Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 176 (2020) 1170–1179

Procedia Computer Science

www.elsevier.com/locate/procedia

24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Predicting the Return of Orders in the E-Tail Industry Accompanying with Model Interpretation

Abdullah Al Imran[a]*, Md Nur Amin[b]

[a]*American International University-Bangladesh, Dhaka, Bangladesh*
[b]*University Jean Monnet, Saint Etienne, France*

Abstract

Electronic Retailing (E-tailing) is one of the most impactful technology trends of recent times. This industry has dramatically enhanced the quality of human lives allowing people to shop online while having the comfort of their homes. In developing countries like Bangladesh, this industry is still rising and creating a significant economic impact. However, there exist a lot of challenges such as the return of orders that affects the growth of an E-tailer and causes revenue losses. This study addresses this most common business challenge in the E-tail industry and performs predictive modeling using 4 different state-of-the-art data mining techniques to help the industry smoothen its curve of growth. Along with predictive modeling, this study also aims to find out the most important features that influence the return of orders.

Keywords: Data Mining; E-Tailing; Order Return; Predictive Modeling; Model Interpretation

1. Introduction

With the evolution of the world wide web, Electronic Retailing or E-tailing has emerged as a new boulevard for the customers to shop in. By definition, E-tailing is the sale of goods and services through the internet. If we only consider the two largest E-tailers in the world, Amazon, and Alibaba, we can see how much impact they have already created in the modern world. However, in a developing country like Bangladesh E-tailing has risen like a sunrise in recent years and is bringing a consistent hike in the economy. On the other hand, customers have also embraced this platform as a medium of a cheaper and convenient way of shopping.

* Corresponding author. *E-mail address:* abdalimran@gmail.com

10.1016/j.procs.2020.09.113

In Bangladesh E-tailing is growing rapidly nowadays due to the easy accessibility of the internet and the world wide web. Particularly with the inhabitants of the capital Dhaka where people always try to avoid the hassle of going to a physical store because of heavy traffic on the roads. Concurrently, to survive in the competitive business environment where customer satisfaction and loyalty are key components to attain new customers and retain the existing ones E-tailers are providing lots of flexibility and agility to the customers.

While the trend is apparent, often customers return products for plenty of reasons. Notably, the propensity of returning an order is much higher online than offline [1] which rises a new business problem for the E-tailing industry. Sending back the product is disproportionately prevalent among online buyers and as a result, dramatically reduces the revenue of E-tailers by hemorrhaging a huge amount of money. Some of the reasons behind this return issue include flexible return policy, difficulty in payment method, impregnable ordering location, damaged, mismatched, incorrect, and delayed delivery of the product. It has become a snag for E-tailers as they are struggling with the swelling rate of returned goods which is a major drain on profitability. The cost of reverse logistics, managing staff and resources, processing returns, and refunds as well as the risk of not being easily resold of the returned items taking a big financial hit on E-tailers as the rate of return spikes.

In this study, we recognize this impactful business problem of the E-tailing industry and aim to propose a solution by providing state-of-the-art (SOTA) predictive modeling approaches and finding the best model. For the modeling purpose, we have used XGBoost, LightGBM, CatBoost, and TabNet along with a traditional Decision Tree algorithm as a baseline. We have collected a comprehensive dataset for this study by drawing a random sample of 10000 instances from the data lake of a leading E-tail platform in Bangladesh. We have also performed unsupervised visual analytics to understand the complexities of the dataset. The primary objective of this study is to compare all the SOTA models, finding the best performer, and also dissecting the best model to reveal the most impactful features that influence the occurrence of order return.

2. Literature Review

To understand the existing research contributions in this problem domain we have performed a rigorous review of the literature. Although we found very few papers that address exactly the same problem as us, we have studied the most similar ones.

