Table of Contents

1. Business context

1.1 Business Understanding

1.1.1 Company background

Nile, a South American e-commerce platform provides an extensive range of consumer goods and focuses on constantly improving its services by regular customer feedback and staying relevant in this competitive market. To do so, they have hired an analytics team to create a predictive model using machine learning to identify customers who are most likely to provide positive reviews. This would help in developing and improving their platform by allocating sufficient resources wherever required and enhance their reputation. They also focus on leveraging the overall experience, increasing customer satisfaction while reducing costs.

1.1.2 Objectives

This project aims to analyse multiple datasets. combine these datasets as per required to generate useful insights and strategies to improve product performance and customer satisfaction. By adopting suitable machine learning model, we must ensure overall customer engagement and product visibility is improved. This is done mainly by focusing on customers who are likely to provide positive feedback for products.

1.1.3 CRISP-DM framework

Using the CRISP-DM framework, Nile can evaluate for most important parameters in a systematic order and develop a predictive model to identify customers who tend to leave positive reviews. This involves understanding of the business objectives, so that required technical approach can be adopted. Next, we need to gather relevant data which is then pre-processed and combined to obtain useful information and patterns. Different machine learning models are employed and model accuracy, scores are compared to optimise customer usability.

2. Data Engineering

2.1 Data Understanding

The dataset consists of multiple CSV files including detailed information on customers, geolocation, order, items, product details, sellers, order reviews. We observe closely and understand that each of them contains different points of view of the business.

2.2 Data Preparation

To achieve a successful predictive model, it is crucial that we clean the data as per required. Therefore, we try to remove any such row or modify those columns which may lead to low performance of our model.

2.2.1 Data Cleaning:

For data cleaning, firstly all datasets were loaded into Pandas data frame and then search for missing values in each column. After careful consideration, we found these areas to work on:

Duplicates: There can be multiple situations due to which duplicate values may arise in a dataset. For instance, when a customer pays using multiple payment methods for a single order in payment dataset and a customer order multiple items in a single order. This is resolved by grouping payments, and total amount of items by order id and aggregating the total amount. In the review's dataset, we obtain multiple reviews of the same product given at different timestamps, based on customer's change in opinions over the time.

Missing Values: Prior to data pre-processing, we had relatively few missing values. However, after applying left join, we observe significant increase in missing values. This is so because a left join retains all values from the primary dataset and again includes the matching values from the secondary dataset. Thus, when corresponding matching record is not found, it is filled with null values thereby increasing missing data.

Misspelling: Another fix essential for our data to be consistent is spelling corrections. We observe mistakes in names of cities, in sellers, customer, and geolocation dataset and thus implement systematic approach to correct them using coding. We use string matching and reference relevant datasets to standardize our dataset.

Joining Datasets for Analysis
Initially, we planned to implement inner join for our primary dataset, but this leads to data loss when corresponding record matches are not found. We switch to left joins for integrating datasets. We have not used the geolocation dataset as it is not relevant and surge data after merging. We have also introduced a translation file which converts Brasilia to English language for the product category to be uniform.

2.2.2 Feature Engineering:

To prepare the data for analysis, we created new related variables and converted string variables to numeric

Total Items: This column represents the total number of items purchased in every order. We obtain this by grouping the data by order id.

Historical Review: We create a new column called historical review to track customers who has review information history over time using the data from review comment title and review comment message due to the high number of missing values in these two columns. To solve this, we define customers who either have a title or a review message was assigned a score of 1, and 0 otherwise.

Time-Based column: We transform time-Based column to amount of time each activity use by minus two different column such as how long it takes to deliver a product.

Product Volume: A new feature was created by calculating the product's volume based on the given dimensions such as width, height and length.

Dummy variables with few different answers: Change dummy variables to numeric using one-hot encoding for order status and customer state.

Dummy variables with many different answers: For Product Category, there are many different types of product categories. If we use one-hot encoding, it may increase noise in the model. We applied frequency encoding to handle the high cardinality of the features.

Dropping columns:
We are removing unnecessary columns for efficiency:

1. ID Columns: Used only for merging and are no longer needed afterwards.
2. Original Activity Columns: Dropped after creating new time-based variables.
3. Zip Code: Redundant due to city and state, with many unique values.
4. Payment Type: after group to resolves duplicates it simplified into grouped categories and unsuitable for one-hot encoding or frequency coding.

