
Analysis of the Effect of Website Sales Quality on Purchasing Decisions on e-commerce Websites

Deden Hardan Gutama ^{1,*}, Izzatul Umami ², Pujo Hari Saputro ³

^{1,3} Alma Ata University, Indonesia

² Universitas Darul Ulum, Indonesia

^{1,*} hardan@almaata.ac.id; ² izzatul.umami@undar.ac.id; ³ pujo@almaata.ac.id

* corresponding author

(Received February 16, 2021 Revised February 23, 2021 Accepted February 27, 2021, Available online March 1, 2021)

Abstract

Both Web-based information infrastructure and marketing activities are dealt with by business-to-consumer electronic commerce. Centered on information systems and marketing literature, this review suggests a research model to explain the effect on consumer loyalty of the dimensions of website quality (system quality, information quality, and service quality). In order to verify the validity of the calculation model, confirmatory factor analysis was performed, and the structural model was also examined to investigate the correlations hypothesized in the study model. In this study, by comparing the Hyperparameter class & Catboost class we can find a number of distributions of individual absolute errors which can be considered as a fairly important factor in the analysis of sales quality on e-commerce websites.

Keywords: CatModel, Hyperparameter, E-Commerce, Sales Quality, RMSE

1. Introduction

In modern times like today, technological advances are growing rapidly and it is deemed necessary to support most human activities [1]. One technology that is felt to be very important is the internet. The Internet is an information medium that is very fast and efficient in disseminating information [2]. The development of the internet has influenced economic development, one of which is the buying and selling process which is usually carried out by face-to-face transactions which has now become very easy with the existence of buying and selling transactions via the internet or called e-commerce [3]. There is e-commerce. Buying and selling transactions no longer have to be done face-to-face but can be done anytime and anywhere without limitation of time and place. E-commerce is a promising business alternative, because with e-commerce, both producers and consumers are given many conveniences [4].

Until now, many e-commerce platforms continue to strive to provide the best for their customers [5]. One of the things that are considered very important to support is by providing an attractive website appearance [6]. Because the e-commerce website is the most important component. When you want to make transactions through e-commerce, customers will definitely visit the website first [7]. Website is a system where information in the form of text, images, sound, etc. is presented in hypertext form and can be accessed by software called browsers [8]. Creating a sense of comfort when visiting the website to view products, will indirectly encourage buying decisions on visitors. If customers feel comfortable accessing the website and making transactions, it will be more frequent and more customers will shop [9]. The increasing number of customers, the easier it is for an e-commerce platform to achieve its sales targets.

2. The Proposed Method

Predictions are made from an ensemble of weak learners in gradient boosting [10]. Unlike a random forest that generates a preference tree for each sample, trees are generated one after the other in gradient boosting [11]. Previous trees are not altered in the model. To improve the next one, findings from the previous tree are used. The gradient

boost on decision trees is a method of machine learning that works to optimize the accuracy of predictions by gradually training more complex models [12]. For predictive models that evaluate ordered (continuous) data and categorical data, gradient boosting is incredibly important [10]. One such example is a credit score forecast that involves numerical characteristics (age and salary) and categorical characteristics (occupation). One of the most powerful methods of building ensemble models is gradient boosting. In several systems of organized data, the integration of gradient boosting with decision trees provides state-of-the-art performance [13]. Developers use these approaches in an iterative fashion to create ensemble models. The algorithm learns the first tree on the first iteration to decrease the training error, seen in figure 1. on the left-hand picture. Typically, this model has a major error; it isn't a smart idea to create really large trees since they overfit the data in boosting.

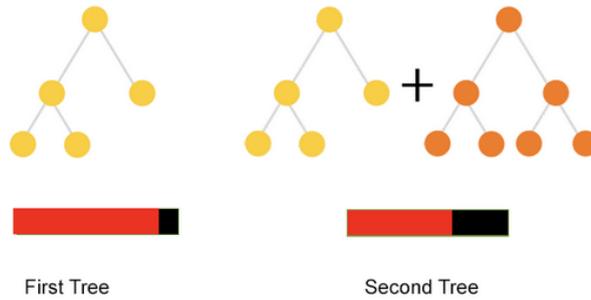


Fig. 1. The first and second trees typically contain major errors.

The second iteration, in which the algorithm learns one more tree to decrease the error created by the first tree, as seen in the right-hand image in figure 1. The algorithm repeats this process until, as we see in Figure 2, it produces a respectable quality mode.

