Managing Risk for Digital Service Subscriptions

A File Hosting Provider

STATUS	BADGE
Restricted	Restricted ⊘

Prof. Delina Ivanova
Lecturer, Schulich School of Business - York University
Director of Analytics, Mistplay

Dhari Gandhi	220227591
Yaosheng Liu	216173163
Vivek Pereira	216421588
Sabrina Renna	220838934
Darren Singh	216236275
Zabihullah Zabi	220546412

August 6, 2023

Table of Contents

List of Figures

List of Tables

1.0 Introduction

In the digital space, a file hosting service provides file storage on remote servers that can be accessed from the Internet. Our client, sanitized to maintain anonymity, is an up-and-coming player in the file hosting space that exceeded an annual total revenue of $1M USD in 2021. A software business that generates more than $20,000 in monthly sales with a recurring billing subscription model, is considered a high-risk business by both payment processors and financial institutions (Chargebacks 911, 2021). Excessive chargebacks (disputes), refunds, and failed payments are all considered indicators of risk. This report aims to leverage exploratory data analysis and predictive modelling to provide key insights to the client on how risk can be minimized using the company's historical payment processing data.

1.1. Account Restriction & Termination with Stripe

Stripe is a multi-national financial services company that is revolutionizing the payment industry by simplifying payments for digital-first companies. Our client has used Stripe since late 2020 as its principal online payments' provider. Stripe is highly sensitive to the risk profile of its merchants and has restricted the status of our client's processing account twice throughout its operations. A "Restricted" account has payouts or payments disabled while the client is responsible for submitting countering information to satisfy the Fraud & Risk team. The account restrictions have resulted in approximately $98,000 USD in lost revenue for our client. Stripe reserves the right that they "may terminate this Agreement and close your Stripe account at any time for any reason effective upon providing you notice in accordance with this Agreement" (Stripe, 2023). Account restriction is the first step in the termination of a Stripe account. Such a termination would have a devastating effect on our client and result in the need for a lengthy and costly procurement of a high-risk processing account with no guarantees of approval. The motivation for this analysis is to maintain good standing with Stripe and protect the financial health of our client. Table 1 provides an overview of account history and financial volumes successfully processed.

Table 1 - Client Sales and Account History

Year	Amount	Total Revenue ($, USD)	Months Active (Processing Payments)	Average Monthly Revenue ($, USD)
1	2020	$160,270	November, December	$80,135
2	2021	$1,229,342	Entire Year	$102,445
3	2022	$945,169	Entire Year*	$78,764
4	2023	$425,186	January to June*	$70,864
*Account restriction occurred.				

1.2. Dataset

The dataset exported from the Stripe Dashboard consists of 91,491 payment events before data cleaning spanning November 1st, 2020, to July 1, 2023. The sanitized data includes 18 columns on subscription description, payment success, card details (types, geography, and funding) and dispute information. A preliminary assessment of the data concluded two intended target variables for risk analysis: "Seller Message" and "Dispute Amount". "Seller Message" provides a readable description of the outcome type and reason a payment failed or succeeded. Failed payments are of most interest. Decline codes can be grouped into user errors, administrative errors, bank decline for risk, processor decline for risk and blocked in Stripe by the client. Disputes comprise only 312 observations of the entire dataset or 0.34%. A small proportion of disputes is to be expected and is still to be monitored closely.

Stripe writes, “The credit card processing industry standard recognizes dispute activity above 0.75% as excessive, but other factors, such as a sudden spike or steep upward trend can trigger placement in a monitoring program before dispute activity reaches the 0.75% threshold (Stripe, n.d.)” A consumer can dispute a payment 120 days beyond the transaction date or even longer depending on their banking institution. Predicting a sudden spike in disputes or dispute ratio could allow the client to anticipate risk reviews, allocate more customer service staff, and deploy mitigations in a proactive way.

1.3. Analysis Objectives

This report is focused on addressing the core business problem of high counts of failed recurring payments and disputes, which trigger Stripe's credit account review process, posing a threat to the client's immediate revenues, the potential long-term ability to accept digital subscription payments and thereby revenue generation. Objectives of this report include:

i. Identifying potential characteristics of transactions that may have a correlation with the outcome of the payment (i.e., does a particular type of card fund often result in a decline code related to a bank decline for risk?),
ii. Dispute win success – The dispute process allows for the Seller, i.e., our client, to submit counter evidence to demonstrate the legitimacy of the payment. If successful, the Dispute is won and not factored into the Dispute Ratio against the business,
iii. Blocklist suggestions to minimize disputes by country and by user,
iv. A predictive model to assess whether a transaction will trigger a decline code associated with risk; and,
v. A predictive model to determine the potential occurrence of a dispute.

2.0 Exploratory Data Analysis

2.1. Data Cleaning

The data cleaning process was common to both the Seller Message and Disputes notebooks. The primary focus was to handle invalid observations and treat missing values appropriately.

Initially, we identified a set of descriptions that were invalid for our analysis. These included ‘Premium Service’, ‘Test Product’, ‘Recurring Payment’, ‘Platinum Template’, ‘PAYMENT – THANK YOU’, and ‘AMAZON AWS’. All observations containing these descriptions were eliminated from our dataset to maintain the integrity and relevance of our analysis.

