MASTER'S PROGRAMME: AIP GROUP ASSIGNMENT

Student Numbers: Please list numbers of all group members	**5663618, 5663426, 5673992, 5616183, 5669443, 5648941**
Module Code:	**IB9BW0**
Module Title:	**Analytics in Practice**
Submission Deadline:	**Monday 2nd December at 12:00:00**
Date Submitted:	**Sunday 1st December**
Word Count:	**2093**
Number of Pages:	**22**
Have you used Artificial Intelligence (AI) in any part of this assignment?	**Yes, AI was used to refine code and structure complex sentences.**

Academic Integrity Declaration

We're part of an academic community at Warwick. Whether studying, teaching, or researching, we're all taking part in an expert conversation which must meet standards of academic integrity. When we all meet these standards, we can take pride in our own academic achievements, as individuals and as an academic community.

Academic integrity means committing to honesty in academic work, giving credit where we've used others' ideas and being proud of our own achievements.

In submitting my work, I confirm that:

- I have read the guidance on academic integrity provided in the Student Handbook and understand the University regulations in relation to Academic Integrity. I am aware of the potential consequences of Academic Misconduct.
- I declare that this work is being submitted on behalf of my group and is all our own, , except where I have stated otherwise.
- No substantial part(s) of the work submitted here has also been submitted by me in other credit bearing assessments courses of study (other than in certain cases of a resubmission of a piece of work), and I acknowledge that if this has been done this may lead to an appropriate sanction.
- Where a generative Artificial Intelligence such as ChatGPT has been used I confirm I have abided by both the University guidance and specific requirements as set out in the Student Handbook and the Assessment brief. I have clearly acknowledged the use of any generative Artificial Intelligence in my submission, my reasoning for using it and which generative AI (or AIs) I have used. Except where indicated the work is otherwise entirely my own.
- I understand that should this piece of work raise concerns requiring investigation in relation to any of points above, it is possible that other work I have submitted for assessment will be checked, even if marks (provisional or confirmed) have been published.
- Where a proof-reader, paid or unpaid was used, I confirm that the proof-reader was made aware of and has complied with the University's proofreading policy.

Upon electronic submission of your assessment you will be required to agree to the statements above

PREDICTING POSITIVE REVIEWS FOR NILE E-COMMERCE: TECHNICAL REPORT

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This report is an in-depth analysis that seeks to understand the ecommerce Nile business problem and find the main drivers to predict good reviews by using Random Forest, Gradient Boosted Decision Trees (GBDT), and Extreme Gradient Boosting (XGBoost) machine learning models on Python. It presents a holistic approach to give optimum results using the SMOTE method, Bayesian Optimisation and Randomized SearchCV. This model will enable Nile to address customer pain points proactively. As a result, the random forest model showed consistent performance with reduced variability, thus providing reliable predictions for strategic decision-making.

BUSINESS UNDERSTANDING

In a business environment, decisions are often based on intuition. Rather than relying purely on theory to base a strategy, data-driven analysis of consumer reviews can make or break a business. It's worth noting that 76% of Gen Zs put more weight on a review online while making a purchase decision (Forbes, 2024). Product quality, description, time, and images attract this generation the most. Thus, any e-commerce platform needs to understand the needs of its target audience and learn every day what influences a customer's decision.

Nile, a South American eCommerce platform, is the focus firm of this report. It has an established database of historical data; this report has infused it into a predictive machine-learning model. As a foundational tool, this model enables Nile to adopt a data-driven approach to understand customer's pain points, better reviews, and sustained business growth.

We followed the timeline shown in Figure 1 to fulfil the task efficiently and effectively.

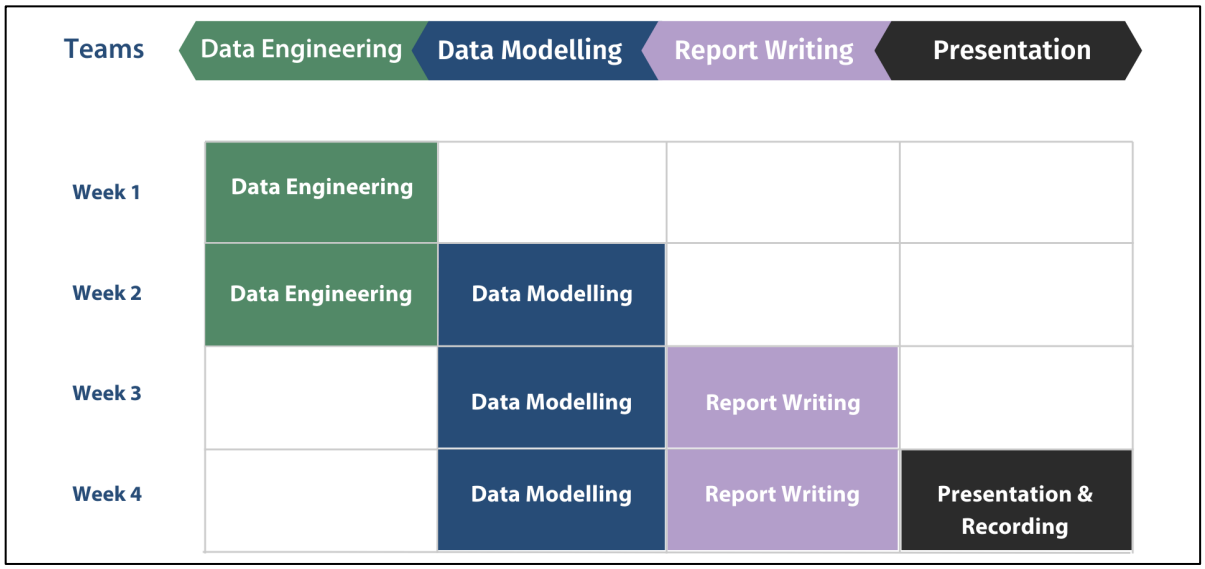


Figure 1: Timeline of CRISP-DM

DATA UNDERSTANDING & DATA PREPARATION

Data Source and Understanding

The initial stage of the analysis focused on thoroughly understanding the datasets provided by Nile. To create a comprehensive and unified dataset, we merged nine distinct data sources: Customer Data, Geolocation Data, Order Items Data, Order Payments Data, Order Reviews Data, Orders Data, Product Data, Seller Data, and Product Category Names. This MS Excel and Tableau (Figure 2) exploration facilitated a distinctive view of the data, enabling us to establish relationships between critical features such as customers, orders, products, and sellers and incorporate additional insights from geolocation and review information. More details on each field can be found in Appendix A.