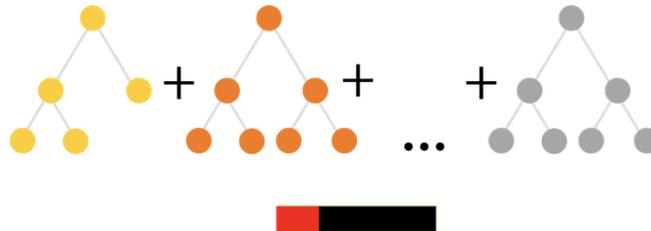


Fig. 2. N-th tree

Gradient Boosting is a tool for every continuous objective function to incorporate this concept [14]. Although regression optimizes using root mean square error, the common classification method uses Logloss [15]. Any variants of LambdaRank are usually applied by ranking tasks. Each stage of Gradient Boosting incorporates two steps: calculating loss function gradients that we want to optimize for each input object & learning the decision tree that predicts loss function gradients [16-17]. A basic operation that can be conveniently performed on the CPU or GPU is typically the first step. The quest for the right decision tree, however, is a computationally taxing process since the GBDT algorithm takes almost all the time.

3. Method

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850	United Kingdom

Fig. 3. Dataset

We can see that the datafile has the information supplied for each transaction. Look at the InvoiceNo and Customer ID from the first entry. Here we can see that one customer with ID 17850 from UK made one order which has InvoiceNo 536365. The customer ordered several products with different stock codes, descriptions, unit prices and quantities. Also, we can see that the Invoice Date is the same for these products. The dataset we use is quite structured and makes sense when it comes to analyzing purchasing decisions. However, the dataset we use has a high enough NaN value for one of the features (see figure 4 below), the customer ID feature has a high enough NaN value, that is, almost 25% of the data in the feature is missing, we assume that this is due to transactions carried out without a customer ID, besides that there are also NaN values in the description feature but with a relatively small amount (below 1%), this is quite reasonable because sometimes the customer or cashier does not leave descriptions when making transactions.

InvoiceNo	0.000000
StockCode	0.000000
Description	0.268311
Quantity	0.000000
InvoiceDate	0.000000
UnitPrice	0.000000
CustomerID	24.926694
Country	0.000000

Fig. 4. NaN Values distribution

Another case we have is How often do we lose customer IDs in transactions? And does the data without description also experience gaps in value in customer ID & item price? The answer is Yes, out of 1454 data that are missing descriptions also do not have customer IDs and item prices. Why do retailers record such entries without further explanation? There appears to be no sophisticated procedure for handling and recording such transactions.

The case in this dataset is bad enough. Prices and number of entries without customer ID can show extreme differences. Since we might want to create a later feature based on historical prices and quantities sold, this is very annoying. Our first advice for retailers is to prepare a strategy for wrong or special deals. And the question remains: Why is a transaction without a customer ID possible. Maybe the Transaction can be done as a search but it would be nice and clean to put up a special ID indicating that this one is a customer without an ID. Because we don't know why the customer or description is missing and we've seen oddities in quantity and price and zero price. In order to avoid interfering with the accuracy in making the data model, we will delete it from the dataset.

Now let's talk about the transaction number or Invoice Number, now we have 22186 different invoice numbers with about 2.2% of that amount being canceled transactions because the invoice has a "C" in the beginning. All cancellations have a negative but positive quantity, the unit price is not zero. With this data, we cannot easily understand why customers make returns and it is very difficult to predict such cases as there may be some hidden reasons why cancellations were made.

```
False    97.81007
True      2.18993
Name: IsCancelled, dtype: float64
```

