UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SCHOOL OF COMPUTING
Faculty of Engineering

Lap Work Assignment

MCSD1143– 01

Advanced Analytics for Data Science
Supervised Learning Models

Lecturer's Name: Assoc Prof Dr Roliana Ibrahim

Student's Name:

Ahmed Hashim Taha Salim (MCS211041)

1.0 Introduction

In today's competitive and saturated market, having an edge over the competition translates into significant benefits for organizations. The battle among suppliers of products and services to expand their consumer portfolio and retain customers has been a constant throughout history. This competition has driven companies to create countless strategies, actions, and models, leading to the emergence of entire disciplines studied at universities worldwide. It is thanks to this competition that statistical techniques and methods are increasingly common in companies. These techniques enable the proper management of large volumes of data, leading to more accurate conclusions and even the ability to anticipate events with a certain degree of probability.

Predictive models are the outcome of grouping statistical techniques and utilizing them to analyse historical and current data, allowing us to make predictions about uncertain events that are yet to happen. Through their application, predictions and probabilities of occurrence can be obtained for each analysed subject. These models are highly valuable in organizations as they provide crucial information for decision-making. With their high levels of precision, predictions can be verified, adjusted, and used to establish future areas of focus, ultimately leading to greater benefits.

Regarding the business insights obtained from the previous assignment, particularly in relation to customer experience through reviews and satisfaction levels, it is worth noting that the term "Customer Experience" has become commonplace in today's business landscape.

This report is based on a database from Olist, an e-commerce company headquartered in Curitiba, Brazil. The company's main activity is to offer sales solutions and services to merchants and companies who wish to sell their products online. They have developed a platform for shopkeepers of all sizes and segments to register their products to be sold at Olist. Their mission, as stated on their website, is to strengthen global trade. This part of the report aims to find ways to increase Olist’s strength in its e-commerce customer's experience using supervised machine learning models. Specifically, our overarching question is: How does item review scores affect the sale of the products? The software used for this purpose is called RapidMiner and version 9.10 was used.

2.0 Dataset Description

The upcoming work will utilize the dataset available on the Kaggle website, specifically the "Brazilian E-Commerce Public Dataset by Olist." This dataset contains information about transactions conducted by clients of the company Olist between the years 2016 and 2018. The dataset was published on September 21, 2018 by Andre Sionek and Olist. It comprises data from 99,941 purchases made on the platform between October 3, 2016, and August 29, 2018.

The dataset is organized into eight CSV (Comma Separated Values) structured tables as seen in schema in Figure 1 below. These tables include a total of 36 mixed variables, consisting of both categorical and numeric types. These variables provide detailed information about each purchase, such as the order's final status, total amount paid, product price, product attributes (e.g., type, description length, number of photos, dimensions, and weight), as well as customer reviews about their experience. For a comprehensive overview of the dataset's variables and their types, please refer to Table 1 below for detailed information.