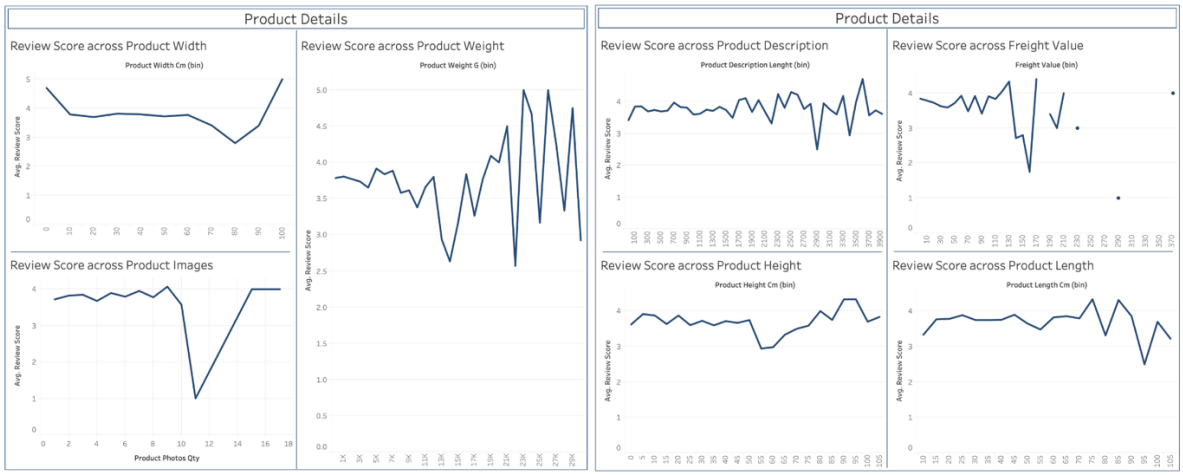


Figure 2: Data Exploration on Tableau

Data Cleaning & Pre-Processing

When understanding the data, we identified the overlaps as duplicates, missing variables, and numerical data in the set. Several pre-processing steps were undertaken in this analysis before analysing the data into machine learning models. The mapping of data sets was done as shown in Figure 3.

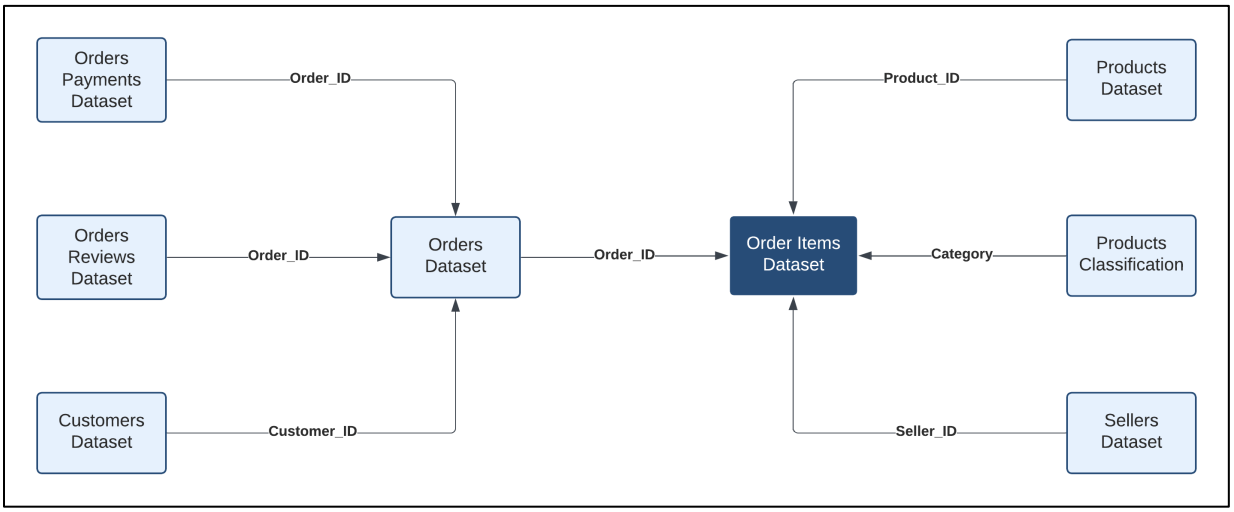


Figure 3: Data Mapping Flowchart

As a part of the exploratory, descriptive analysis of the percentage as a metric, we checked the database, its missing values and duplicates in all columns-rows, and the unique values for indexing each row (Refer Appendix B for % values). A standard scaling approach was applied to address the varying ranges of negative and positive numerical data fields, which can negatively affect model performance with higher values. This normalisation method used in Python transforms numerical data into a mean of 0 and a standard deviation of 1 with a range of -1 to 1 to work with negative values rather than focusing on far outliers, ensuring all features contribute equally to the model training process (Bolikulov, Nasimov, Rashidov, Akhmedov, & Cho, 2024).

In predictive modelling, segmentation breaks the dataset into subgroups with comparable traits. In addition to improving prediction accuracy, well-designed segmentation offers comprehensible patterns that can aid in decision-making (Provost & Fawcett, 2013). Here, we grouped the pre-existing 71 product categories into ten subgroups. This reduction in nominal categories, which do not have a natural order, subgrouping reduces sparsity in the model as we minimise the number of columns with 0 values when implementing one-hot encoding (Figure 4), improving model performance.

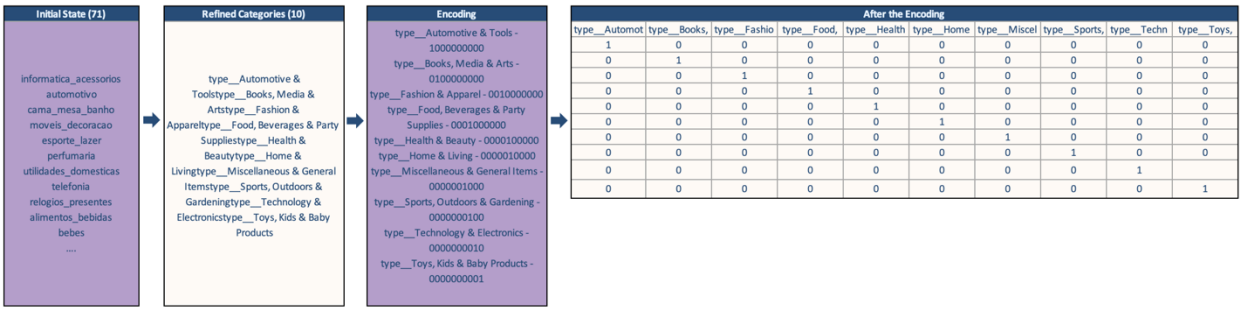


Figure 4: One-Hot Encoding

Binary coding choice, with 1 indicating the presence of a category and 0 indicating its absence, condenses multiple categories in columns to make the data more suitable for machine learning models. This approach ensures that each category is represented without introducing redundancy, which is critical for our chosen decision tree approach.

