Report on Capstone Midterm Project

Introduction

The aim of this project is to create a classification model for predicting whether an online purchase order carries a high risk of default payment. This report details the preprocessing steps, the chosen classification method, and the evaluation of the model's performance.

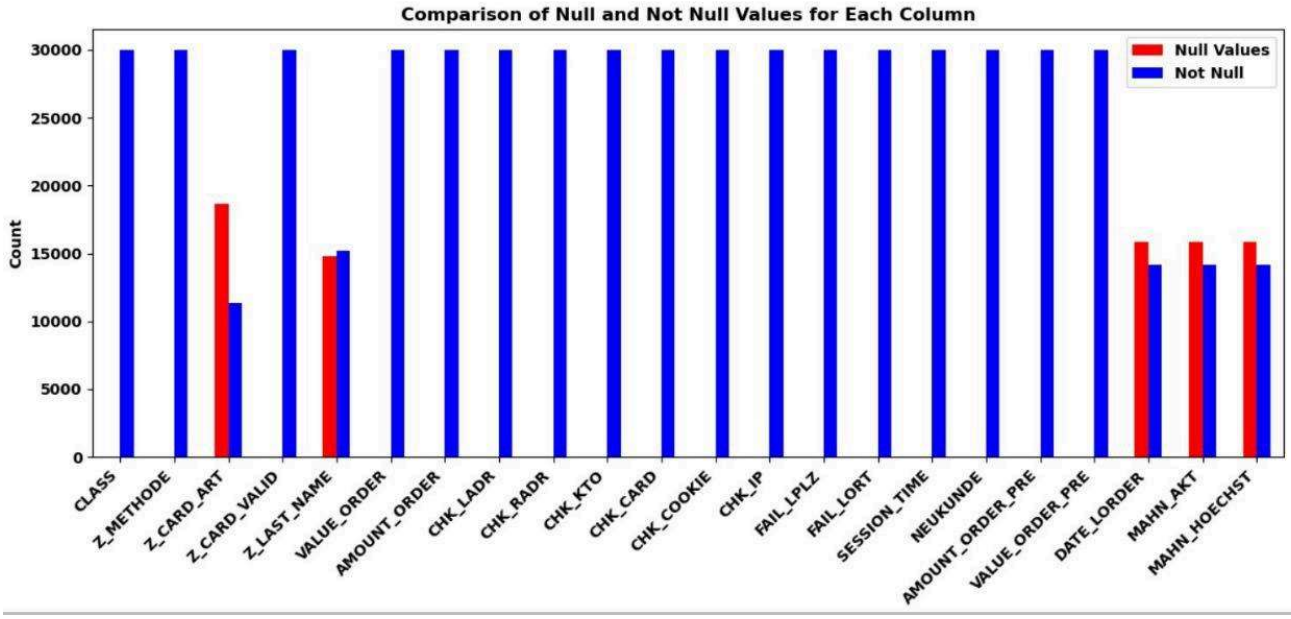
Description of the Dataset

The dataset comprises various attributes related to online orders, encompassing customer information, order details, and transaction characteristics. It contains information such as the order ID, payment method, value of the order, and the presence of associated data like email addresses, phone numbers, or birth dates. Additionally, the dataset includes indicators of customer behavior, such as newsletter subscription, previous order history, and any potential failures or discrepancies in address information. Furthermore, attributes like session duration, weekday of the order, and the presence of new customers offer insights into ordering patterns and customer engagement. The dataset also features flags indicating similarity or discrepancies in delivery and invoice addresses, as well as checks for repeated orders within a short timeframe or across various parameters like account or card numbers. Overall, this dataset provides a comprehensive view of online ordering dynamics, customer interactions, and transactional patterns, which can be leveraged for various analytical and predictive tasks in e-commerce and customer relationship management.