Jianbo et al. [2] proposed a generic framework for an E-tail product return prediction named HyperGo. This approach aimed to predict the intention of the customers to return after they have put together the shopping basket. HyperGo is based on a novel hypergraph representation of historical purchase and returns records. It effectively leverages the rich information of basket composition and. Because of being linearly dependent on the size of the output cluster and polylogarithmically dependent on the volume of the hypergraph, this algorithm is very time efficient which is a major advantage of the algorithm. The authors compared the performance of HyperGo against k-NN, JacWght, and JacNorm. HyperGo yielded outperforming results in terms of AUC, Precision, and F-0.5 score.

Hailong et al [3] proposed a data-driven model to predict the volume of the order return. They found that the reason for the retail industry to develop a predicting model can be grouped into operational and categorical issues. They proposed a model by taking into account four factors: sales volume, time, retailer, and product type and applied high dimensional machine learning methods such as the Least Absolute Shrinkage and Selection Operator (LASSO), LARS-OLS Hybrid, Elastic Net, Random Forest, and Gradient Boosting to capture the nonlinearity. LASSO achieved the highest MSE score of 161.908 and 171.824 in train and test respectively.

Nachiketa et al [4] proposed an analytical model on how product review affects the probability of product return. They demonstrated how product reviews of consumers can lead to a lower return rate by examining the precision of product quality and prior uncertainty of product quality on return probabilities. A product that has an average rating than true rating is returned more often, on the other hand, unbiased reviews decrease the probability of returning.

Fraudulent behavior and damaged returns cost the Fashion e-commerce huge losses related to logistics and liquidation cost. If the return of products can be predicted even before the order takes place, it could greatly decrease the rate of return by taking preemptive measures. Sajan [5] et al proposed a hybrid dual model using deep neural networks to detect returnable products in real-time. Gradient Boosted classifier as a baseline classifier resulted in a 71.1% AUC score and was further enhanced by 8% by applying feature engineering. A deep neural network using aggregated embeddings of product outperformed the baseline model by increasing AUC and Precision by 83.2% and 74% respectively.

While prohibiting the return of product is not an option and to prevail in the competition, it is a cardinal part of a business model. Customers have the ease given by the retailers to return a product after purchasing for a certain time. Many studies have been corroborated that lenient return policy has a positive impact on customer satisfaction and buying behavior. Patrick et al [6] proposed a decision support system that can identify high product return rates to intervene before a problematic transaction takes place.

Successful implementation of reverse logistics networks requires different strategic, tactical, and operational levels in supply management systems. It is an important decision-making process when managing the recovery of products. Hess et al. [7] proposed a model for reversing logistic cost in the supply chain management system focusing management of the recovery of products that are no longer used nor desired by the customers.

Daria et al [8] has addressed the issue of order return for the apparel products. They have shown that predicting the return before a product launch can increase the profit substantially. In terms of apparel products, they found that color-pattern-shape and other intangibles related to high return rates seem to be captured well by deep learning techniques. Therefore, they derived an optimal policy to manage return as well as further enhance the predictive ability to observe the discrepancy in online and offline sales of a large European retailer using deep learning over the product images. It enabled the retailers to achieve a 21.2% increased profit having prior knowledge of the true return rate.

The aforementioned studies present different approaches to order return issues from different perspectives. From their studies, it is also apparent that this problem causes a significant amount of losses to the industry. In this study, we are also challenging this business problem from a classification perspective and also harnessing the techniques of model interpretation to reveal the most impactful causes.

3. Data Engineering

In an applied data mining research, data engineering combines all the data related processes such as data collection, pre-processing, feature engineering, and data transformation. Each process of data engineering affects the rest of the experiment and impacts the prediction results crucially. In this section, we are going to briefly discuss each data engineering process conducted in this study.

