Table of Contents

1. Business Understanding

This report presents the development and deployment of a predictive model to identify customers' reviews for Nile, one of the leading South American eCommerce platforms. It is important in online retail to maintain positive reviews and convert negative ones to positive since these greatly determine customer trust and buying behaviours. (Godara , 2024). Using data from the Nile's database, our objective is to devise a model that will help predict customer reviews.

Using advanced machine learning techniques, our team has devised a robust and resource-efficient solution that aligns with the goals of Nile to enhance customer engagement and review management.

Data preparation and engineering, selecting the appropriate machine learning algorithm that would forecast customer reviews, and what potential impact the model would have on Nile's business strategy are discussed in this report. This report will follow the CRISP-DM framework by providing the details of data engineering, modelling, evaluation, and actionable recommendations to enhance Nile's review management system.

2. Data Understanding & Preparation

2.1. Datasets Used

We used multiple datasets from Nile's database, including customer details, orders, payments, product categories, and reviews, merging them via primary and foreign keys. Not all datasets were relevant, so we focused on key variables for building the machine learning model.

Key datasets:

Dataset	Description
olist_customers_dataset	Customers' state, Customers' unique ID
olist_orders_dataset	Order status, Delivery date
olist_order_reviews_dataset	Review scores
olist_order_payment_dataset	Payment method, total payment amount, and instalment information.
olist_order_items_dataset	Product category and price information.

Table 1

2.2. Data Preprocessing

Preparing Data for Modelling: The following steps were performed to prepare the data for modelling:

1. **Handling Missing Values:** Dropped the rows with missing "review_score" and critical features. Dropped the rows which were giving error values. We did not replace them as the data size was already good enough
2. **Feature Engineering**:
   i. Created a binary target variable: review_score_binary (1 for positive reviews, 0 for negative reviews). This is because we used binary classification as we thought this would be the best way to categorise the decision variable into 2 classes.

   ii. Added derived features such as delivery_date_difference, which gave the measure of punctuality. It gave us a deciding parameter since the delivery time is such an important aspect of how the customer would rate the order on the ecommerce platform.

   iii. Made the installment payments binary to show the flexibility of the customers for the column "payment_installments_binary".

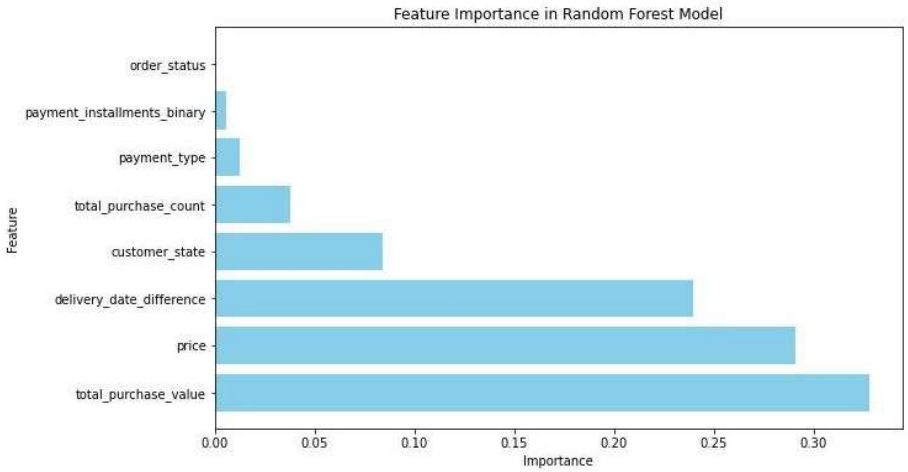

Figure 1 Feature Priority for The Model

3. **Categorical Encoding**: Transformed features using Label Encoding for payment_type, order_status, and customer_state. Label Encoding will enable us to give more numerical and effective results.
4. **Standardisation**: Scaled numerical variables' price and payment_value to ensure equal weightage for features.