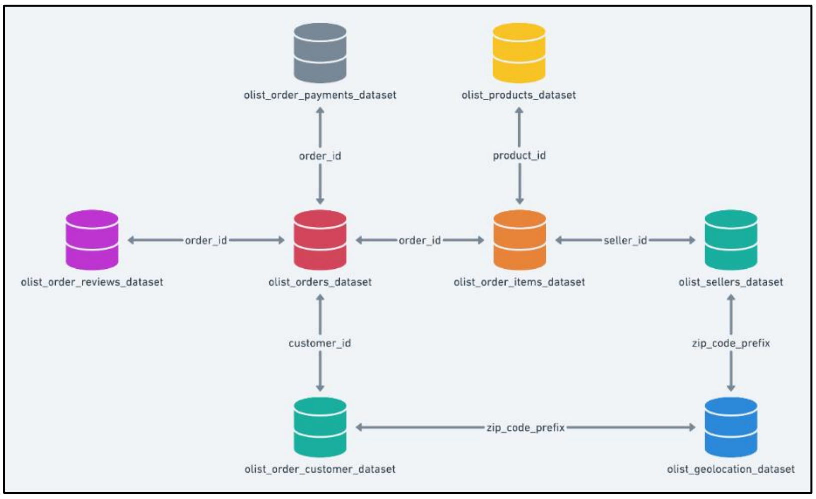
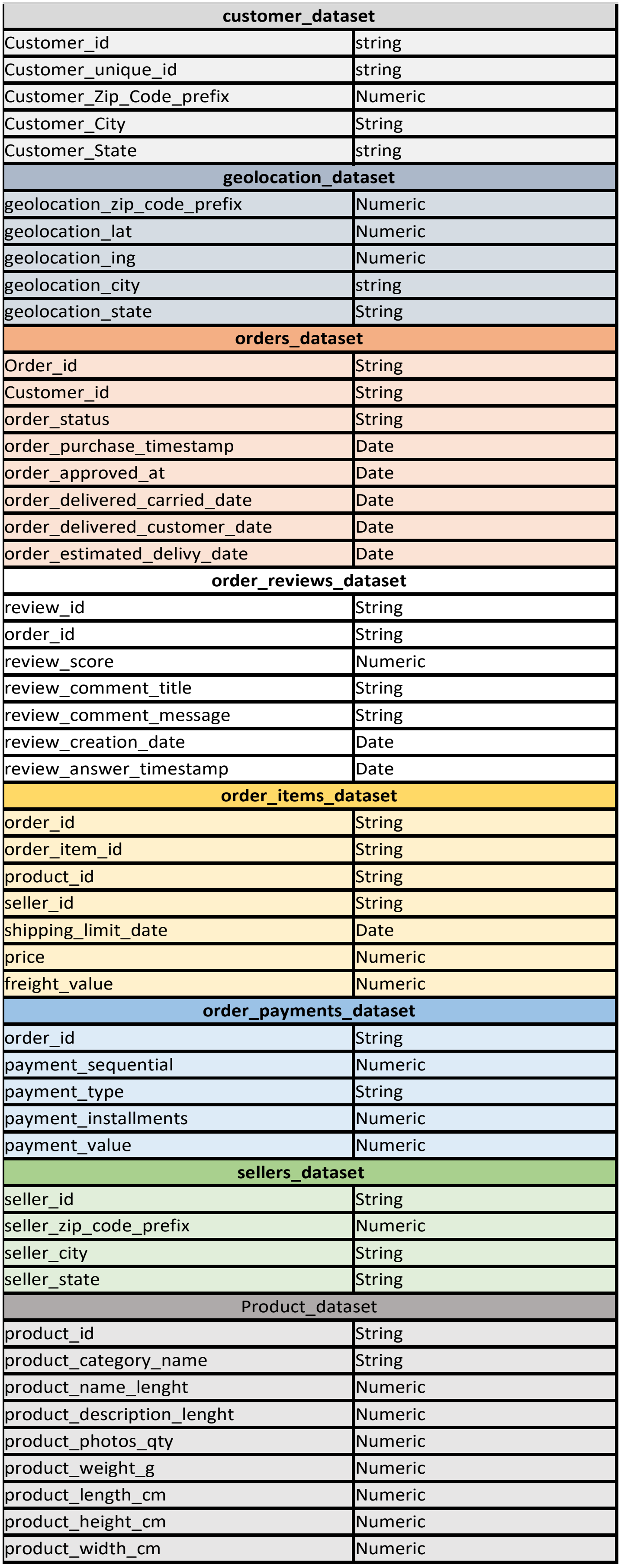