Next, we handled missing values across different columns. Missing records in the ‘Card Address State’, ‘Card Address Country’, ‘Card Issue Country’, ‘Seller Message’, and ‘Card Address City’ columns were discovered. Considering these missing entries represented less than 1% of the overall data, we decided to drop these records to prevent any potential bias or inaccuracy. Furthermore, in the ‘Description’ column, we found several missing entries. Our domain knowledge indicated that these entries should be labeled as ‘promo’ and therefore all null entries in this column were replaced with ‘promo’. We also identified a single record with null ‘Seller Message’, which was dropped to preserve the quality of the dataset.

Lastly, no notable outliers were found during the cleaning process indicating that the dataset has been pre-processed, cleaned, and is ready for further analysis.

2.2. Target Variable 1: Seller Message

2.2.1. Descriptive Statistics and Visualizations

At the outset, we focused on columns unrelated to disputes as disputes constituted a minor part of the data and did not significantly impact this phase of the analysis. Although this could fall under the umbrella of feature engineering, the exclusion simplifies our exploratory data analysis and thus, is worthy of a mention at this stage.

Next, we conducted descriptive statistics for each column, but given the discrete and categorical nature of the remaining data, there was limited unique information to report. However, we identified significant class imbalances in our data. The records of credit and debit cards vastly outnumbered those of prepaid cards. Additionally, the seller messages exhibited a stark imbalance with nearly 70% of all entries corresponding to the message 'Payment Complete'. Figure 1 illustrates the card funding imbalances and Figure 2 on the following page illustrates the imbalances among the seller messages.

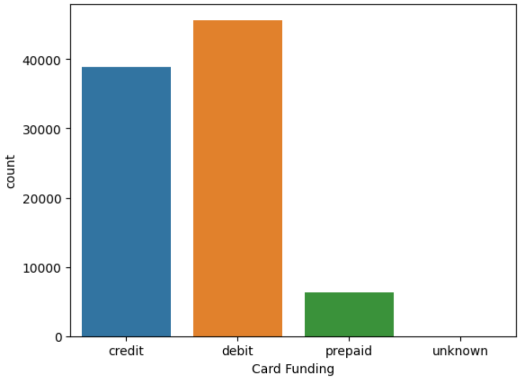


Figure 1 – Card Funding Distribution for Client Payment Data

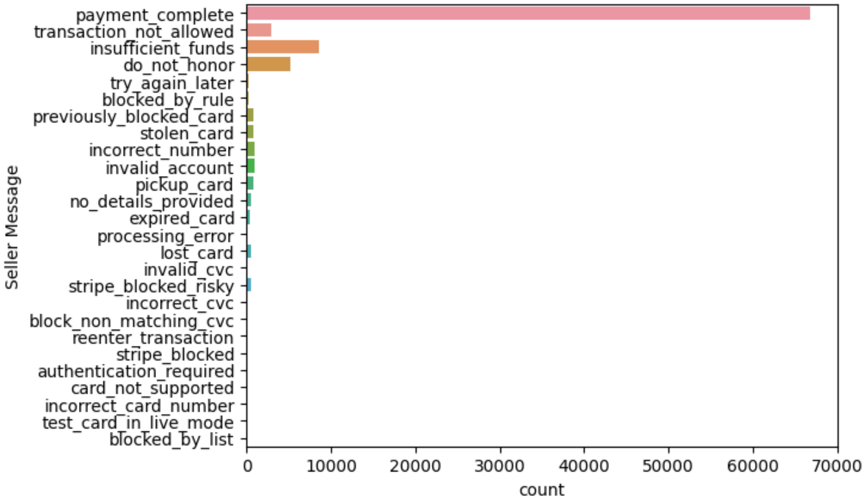


Figure 2 – Seller Message Distribution for Client Payment Data

We explored various relationships between several variables, and although these are detailed in the Jupyter notebooks, we will discuss only those pertinent to our current analysis. Specifically, we investigated the relationship of each seller message with the different types of card funding, as well as the several types of descriptions. A sample of representative plots of these relationships are provided in Figure 3.