Data preprocessing steps include:

- Dropping irrelevant columns such as 'B_EMAIL', 'B_TELEFON', etc.
- Transforming null/missing/unknown values to proper NaN values.
- Missing values were identified and handled appropriately. Null values in certain columns, such as 'Z_CARD_ART' and 'Z_LAST_NAME', were replaced with meaningful placeholders based on contextual understanding. Notably, the total number of null values in 'Z_CARD_ART' was found to be 18654. It was observed that these null values occur when the 'Z_METHODE' is not a card method (i.e., credit or debit_note). Since the number of null values is equal to the total null values for 'Z_CARD_ART' when 'Z_METHODE' is not a card method, the decision was made not to drop the 'Z_CARD_ART' column, as its null values are justified.
- We also dropped 10 columns of ANUMBER_1 to ANUMBER_10 which were not useful in the prediction.
- Splitting the 'DATE_LORDER' field into numerical formats (day, month, year).
- Applying one-hot encoding to categorical values.

Here is a Barplot of our data after deleting the unnecessary columns which shows a distribution of null vs non null values:



Classification Method

Logistic Regression was selected as the classification algorithm due to its simplicity, interpretability, and effectiveness in binary classification tasks. Additionally, class weights were adjusted to handle the dataset's penalty on predictions and the uneven costs associated with misclassifications.

Solutions, Findings, and Results

The logistic regression model was trained on the preprocessed dataset and evaluated using a separate testing set. Various performance metrics, including accuracy, custom cost, and confusion matrix, were computed to gauge the model's efficacy. However the classes were not balanced and it affected the model's performance, but we did not address the class imbalance issue since it was not required as a deliverable. As an examination we swapped the penalty values for high risk and low risk and the model predicted all values as low risk which would indicate an overfitting problem for the trained model.