Figure 1: Data Schema

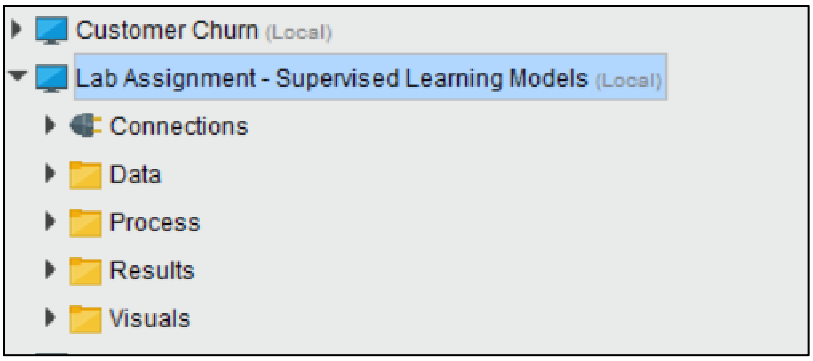
Table 1: Descriptions of Attributes



customer_dataset	
Customer_id	string
Customer_unique_id	string
Customer_Zip_Code_prefix	Numeric
Customer_City	String
Customer_State	string
geolocation_dataset	
geolocation_zip_code_prefix	Numeric
geolocation_lat	Numeric
geolocation_ing	Numeric
geolocation_city	string
geolocation_state	String
orders_dataset	
Order_id	String
Customer_id	String
order_status	String
order_purchase_timestamp	Date
order_approved_at	Date
order_delivered_carried_date	Date
order_delivered_customer_date	Date
order_estimated_delivy_date	Date
order_reviews_dataset	
review_id	String
order_id	String
review_score	Numeric
review_comment_title	String
review_comment_message	String
review_creation_date	Date
review_answer_timestamp	Date
order_items_dataset	
order_id	String
order_item_id	String
product_id	String
seller_id	String
shipping_limit_date	Date
price	Numeric
freight_value	Numeric
order_payments_dataset	
order_id	String
payment_sequential	Numeric
payment_type	String
payment_installments	Numeric
payment_value	Numeric
sellers_dataset	
seller_id	String
seller_zip_code_prefix	Numeric
seller_city	String
seller_state	String
Product_dataset	
product_id	String
product_category_name	String
product_name_lenght	Numeric
product_description_lenght	Numeric
product_photos_qty	Numeric
product_weight_g	Numeric
product_length_cm	Numeric
product_height_cm	Numeric
product_width_cm	Numeric

3.0 Data Processing

The data processing stage is crucial for conducting any supervised machine learning analysis. In this section, we will describe the tasks undertaken and the attributes used to implement the analysis criteria, four subfolders were created, where “Data” is used to import the initial Olist datasets, “Process” is used to store the operators process, “Results” stored the output of the Process using “Store” operator and “Visuals” where the analytics visualisations were saved.



Customer Churn (Local)
Lab Assignment - Supervised Learning Models (Local)
Connections
Data
Process
Results
Visuals

Figure 2: Subfolders

The initial step of this investigation involved loading the various data sets into the RapidMiner software using the "Read CSV" operator. Additionally, the "Join" operator was used to combine the data sets based on their Primary/Foreign keys and facilitate working with the structured data arrays. During this process, one of the data sets named "olist_geolocalization_dataset" was excluded as it contained irrelevant data for the analysis. As a result, the final dataset consisted of seven files instead of the original eight. The following Figure 3 below shows the process of reading and joining the datasets, this process is saved within “Process” subfolder under the name “1. Load and Join the datasets (7 tables)” and its result is stored “Store” operator in Results subfolder.

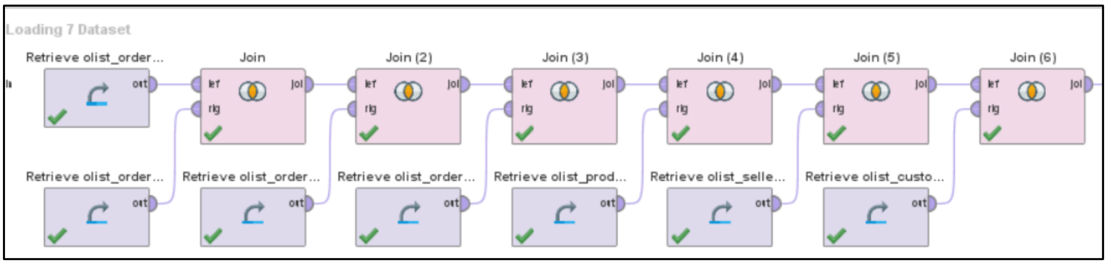


Figure 3: Read CSV, Join Operators

Next, the "Generate Attribute" operator was employed to create additional variables, and the "Select Attributes" operator was used to choose specific variables for inclusion. The following Figures 4,5 are showing the steps of the process and the functions of 11 new variables.