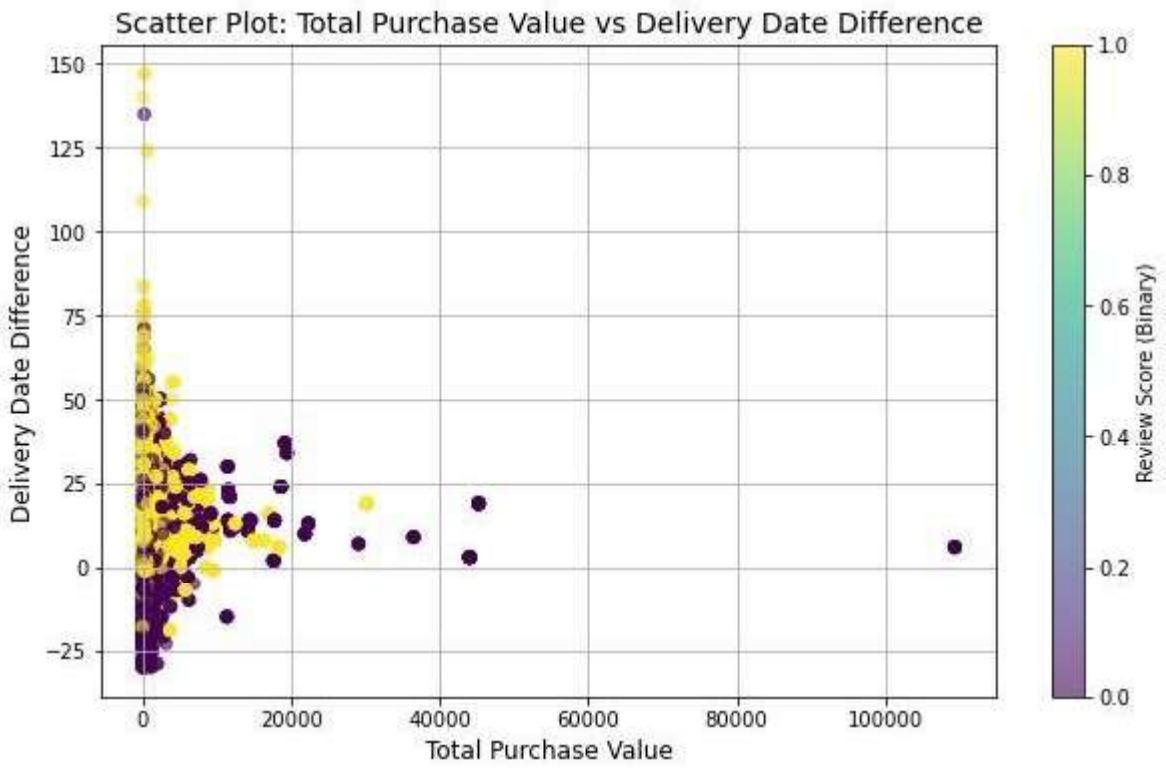

Figure 2 Scatter Plot of Total Purchase Value and Delivery Date Difference

Here, we can see that as the delivery date difference is negative (the product is being delivered late by the number of days mentioned), there are greater negative reviews and as the delivery date difference goes on increasing (the product arrives those many days before the estimated time) there are greater positive reviews.

2.3. Addressing Class Imbalance

As can be seen from the graph below, the dataset used to predict customer reviews at Nile was highly imbalanced between the positive reviews of class 1 and negative reviews of class 0. This was a challenge to the machine learning models since they always favour the majority class. Therefore, poor recall and underperformance could be identified in instances of the minority class.

SMOTE synthesises examples for the minority class, which works well in this case, as suggested by He & Garcia (2009). In contrast with the traditional oversampling that simply replicates the existing data, SMOTE creates new, realistic data points interpolating the existing minority class samples.

SMOTE makes sure that both positive and negative reviews are balanced equally in this dataset. The Random Forest model by Chawla et al. (2002) can therefore learn the underlying patterns in the minority class, thus improving recall, which is very important for our goal of predicting positive reviews while trying to keep a balanced performance across both classes.

As we can see in the below map, most of the states have average reviews > 3.5 which gives us a visual example of data imbalance.