Fig. 5. False & True Transaction

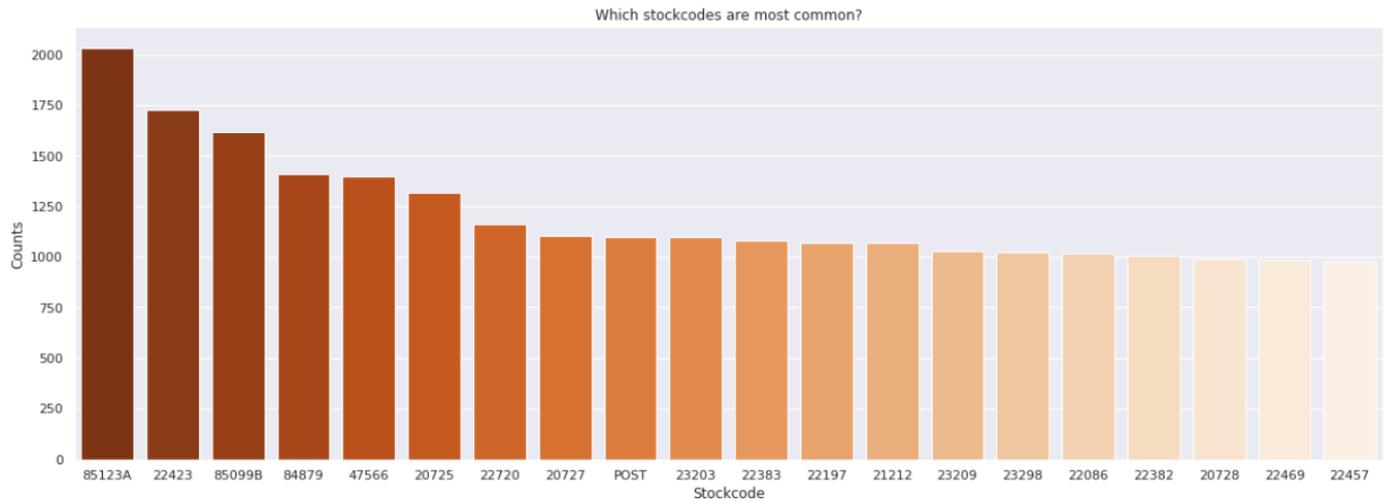


Fig. 6. Most Common Stock codes

As we can see in figure 5, this is the most popular stock code. From 3663 data, stocks with code 85123A are the most attractive. Most stockcodes are very rare (Fig. 6). This suggests that the retailer is selling a lot of different products and that there is not strong specialization for a particular stock code. However, we have to be careful as this does not mean that retailers are not specialized considering a particular type of product. Stock codes can be very detailed indicators that do not provide information about the type, for example a snack may have very different color variants, names and shapes but they are all snacks. What about the length of the stock code with the numeric code that each code has? As we have attached in figure 7, although the majority of the sample has a stockcode that consists of 5 numeric characters, we can see that there are other events as well. The length can vary between 1 and 12 and there are stock codes with no numeric characters at all.