3.1. Data Description

The dataset has been collected from the data lake of one of the renowned and leading online shops in Bangladesh. It includes a random sample of 10000 instances of unique orders from different locations of Bangladesh. The raw data is consisting of 12 columns including the target column. The columns are - ***"order_id":*** a unique identification number for each order; ***"order_date":*** the date of order placement; ***"confirmation_date":*** the date of the order confirmation by a phone call; ***"order_type":*** the type of order. There are 2 types of orders such as urgent delivery with extra payment and regular delivery with usual payment; ***"order_location":*** the location of the order. There are 69 unique locations including urban, suburban, and rural areas; ***"order_medium":*** the medium through which the order has been placed. There are 5 unique mediums such as android-app, mobile-web, desktop-web, phone calls, and others; ***"cart_order":*** is the order is a cart order or single product order; ***"promotional_order":*** if the order is a promotional order or not; ***"order_amount":*** the total amount of the order in BDT; ***"payment_method":*** the method of payment. There are 6 payment methods such as Cash on delivery (MPD, MPC), Mobile Payment Services (MPS), Online Payment System (OPS), Account Wallet Cash (AWC), and Monthly EMI (EMI); ***"shipping_charge":*** the amount of charge for shipping the order in BDT. The charge varies from location to location; ***"is_return":*** is the order returned or not. This is the target label. The dataset is significantly imbalanced which makes this study very challenging to tackle while training the predictive models. Among the 10000 instances, we have 85% (8465) instances labeled as "No" and 15% (1535) instances labeled as "Yes". However, we have embraced the resampling technique to add another dimension to make our experiment rigorous and unbiased.

3.2. Data Resampling

Data resampling is the most prominent strategy to tackle the imbalanced data issues. There are two most frequently used resampling strategies which are oversampling and under-sampling. In the oversampling strategy, synthetic data points are generated for the minority class to balance with the majority class. The drawback of this strategy is that it adds impurity to the dataset. In the under-sampling strategy, some instances from the majority class are dropped randomly and uniformly to make balance with the minority class. Though this strategy leads to loss of information, it keeps the authenticity of the original data. In terms of our study, we have chosen to use the random under-sampling strategy [9] to make a balanced version of the dataset. We have under-sampled the majority class by randomly picking samples without replacement. As a result, the resampled balanced version of the data contains 3070 instances where each of the class includes an equal number of 1535 of instances.

3.3. Feature Engineering

Feature engineering is the process of deriving additional columns or features from the existing ones to improve the learning performance of machine learning models. Since our dataset does not contain a large number of features, we can derive some additional informative features that can help the classification models perform better. For this purpose, we have derived 7 additional features from the time-series columns. The features are: ***'order_hour':*** the hour of order placement; ***'order_minute':*** the minute of order placement; ***'order_day_name':*** the day of order placement; ***'confirmation_hour':*** the hour of order confirmation; ***'confirmation_minute':*** the minute of order confirmation; ***'confirmation_day_name':*** the day of order confirmation; and ***'order_confirmation_gap':*** the time gap between order placement and confirmation date. After deriving the new features, we have dropped the unnecessary and redundant features such as “order_id”, “order_date”, and “confirmation_date”. Finally, we are left with 15 features and a target column where 8 features are categorical and 7 features are numerical.

3.4. Feature Encoding and Transformation

Feature encoding and transformation is another imperative step before the modeling phase. To create informative numerical representations and also keeping the cardinality issue in our mind we have used the James-Stein encoder to encode the 8 categorical features. The James-Stein encoder can be formalized as follows:

$$JS_i = (1 - B) \times mean(y_i) + B \times mean(y) \tag{1}$$

where, $B = \frac{var(y_i)}{var(y_i) + var(y)}$ (2)

For feature value i, the James-Stein encoder returns a weighted average of the mean target value for the observed feature value i and the mean target value regardless of the feature value. Here, weight for B is determined in such a way that if the estimate of $\mathrm{mean}(y_i)$ is unreliable, more weight is put on $\mathrm{mean(y)}$. The target label is encoded using 0 and 1, where 0 means the order will not return and 1 means the order will be returned.

For the rest of the 7 numeric features, we have applied power transformation [10] which will make data more Gaussian-like and help the model tackle the issues related to heteroscedasticity.