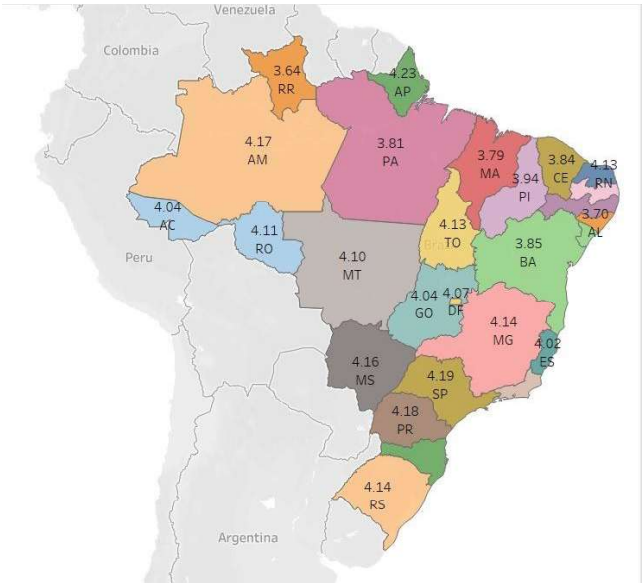

Figure 3 Average customer review by state

3. Modelling

3.1. Model Selection

For this predictive task, we have referred to Han, Kamber & Pei (2012) and Kuhn & Johnson (2013). The Random Forest Classifier was chosen as the final model due to its interpretability, robustness and ability to manage complex datasets with high-dimensional and diverse features. It operates as an ensemble learning method that builds multiple decision trees during training (Breiman, 2001) and combines their outputs through majority voting (classification) or averaging (regression). This approach inherently enhances the model's predictive performance and general accuracy.

3.2. Key aspects of Random Forest -

i. ***Versatility in Handling Features:***
   It works easily both with numeric and categorical features without a requirement of intensive feature engineering. In that respect, this dataset combines two types of variables, one numerical, the payment value, and the other categorical, payment type. This will increase the suitability of our application.

ii. ***Resilience to Noise:***
   As it uses an aggregation from multiple trees to do a prediction, random forests will therefore be more insensitive to any small effects within the data-noise.

iii. ***Feature Importance and Insights:***
   One of the strengths of the Random Forest algorithm is ranking features by their contribution to the predictions. In the case of Nile's eCommerce dataset, this allows us to identify key drivers of positive reviews, such as delivery_date_difference and total_purchase_count, which provides actionable managerial insights

iv. ***Robustness to Overfitting:***
   - Two main mechanisms mitigate overfitting:
     - Bagging (Bootstrap Aggregation): Each tree is trained on a random subset of the data, which reduces the model's dependency on specific data points.
     - Random Feature Selection: At each split, only a subset of features is considered, which prevents the model from being overly influenced by dominant variables.
   - These properties make Random Forest highly effective in capturing the underlying patterns in the data without memorising noise or irrelevant details.

3.3. Comparison with Other Models

i. **Gradient Boosting Classifier (GBC):**
   This algorithm builds trees sequentially, correcting predecessor errors. It offers high accuracy and models non-linear relationships but is computationally expensive and prone to overfitting without tuning.
   a. As there was no hyperparameter tuning in this model, **Random Forest** was preferred for its faster training and robust default performance.

ii. **Logistic Regression:**

a. Logistic Regression is simple and interpretable for binary classification but struggles to capture complex feature relationships due to its linear nature.

b. The assumption of feature independence further limits its applicability to this dataset where interactions between variables like payment_value and delivery_date_difference are critical for predicting review scores.

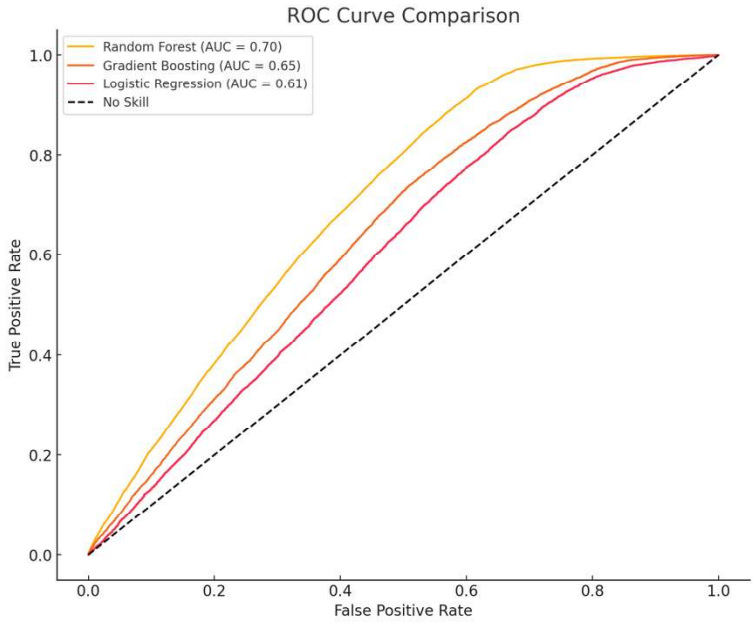

Figure 4 ROC Curve Comparison

According to the ROC curve, the area under the curve is highest for Random Forest statistically making it the best model for our data.

