generate fresh business insights and aid in the continual improvement of business theories. Matter of fact, the 8C feature framework may also be used to conduct qualitative research on those new features.

References

[1] Chang, Pei-Chann, and Yen-Wen Wang. "Fuzzy Delphi and back-propagation model for sales forecasting in PCB industry." Expert systems with applications 30.4 (2006): 715-726.

[2] Xia, Min, and Wai Keung Wong. "A seasonal discrete grey forecasting model for fashion retailing." Knowledge-Based Systems 57 (2014): 119-126.

[3] Gilbert, Kenneth. "An ARIMA supply chain model." Management Science 51.2 (2005): 305-310.

[4] Chu, Ching-Wu, and Guoqiang Peter Zhang. "A comparative study of linear and nonlinear models for aggregate retail sales forecasting." International Journal of production economics 86.3 (2003): 217-231.

[5] Liu, Yue, et al. "Wrapper feature selection optimized SVM model for demand forecasting." 2008 The 9th International Conference for Young Computer Scientists. IEEE, 2008.

[6] Yue, Liu, et al. "Demand forecasting by using support vector machine." Third International Conference on Natural Computation (ICNC 2007). Vol. 3. IEEE, 2007.

[7] Gao, Junjie, et al. "A hybrid optimization approach to forecast automobile sales of China." Advances in Mechanical Engineering 9.8 (2017): 1687814017719422.

[8] Chen, Tianqi, et al. "Xgboost: extreme gradient boosting." R package version 0.4-2 1.4 (2015): 1-4.

[9] Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in neural information processing systems 30 (2017).

[10] Ramachandra, H. V., et al. "Machine Learning Application for Black Friday Sales Prediction Framework." 2021 International Conference on Emerging Smart Computing and Informatics (ESCI). IEEE, 2021.

[11] Elith, Jane, John R. Leathwick, and Trevor Hastie. "A working guide to boosted regression trees." Journal of animal ecology 77.4 (2008): 802-813.

[12] Fan, Jianqing, and Yingying Fan. "High dimensional classification using features annealed independence rules." Annals of statistics 36.6 (2008): 2605.

[13] Li, Jiefei, et al. "Feature engineering and tree modeling for author-paper identification challenge." Proceedings of the 2013 KDD Cup 2013 Workshop. 2013.

[14] Taherdangkoo, Mohammad, et al. "An efficient algorithm for function optimization: modified stem cells algorithm." Central European Journal of Engineering 3.1 (2013): 36-50.

[15] Mabu, Audu Musa, Rajesh Prasad, and Raghav Yadav. "Mining gene expression data using data mining techniques: A critical review." Journal of Information and Optimization Sciences 41.3 (2020): 723-742.

[16] Doloca, Adrian. "Feature selection for texture analysis using genetic algorithms." International journal of computer mathematics 74.3 (2000): 279-292.

[17] Heaton, Jeff. Automated feature engineering for deep neural networks with genetic programming. Diss. Nova Southeastern University, 2017.

[18] Erickson, Nick, et al. "Autogluon-tabular: Robust and accurate automl for structured data." arXiv preprint arXiv:2003.06505 (2020).