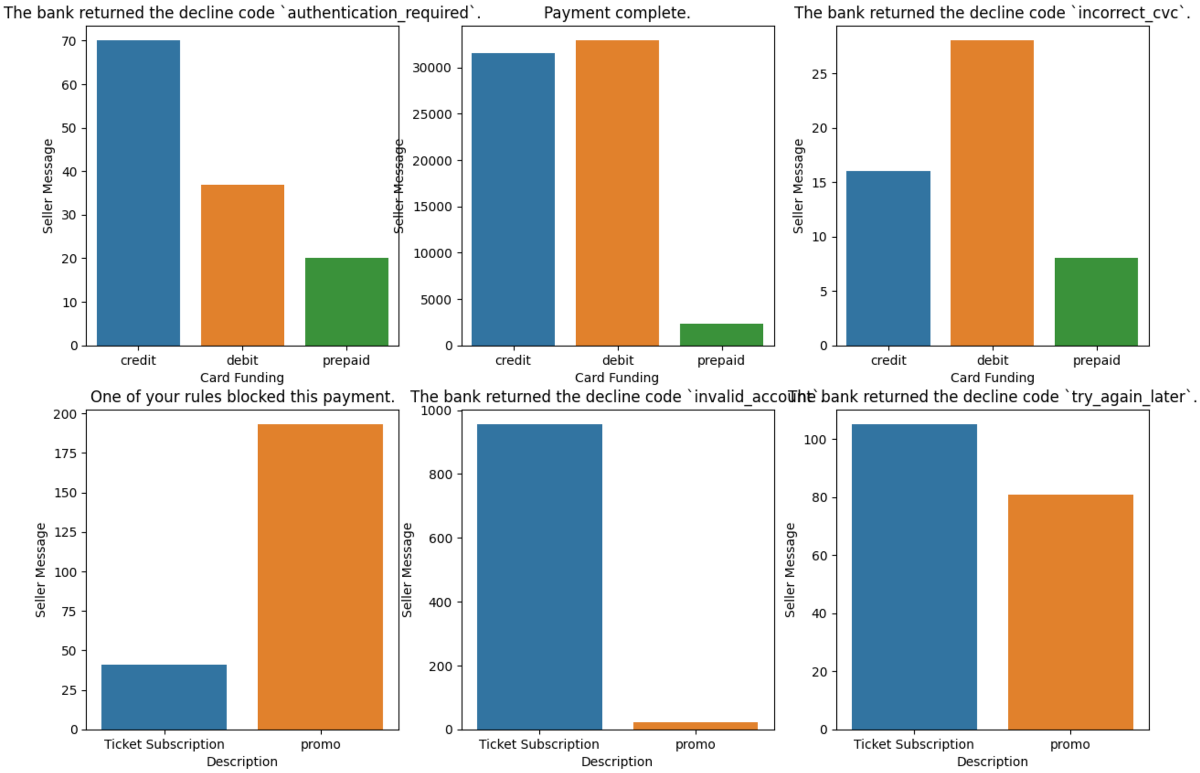


Figure 3 - Plots of counts of Card Funding and Description for various seller messages (Plot subtitles indicate the type of Seller Message being examined)

2.2.2. Results and Interpretation

From Figure 3, it is evident that there are correlations between ‘Seller Message’, ‘Description’, and ‘Card Funding’. The counts of card funding types vary for different seller messages, suggesting a non-linear relationship. Similarly, the ratio of descriptions for different seller messages does not appear to be consistent, implying a similar non-linear relationship. While these relationships do seem non-linear, further tests are needed for confirmation. However, the analysis so far does suggest a relationship between these variables in the dataset. This observation indicates that it might be possible to predict seller messages based on ‘Description’/ ‘Card Funding’ and other available data.

It is important to note that we did not evaluate the relationships between all variables and Seller Message. The discovered relationships will be used as a starting point to employ modelling techniques which will then be used to identify other potential correlations with ‘Seller Message’ and any useful, underlying trends/patterns.

2.3. Target Variable 2: Disputes

When assessing disputes, key relationships for consideration included:

1. Description: What offer type (promotion vs. ticket subscription) reflects a higher percentage of disputes? Promotional offers are reduced prices on new subscriptions i.e., a new user or a reactivation whereas ticket subscriptions refer to a simple renewal.
2. Value: Are certain subscription packages more prone to disputes?
3. Dispute Reason: What are customers citing for the dispute?
4. Dispute Status: Are we able to reclaim any disputes? Can we do more to submit evidence?
5. Blocklists: What countries or users should we no longer accept subscriptions from?

The answers to these questions and the accompanying visualizations are provided in Section 2.3.1.

2.3.1.Visualizations and Recommendations

The analysis begins with a plot to best understand which of the offer types reflect the greatest volume and value of transactions. Ticket subscriptions are found to generate the bulk of the revenue. Given the imbalanced representation, the dispute counts are normalized by offer total transaction counts by type. On average, it can be concluded that 0.52% of promotions result in a dispute as compared to only 0.35% for ticket subscriptions. Figure 4 (left) demonstrates that promotions also result in the greatest variability in dispute ratio and likely account for surges in disputes (unfavourable to the processor). Ticket subscriptions are more predictable over time. On average, total disputes are decreasing over time coinciding with the implementation of the payment verification process through Spreedly. To mitigate risk, it is recommended that promotional offers are not released to the entire email campaign list rather, the marketing is segmented and released slowly over time.

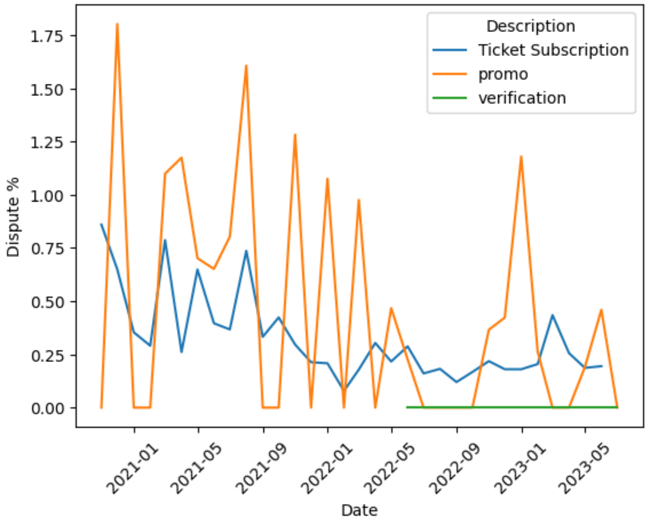


Figure 4 - Dispute Percent by Description

Generally, in ecommerce, the larger the transaction value, the higher the risk for fraud. The data behaves in accordance with this anticipated rule except for the seldom offered $249 lifetime plan as shown in Figure 5. The $249 lifetime are least profitable and by definition do not require renewal. The $94-priced package corresponds to a promotional offer for a two-year plan. It is anticipated that the duration between plan renewals results in a lack of recognition of the charge on bank statements and thereby, more disputes. It is recommended that the maximum plan duration to be offered by the client does not exceed 365 days.

