Analysis of Order Ratings, Sales, and Customer Relationship on a Brazilian E-Commerce Website

Authors: Adam Foley, Mann Purohit, Nhat Pham, Sarvagna Shukla

I. Summary:

A. Business Question

Brazil has the biggest and fastest growing e-commerce market in Latin America and the Caribbean, concentrating 34 percent of the market in the region in 2020 [1]. Since the market is dominated by numerous large-size marketplaces rather than by a few like in the US (Amazon, Ebay etc.), Brazilian merchants face the operational burden of integrating with those marketplaces [2]. Olist was founded to solve that problem and help merchants connect to a larger customer base. The platform provides tools that support all the stages of an e-commerce operation, from managing inventory to fulfillment, customer services, and payments [3]. As a result, the driver for the company's business is to attract more clients and raise the quality of the process. The motivation in this project is to support this effort through exploring Olist's database.

B. Data Mining Solution

1. Description of the dataset:

The data provided by Olist[4] includes 8 tables (Figure 1, Appx.) of data on all areas of a transaction including customers, sellers, orders, products, payments, etc. Although they provide ways to connect these tables through "keys", in many instances it was not in a tidy format or easy to query. Therefore, we performed data cleaning and tidying for some of the tables. For example, in the payments tables (Figure 2, Appx) we consolidated the table to have a composite key of order_id and payment_type. Therefore, we had to find all instances of a payment type for each order and aggregate their payment installments and payment value.

Once we decided on the schema and keys we would like to use, we leveraged a MySQL instance hosted on AWS in order to store our database as there were over 400k rows. The reason we decided to go with a MySQL instance was to ensure that all of the table tidying we did was captured in each team member's work which ensured consistency amongst the various problems we were trying to solve. In Figure 3 (Appx.), you will see the final schema that we decided on and the fields we needed to answer the questions we had. All of the code for processing and loading the data into MySQL can be found in this notebook.

2. Project goals:

Based on the multi-level datasets available, we provided a comprehensive understanding of Olist's business through four learning objectives. First, we examined customer satisfaction based on analysis of order review scores. Our final results indicate that issues with the delivery process such as long delivery time and late deliveries, high total freight and price value, and large orders negatively impacted customer experience. Second, we performed a sentiment analysis of product reviews, and were able to create a reliable Natural Language Processing model that predicts a customer review's sentiment given a textual message. Third, we would like to target larger segments of customers by knowing their shared values, likes and buying behavior. Our customer segmentation analysis identified different clusters of customers, yet without labels and limited data it was difficult to discern each cluster's traits. Finally, we examined Olist's sales activities and discovered patterns on seasonality and trend from historical sales which might be important for the logistic concept. We also identified important product categories, differences in performance across different geographical markets, and areas that Olist should focus on to improve and expand its business.

II. Data Analysis and Modeling

A. Customer Satisfaction Analysis:

Surveys report that 93% of consumers are influenced by online reviews when making purchase decisions [5]. By leveraging its order review data, Olist can gain insights into its customers' wants and needs, and therefore enhance its services and products to provide better customer experience and attract more merchants to the platform. Through EDA and machine learning models for classification, we are interested in identifying key features that affect a good or bad review score, using information on reviews, orders, customers, sellers, products, and payments.

1. Exploratory Data Analysis

We started by looking at the distribution of review score ratings of all the orders in our dataset. The majority of our orders were given 4 and 5 scores (Figure 4, Appx.). To improve the imbalanced distribution, we categorize 1, 2, 3 scores as negative scores and 4, 5 scores as positive scores. Next, the team examined the attributes that could potentially impact review scores.

We first assessed factors related to the delivery process, including delivery time (number of days it takes from time of purchase to actual delivery date) and late

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test

delivery time (difference in estimated and actual delivery date). We found that orders that were delivered late received significantly lower scores on average than those delivered on time (Figure 5, Appx.), and that the longer it takes for an order to arrive, the less happy the customers are (Figure 6, Appx.). Also, mean delivery time among all orders is 10 days, much higher than Amazon (3.39 days average, 2017 data) [6]. This is a KPI Olist might want to improve to maintain its competitiveness in the long term.

The team also reviewed other order-based characteristics. We saw that large orders (those with multiple items or multiple sellers) have lower scores on average than single-item orders (Figure 7, 8, Appx.). This could be because large orders are more prone to fulfillment errors. Yet, 86% of all orders are single-item-orders so the pattern might not be representative. Meanwhile, we found no difference in the average number of photos, product description length, and product name length per order across different review scores. Proportion of negative and positive reviews is similar across different payment types, indicating that customers' experience do not depend on payment methods chosen. When reviewing sellers' profiles, we saw that sellers with fewer negative reviews are the popular ones (high order volumes) who also have a low proportion of delayed orders (Figure 9, Appx.).

2. *Data Cleaning & Feature Engineering*

As mentioned above, we assigned a binary variable for 'positive' reviews (4, 5 scores) and 'negative' reviews (1,2,3 scores). The number of positive and negative reviews was still imbalanced, and we will talk about the effect it had on our models in the *Modeling & Evaluation* section. In addition, we created some aggregate measures to assess their potential relationship with review scores, including order value, item count per order, seller count per order, delivery time, order counts per product, and order count per seller.

We noticed several numerical features that were heavily positively skewed. These features were not transformed for the first few models. After that, models were rerun using log-transformed data to see whether this improves model accuracy. Next, multicollinearity was checked for by assessing correlations between predictor features, as this can cause issues with some models. The multicollinearity matrix showed that this is not an issue. We also performed one-hot encoding for categorical variables such as payment type and purchase day (of week) to prepare for modeling. We then ended up with 27 attributes to be considered for this analysis. Finally, because independent features were on different scales, they were transformed and normalized using the StandardScaler function in sklearn.

3. *Modeling & Evaluation*

We split the dataset into 80% for the training set and 20% for the testing set, then used Random Forest and Logistic Regression algorithms for this analysis. These models were chosen as appropriate for modeling our binary response variable, "negative" versus "positive" review scores.

a. *Evaluation Metrics*

As the main evaluation metrics, we used the F1 scores of both classes and Area Under the Curve (AUC). The F1 score calculates the harmonic mean between precision and recall and is a suitable measure because there is no preference for false positives or false negatives in this case. AUC is a common, easily interpretable metric for binary classifications. Graphs can be plotted together for comparison across different models to identify the best performer. (See Figure 10, Appx. for a comparison of all models).

b. *Logistic Regression*

Running the Logistic Regression (LR) model with 27 features gave us a fairly good weighted average F1 score of 0.78. However, it is notably worse at predicting negative than positive reviews, with the recall rate being 0.26 for the negative class and 0.96 for the positive class. The AUC has a value of 0.704. Figure 11 (Appx.) shows the top 10 feature importance for the model. As expected, delivery time and whether or not the order is delayed are the most important features, followed by seller count per order and item count per order, and whether the order used boleto as the payment type.

Applying log transformation to right-skewed features and rerunning LR, we achieved an AUC of 0.698, which underperformed the basic LR model. In attempts to improve our Logistic Regression model, we performed random oversampling and undersampling on the imbalanced original data set and rerun the model. LR with oversampled data results in an AUC of 0.707, and while the recall rate improved (0.49), the precision rate for both classes worsened noticeably (0.44 for positive class) compared to the original model. LR with undersampled data gave us similar results.

c. *Random Forest*

We also tried running a basic Random Forest model to see if the accuracy is improved. Random Forest classifier algorithm is a set of decision trees from randomly selected subsets of the training set. It aggregates the votes from different decision trees to decide the final class of the test