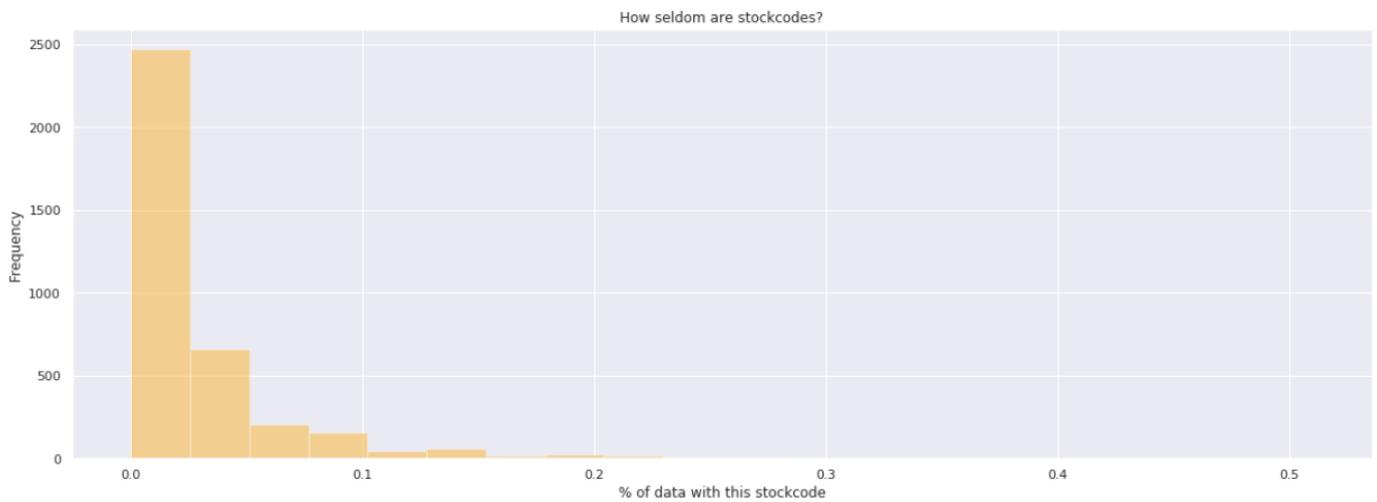


Fig. 7. Stock codes Seldom

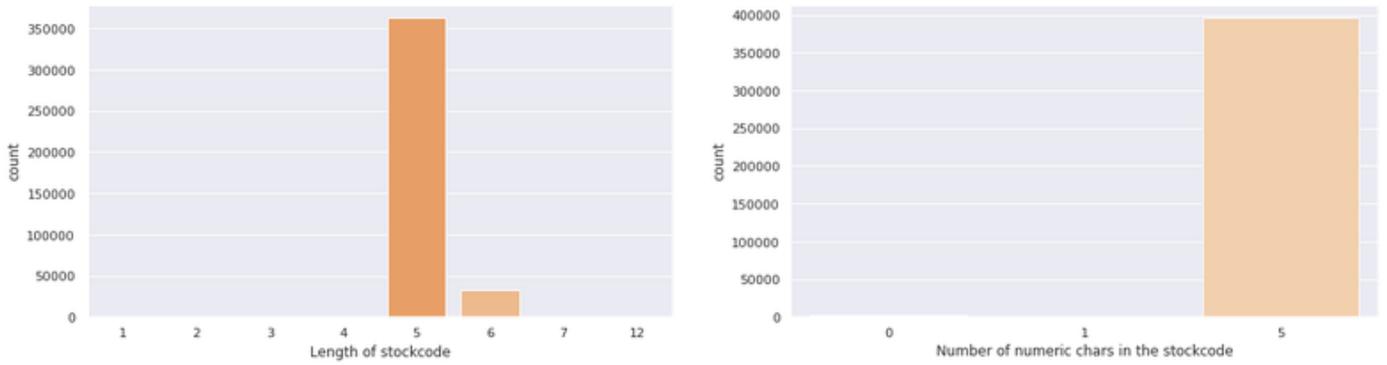


Fig. 8. Length and number of numeric chars in the stockcode

It is almost the same as the stock code in that several descriptions correspond to a similar type of product. Do we see the lunch bags that often appear? We also often have color information about products. Furthermore, the most common descriptions seem to confirm that the retailer is selling a variety of different types of products. All descriptions appear to consist of uppercase characters. OK, now we're going to do some additional analysis on the description by calculating the length and number of lowercase characters.