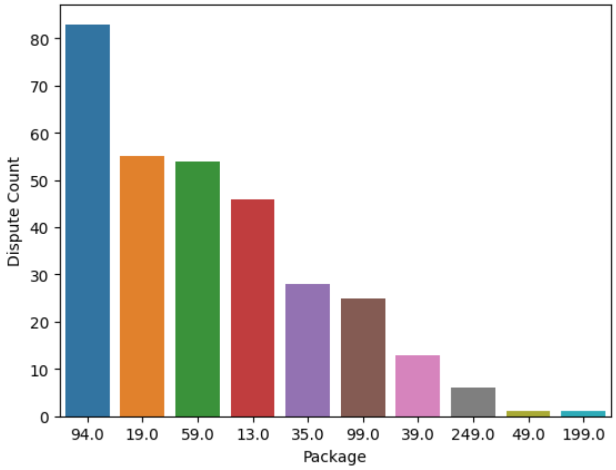


Figure 5 - Dispute Count by Subscription Package

By volume alone, ticket subscriptions account for the most dispute events. Fraudulence is by far the largest reported dispute reason accounting for over 250 of the 312 reported disputes. For the purposes of our model, preventing disputes has become synonymous with preventing fraud. The over-representation of fraud is anticipated to be a result of unrecognized billing description. The visualization is provided in Figure 6 on the following page.

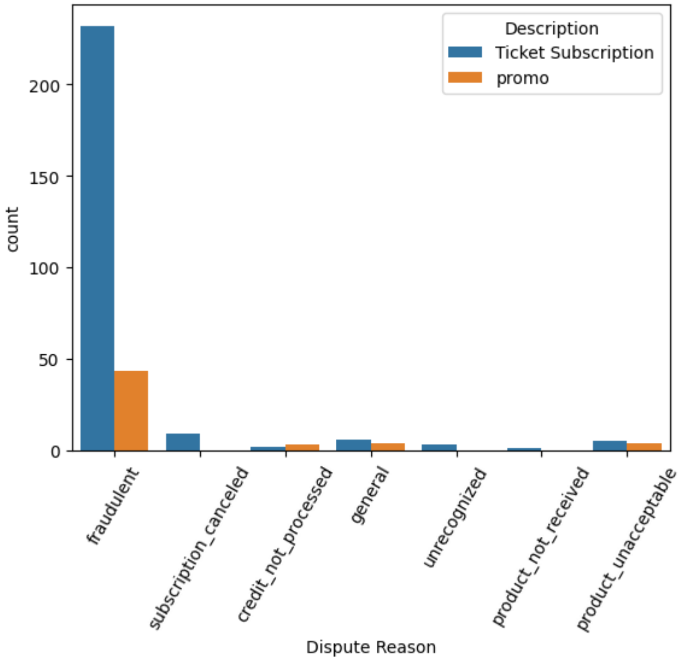


Figure 6 - Dispute count by reported dispute reason.

Additionally, once a dispute is launched, 85.6% of disputes are lost as shown in Figure 7. Therefore, it is anticipated that banks tend to lean heavily in favour of the consumer. Disputes also require additional administrative work to respond to and often remain uncontested by our client. Therefore, in addition to an increased effort to respond to disputes, it is recommended that dispute prevention via predictive modelling or alternate methods is explored.

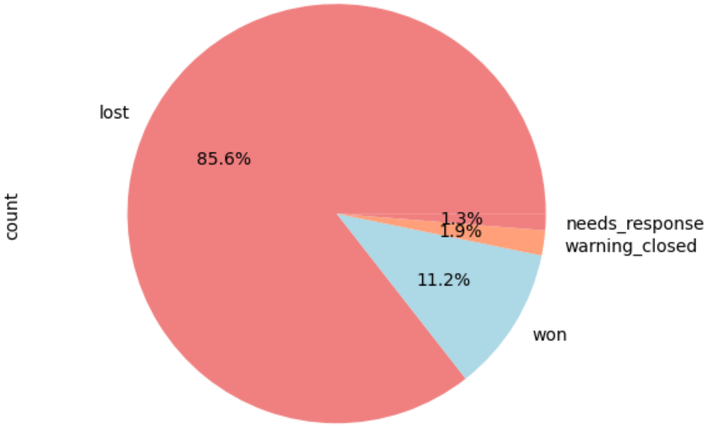
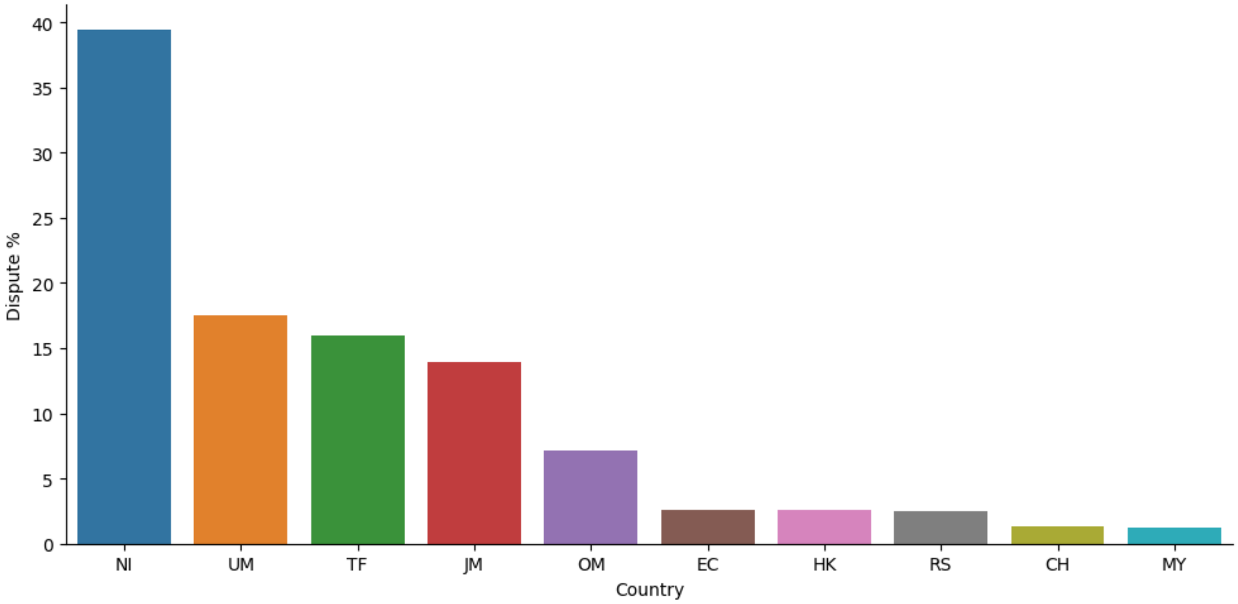