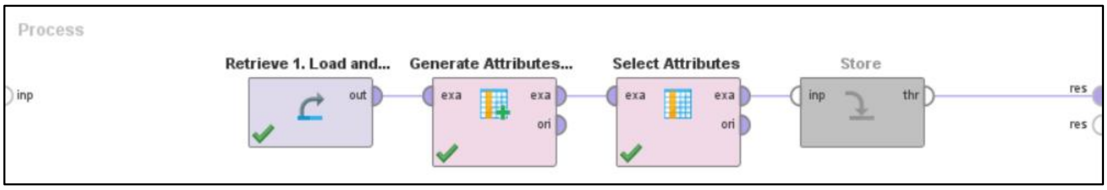
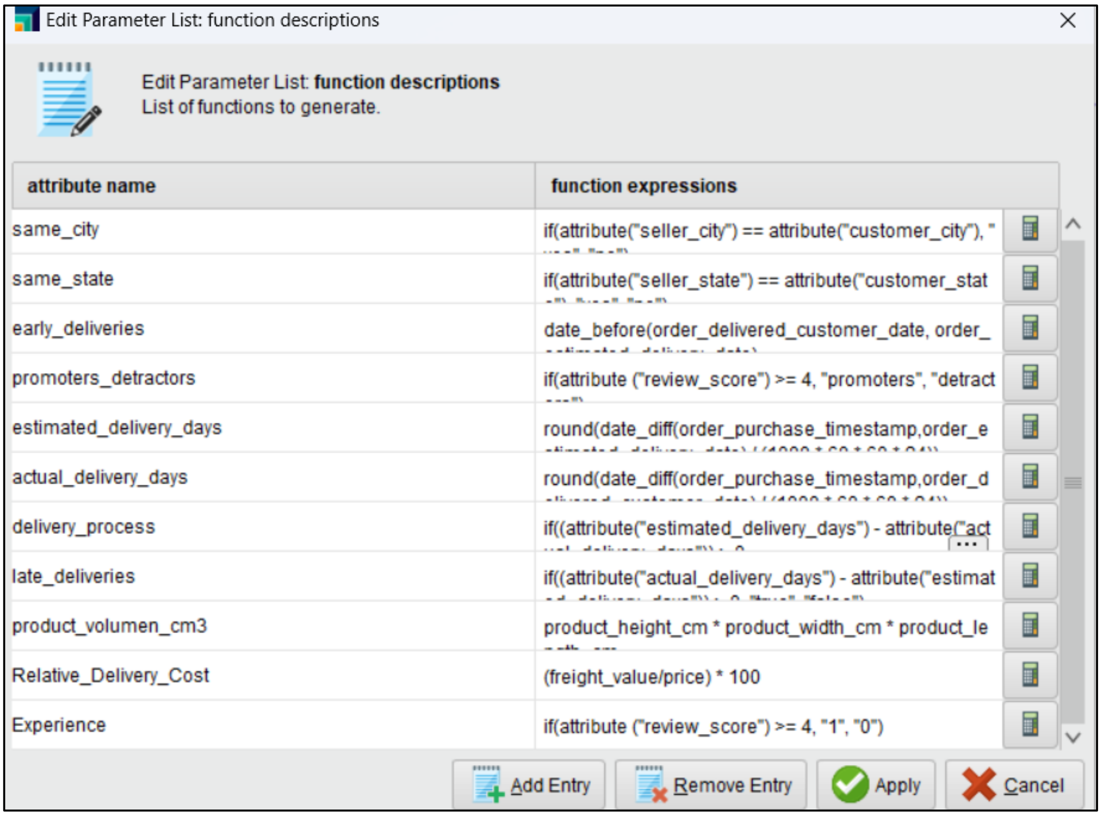