4. Unsupervised Visual Analytics using t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) [11] is an unsupervised, non-linear algorithm that is primarily used for visualizing high-dimensional data in a low dimensional space. It can give us an intuition of how the data is arranged in a high-dimensional space and also about its complexity. The t-SNE algorithm calculates the probability of similarity between a pair of points in high-dimensional space and maps the probability of similarity of points in the corresponding low-dimensional space. It then tries to minimize the sum of Kullback-Leibler divergence of overall data points using the gradient descent method for a perfect representation of data points in lower-dimensional space.

We have applied t-SNE on both original, and balanced versions of datasets and projected the data points into 2D and 3D embedding space. To evaluate how the data points are clustered together, we have used the Calinski-Harabasz Index, Davies-Bouldin Index, and Silhouette Coefficient. In terms of Calinski-Harabasz Index, the score is higher when clusters are dense and well separated; in terms of Davies-Bouldin Index, values closer to zero indicate a better partition; and Silhouette Coefficient is bounded between -1 to +1 where +1 is for highly dense clustering and scores around zero indicate overlapping clusters.

The following figure 1 shows the visual representation of the original (imbalanced) data into 2D and 3D embedding space.

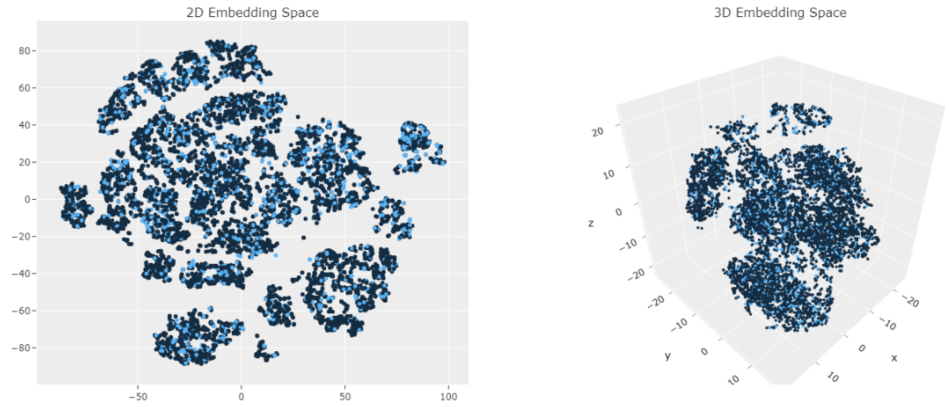

Fig. 1. Original (imbalanced) data in 2D and 3D embedding space

From the above visualization, we can see that all the data points of different labels are very close to each other and they do not show any significant dissimilarities. For the 2D embedding space the Calinski-Harabasz Index, Davies-Bouldin Index, and Silhouette Coefficient have been calculated as 8.800, 22.501, and 0.000. And for the 3D embedding space, the scores are 16.743, 16.749, and -0.002 respectively. In all metrics, the scores also express that there is no better partition between the data points and there are plenty of overlapping clusters.

The following figure 2 shows the visual representation of the balanced data into 2D and 3D embedding space.