Figure 7 - Dispute status from 2022 to 2023

Canadian banks perceive sales in nations anywhere outside the United States, European Union, Canada, Japan, or Australia as high-risk (Chargebacks 911, 2021). As our client is an online service provider, our client has the capacity to sell globally. However, the analysis concludes it is not advisable to do so. When normalized by sales value,



demonstrates that countries: Oman, Jamaica, French Southern Territories, United States Minor Outlying Islands and Nigeria that dispute represent 7.1% to 39.4% of total national sales. These nations only correspond to a low volume of sales ($2,460) but can reduce the total dispute count by 5 or 1.82%. Ceasing operations in these countries also generally reduces the risk profile of the client's company. It is advised that these nations are added to the blocklist at the earliest opportunity.

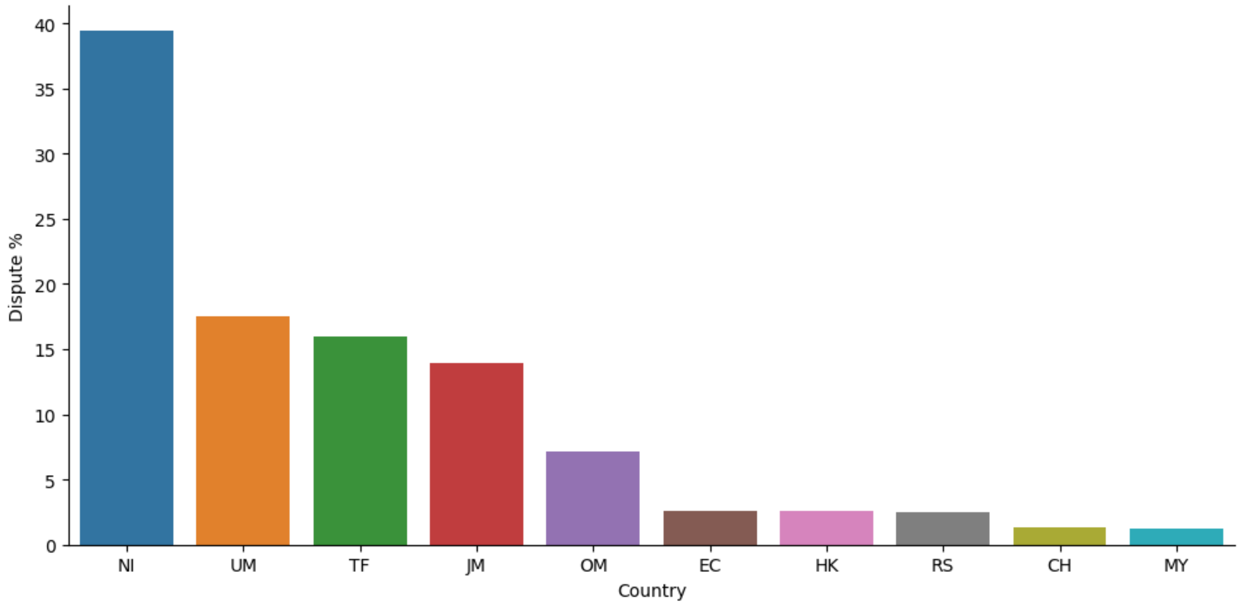


Figure 8 - Countries for Blocklist: High Proportions of Disputes Normalized by Total Sales Volume

A similar analysis was conducted by card fingerprint to assess repeat disputers i.e., users that fail to cancel and dispute multiple months at once. A list of card fingerprints with one or more disputes can be found in Table *2***Error! Reference source not found.**. Repeat disputers account for 53 of the 312 total disputes or 17.0%. It is recommended that these cards are added to the blocklist, and the corresponding customer data is used to deny service.