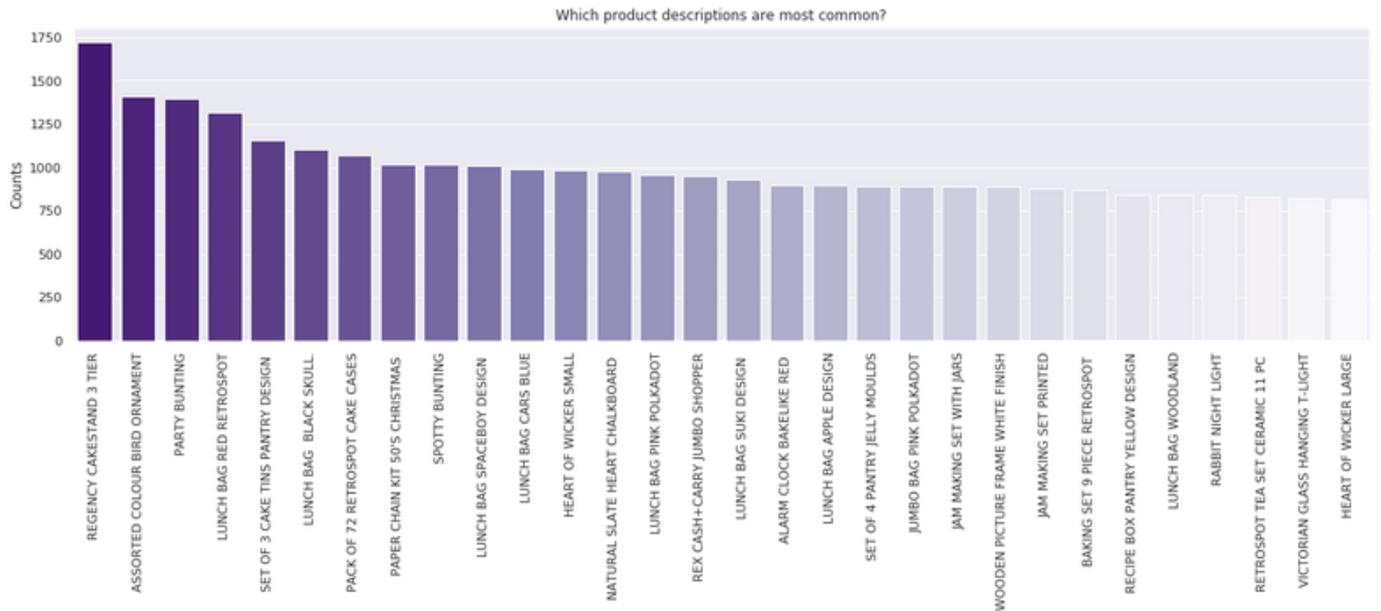


Fig. 9. Most Common Description

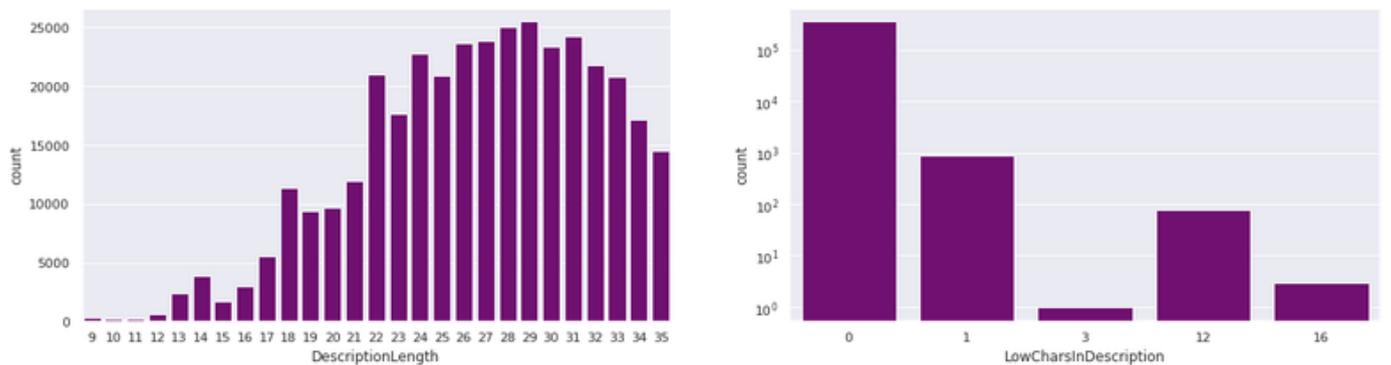


Fig. 10. Description Length & Low Chars

3.1. Focus on daily product sales

Since we want to predict the daily number of product sales, we need to calculate the daily aggregation from this data. For this purpose we need to extract temporal features from InvoiceDate. In addition, we can calculate the income earned from transactions using unit prices and quantities. Since the main task of this kernel is to predict the number of products sold per day, we can add up the daily quantities per product stock code. This way we lost information regarding the customer, country and price information but we will recover it later during this study.

Date	Year	Quarter	Month	Week	Weekday	Dayofyear	Day	StockCode	Quantity	Revenue
2010-12-01	2010	4	12	48	2	335	1	10002	60	51.00
2010-12-01	2010	4	12	48	2	335	1	10125	2	1.70
2010-12-01	2010	4	12	48	2	335	1	10133	5	4.25
2010-12-01	2010	4	12	48	2	335	1	16014	10	4.20
2010-12-01	2010	4	12	48	2	335	1	16016	10	8.50

Fig. 11. Revenues Aggregate

	Quantity	Revenue
count	195853.000000	195853.000000
mean	14.964244	28.181114
std	18.809496	43.938183
min	1.000000	0.120000
25%	3.000000	6.950000
50%	9.000000	15.300000
75%	20.000000	30.600000
max	411.000000	1266.300000

Fig. 12. Distribution of quantities and revenues

As we can see with the min and max values, the target variable shows extreme outliers. If we want to use it as a target, we have to exclude it because it will mislead our validation. Since we like to use pre-stop, this will directly affect predictive model training as well. Now, we only use the target range data occupied by 90% of the data entries. This is the first and simple strategy to exclude heavy outliers but we must always be aware of the fact that we have lost some of the information provided by the remaining% that we have excluded. It is generally useful to understand and analyze what caused this, and we are missing 5258 entries.

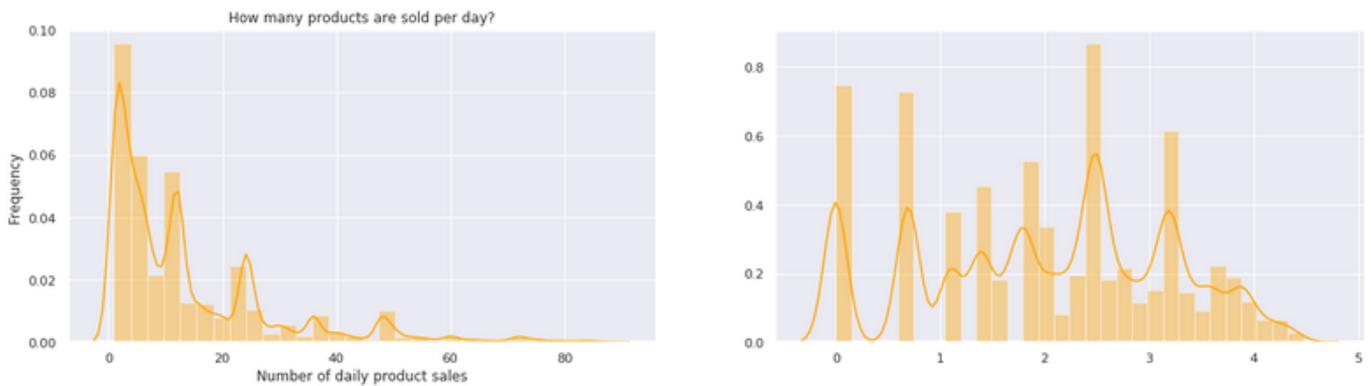


Fig. 13. Product sold per days

In figure 13 above, we can see that the distribution is right skewed. Lower values are more common. Moreover, the daily sales numbers appear to be multimodal. Daily sales of 1 are common as well as the sums of 12 and 24. This pattern is very interesting and leads to the conclusion that quantities can often be divided by 2 or 3. In short we can say that certain products are often purchased as single quantities or in small batches.