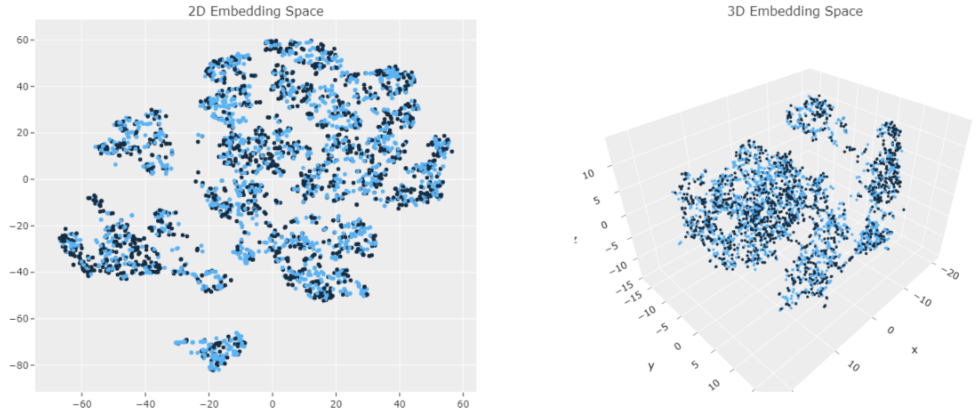

Fig. 2. Resampled (balanced) data in 2D and 3D embedding space

In figure 2, we can again see the same sort of scenarios like figure 1. The data points are densely plotted with each other and do not show any significant dissimilarities. For the 2D embedding space the Calinski-Harabasz Index, Davies-Bouldin Index, and Silhouette Coefficient have been calculated as 3.627, 26.970, and 0.002. And for the 3D embedding space, the scores are 3.429, 28.441, and 0.002 respectively.

From the above visual analytics, we can intuitively understand that the data points are very similar and overlapping to each other and do not show any significant differentiable patterns. This fact indicates that the supervised classification algorithms are going to struggle to model the data with good scores.

5. Supervised Learning Methodology

For modeling purposes, we have chosen to use different state-of-the-art algorithms that have been yielding outstanding results and outperforming the traditional data mining algorithms. In this section, we are going to briefly discuss the algorithms, hyperparameter optimization of those algorithms, and choosing the correct evaluation metrics. All the algorithms have been applied over the two versions of the dataset and the hyperparameters have been optimized for each of the modeling tasks.

5.1. Data Segregation

Before jumping into the modeling phase, we have segregated the datasets into a train and test set. From each version of the datasets, we have taken 80% of the data points for training purposes and 20% for the validation purpose. While drawing the samples, we have used stratified random sampling to keep the balance of positive/negative class between the train-validation datasets. From the original (imbalanced) dataset, 8000 data samples have been drawn for training and 2000 samples for validation whereas for the resampled (balanced) dataset 2456 data samples have been drawn for training and 614 samples for validation.

5.2. Classification Algorithms

In this study, we have applied 4 state-of-the-art (SOTA) classification algorithms and 1 traditional algorithm as a baseline. From the t-SNE analysis of our datasets, we have found that the relationships between the data points are very complex and tough to separate. That is why we have chosen the algorithms that have been proven to learn from complex relationships. As a baseline, we have used a well-tuned Decision Tree [12]. Among the 4 SOTA algorithms, XGBoost [13], LightGBM [14], and CatBoost [15] are the tree-based ensemble boosting algorithms. The remaining one is TabNet [16] which is a deep tabular data learning network. Using the Decision Tree model, we will assess the training and predictive performance of the traditional algorithm and set a baseline performance score for the advanced SOTA model. Later, we will observe how well the SOTA models perform against the traditional models. On the other hand, XGBoost, LightGBM, and CatBoost have been showing outperforming results in most of the data mining competitions and applied research compared to the traditional models. If we talk about the deep learning (DL) algorithms, there is a different scenario that DL algorithms do not perform well on the tabular data. That is why, we have used TabNet, the most recently developed DL algorithm by the google engineers that shows outperforming results in terms of tabular data. In this study, we will compare the training and validation results of all the models and find the best performing one.

5.3. Bayesian Hyperparameter Optimization

Hyperparameter optimization is the process of finding the optimal combination of the hyperparameters that minimize cost functions. Setting the right combination of hyperparameters and the right hyperparameter values for the model significantly improves the performance of the model in most of the cases. Without proper setting of model hyperparameters, it can show overfitted and biased results. In this study, we have used a distributed Bayesian optimization [17] approach for finding the optimal hyperparameter combination and values for each of the algorithms. The purpose of choosing a Bayesian approach is that it efficiently finds better hyperparameters in less time because it reasons about the best set of hyperparameters to evaluate based on the past trials.