Feature Engineering

In the machine learning spectrum, feature engineering gives a model more meaningful insights into latent patterns and available data. Temporal or time-based (Figure 5) features such as "time since order place" and "date difference" were engineered to provide insights into customer behaviour trends. These temporal features were computed by mapping data across related columns, creating variables that made unusable sets into actionable features (Refer to Appendix C).

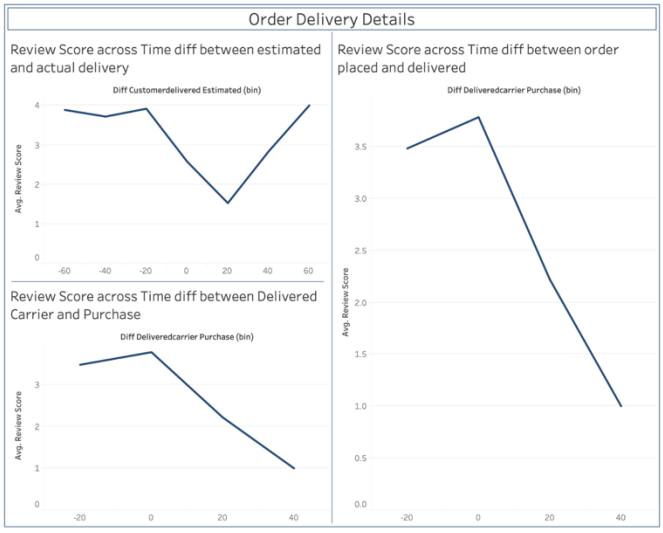


Figure 5: Review Scores across Temporal Features

Feature selection and engineering were integral to refining the dataset for optimal model performance. Additionally, feature importance scores derived from preliminary model iterations helped prioritise excluding irrelevant variables like Geolocation that were eventually dropped in the merged dataset.

Since GBDT and XGBoost are less sensitive to high dimensionality than linear models, reducing dimensions will create redundancy. Hence, we included 23 features in the model (Refer to Appendix C).

MODELLING

Predictive modelling involves building a simplified representation of reality to capture critical relationships between the target variable and descriptive attributes for a specific purpose (Provost & Fawcett, 2013). For the Nile project, the creation of models from data aligns with the concept of model induction, where the goal is to generate predictive models based on the provided dataset. Given the nature of the business problem, which involves categorising customer behaviour as either Class 1 (“Good Reviews" (4-5)) or Class 0 (“Bad Reviews" (1-3)), classification models were chosen.

Model Selection

Decision trees can be broadly categorised based on the output variable type: classification trees, which predict categorical outputs, and regression trees, which predict numerical outputs. In this project, we exclusively focused on classification models, including Gradient Boosted Decision Trees (GBDT), Random Forest, and Extreme Gradient Boosting (XGBoost), as these methods are tailored to provide binary predictions. We identified that regression trees were accurate for predicting continuous numerical variables but unsuitable for the objective, requiring precise, discrete classifications of Class 1 or Class 0 rather than continuous values. This focus ensures that the models align with the binary nature of the dependent variable in the dataset.

Data Splitting

Random splitting was deemed the best strategy for splitting the dataset. This ensures that the data is distributed uniformly across subsets (Muraina, 2022). Our method of splitting the data followed a 70-30 split of training to testing data. Although the traditional practice is a split of 80-20 (training: testing), our approach of a 70-30 split was derived from assessing our data volume and the scores the model provided. We found that our scores were better with this split because the testing had more data to predict. Training was not needed as much due to the creation of synthetic data from oversampling.

Model Training & Validation

Nile’s customer review prediction model, which we developed, has hyperparameter tuning, which is critical in improving performance. Previously, we leveraged Bayesian optimisation for

hyperparameter tuning (Refer to Appendix F), a probabilistic model-based approach. However, we have adopted a random search method to achieve similar objectives.

We built two models on the same set of training data. First, there is hyperparameter tuning with the Randomized SearchCV technique, and second, there is hyperparameter tuning with a Bayesian Optimisation model. Randomised SearchCV, despite its simplicity, showed the best results, which will be further discussed in the results and evaluation section. Bayesian optimisation inherently balances exploration and exploitation (Turner(Twitter), et al., 2021) , reducing entropy and expected improvements in the model. However, a randomised search gave better results for Random Forest, GBDT, and XGBoost in this case. Random search enables rapid experimentation and avoids the overhead of building and maintaining a surrogate model, as required in Bayesian optimisation (Shi & Rasheed, 2010).

The main objective of using decision trees is to identify an underlying pattern. The Synthetic Minority Oversampling Technique (SMOTE) has been used to solve the problem of overrepresentation of Class 1 in the data set. The algorithm addresses data imbalance within the training dataset. Instead of duplicating instances from underrepresented Class 0, SMOTE creates synthetic samples to develop the minority class. These synthetic reviews are generated by interpolating feature values of similar reviews within a defined neighbourhood, ensuring they reflect realistic patterns and variations observed in the data (Castro, 2020). By rebalancing the dataset, SMOTE enhances the model's ability to learn from positive and negative reviews, improving prediction accuracy and reducing bias.

MODEL EVALUATION & RESULTS

The Random Forest Method gave us the most precise results. This methodology combines each series of tree classifiers to cast a unit vote for the most popular tree class. Then it combines these results to get final values in the optimum model. This possesses accuracy and tolerates outliers and noises in the model (Liu, Wang, & Zhang, 2012).

The model was evaluated comprehensively using multiple metrics to assess its performance. We decided to give precision the highest weightage, while accuracy was considered the second most important metric (Osisanwo, et al., 2017). Precision had more weight in the analysis because incorrectly predicting a “good” review (False Positive) could lead to inefficient resource allocation, such as wasted efforts in marketing and discount codes. Precision is a better use-case in our scenario than recall, as it helps ensure that the model predicts reliable positive reviews, once again recognising the objective stated in the brief.

Accuracy was used as a supporting metric to measure score reliability, further reinforced by our well-balanced data. Moreover, it is a metric that can be easily conveyed to non-technical stakeholders by assessing the model's performance simplistically. Additionally, we used metrics like recall, F1-score, and AUC-ROC(Refer to Appendix D) to draw insights from the model that can paint a well-rounded, broader picture.