3.2. Predicting Daily Product Sales

In this study, we used catboost as a predictive model. The prediction of daily amounts and income is a regression task and consequently we will use the catboost regressor. The disadvantages and the metric we want to use are root mean square error (RMSE):

$$E = \sqrt{\frac{1}{N} \sum_{n=1}^N (t_n - y_n)^2}$$

It calculates the error between the target value t_n and the predicted value y_n per sample, taking the squares to ensure that the deviation is positive and negative, contributing to the sum in the same way. Then the mean is taken by dividing it by the total N samples (entries) in the data. And finally to get the impression of error for a single prediction, the roots are taken. We'll remember working with the loss function and this metric knowing that it is heavily influenced by outliers. If we have some predictions that are far off target, they will guide the mean towards higher values as well. Therefore, it is possible that we will make good predictions for most of the samples but the RMSE is still high due to high errors for a small portion of the sample.

Since the data covers only one year and we have a high increase of products sold during the pre-christmas period, we need to select validation data carefully. We'll start with data validation that covers at least the full 8 weeks (+ days remaining). After creating a new feature by exploring the data, we will use a sliding window time series validation which will help us understand whether the model can complete the prediction task during both times: the pre-Christmas season and the non-Christmas season.

Hyperparameter Class

This class holds all the important hyperparameters we have to set before training such as loss function, evaluation metric, maximum tree depth, maximum number of trees (iterations) and l2_leaf_reg for regularization to avoid overfitting.

```
class CatHyperparameter:
    def __init__(self,
                 loss="RMSE",
                 metric="RMSE",
                 iterations=1000,
                 max_depth=4,
                 l2_leaf_reg=3,
                 seed=0):
        self.loss = loss,
        self.metric = metric,
        self.max_depth = max_depth,
        self.l2_leaf_reg = l2_leaf_reg,
        self.iterations=iterations
        self.seed = seed
```

Fig. 14. Hyperparameter Class

Catmodel Class

This model acquires training & validation sets as data or pkita data frames for feature X and target Y together with one week. This is the first week of our data validation and the other weeks above are also used. It trains the model

and can show the learning process as well as the importance of features and multiple images for analysis of results. That's the fastest choice we can make for playing around.

Hyperparameter-Search Class

This is a class for hyperparameter search that uses Bayesian Optimization and Gaussian Process Regression to find the optimal hyperparameter. We decided to use this method because scoring calculations for one cat family model could be expensive. In this case the bayesian optimization can be a plus. Since this optimization method takes time, you should also try random search as it may be faster.

Time Series Validation Catfamily

This model stores information about how to divide the data into validation chunks and organize training with sliding window validation. Additionally, it can return the score as the mean of all of its model RMSE scores.

4. Model & Result Analysis

Date	Year	Quarter	Month	Week	Weekday	Dayofyear	Day	StockCode	Quantity	Revenue
2010-12-01	2010	4	12	48	2	335	1	10002	60	51.00
2010-12-01	2010	4	12	48	2	335	1	10125	2	1.70
2010-12-01	2010	4	12	48	2	335	1	10133	5	4.25
2010-12-01	2010	4	12	48	2	335	1	16014	10	4.20
2010-12-01	2010	4	12	48	2	335	1	16016	10	8.50

Fig. 15. Daily Data Feature

Now we will see how well this model performs without feature engineering and hyperparameter tracing. in this study and in the interactive model you can see that the loss models have met. We found that the root mean square error evaluated in the validation data was 1.065775039230805. This value is high, but as already mentioned, RSME is affected by outliers and we have to look at the distribution of the individual absolute error:

1. We can see that the absolute error distribution of the single prediction is skewed to the right.
2. The median (black) single error is half of the RMSE score and significantly lower.
3. By plotting targets versus predictions, we can see that we are making a higher error for validation entries that have a high true quantity value above 30. A strong blue line indicates an identity where the prediction is closer to the target value. To improve it, we need to make better predictions for the actual high quantity of products during the validation time.
4. We can see that the stock code as well as the product description is very important. They have no color because they are not numeric and have no low or high values.
5. Weekdays are also an important feature. We've seen this by exploring the data. Low scores from Monday to Thursday are the days when retailers sell most of the products. Conversely, high scores (Friday to Sunday) generate few sales.
6. Look at weekdays to understand this plot: Low scores (0 to 3) correspond to Monday, Tuesday, Wednesday, and Thursday. These are the days with a high number of product sales (high quantity target value). They are colored blue and pushed to a higher sharper value and consequently to a higher predicted quantity value. Higher workday values correspond to Friday, Saturday, and Sunday. They are colored red and push towards negative sharp values and to lower predicted values. This confirms the observations we made during the exploration work day and the daily quantity counts.
7. StockCode and Description are important features but also very complex. We've seen that we have nearly 4000 different stock codes and even more descriptions. To improve, we should try to engineer features that describe the product more generally.