2.2.3 Data Normalization:

We need to supply values across different features on a similar scale for evaluation. This approach helps prevent any bias cause from features having smaller or larger ranges. To ensure numeric values are on the same scale, we apply min-max normalisation, which scales values between 0 and 1.

3. Analytical Modelling

Target Variable and Features:

Positive feedback was indicated by assigning a value of 1 to review scores of 4 and 5 datasets. A value of 0 (indicating negative feedback) was assigned to review scores in datasets 1-3. To guarantee a distinct division between predictors and results, the objective was not included in the dataset and the remaining attributes were utilized as input variables (X).

3.1 Modelling

Algorithm Approach

We tried using four classification algorithms for modelling, Logistic Regression (LogR), Random Forest (RF), Gradient Boosted Decision Trees (GBDT), Extreme Gradient Boosting (XGB). The main aim was to balance the scalability, performance, and interpretability. Each of them was trained to establish performance metrics, which were accuracy, macro precision, macro recall, and macro F1-score. We cannot use Support Vector Machine due to problems with scaling up the large dataset. DTC was easy to understand but lacked toughness and stability.

3.2 Addressing Data Imbalance:

The dataset was highly imbalanced with 79.4% of the reviews being positive in the dataset. Macro-averaged precision assigned equal weights to both classes, therefore reducing bias towards the majority class and fairly evaluating the minority class at the same time. To offset class imbalance and measure performance equally across every class irrespective of their score, macro precision was the target metric to minimize anticipation bias on positive predictions for both classes.

X variables		1	2	3
historical_review		√	√	
price		√	√	√
freight_value		√	√	√
total_items		√	√	√
customer_history		√	√	
product_name_lenght		√	√	
product_description_lenght		√	√	
product_photos_qty		√	√	√
product_weight_g		√	√	√
product_length_cm		√	√	
product_height_cm		√	√	
product_width_cm		√	√	
payment_sequential		√	√	
payment_installments		√	√	√
payment_value		√	√	√
time_to_approve		√	√	
time_to_deliver_to_carrier		√	√	
time_to_deliver_to_customer		√	√	√
early_delivery_days		√	√	
review_response_time		√	√	
time_to_seller_to_carrier		√	√	
product_volumn		√	√	
status_delivered		√	√	
state	One-hot encoding	√		√
	Frequency encoding		√	
customer_city	One-hot encoding			
	Frequency encoding	√	√	
product_category_name_freq		√		√
inner join		√	√	
left join in review, item, payment				√

Table 1: Different data frame using for Modelling

Initially, we run the model using inner join with all the columns obtained from the data engineering process with respect to avoiding occurrences of missing values. From data frame1 and 2, we examine using frequency coding and one-hot encoding to compare the scores. The one with hot encoding gets a higher score in most cases except for random forest results from overfitting due to high dimensionality and sparsity.

In set of data frame 3, we use a correlation matrix to identify which X variables were most correlated with the target variable y, positively and negatively, and select the most relevant X variable.

- Time-related variables, "time delivery to customer" had the highest correlation with a score of (value).
- Product volume instead of separate dimensions such as width, length, and height.
- Data related to products, namely, product name length, description length, and photos quantity. We select the highest score.

- Data related to payment, like payment instalment and sequential, we select the highest score.

The results of this set of X5 variables performed best, particularly in the GBDT model.

The investigation found that price, freight value, total items, product photos quantity, product weight, payment instalment, payment values, time delivery to customer, product volume, state of customer, and the frequency of product category name were the most critical factors in predicting customer review score.

3.3 Model Performance and Results

Algorithm	Accuracy of Training Data	Accuracy of Test Data
Logistic Regression	0.8181	0.8184
Random Forest	0.8010	0.8008
GBDT	0.8260	0.8201
XGBDT	0.8262	0.8199

Table 2: Accuracy of Algorithms Used

Data	Precision	Recall	F1 Score
Training Data	0.795	0.575	0.582
Testing Data	0.843	0.517	0.478

Table 3: GBDT Performance Evaluation on Testing and Training Data

The results show that GBDT model has higher accuracy than others in forecasting customer who tend to leave positive reviews, with accuracy of 82.00%. However, LogR, RF, and XGB also perform well, with respective accuracies of 81.84%, 80.08%, and 81.99%.