3.4. Direct advantage of using Random Forest for Nile

i. **Class Imbalance Handling:**

Class imbalance in the dataset, with fewer negative reviews (class 0), was addressed using SMOTE. Random Forest's flexibility and resilience make sure that the oversampled minority class (negative reviews) is well-represented in the training process further enhancing recall.

ii. **Dynamic Feature Relationships:**

Predicting review scores involves complex, non-linear relationships among features such as payment behaviour, delivery time, and customer state. Random Forest's collection of decision trees excels in modelling these interactions without requiring explicit feature transformations.

iii. **Interpretability for Business Decisions:**
Random Forest provides feature importance rankings, enabling Nile to identify actionable insights, such as improving delivery punctuality (delivery_date_difference) or incentivising frequent customers (total_purchase_count).
iv. **Scalability:**
Random Forest can scale effectively with large datasets and high-dimensional data, making it suitable for Nile's growing dataset and future expansion.

3.5. Hyperparameter Tuning

No hyperparameter tuning was performed to keep the model straightforward. Default parameters yielded sufficient performance for our objectives.

4. Evaluations

4.1. Model Performance

The Random Forest Classifier yielded the following metrics:

- **Accuracy**: 0.77
- **Precision**: 0.75
- **Recall**: 0.80
- **F1-Score**: 0.78

Model Performance:

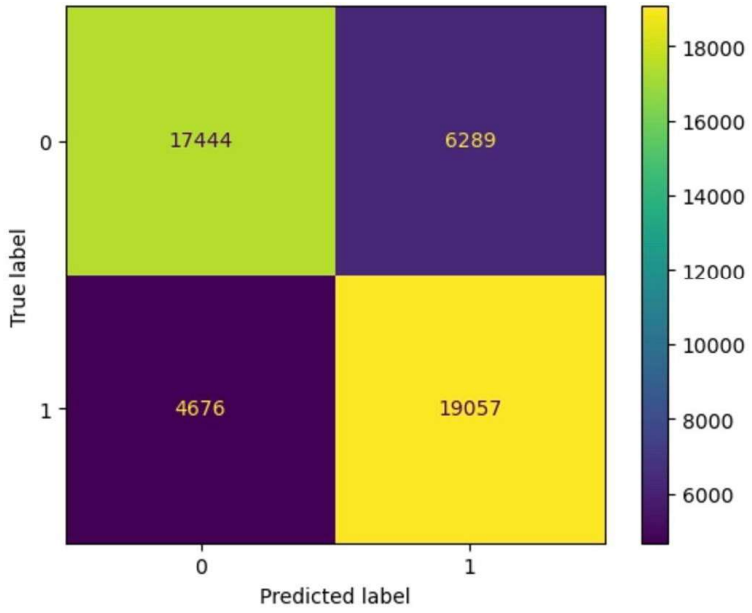

Figure 5 Confusion Matrix

These metrics collectively provide a comprehensive understanding of the model's performance highlighting strengths and areas for improvement.

4.2. Technical Analysis of Metrics

1. **Recall (0.80)**: Recall is the proportion of actual positive reviews that were correctly identified by the model.

Formula:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} = \frac{19,015}{19,015 + 4,718} \approx 0.80$$

Interpretation: A recall of 0.80 indicates that the model successfully identifies 80% of all actual positive reviews. This is critical for our objective of ensuring that customers likely to leave positive reviews are targeted effectively with loyalty incentives.

2. **Precision (0.75)**: Precision measures the proportion of predicted positive reviews that are actually positive.

Formula:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} = \frac{19,015}{19,015 + 6,281} \approx 0.75$$

Interpretation: A precision of 0.75 means that 75% of the reviews predicted as positive were genuinely positive. While this is a good result, the presence of false positives (6,281 cases) indicates that some resources might be misallocated to customers who are less likely to leave positive reviews.

3. **F1-Score (0.78)**: The F1-Score is the harmonic mean of precision and recall, providing a balanced measure of the model's ability to predict positive reviews.

Formula:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0.75 \times 0.80}{0.75 + 0.80} \approx 0.78$$

Interpretation: The F1-Score reflects the model's overall reliability, balancing precision and recall. With an F1-Score of 0.78, the model demonstrates strong performance, emphasising that it handles both false positives and false negatives effectively.