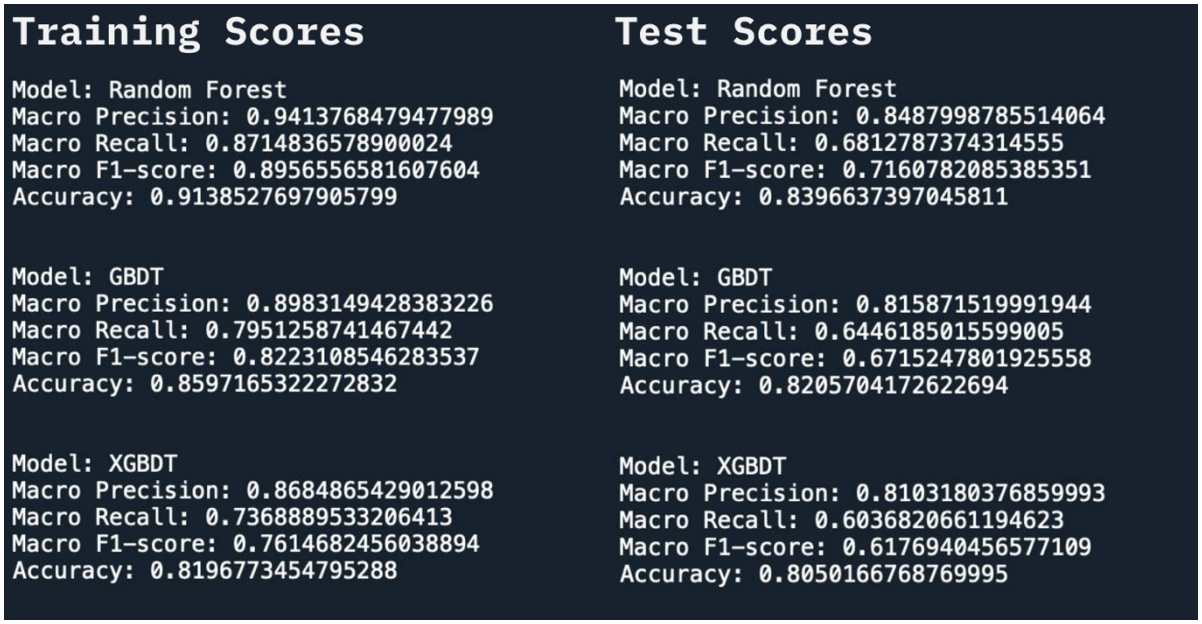


Figure 6: Model Results

Random Forest analysis had the best precision on the training set with a score of 0.941 on a 0-1 scale. With a recall value of 0.871 and an F1-score of 0.895, the model is performing exceptionally well. The high training values imply that the model identifies the inherent pattern. However, potential overfitting may need to be verified against test results. Here, precision dropped to 0.848, with a significant fall in the Recall score of 0.681 and macro of 0.716. However, it is worth noting that the features fitted in the model proved to be 83.97% accurate.

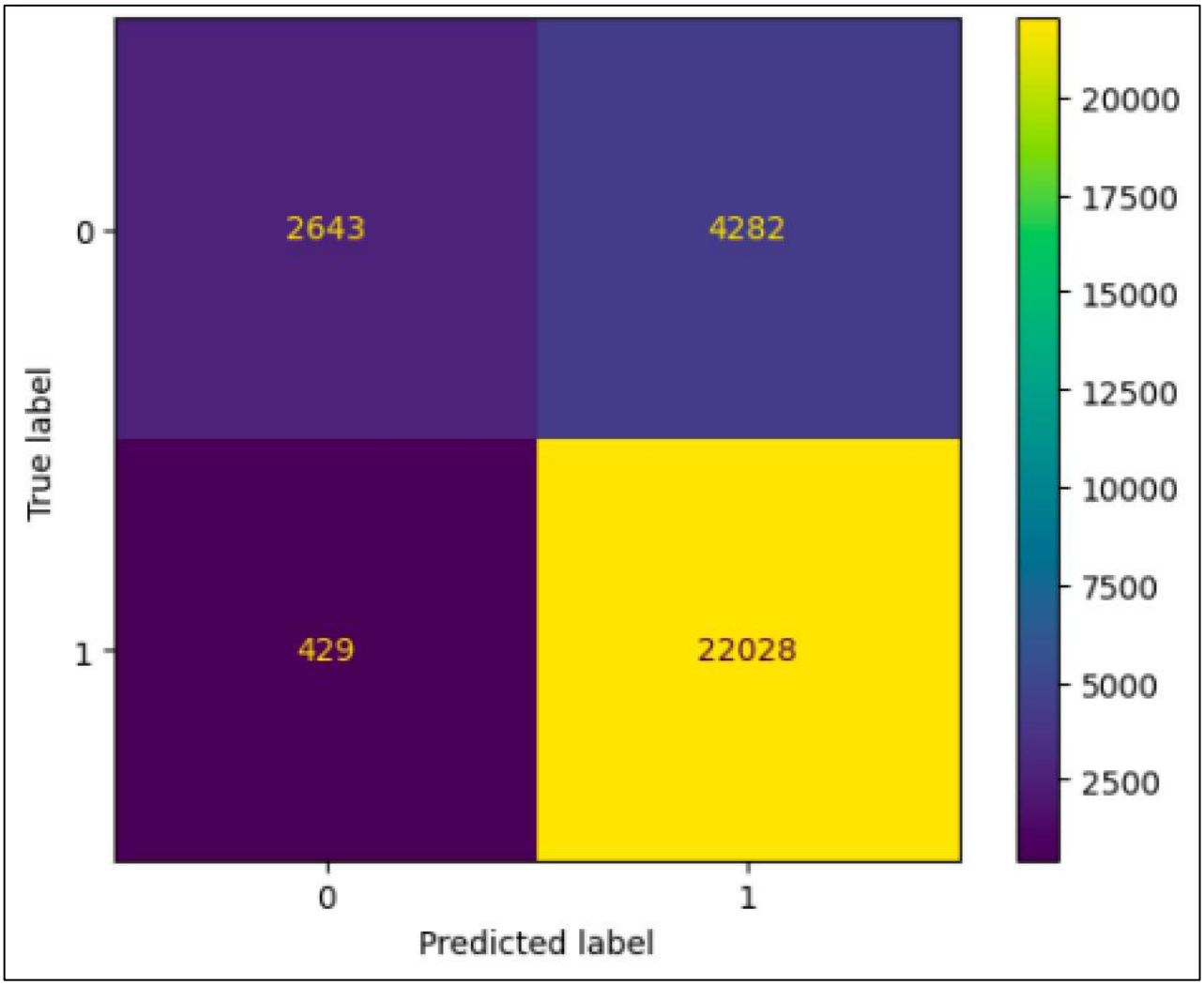


Figure 7: Confusion Matrix of Random Forest Method

Random Forest strongly relates to true positives (22,028) and minimises the misclassification of negative reviews as positives, i.e., keeping false positives at a low of (4,282).