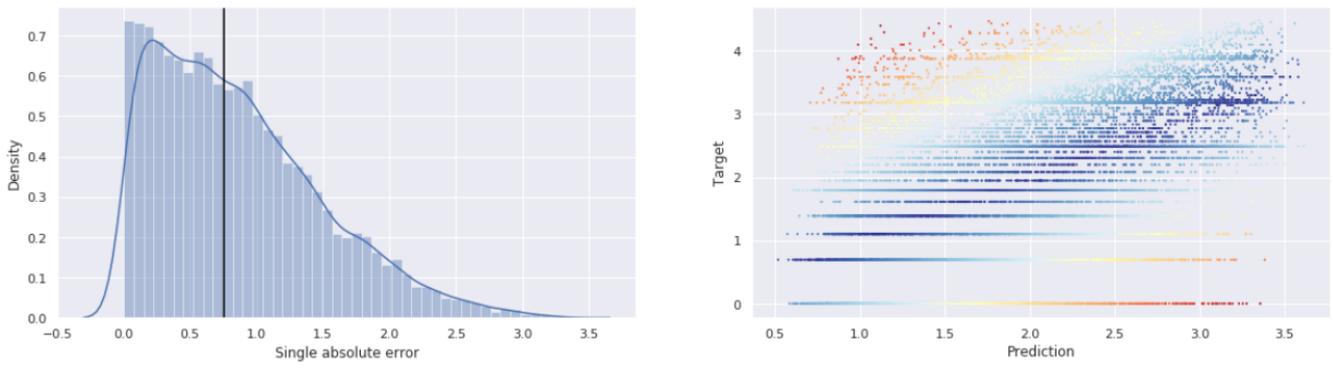


Fig. 16. Median Absolute Error

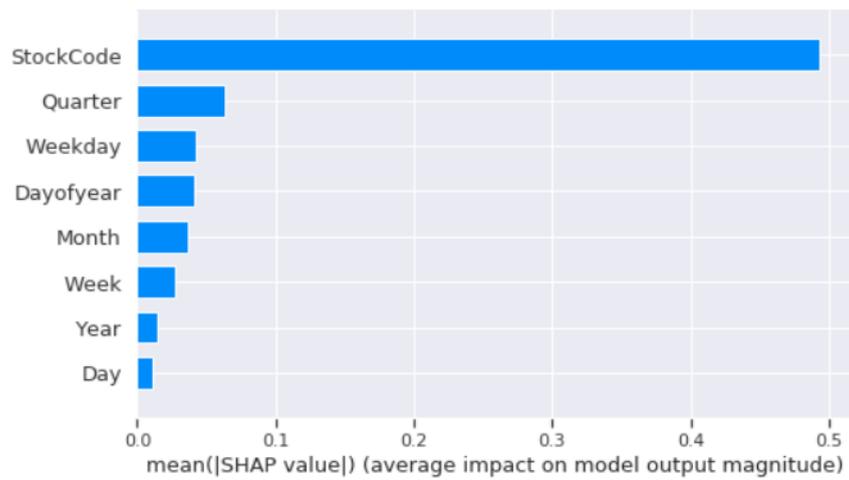


Fig. 17. Mean (SHAP Value)

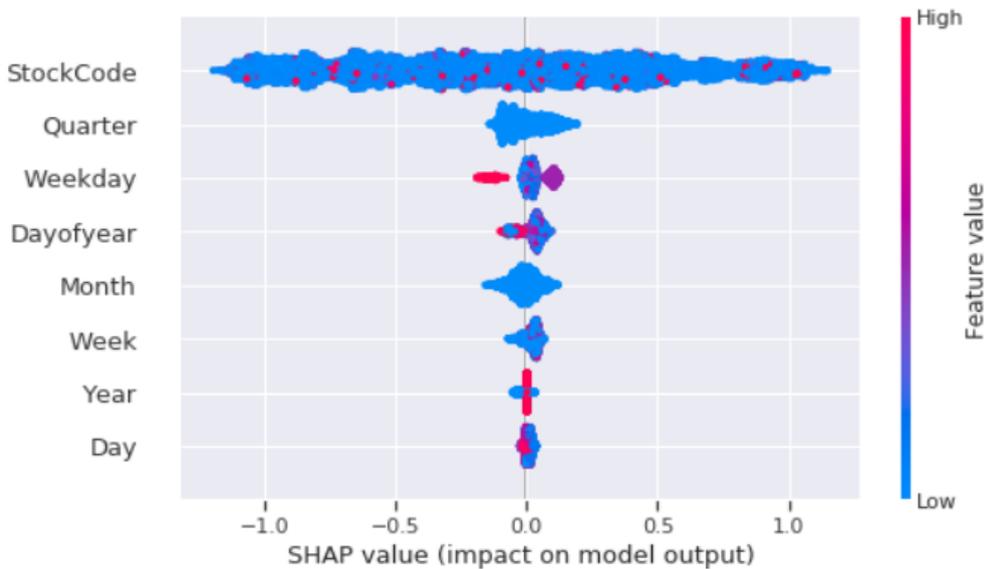


Fig. 18. SHAP Values (impact on model output)

5. Conclusion

In this study, the distribution of individual absolute errors has been successfully analyzed, retailers can easily control the quality of sales on their e-commerce websites. the distribution of the individual absolute error that we found may be just the tip of the iceberg which means there is still a high probability that there are other errors that can be found using other models, but this does not cover the fact that the model we used is considered successful. The RMSE results we got also support the results we got. Further research could apply the research model to examining other types of online retailers, as customer perceptions of website quality are context-dependent and thus their detailed effect on customer satisfaction may be related to specific products and services.