Table 2 - User Blocklist: Repeat Disputes by Card Fingerprint

No.	Card Fingerprint	Dispute Count
1	vXP4eIMWzzuVqall	4
2	a5WgSGkrhTQsDmKv	4
3	uFyMonHAV7xE2Vwv	4
4	SIs7e4IOMXQKLOR5	3
5	zfUrOHR39mSyrnDT	3
6	WuIVsgA7NtX5fsCn	3
7	nwXYSxirYlHYjlp0	3
8	t4d2L22cnqANDlJF	3
9	Er6U5PFSYm1PTf5L	2
10	cCPVQb3bhilEFeM8	2
11	8t7n5iV95T8y5VMf	2
12	wY1fStlSXfxBUCQ1	2
13	tfTqYiBJivAEJsF7	2
14	MXa1KHWh9vf1YD5E	2
15	Y8yKc7xjihE5Hn4E	2
16	Xj6rATnS5qTbXSPU	2
17	kUXz2g9du3JKY2TI	2
18	VYJNqxgv7g9cW59d	2
19	doXYc3dPNGlOkjEz	2
20	WMKzTOsToOQyCM5x	2
21	vDiUMgBUGY7X71oS	2
	Total Repeat Disputers	**53**

3.0 Feature Engineering

3.1. Seller Message

To begin, the 23 unique seller messages were grouped into six categories. This was done to simplify the analysis and to potentially increase the performance of models by having them predict fewer possible outcomes. Furthermore, the date created was broken down into its constituent components of Day, Month and Year.

The seller messages were also further classified into a binary output, where 1 represents a seller message that could be potentially concerning to the business and 0 represents one that would not be concerning to the business. This further classification was used for the Logistic Regression model, while the other models used the six-category seller message as their target.

We developed unique data pipelines for each model, with only slight variations tailored to suit specific needs. Most notably, the logistic regression model was set apart by its distinctive target variable - binary classified seller messages. The input features for the data pipelines are all categorical variables, this removes the need for the scaling of numerical variables. The input features all had a one hot encoding applied to them in the data pipeline to numerically encode them. This allowed for the models to properly interpret the input categorical features.

3.2. Disputes

Initially, the 'Created (UTC)' column was parsed into a datetime object and renamed to 'Sale Date'. This data was further broken down into separate columns representing the 'Year', 'Month', and 'Day' of each transaction, thus enabling us to analyze temporal trends in the data.

In addition, a new feature was engineered, named 'Dispute'. This binary variable was used as the target for the models, where '1' signifies a disputed transaction and '0' indicates a non-disputed one. This column was derived from the 'Disputed Amount' column by marking any transaction with a non-zero, disputed amount as '1'.

4.0 Model Development

4.1. Predicting Seller Message

In our investigation, we aimed to predict the general seller message based on a set of categorical input features like the description, status, card brand, card funding, card address city, card address state, card address country, card issue country, amount, year, month, and day.

In our initial analysis, we explored several models: Logistic Regression, K-Nearest Neighbours, Random Forest, and Decision Tree. Each model was evaluated based on its performance metrics such as accuracy, precision, recall, and F1 score.

Logistic Regression Model

Logistic Regression works best with a binary output; thus, we simplified seller message categories into concerning (value 1) and not concerning (value 0). The model was able to achieve an: accuracy, precision, recall and f1 score, measuring: 0.952, 0.950, 0.952 and 0.951 respectively. Despite the model's high performance, it may be over predicting or be overfit to the data.

KNN Model

The KNN model achieved high accuracy, precision, recall and f1 score, measuring: 0.946, 0.944, 0.946 and 0.945 respectively. This high performance is very promising; however, the performance is so good that it may be indicating overfitting.

Random Forest Model

The Random Forest model's performance was evaluated through its predictions on the test set. The results were outstanding, scoring 0.967, 0.966, 0.967 and 0.966 on accuracy, precision, recall and f1 score respectively. However, though this level of performance is remarkable, it raised concerns about potential overfitting or data leakage. These results could also be indicative of the model having simply memorized the training data, rather than learning it to generalize from, meaning that the model may struggle when introduced to new, unknown data that does not match the patterns in the training data exactly.

Decision Tree Model

A decision Tree Classifier was used due to its ability to handle categorical data and generate interpretable models. Our first approach was to train the model without any restrictions on the tree depth to understand the feature importance and discern which ones contribute most to the model's decision-making process. For this purpose, the features were one-hot encoded and passed into the decision tree. Our initial evaluation of the model with a limited depth (arbitrarily chosen to be 5 to obtain a baseline) yielded promising results with an accuracy of 0.887, precision of 0.879, recall of 0.887, and F1-score of 0.866. These metrics indicated a well-performing model comparatively; however, the possibility of overfitting is still ever present, more so with a Decision Tree.

4.2. Hyperparameter Tuning and Cross-Validation

Following the initial model builds, we conducted hyperparameter optimization and cross-validation on all models: Logistic Regression, K-Nearest Neighbours, Random Forest, and Decision Tree, to enhance their predictability.

Logistic Regression Model

Our Logistic Regression model underwent hyperparameter tuning, with parameters such as penalty, fit intercept, solver, and maximum iterations being adjusted. We identified the best parameters to be 'newton-cg' for solver, 'l2' for penalty, 100 for maximum iterations, and fit intercept set to True. Following the adjustments, the model demonstrated improved performance metrics. The accuracy was 0.9529, precision was 0.9513, recall score was 0.9529, and the F1 score was 0.9518. Despite the enhancement, a five-fold cross-validation was performed, revealing consistent scores of [0.9498, 0.9469, 0.9498, 0.9460, 0.9493], indicating a risk of overfitting.

KNN Model

For the KNN model, we adjusted parameters including the number of neighbors, weights, algorithm, leaf size, and the power parameter for the Minkowski metric. The optimal parameters were distance for weights, 1 for the power parameter, 5 for the number of neighbors, 40 for leaf size, and 'kd_tree' for the algorithm. After these adjustments, the model's accuracy increased to 0.9466, with precision, recall, and F1 scores reaching 0.9449, 0.9466, and 0.9455 respectively. A five-fold cross-validation yielded accuracy scores of [0.9408, 0.9398, 0.9442, 0.9403, 0.9416], a consistent and improved performance hinting at the possibility of overfitting.