There are different assessment criteria using to utilized to assess the GBDT model, which are accuracy of 82.67%, a precision score of 77.22%, a recall score of 59.45%, and a F1-score of 61.10%.

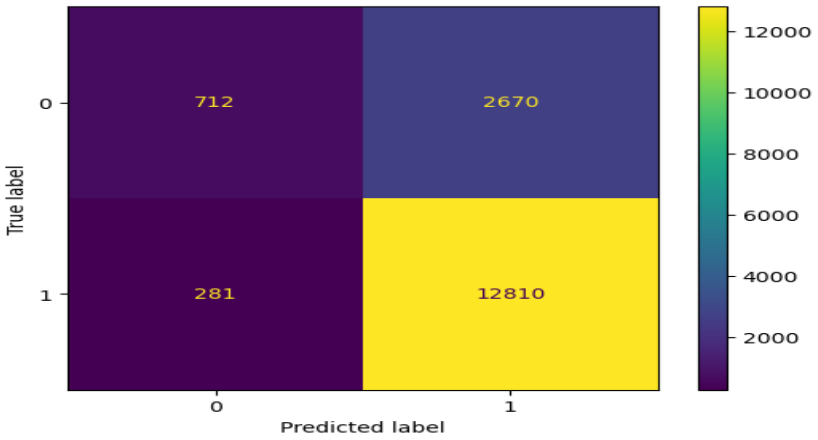

Fig. 1: GBDT Confusion Matrix Interpretation

The confusion matrices of the models evaluate how well the models predicts customer feedback. The GBDT model revealed that it accurately predicted 12,810 customer who were correctly identified as suitable for requesting feedback and provided a positive review score, and 721 customer of false negatives who were correctly identified as unsuitable for requesting feedback. However, the number of error shows in 281 customers for false negative who were incorrectly identified as unsuitable for requesting feedback, but who would have provided a positive review score if asked. Additionally, we should specifically focus on the number of customers who were incorrectly identified as suitable for requesting feedback but provided a negative review score in false positive around 2,670 customers.

3.4 Model Summary

According to indicators from a good precision rate, the GBDT model was able to accurately forecast both the customer who tend to giving a positive and negative review score. Moreover, an F-1 score indicates that there should be an improvement in balancing precision and recall.

3.5 Evaluation

3.5.1 Evaluation and Metrics:

After retraining with optimal parameters, models were evaluated again. Despite outstanding macro precision, the accuracy was suboptimal due to high false positives, many clients predicted to leave positive feedback provided unfavourable reviews. Further improvements may include noise reduction, feature refinement to increase predictive performance.

3.5.2 Future plan for evaluation

To improve our model, we include other variables in the model, mean review scores for product ID, customer unique ID, customer city, seller city, and seller ID. This makes prediction more precise but can also lead to problems related to multicollinearity and data leakage, concerning linear regression models. Initial attempts in incorporating those variables caused performance issues on test data, the future work involves tackling those challenges to gain effectiveness in model's predictive power.

4. Implementation

It is important to ensure that it functions correctly with real-time predictions, and large datasets, as Nile is a big e-commerce platform that data runs all the time. For now, we prefer GBDT due to model accuracy and scores obtained with our computational power, but later as the dataset increases, we may use XGB algorithm for complex optimisations in future scenario.

4.1 Deployment

In real-world situations, it may detect any degradation over time as different sets of datasets are changed, particularly, processing time. The current model performs effectively with less time of processing, but to accommodate the company's growth, which might bring a huge amount of data processing, the model needs to be adjusted for proper processing time, trading off the efficient performance of high traffic and scale.