Figure 4: Generate New Attributes



Edit Parameter List: **function descriptions**
List of functions to generate.

attribute name	function expressions
same_city	if(attribute("seller_city") == attribute("customer_city"), "
same_state	if(attribute("seller_state") == attribute("customer_stat
early_deliveries	date_before(order_delivered_customer_date, order_
promoters_detractors	if(attribute ("review_score") >= 4, "promoters", "detract
estimated_delivery_days	round(date_diff(order_purchase_timestamp,order_e
actual_delivery_days	round(date_diff(order_purchase_timestamp,order_d
delivery_process	if((attribute("estimated_delivery_days") - attribute("act
late_deliveries	if((attribute("actual_delivery_days") - attribute("estimat
product_volumen_cm3	product_height_cm * product_width_cm * product_le
Relative_Delivery_Cost	(freight_value/price) * 100
Experience	if(attribute ("review_score") >= 4, "1", "0")

Figure 5: New Attributes & Function expressions

The eleven new variables as shown in Figure 5 where extracted from the union of datasets from previous process and others that were created from the transformation of others.

In the data processing stage, several variables related to the waiting time between purchase and delivery were generated using the "Generate Attribute" operator. These variables include the columns "estimated_delivery_days” and "actual_delivery_days" as illustrated in Figure: 5.

The " estimated_delivery_days " column indicates the number of days that have passed between the day customers placed the purchase order (retrieved from the "order_purchase_timestamp" column) and the day the platform provided as the estimated arrival day (from the "order_estimated_delivery_date" column). Similarly, the " actual_delivery_days " column represents the number of days it took for the order to be delivered to the customer (from the "order_delivered_customer_date" column).

The first categorical variable is called " delivery_process," which categorizes the orders as either “In Advance”, "On Time" or "Late" based on the comparison of the estimated delivery days and the actual delivery days. If the difference is less than zero, indicating a delay in delivery, the variable is assigned the value "Late." If the difference is exactly zero, indicating an on-time delivery, the variable is assigned the value "On Time". Otherwise, if the difference is greater than zero and not equal to one, indicating an early delivery, the variable is assigned the value "In Advance.

The second variable is a quantitative variable called "late_deliveries." It takes a value of 1 if the delivery was made later than the estimated date communicated to the customer and 0 if it was made within the stipulated timeframe. Consequently, the "early_deliveries" variable represents the opposite scenario, resulting in a qualitative variable with values of either "true" or "false."

On the other hand, "same_city" and “same_state” were incorporated as a binomial variable to show (with 1 and 0) if the client and the seller are from the same city/state or not. This seeks to identify if the distance between supply and demand is a relevant driver for the experience of the users. Also, it was investigated if the size of the product was a relevant factor for the qualify the experience, for this reason the variable "product_volumen_cm3" was included, which arises from the multiplication of the height, width, and depth in centimeters of the product bought.

Subsequently, one variable "Relative Delivery Cost" were added to corroborate if the cost of delivery over the total amount paid was relevant. And, finally, the variable to be predicted was created: “Experience”. This can take two values: promoter or detractors, the promoters are the customers with review scores 4 and 5 in the survey; the detractors, therefore, are the users who scored between 1 and 3. Table 2 below illustrates the selection of the 29 variables along with the new ones generated.

Table 2: Attributes Selection

order_id	same_city
order_purchase_timestamp	same_state
order_approved_at	product_category_name
order_delivered_carrier_date	product_description_lenght
order_delivered_customer_date	product_photos_qty
order_estimated_delivery_date	product_weight_g
estimated_delivery_days	product_volumen_cm3
actual_delivery_days	Relative_Delivery_Cost
early_deliveries	Experience
late_deliveries	delivery_process
payment_type	promoters_detractors
payment_value	seller_state
Price	seller_city
freight_value	customer_state
customer_city	order_status

Once the bases were unified and the variables to be considered selected, the next stage began. remove invalid values. There, more than 2000 orders that had values were eliminated null in the field "order_delivered_customer_date" and. In addition, only payment orders were considered with status delivered to have as many dates as possible with non-empty values and, in the cases in which the date of approval of the order was null, it was not selected.