Random Forest Model

Hyperparameter tuning of the Random Forest model involved adjusting the criterion, maximum depth, minimum samples split and leaf, and number of estimators. The optimal parameters were found to be 'gini' for criterion, 30 for maximum depth, 4 for minimum samples split, 1 for minimum samples leaf, and 50 for the number of estimators. Despite the adjustments, the model's accuracy decreased to 0.7698. Precision was 0.8011, recall was 0.7698, and the F1 score was 0.6936. The five-fold cross-validation returned accuracy scores of [0.7791, 0.8120, 0.7909, 0.8196, 0.8683], revealing a significant variance and the potential overfitting of the initial model.

Decision Tree Model

Our Decision Tree model underwent hyperparameter tuning, where we adjusted parameters like criterion, maximum depth, minimum samples split, and leaf. The optimal parameters were found to be 'entropy' for criterion, 30 for maximum depth, 2 for minimum samples split, and 1 for minimum samples leaf. The adjusted model displayed better performance metrics with accuracy, precision, recall, and F1 scores of 0.9434, 0.9437, 0.9434, and 0.9432 respectively. After the adjustments, we performed a five-fold cross-validation on our training data, yielding accuracy scores of [0.9384, 0.9376, 0.9416, 0.9394, 0.9333]. This consistency in results across multiple folds of data indicates the model's robustness and its ability to generalize well to unseen data.

4.3. Model Selection and Justification

In the process of selecting the most suitable model for our predictions, we investigated several machine learning models including Logistic Regression, KNN, and Random Forest. However, after a meticulous process of model exploration, we have chosen the Decision Tree Model as our preferred model for predicting seller messages. Our choice is primarily driven by three key factors: performance metrics, model interpretability, and robustness against overfitting.

Firstly, the decision tree’s innate ability to handle categorical data efficiently made it a suitable choice given the nature of our input features. This was further affirmed by the model’s initial performance that displayed a balance of accuracy, precision, recall, and F1-score, which was superior to the other models evaluated during our initial explorations. It is also important to note that its precision and recall metrics, crucial in our context to reduce both false positives and false negatives, were considerably high assuring us of its capability to accurately predict concerning seller messages.

Secondly, the decision tree model is favored for its transparency and interpretability. It offers valuable insights into the decision-making process, providing an understandable representation of how different features contribute to the final prediction. Moreover, we witnessed a substantial enhancement in performance after conducting hyperparameter tuning. By optimizing certain parameters, we were able to improve the model’s accuracy, precision, recall, and F1-score.

Finally, the decision tree model demonstrated robust generalization capabilities, as evidenced by the consistent results during the 5-fold cross-validation process. This suggests the model’s ability to perform well on unseen data, making it a reliable tool for practical applications. It also signifies that our decision tree model is less likely to suffer from overfitting.

While the Random Forest model demonstrated seemingly perfect results on the validation set, its performance significantly dropped post hyperparameter tuning, and it showed significant variance in cross-validation, which indicates potential overfitting and raises concerns about its ability to generalize to new data. The Logistic Regression and KNN models, although showing improvement after tuning, still did not reach the balanced high performance of the Decision Tree model and exhibited possible signs of overfitting.

To sum up, the Decision Tree model demonstrates an excellent balance between performance and robustness, allowing for reliable, accurate predictions while also offering clear insight into the decision-making process.

4.4. Predicting Disputes

Our approach to interpreting disputes employed several models: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Gradient Boosting Classifier, and Neural Network. Each model was evaluated based on performance metrics such as accuracy, precision, recall, and F1 score.

KNN Model

KNN, a non-parametric algorithm, was utilized due to its simplicity and effectiveness when dealing with multiclass problems. For each transaction, the model identifies the K closest transactions in the feature space and assigns the most common dispute category among these neighbors. A range of K values were tested through grid search, and K=20 provided the best performance. The model achieved an accuracy of 0.576, with a precision of 0.509, recall of 0.537, and F1-score of 0.523, indicating that the model was somewhat able to balance between correctly predicting true positives (precision) and not overlooking actual positives (recall). These metrics show mediocre performance for dispute predictions.

SVM Model

The SVM algorithm was also used due to its effectiveness in high-dimensional spaces, even in cases where the number of dimensions exceeds the number of samples. SVM uses a subset of the training points to build a hyperplane that maximizes the margins between the two classes. The model demonstrated an accuracy of 0.576, a precision of 0.507, a recall of 0.704, and an F1-score of 0.589. This indicates that the model is less likely to overlook true positives in comparison with the KNN model, however, the low precision score also showcases the model predicting a relatively high rate of false positives. This can lead to predicting outcomes that may end in unnecessary resources spent on investigating disputes.

Gradient Boosting Classifier

The Gradient Boosting Classifier is a powerful machine learning algorithm that builds an ensemble of decision trees in a stage-wise fashion and generalizes them by allowing optimization of an arbitrary differentiable loss function. This model yielded perfect accuracy, precision, recall, and F1-score. However, though this level of performance is remarkable, the model’s results most likely suggested that it was overfitting to the data. An overfit model would perform well on training data but could potentially perform poorly on unseen data.