Overall, other models, i.e. GBDT and XBDT, have a lower precision (Figure 6 & Appendix E). Hence, the Random Forest method has the most significant results for the model created with respect to the target variable.

CONCLUSION

This predictive model extracts and cleans factors influencing the target variable with encoded product categories as relevant features. Binary techniques with SMOTE, one-hot coding, randomising and Bayesian techniques provide a balanced, consistent, and efficient model for training. The classification outcomes will make Nile to make an actionable data-driven strategy and identify the factors contributing to negative customer feedback, eventually providing insights for customer satisfaction, reviews and retention.

Deployment and Limitations

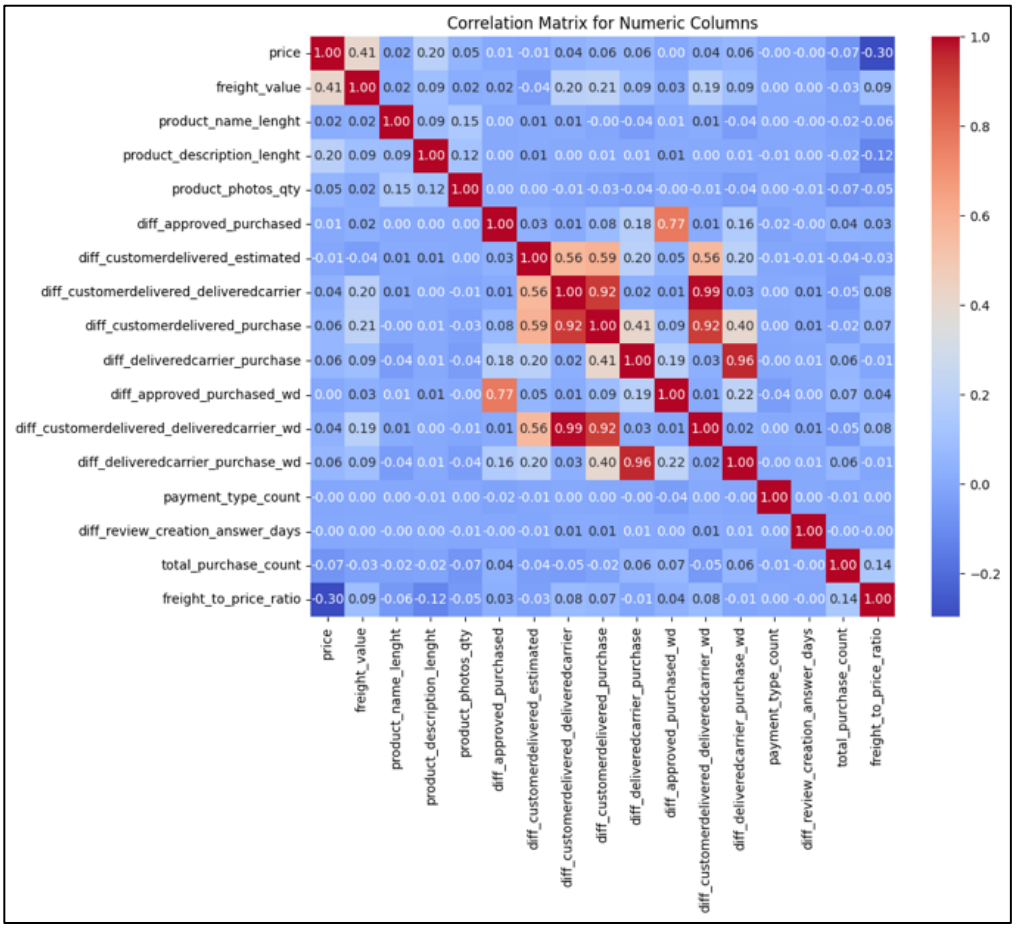


Figure 8: Correlation Matrix of Top 20-Features

In the correlation heat map, we can see that there are redundant variables. Although significant features were regenerated, co-dependent features could be dropped. For future iterations, we can remove highly correlated features and reduce noise in the data. In models built using SMOTE,

there is a risk of artificial generating of data patterns. This may impact the generalisability of the model. Careful considerations were made during feature engineering, but some latent patterns or interactions may eventually be left unexplored. Integrating a real-time pipeline and retraining mechanism would require more work in a real-world scenario.

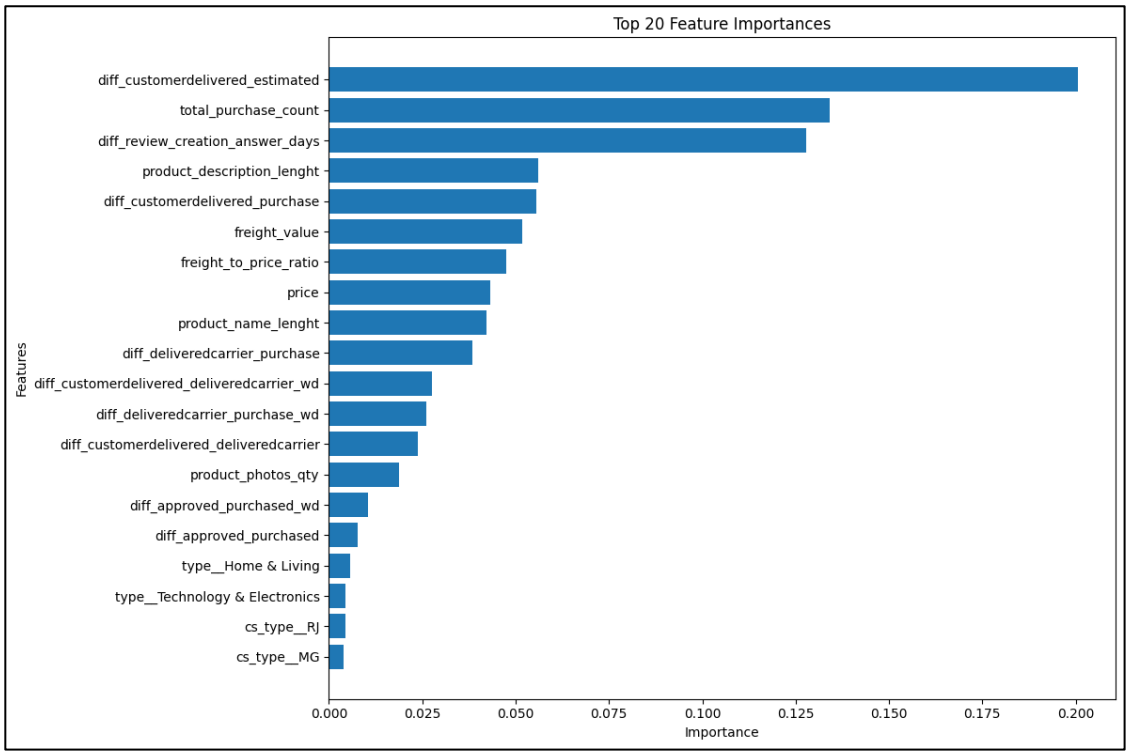


Figure 9: Ranking of Selected Features

As we can see in the graph, this report has concluded that Days between the date of delivery and the estimated date of delivery, Days between the date of notification sent for review and the date when the review was given, Length of Product Description Days between the delivery date and the date of order purchase and Freight value for the item were the top features responsible for predicting target variable (good reviews). The key recommendations for Nile include review analysis on their website home page, establishing a social proof of credibility with more reviews and incentivising genuine reviewers.