References

- [1] T. Jelassi and S. Leenen, "An e-commerce sales model for manufacturing companies: A Conceptual framework and a European example," *Eur. Manag. J.*, vol. 21, no. 1, pp. 38–47, 2003, doi: 10.1016/S0263-2373(02)00151-2.
- [2] E. Kirby-Hawkins, M. Birkin, and G. Clarke, "An investigation into the geography of corporate e-commerce sales in the UK grocery market," *Environ. Plan. B Urban Anal. City Sci.*, vol. 46, no. 6, pp. 1148–1164, 2019, doi: 10.1177/2399808318755147.
- [3] A. Davis and D. Khazanchi, "An Empirical Study of Online Word of Mouth as a Predictor for Multi-product Category e-Commerce Sales," *Electron. Mark.*, vol. 18, no. 2, pp. 130–141, 2008, doi: 10.1080/10196780802044776.
- [4] D. Bruce and W. F. Fox, "E-Commerce in the Context of Declining State Sales Tax Bases," *Natl. Tax J.*, vol. 53, no. 4 PART 3, pp. 1373–1388, 2000, doi: 10.17310/ntj.2000.4s2.06.
- [5] D. Wei, P. Geng, L. Ying, and S. Li, "A prediction study on e-commerce sales based on structure time series model and web search data," *26th Chinese Control Decis. Conf. CCDC 2014*, vol. i, pp. 5346–5351, 2014, doi: 10.1109/CCDC.2014.6852219.
- [6] W. F. Fox, "State and Local Sales Tax Revenue Losses from E-Commerce : Updated Estimates," vol. 4170, no. September, 2001.
- [7] J. Ben Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," *Data Min. Knowl. Discov.*, vol. 5, no. 1–2, pp. 115–153, 2001, doi: 10.1007/978-1-4615-1627-9_6.
- [8] X. Lin, X. Wang, and N. Hajli, "Building E-Commerce Satisfaction and Boosting Sales: The Role of Social Commerce Trust and Its Antecedents," *Int. J. Electron. Commer.*, vol. 23, no. 3, pp. 328–363, 2019, doi: 10.1080/10864415.2019.1619907.
- [9] E. Williams and T. Tagami, "Energy use in sales and distribution via e-commerce and conventional retail: A case study of the Japanese book sector," *J. Ind. Ecol.*, vol. 6, no. 2, pp. 99–114, 2002, doi: 10.1162/108819802763471816.
- [10] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: Gradient boosting with categorical features support," *arXiv*, pp. 1–7, 2018.
- [11] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: Unbiased boosting with categorical features," *Adv. Neural Inf. Process. Syst.*, vol. 2018-December, no. Section 4, pp. 6638–6648, 2018.
- [12] J. T. Hancock and T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00369-8.
- [13] Y. Zhang, Z. Zhao, and J. Zheng, "CatBoost: A new approach for estimating daily reference crop evapotranspiration in arid and semi-arid regions of Northern China," *J. Hydrol.*, vol. 588, no. April, p. 125087, 2020, doi: 10.1016/j.jhydrol.2020.125087.
- [14] G. Huang *et al.*, "Evaluation of CatBoost method for prediction of reference evapotranspiration in humid regions," *J. Hydrol.*, vol. 574, no. December 2018, pp. 1029–1041, 2019, doi: 10.1016/j.jhydrol.2019.04.085.
- [15] L. Diao, D. Niu, Z. Zang, and C. Chen, "Short-term weather forecast based on wavelet denoising and catboost," *Chinese Control Conf. CCC*, vol. 2019-July, no. 2018, pp. 3760–3764, 2019, doi: 10.23919/ChiCC.2019.8865324.

- [16] F. Zhang and H. Fleyeh, "Short term electricity spot price forecasting using catboost and bidirectional long short term memory neural network," *Int. Conf. Eur. Energy Mark. EEM*, vol. 2019-September, pp. 1–6, 2019, doi: 10.1109/EEM.2019.8916412.
- [17] J. Hancock and T. M. Khoshgoftaar, "Medicare Fraud Detection using CatBoost," *Proc. - 2020 IEEE 21st Int. Conf. Inf. Reuse Integr. Data Sci. IRI 2020*, pp. 97–103, 2020, doi: 10.1109/IRI49571.2020.00022.