5.4. Evaluation Metrics

We have used 8 metrics to evaluate the performance of the models. The metrics are True Positive Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), True Positive Rates (TPR) aka. Recall, ROC-AUC Score, F2-Score, Precision, and Accuracy. Since the dataset has class imbalance issues, all the metrics do not serve the business purpose of the problem. For example, accuracy and precision will not reflect the true learning and predictive capability of the models. Moreover, the business objective is not to misclassify the instances that have a high probability to return the order. Therefore, we have chosen TPR, ROC-AUC Score, and F2-Score to compare the performance of the models and make a decision for the best model.

6. Result Analysis

In this study, we have conducted two experiments over the two different versions of the dataset. The results of the experiments have been recorded in terms of training and validation scores for each of the aforementioned metrics. The following table 1 shows the results for the original (imbalanced) data.

Table 1. Results for the original (imbalanced) data.

		TNR	FPR	FNR	TPR	ROC-AUC	F2-Score	Precision	Accuracy
DecisionTree	Training	0.736	0.264	0.545	0.455	0.595	0.385	0.238	0.693
	Validation	0.753	0.247	0.547	0.453	0.603	0.389	0.249	0.707
XGBoost	Training	0.681	0.319	0.305	0.695	0.688	0.539	0.283	0.683
	Validation	0.686	0.314	0.349	0.651	0.669	0.510	0.273	0.681
LightGBM	Training	0.599	0.401	0.213	0.787	0.693	0.563	0.263	0.628
	Validation	0.595	0.405	0.264	0.736	0.666	0.528	0.248	0.617
CatBoost	Training	0.606	0.394	0.261	0.739	0.673	0.535	0.254	0.627
	Validation	0.608	0.392	0.313	0.687	0.648	0.502	0.241	0.621
TabNet	Training	0.601	0.399	0.270	**0.730**	**0.666**	**0.527**	0.249	0.621
	Validation	0.605	0.395	0.254	**0.746**	**0.676**	**0.527**	0.255	0.627

From table 1, we can see that in terms of TPR, ROC-AUC, and F2-Score all the models have performed better than the Decision Tree baseline. Among the tree-based ensemble boosting algorithms, LightGBM has yielded slightly better results than XGBoost and CatBoost. On the other hand, TabNet has yielded outperforming results than all other models. However, we can observe that in some cases LightGBM performed better than TabNet in terms of training scores but not in validation. Since the train-validation scores are more stable in TabNet we will select TabNet as the best model for the original (imbalanced) data.

Table 2 shows the performance of the models for the resampled (balanced) data. From table 2 we can see very similar types of results like table 1, however, important notes can be taken for Decision Tree and TabNet. In the balanced data, the Decision Tree performs far better than the performance on the imbalanced one. It performs very close to the tree-based ensemble boosting models. From this phase, we can infer that these types of advanced tree-based ensemble methods may perform better on imbalanced data. As like previous, LightGBM still yielded a bit better result than XGBoost, LightGBM, and CatBoost. On the other hand, TabNet has shown a significant improvement in terms of TPR (=0.83) and F2-Score (=0.76). However, LightGBM performs better ROC-AUC score (=0.67) than TabNet. As TabNet performs significantly better in TPR and F2-Score, we choose it as the best model for the resampled (balanced) data.

Table 2. Results for the resampled (balanced) data.

		TNR	FPR	FNR	TPR	ROC-AUC	F2-Score	Precision	Accuracy
DecisionTree	Training	0.580	0.420	0.271	0.729	0.654	0.708	0.634	0.654
	Validation	0.570	0.430	0.280	0.720	0.645	0.699	0.626	0.645
XGBoost	Training	0.592	0.408	0.250	0.750	0.671	0.727	0.648	0.671
	Validation	0.580	0.420	0.264	0.736	0.658	0.727	0.637	0.658
LightGBM	Training	0.610	0.390	0.276	0.724	0.667	0.708	0.650	0.667
	Validation	0.619	0.381	0.277	0.723	0.671	0.708	0.655	0.671
CatBoost	Training	0.562	0.438	0.211	0.789	**0.675**	0.755	0.643	0.675
	Validation	0.528	0.472	0.280	0.720	**0.624**	0.755	0.604	0.624
TabNet	Training	0.470	0.530	0.179	**0.821**	0.645	**0.767**	0.608	0.645
	Validation	0.489	0.511	0.169	**0.831**	0.660	**0.767**	0.619	0.660

We have also extracted the top 5 features and their importance from the trained TabNet model to understand the most impactful features. Table 3 shows the top 5 features according to their importance for both of the datasets.