The future scope of the model can be refined to provide better precision, recall and accuracy for a scalable environment, making it a robust model in its training stage for overall improvement and long-term implementation. To have better predictions, text analysis methods on reviews written and make it into a holistic model in future.

REFERENCES

Bolikulov, F., Nasimov, R., Rashidov, A., Akhmedov, F., & Cho, Y.-I. (2024). Effective Methods of Categorical Data Encoding for Artificial Intelligence Algorithms. *Mathematics, https://doi.org/10.3390/math12162553*, 2553.

Provost, F., & Fawcett, T. (2013). *Data Science for Business : What You Need to Know about Data Mining and Data-Analytic Thinking* (1 ed.). Sebastopol, United States of America: O'Reilly Media, Incorporated.

Turner(Twitter), R., (Facebook), D. E., Michael McCourt (SigOpt, a. I., (Valohai), J. K., (Valohai), E. L., (4Paradigm), Z. X., & (ChaLearn), I. G. (2021). Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020. *Proceedings of Machine Learning Research, 133*, 3–26.

Candelieri, A., Ponti, A., & Archetti, F. (2023). Uncertainty quantification and exploration–exploitation trade-off in humans. *Journal of Ambient Intelligence and Humanized Computing, 14*, 6843–6876.

Castro, E. P. (2020). An Examination of the Smote and Other Smote-based Techniques That Use Synthetic Data to Oversample the Minority Class in the Context of Credit-Card Fraud Classification . *Technological University Dublin, Masters Dissertation*(10.21427/wj33-n221).

Shi, L., & Rasheed, K. (2010). A Survey of Fitness Approximation Methods Applied in Evolutionary Algorithms. In Y. T.-K. Goh (Ed.), *Adaptation Learning and Optimization* (pp. 3–28). Verlag Berlin Heidelberg: Springer.

Forbes, M. P. (2024, Feb 8). *Online Review Trends Affecting Today's Consumers*. Retrieved from Forbes Business Council : https://www.forbes.com/councils/forbesbusinesscouncil/2024/02/08/online-review-trends-affecting-todays-consumers/

Osisanwo, J.E.T., A., O, A., O., H. J., O., O., & J., A. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology (IJCTT)*, 128-138.

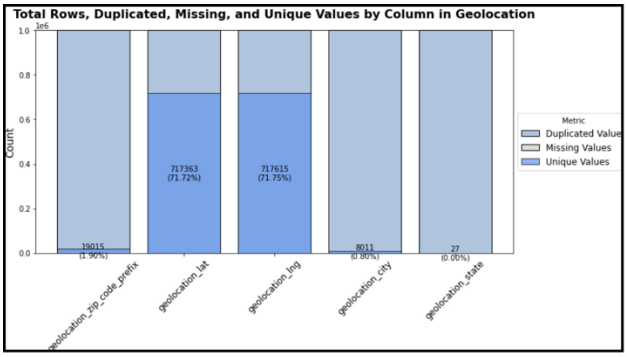
Muraina, I. O. (2022). Ideal Dataset Splitting Ratios in Machine Learning Algorithms: General Concerns for Data Scientists and Data Analysts. *International Mardin Artuklu Scientific Researches Conference*, 496-504.

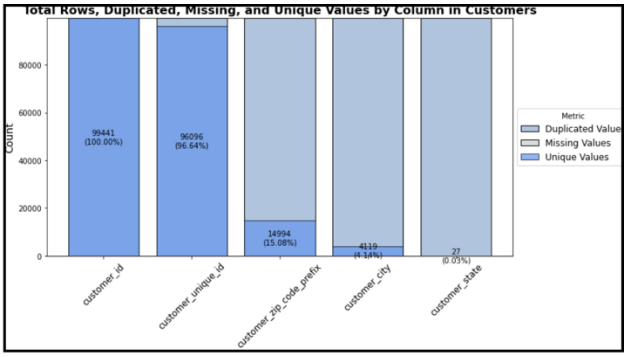
APPENDIX

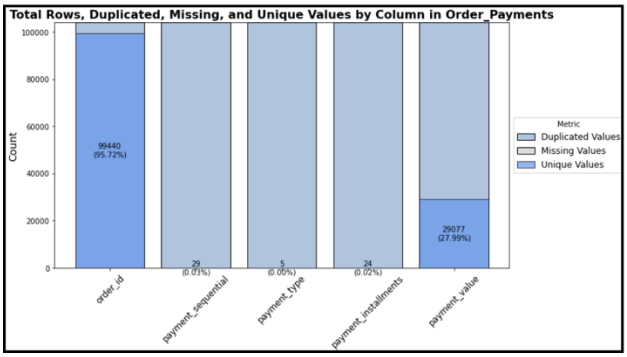
Appendix A: Types of Data

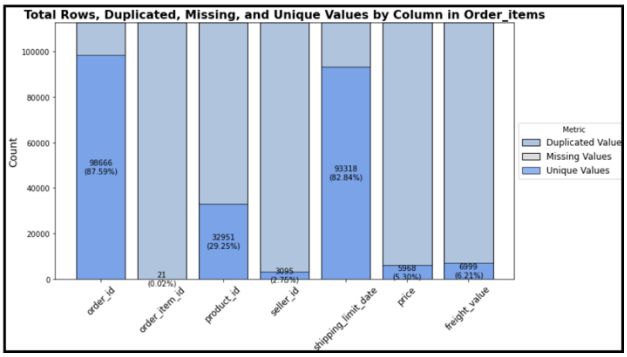
Column Name	Data Type	Column Name	Data Type
shipping_limit_date	Date	order_id	String
order_estimated_delivery_date	Date	product_id	String
review_creation_date	Date	seller_id	String
price	Float	Customer_ID	String
freight_value	Float	order_status	String
payment_value	Float	seller_city	String
order_item_id	Integer	seller_state	String
seller_zip_code_prefix	Integer	customer_unique_id	String
customer_zip_code_prefix	Integer	customer_city	String
review_score	Integer	customer_state	String
payment_sequential	Integer	geolocation_state	String
payment_installments	Integer	payment_type	String
product_name_lenght	Integer	product_category_name	String
product_description_lenght	Integer	order_purchase_timestamp	Timestamp
product_photos_qty	Integer	order_approved_at	Timestamp
product_weight_g	Integer	order_delivered_carrier_date	Timestamp
product_length_cm	Integer	order_delivered_customer_date	Timestamp
product_height_cm	Integer	review_answer_timestamp	Timestamp
product_width_cm	Integer		