Appendix

Algorithm	Precision	Recall	F1 Score
Logistic Regression	0.795	0.575	0.582
Random Forest	0.843	0.517	0.478
GBDT	0.797	0.601	0.620
XGBDT	0.815	0.595	0.613

Table 4: Algorithm Performance Evaluation on Training Data

Algorithm	Precision	Recall	F1 Score
Logistic Regression	0.794	0.576	0.584
Random Forest	0.845	0.516	0.477
GBDT	0.768	0.594	0.610
XGBDT	0.783	0.585	0.598

Table 5: Algorithm Performance Evaluation on Test Data

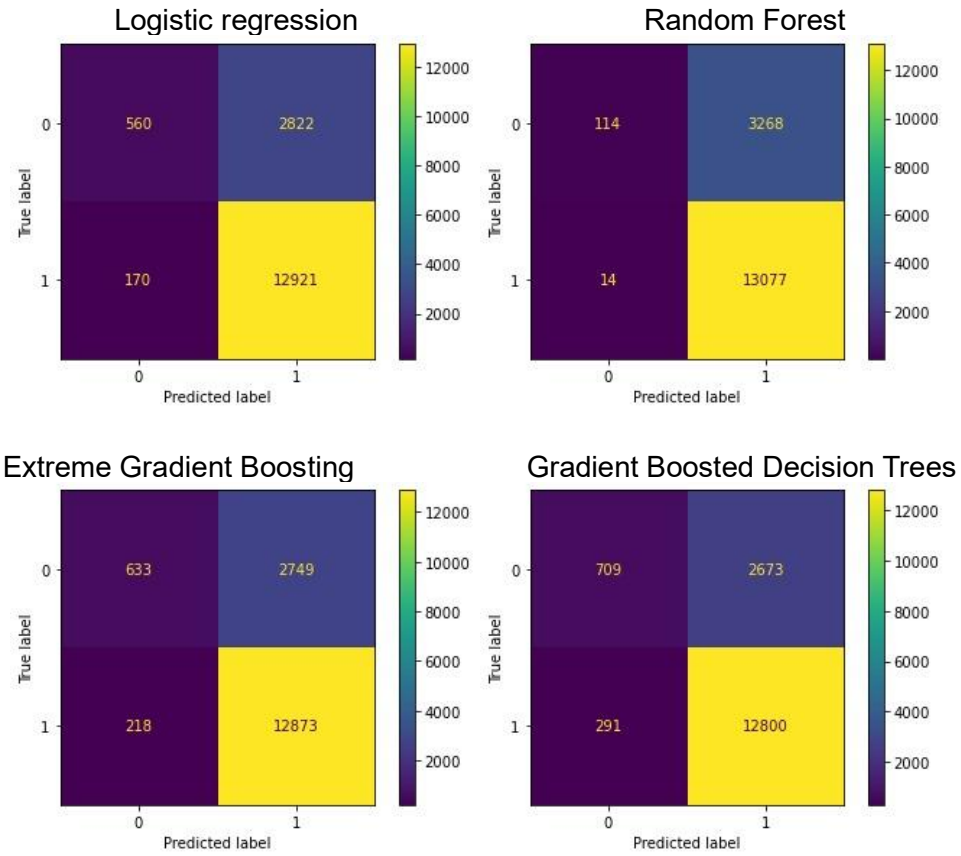

Fig. 2: Confusion Matrix Interpretation of data frame 3

References

Hotz, N. (2018). *What is CRISP DM? - Data Science PM*. [online] Data Science PM. Available at: https://www.datascience-pm.com/crisp-dm-2/?form=MG0AV3 [Accessed 29 Nov. 2024].

Tableau. (2024). *Guide To Data Cleaning: Definition, Benefits, Components, And How To Clean Your Data*. [online] Available at: https://www.tableau.com/learn/articles/what-is-data-cleaning?form=MG0AV3 [Accessed 29 Nov. 2024].

DataSpace Academy. (2024). *Data Cleaning 101: Key Dos & Don'ts for Flawless Results - DataSpace Academy*. [online] Available at: https://dataspaceacademy.com/blog/data-cleaning-101-key-dos-donts-for-flawless-results?form=MG0AV3 [Accessed 29 Nov. 2024].

Team, D. (2022). *Data Cleaning Tutorial*. [online] Datacamp.com. Available at: https://www.datacamp.com/tutorial/tutorial-data-cleaning-tutorial?form=MG0AV3https://www.datacamp.com/tutorial/tutorial-data-cleaning-tutorial?form=MG0AV3 [Accessed 29 Nov. 2024].

Brownlee, J. (2019). *A Tour of Machine Learning Algorithms*. [online] MachineLearningMastery.com. Available at: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/?form=MG0AV3 [Accessed 29 Nov. 2024].

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science, 2(160). Retrieved from https://link.springer.com/article/10.1007/s42979-021-00592-x