One of the consequences of the filter to include only the shipments made and the removal of null values is that the atypical values or outliers that came from the given by wrongly imputed dates. Figure 6 below shows the third process which is Data Cleaning after retrieving the selected attributes described in Table 2 previously.

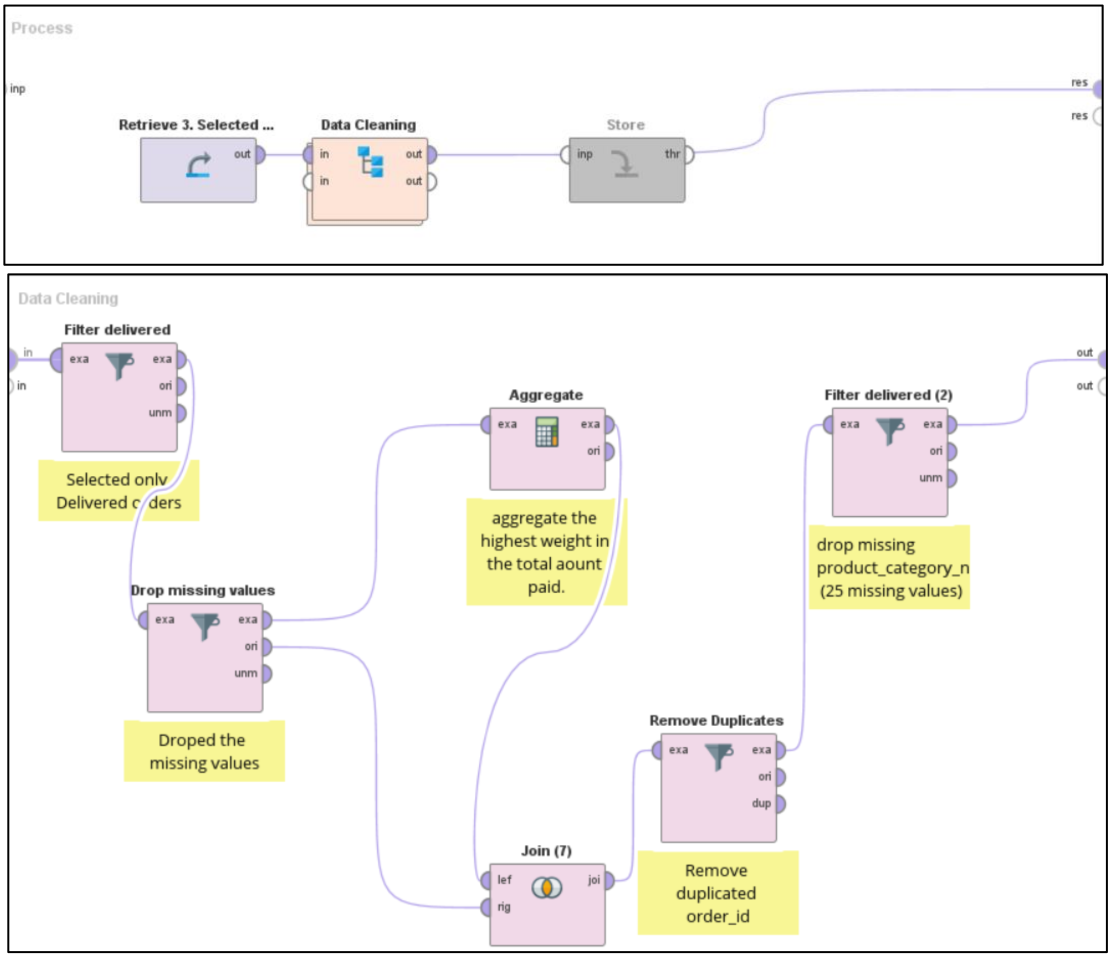


Figure 6: Data Cleaning

There were also customers who chose to use multiple payment methods or vouchers. This caused orders to be duplicated, along with the survey responses from those customers. This issue was resolved by selecting the payment method with the highest weight in the total amount paid. After making this change, the total payments were adjusted to reflect that customer paid the full amount using only that payment method. This ensured that the Relative Delivery Cost variable was not affected.