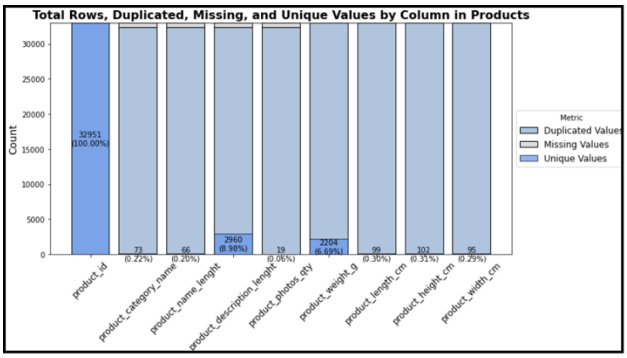
Appendix B: Percentage of Duplicated, Missing and U nique values

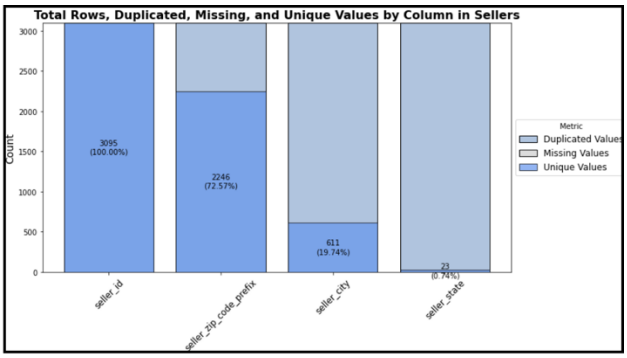


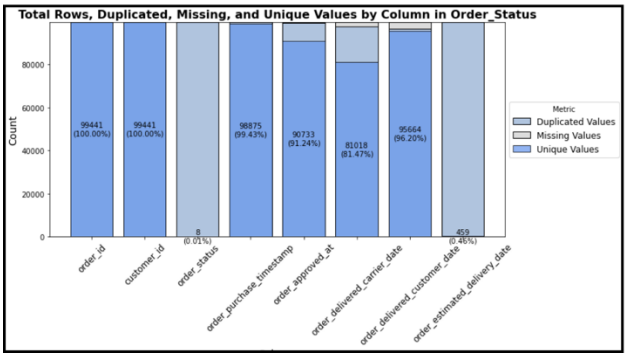


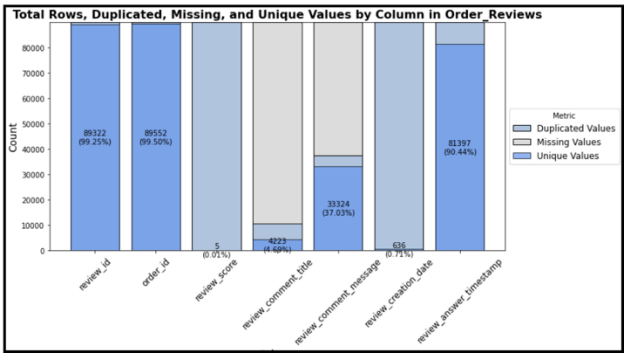








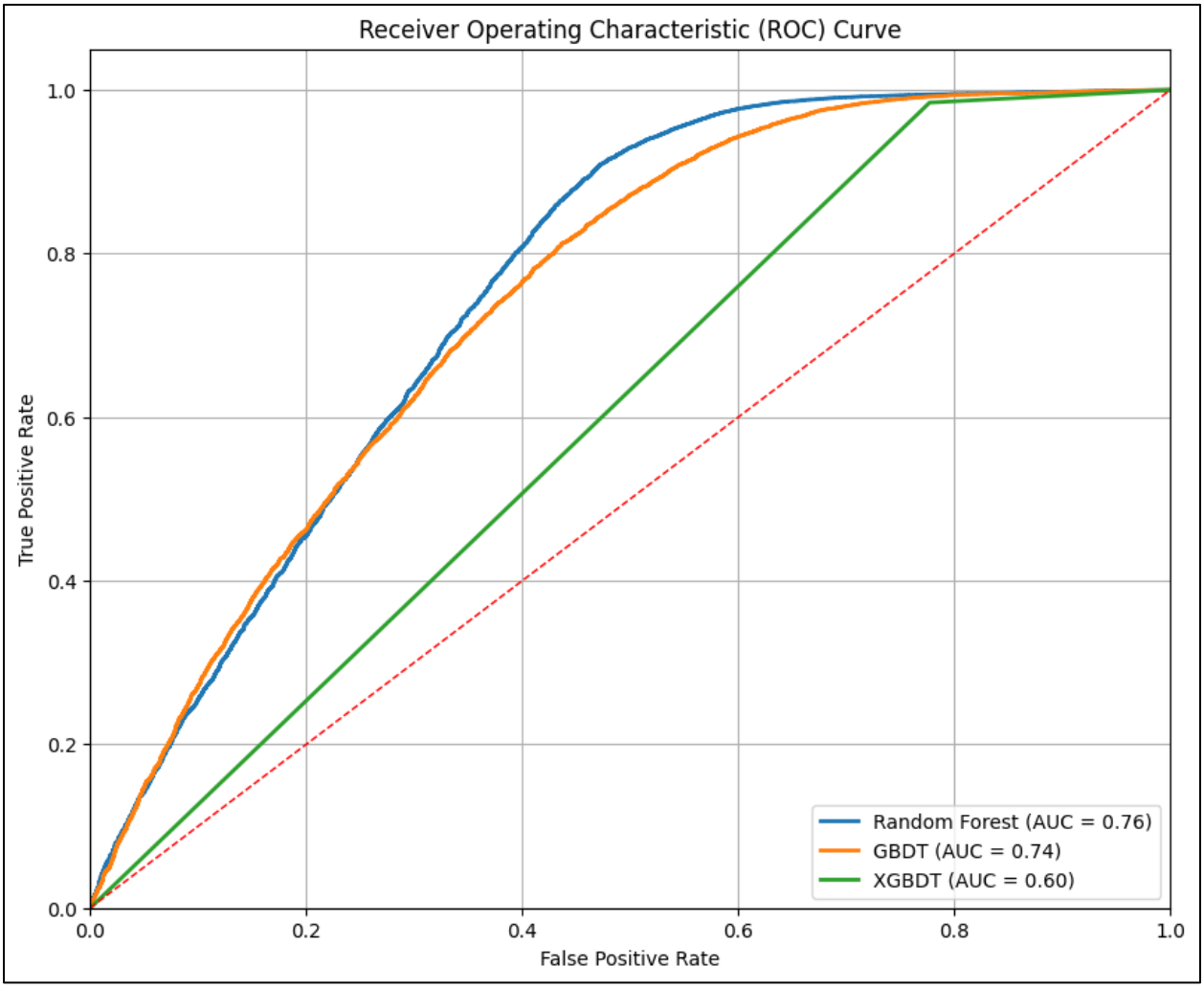




Appendix C: Features Used in the Model

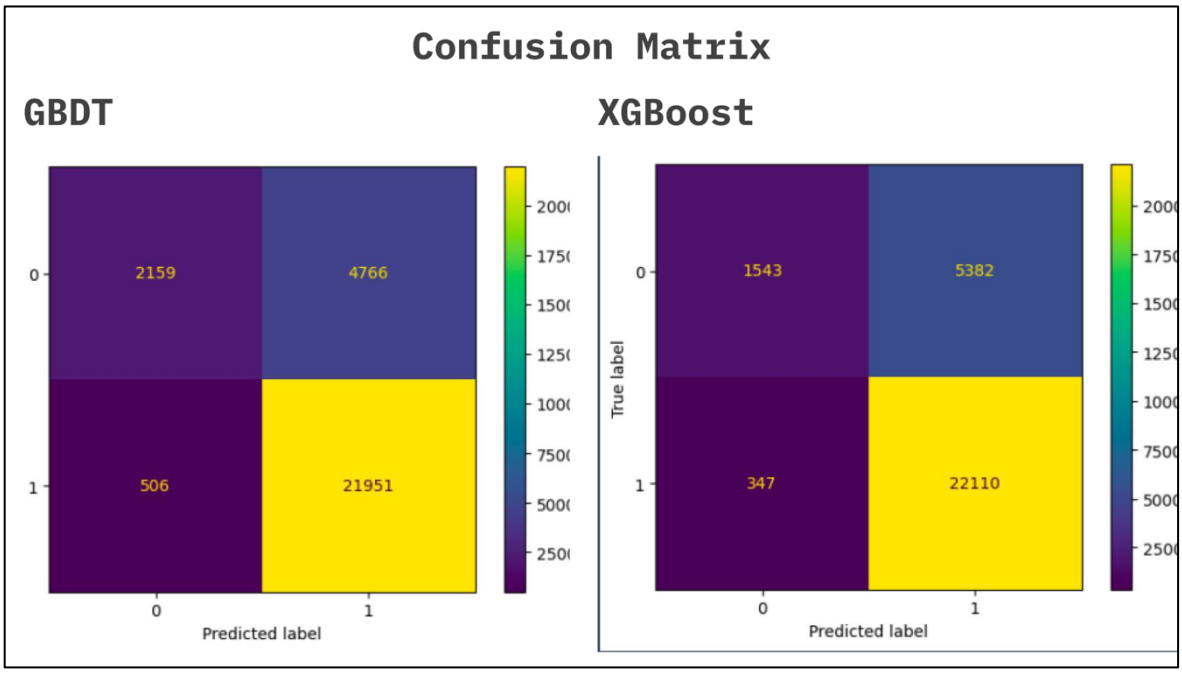
S. No	Field	Description
1	price	Price for the item
2	freight_value	Freight value for the item
3	product_payment_value	Product cost (Freight + Price)
4	product_name_lenght	Length of Product Name
5	product_description_lenght	Length of Product Description
6	product_photos_qty	Count of Product Photos
7	seller_state	State of the Seller
8	diff_approved_purchased	Days between date of purchase and date of order approval
9	diff_customerdelivered_estimated	Days between date of delivery and estimated date of delivery
10	diff_customerdelivered_deliveredcarrier	Days between date of delivery and date of carrier delivery
11	diff_customerdelivered_purchase	Days between delivery date and date of order purchase
12	diff_deliveredcarrier_purchase	Days between date of carrier delivery and date of order purchase
13	diff_approved_purchased_wd	Weekdays between date of purchase and date of order approval
14	diff_customerdelivered_deliveredcarrier_wd	Weekdays between date of delivery and date of carrier delivery
15	diff_deliveredcarrier_purchase_wd	Weekdays between date of carrier delivery and date of order purchase
16	payment_type_count	Distinct count of payment types for an order
17	review_score	Review Score for the order