4. **Accuracy (0.77)**: Accuracy measures the overall proportion of correctly classified reviews.

Formula:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}} = \frac{19,015 + 17,452}{47,466} \approx 0.77$$

Interpretation: While an accuracy of 77% indicates the model performs well overall, it does not account for the class imbalance. Metrics like recall and precision are better indicators for our specific objective.

5. Recommendations

To further enhance the performance and reliability of our ML model for predicting customer reviews, the following technical recommendations focus on refining key aspects such as recall, precision, and overall robustness:

i. **Recall Enhancement**

The objective of recall enhancement is to improve the model's ability to identify positive reviews (true positives). This is achieved through hyperparameter tuning, adding customer-centric features like repeat purchase frequency, sentiment analysis and using ensemble learning with models like Gradient Boosting or XGBoost. These strategies aim to increase recall, ensuring the model reliably targets customers likely to leave positive reviews.

ii. **Precision Improvement**

The goal of precision improvement is to reduce false positives and optimise resource allocation. Strategies include adjusting the class_weight in Random Forest to penalise false positives, using cost-sensitive algorithms, and employing advanced feature engineering (e.g., analysing delivery time deviation or loyalty status). Outlier detection methods address data noise, ensuring predicted positives are genuine and improving the efficiency of incentive campaigns.

iii. **Dynamic Threshold Optimisation**

Dynamic thresholding aims to align classification thresholds with business goals, such as prioritising precision for high-cost campaigns or recall for broader outreach. This involves analysing the Receiver Operating Characteristic (ROC) curve to determine the optimal balance between precision and recall and implementing custom thresholding using weighted F1-scores or cost-based functions to adjust thresholds dynamically. The expected outcome is a tailored prediction strategy that maximises campaign effectiveness and cost-efficiency.

iv. **Model Robustness and Overfitting Prevention**

To enhance the model's generalisability to unseen data, strategies for improving robustness and preventing overfitting include using stratified K-fold cross-validation to ensure balanced class representation and applying regularisation techniques like limiting tree depth or controlling leaf nodes. For neural networks, dropout layers can mitigate overfitting, while ensuring sufficient trees in Random Forest reduces variance. These measures aim to enhance robustness, ensuring consistent performance on future datasets.

v. **Metrics Refinement**

Metrics refinement involves incorporating additional evaluation metrics for a more comprehensive model assessment. This includes analysing the precision-recall curve to evaluate performance on imbalanced datasets where accuracy can be misleading, and

using weighted metrics like recall, precision, and F1-score to reflect the business value of predictions. Additionally, calibration curves are employed to assess and adjust prediction probabilities, ensuring reliability for high-stakes decisions such as incentive allocation. These refined metrics enable nuanced performance evaluation, guiding targeted improvements.

vi. **Scalability and Future Extensions**

To ensure scalability, the model is designed to handle Nile's growing customer base and dataset. Strategies include incremental learning for updates without retraining, automated feature selection (e.g., RFE, SHAP), and transfer learning with pre-trained models to enhance predictive power. These approaches future-proof the model for evolving customer behaviour and larger datasets.

6. Conclusion

This report demonstrates the successful application of machine learning to predict customer reviews on Nile's eCommerce platform. The **Random Forest Classifier** with SMOTE effectively addresses class imbalance, achieving a recall of 0.80 and an F1-Score of 0.78. By identifying customers likely to leave positive reviews, Nile can allocate resources efficiently, enhance customer satisfaction, and sustain a competitive edge.

Future iterations of this model could incorporate additional features like purchase frequency and sentiment analysis from textual reviews. Regular monitoring and retraining will ensure model relevance in dynamic business conditions.

Through these insights and recommendations, Nile can foster customer loyalty, reduce negative feedback, and improve overall brand perception.

7. Appendices

References

Breiman, L. (2001). *Random Forests.* Machine Learning, 45(1), 5–32.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique.* Journal of Artificial Intelligence Research, 16, 321–357.

Godara, R. S. (2024). *Impact of Customer Reviews on eCommerce Trust and Sales.* Journal of Digital Commerce, 18(3), 245–260.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques.* 3rd Edition, Elsevier.

He, H., & Garcia, E. A. (2009). *Learning from Imbalanced Data.* IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284.

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling.* Springer.

List of Tables

List of Figures