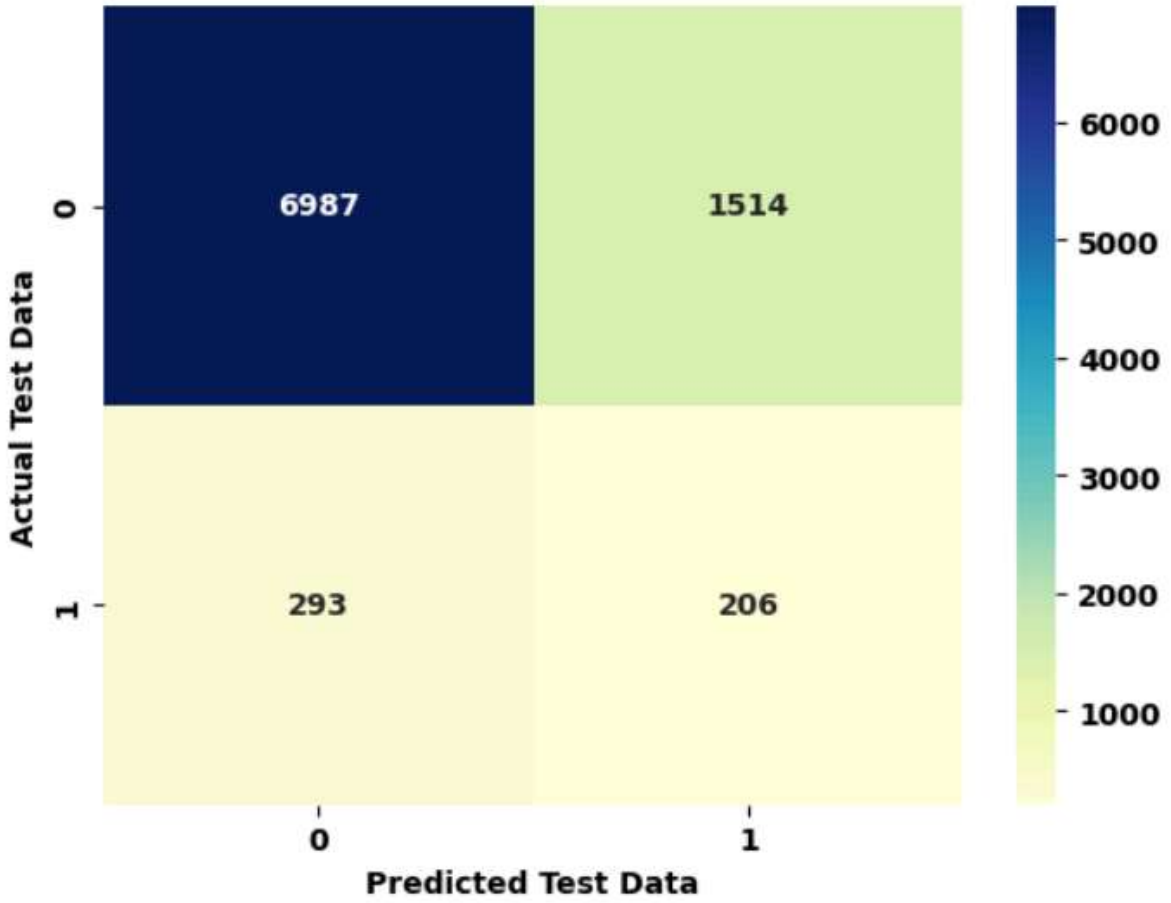
Explanation of the Cost Function

The custom cost function used in this project is a modified cross-entropy loss function that accounts for the specific costs outlined in the provided cost matrix. This matrix reflects the costs associated with misclassifications, considering the different consequences and financial implications of predicting high-risk and low-risk orders incorrectly.

By incorporating the cost matrix into the cost function, the model is trained to minimize the expected financial losses for the online trader, aligning with business objectives and risk management strategies. As requested in the deliverables we treated the high risk incorrect prediction 10 times more costly as the low risk class incorrect prediction.

Explanation of Evaluation Metrics

- **Confusion Matrix:** The confusion matrix provides a tabular representation of the model's predictions compared to the actual labels. It shows the number of true positives, true negatives, false positives, and false negatives, allowing for a detailed analysis of the model's performance.



- **Classification Report:** The classification report presents a summary of various evaluation metrics, including precision, recall, F1-score, and support, for each class. It provides insights into the model's performance for both high-risk and low-risk orders.

The Classification Report for model:

	precision	recall	f1-score	support
False	0.96	0.82	0.89	8501
True	0.12	0.41	0.19	499
accuracy			0.80	9000
macro avg	0.54	0.62	0.54	9000
weighted avg	0.91	0.80	0.85	9000

Conclusion:

Based on the classification report, the model exhibits notable discrepancies in performance between the two classes. Specifically, for the negative class (False), the model demonstrates strong precision (0.96) and moderate recall (0.82), resulting in a high F1-score of 0.89. This indicates that the model effectively identifies instances belonging to the negative class, with a high level of confidence in its predictions. However, for the positive class (True), the model's performance is considerably poorer, with low precision (0.12) and recall (0.41), leading to a low F1-score of 0.19. This suggests that the model struggles to accurately identify instances belonging to the positive class and frequently misclassifies them as negative.

Overall, the model achieves an accuracy of 0.80, indicating that it correctly classifies 80% of the instances in the dataset. However, it's crucial to consider the class imbalance, particularly since the negative class dominates the dataset. The macro-average and weighted-average F1-scores, at 0.54 and 0.85 respectively, highlight the overall performance of the model across both classes.

In conclusion, while the model demonstrates strong performance in identifying instances of the negative class, its ability to accurately classify instances of the positive class is severely lacking. Further improvements in the model's ability to detect instances of the positive class are necessary to enhance its overall effectiveness and reliability in real-world applications.

References:

- Data preprocessing: A comprehensive step-by-step guide. Technology & Software Development Blog | Future Processing. (2024, January 31). https://www.future-processing.com/blog/data-preprocessing-a-comprehensive-step-by-step-guide/

- ● Coursesteach. (2023, October 21). Deep learning (part 7)-logistic regression cost function Logistic Regression Cost Function | Medium

- Mutuvi, S. (2021, September 24). *Introduction to machine learning model evaluation*. Medium. https://heartbeat.comet.ml/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f

- Raja, A. Z. (2023, February 25). *Model selection and training: Choosing the right model for your data*. Medium. https://alizahidraja.medium.com/model-selection-and-training-choosing-the-right-model-for-your-data-b44958d1b4be

- Coursesteach. (2023, October 21). Deep learning (part 7)-logistic regression cost function Logistic Regression Cost Function | Medium

Member Contributions:

- **Elham Peimani:** Handled the preprocessing section.
- **Parthiban Subramani:** Handled the visualization parts of the code.
- **Nisargkumar Mahyavanshi:** Implemented the model.
- **Pooya Fazeli:** Documented the project and participated in the cost function implementation.