Table 3. Results for the resampled (balanced) data.

	TabNet on Original (imbalanced) Data		TabNet on Resampled (balanced) Data	
rank	columns	importance	Columns	importance
1	order_location	0.427213	order_location	0.506463
2	payment_method	0.386938	payment_method	0.217543
3	confirmation_day_name	0.087183	order_day_name	0.140589
4	order_medium	0.038991	confirmation_day_name	0.051143
5	promotional_order	0.027802	confirmation_hour	0.044241

From the feature importance table, we can see that ***"order_location"*** and ***"payment_method"*** are the most influential features in both of the datasets. Since, the rest of the features are not very significant contributor, we have given a deeper dive into these two features. Our findings are:

a. order_location: 56% (=866) instances among the 1535 instances who have returned orders are located in Dhaka. Since, Dhaka is the most populous city in Bangladesh and most of the orders are placed from Dhaka, it has the highest return influence. Moreover, we have found that after Gazipur, Chittagong City, Rangpur City are the most order return places after Dhaka and all of these are big city areas.

b. payment_method: 98% (=1508) instances among the 1535 instances who have returned orders have used cash on delivery (MPD) method for orders. That means those who do not intend to pay in advance are very likely to return the order.

c. order_location and payment_method: 55% (=849) of the instances that are located in Dhaka and also used cash on delivery returned the orders.

We have also performed another model-agnostic approach called Morris Sensitivity Analysis to find the most influential features. The Morris method computes global sensitivity using a set of local derivatives taken at points sampled throughout the parameter space. It gives a sensitivity score describing the relative importance of each column in determining output variability. The following figure 3 shows the rank and Morris sensitivity score for each of the features.

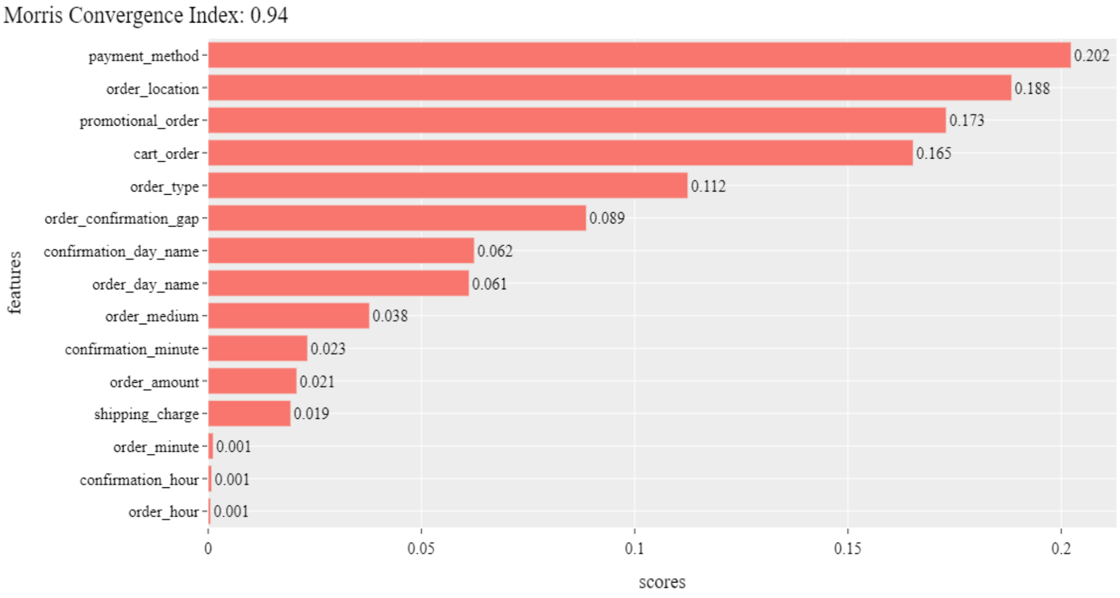

Fig. 3. Morris sensitivity score

From fig 3 we can observe that in terms of Morris sensitivity the “payment_method” and “order_location” are still the top two influential features. The next top two influential features are “promotional_order” and “cart_order”. We have also given a deeper dive into these two features. The findings are:

a. promotional_order: 51% (=789) instances among the 1535 instances who have returned took promotional orders. Selling out of trend products or rejected products through promotion might lead to return the orders.
b. cart_orders: 62% (=953) among the 1535 instances who have returned orders did not place cart orders. They mostly bought single items through the “Buy Now” option. It is apparent that when people order multiple products in a single order, they are less likely to return the orders.

7. Conclusion

The purpose of the study was to model one of the most common business problems in the E-tailing industry named order return. In the study, we have applied four SOTA algorithms along with one traditional baseline algorithm to model the prediction problem. Among all the models XGBoost and CatBoost performed very similarly while LightGBM produced a better result. The deep learning-based algorithm, TabNet, yielded the best results among all of the models. Since the dataset contained class imbalance issues, a new version was created using stratified random sampling. The best results were recorded from the balanced data where TabNet outperformed all other models with better TPR (=0.83), F2-Score (=0.76), and ROC-AUC score (=0.67). “order_location” and “payment_method” was found to be the most influential features for modeling.

In the future, we would like to employ approaches from anomaly detection methods and other state-of-the-art deep learning methods.

References

[1] Dzyabura, Daria, et al. Leveraging the Power of Images in Managing Product Return Rates. No. w0259. 2019.

[2] Li, Jianbo, Jingrui He, and Yada Zhu. "E-tail product return prediction via hypergraph-based local graph cut." Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018.

[3] Cui, Hailong, Sampath Rajagopalan, and Amy R. Ward. "Predicting product return volume using machine learning methods." European Journal of Operational Research 281.3 (2020): 612-627.

[4] Sahoo, Nachiketa, Chrysanthos Dellarocas, and Shuba Srinivasan. "The impact of online product reviews on product returns." Information Systems Research 29.3 (2018): 723-738.

[5] Kedia, Sajan, Manchit Madan, and Sumit Borar. "Early Bird Catches the Worm: Predicting Returns Even Before Purchase in Fashion E-commerce." arXiv preprint arXiv:1906.12128 (2019).

[6] Urbanke, Patrick, Johann Kranz, and Lutz Kolbe. "Predicting product returns in e-commerce: the contribution of mahalanobis feature extraction." (2015).

[7] Hess, James D., and Glenn E. Mayhew. "Modeling merchandise returns in direct marketing." Journal of Direct Marketing 11.2 (1997): 20-35.

[8] Dzyabura, Daria, et al. Leveraging the Power of Images in Managing Product Return Rates. No. w0259. 2019.

[9] Lemaître, Guillaume, Fernando Nogueira, and Christos K. Aridas. "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning." The Journal of Machine Learning Research 18.1 (2017): 559-563.

[10] Yeo, In-Kwon, and Richard A. Johnson. "A new family of power transformations to improve normality or symmetry." Biometrika 87.4 (2000): 954-959.

[11] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.Nov (2008): 2579-2605.

[12] Quinlan, J. Ross. "Simplifying decision trees." International journal of man-machine studies 27.3 (1987): 221-234.

[13] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

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

[15] Prokhorenkova, Liudmila, et al. "CatBoost: unbiased boosting with categorical features." Advances in neural information processing systems. 2018.

[17] Arik, Sercan O., and Tomas Pfister. "TabNet: Attentive Interpretable Tabular Learning." arXiv preprint arXiv:1908.07442 (2019).

[18] Bergstra, James, Daniel Yamins, and David Daniel Cox. "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures." (2013).