18	diff_review_creation_answer_days	Days between date of notification sent for review and date when review was given
19	customer_state	State of the Customer
20	Category	10 Product Category for the item
21	order_id_product_id	Unique key for indexing
22	total_purchase_count	Number of orders bought by the customer
23	freight_to_price_ratio	Freight over price per item

Appendix D: Receiver Operating Characteristic Curve



(Shows random forest gives best results)

Appendix E: GBDT & XGBoost Confusion Matrix



Appendix F: Bayesian Hyperparameter Tuning Results

Bayesian Method

Training Scores

Model: RF
Macro Precision: 0.8307950160373052
Macro Recall: 0.6419234714927275
Macro F1-score: 0.646372845047553
Accuracy: 0.7572457314435482

Model: GBDT
Macro Precision: 0.8965941378085205
Macro Recall: 0.7849823417393205
Macro F1-score: 0.8125873041830806
Accuracy: 0.8534949998727703

Model: XGB
Macro Precision: 0.8921173719050681
Macro Recall: 0.7867190831195838
Macro F1-score: 0.8135983339242978
Accuracy: 0.85346955393267

Test Scores

Model: RF
Macro Precision: 0.8178279813359781
Macro Recall: 0.5918615680926678
Macro F1-score: 0.6003694418714958
Accuracy: 0.8013749914913892

Model: GBDT
Macro Precision: 0.822809503770283
Macro Recall: 0.6379297651717546
Macro F1-score: 0.6635951934118671
Accuracy: 0.8192771084337349

Model: XGB
Macro Precision: 0.8054435238611148
Macro Recall: 0.6388512438291615
Macro F1-score: 0.6640162951239907
Accuracy: 0.8167925941052345

Appendix G: Table of Figures

**All figures are the Authors' own creations.*