Neural Network Model

A neural network model was employed given its capacity to model complex, non-linear relationships, and the ease that it comes with in terms of handling high dimensional data. The model consisted of one input layer, a hidden layer, and an output layer. Each layer applying the ReLU activation function, except for the output layer, which used a sigmoid activation. The initial evaluation revealed promising outcomes with a test accuracy of 0.9966 and test loss of 0.0224. This high-performance accuracy likely has to do with the imbalanced dataset, where the “non-disputed” payments significantly outnumber the disputes. It is suspected that this model that achieves high accuracy by primarily predicting the majority class but fails to adequately learn the distinguishing characteristics of the minority class.

4.5. Hyperparameter Tuning and Cross-Validation

KNN Model

In this model, hyperparameter tuning is carried out using GridSearchCV to optimize the 'n_neighbors' hyperparameter in the K-Nearest Neighbors (KNN) classification algorithm. The tuning process involves evaluating different 'n_neighbors' values (5, 10, 20, 30, 50, and 100) through 5-fold cross-validation. Model validation is performed using a train-test split approach, where the KNN model is trained on the resampled training data and evaluated on the test set. The results show that the best 'n_neighbors' value for this model is 20.

SVM Model

To find the best combination of hyperparameters in the SVM Model, a cross-validation process with 5-fold validation was conducted. The hyperparameters under consideration were the regularization parameter C and the kernel type. C controls the trade-off between maximizing the margin and minimizing the classification error, while the kernel type determines the type of decision boundary used by the SVM. After the cross-validation, the best hyperparameters were identified to be C=0.1 and the 'rbf' kernel. These hyperparameters were then used to train the final SVM model which performed reasonably well outside of the relatively high prediction of false positives.

Gradient Boosting

The hyperparameters under consideration were the number of estimators and the learning rate. The dataset was initially split into training and test sets, and the training data was further resampled using RandomUnderSampler to address class imbalance. The hyperparameters were then tuned using 20-fold cross-validation to identify the optimal combination of n_estimators and learning_rate. After the hyperparameter tuning, the best model was trained using the selected hyperparameters, n_estimators=100 and learning_rate=1. Subsequently, the model's performance was evaluated on the test set, revealing perfect results that indicated the model is overfit and likely to perform poorly on other data sets.

Neural Network

The Hyperband tuner is used with the defined parameters, focusing on optimizing the validation accuracy. The tuner.search method is called on the training data with a validation split of 10% for hyperparameter tuning. The best hyperparameters are retrieved from the tuning process, and the final model is built using them. The final model is then trained on the entire training set using the optimal hyperparameters, with a validation split of 10% to monitor the training. Although cross validation was not explicitly used, a validation split was applied in both the hyperparameter tuning and final training phases providing a way to internally validate the model later.

4.6. Model Selection and Justification

KNN's simplicity, being a non-parametric algorithm, allows it to not make strong assumptions about the underlying data distribution. This simplicity also extends to the model's interpretability, where predictions are based on the most common category among the K nearest neighbors, making it easier to explain and understand. Compared to models like Gradient Boosting Classifier and Neural Networks, KNN might be less prone to overfitting and thereby, better able to predict disputes. The tuning process that identified the best 'n_neighbors' value through GridSearchCV tailored the model to the specific problem, and although the performance metrics demonstrate mediocre performance, the model outperformed the SVM in terms of false positives while avoiding issues with overfitting and imbalanced data. For these reasons, the KNN model was selected.

5.0 Conclusion

The analysis of the file hosting client's payment processing data has revealed insights into risk factors: failed payments and disputes. Key findings include the identification of non-linear relationships for predictive modeling of seller message, detection of imbalances in card funding types, understanding dispute patterns, strategies to mitigate risk in subscription packages, and the need for implementing blocklists for high-risk countries and users. Overall, there is an observable downward trend in disputes aligned with payment verification with further declines anticipated if recommendations are employed.

In the investigation to predict the seller message, several models were explored including Logistic Regression, K-Nearest Neighbours, Random Forest, and Decision Tree, each evaluated on performance metrics such as accuracy, precision, recall, and F1 score. Ultimately, the Decision Tree model was chosen as the preferred approach for predicting seller messages, primarily driven by its balanced performance metrics, interpretability, and robustness against overfitting, while demonstrating an excellent ability to handle categorical data efficiently and provide accurate, reliable predictions.

The study also explored predicting disputes through various machine learning models, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Gradient Boosting Classifier, and Neural Network. While the KNN model was selected for its balance between simplicity and accuracy, challenges such as overfitting and imbalanced data handling were also exposed, suggesting areas for future research and improvement.

By leveraging these findings and implementing the recommended models, the file hosting client can minimize risk, protect revenue, and continue to foster a healthy relationship with Stripe for ongoing stability in the digital payment landscape.

6.0 References

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), 175-185.

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

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Schölkopf, B., & Smola, A. J. (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

Rokach, L., & Maimon, O. (2014). Data mining with decision trees: theory and applications (2nd ed.). World scientific.

Sperandei, S. (2014). Understanding logistic regression analysis. Biochemia medica, 24(1), 12-18.’

Chargebacks 911. (2021, July 7). *High-Risk Business*. Retrieved from Chargebacks 911: https://chargebacks911.com/high-risk-business/

Stripe. (2023, July 13). *Stripe Services Agreement — Canada*. Retrieved from Stripe: https://stripe.com/en-ca/legal/ssa

Stripe. (n.d.). *API Reference*. Retrieved from Stripe API: https://stripe.com/docs/api/cards/object#card_object-fingerprint

Stripe. (n.d.). *Stripe Docs* . Retrieved from Measuring disputes: https://stripe.com/docs/disputes/measuring