The data cleaning process resulted in a total of 90,907 orders to be classified based on the obtained experience, along with 28 variables that will be used to feed the selected models to achieve the best results.

4.0 Analysis of data

The results of the data processing led to almost 91,000 orders compared to the original dataset with 112,516 orders, in order to predict according to the experience obtained and variables that will feed the models selected. However, the cleaned dataset has 28 variables of mixed origin. In the next lines, a descriptive analysis of the quantitative variables and qualities considered for the analysis.

The first variable to define is order_id. This works as the primary key in the database of data since it identifies each purchase order sent to customers with a unique name. This variable was used as ID in RapidMiner to make predictions and it is of type nominal.

On the other hand, we present the attribute to predict: Experience. As we mentioned in the previous section, this arises as a result of the transformation of the numerical variable score, and it is of the binomial type since it can take two values, promoter and detractor. In RapidMiner, this variable was defined as a label to indicate that it was on which it should calculate the performance of the models. Figure 7 shows the frequency and the percentages that were the base from which the algorithms carry out the predictions.

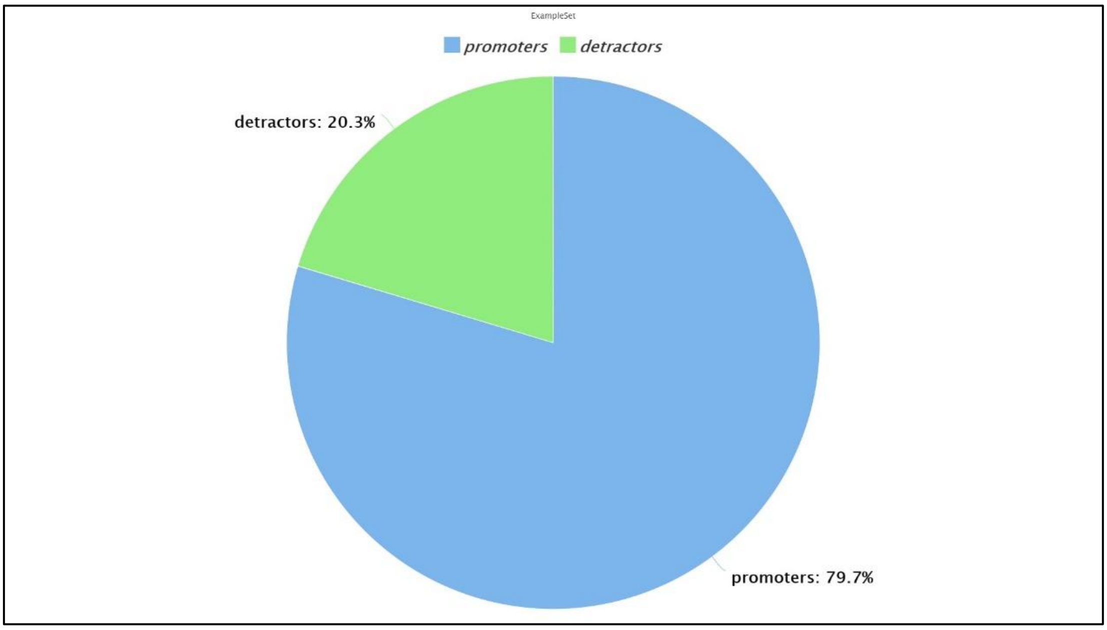


Figure 7: Experience

The customers of this dataset made their purchases with four different forms of payment, and this was recorded in the variable payment_type. The types of payment used were: credit card, debit card, boleto and vouchers. The participation of each means of payment on the total purchases can can be seen below in Figure 8.

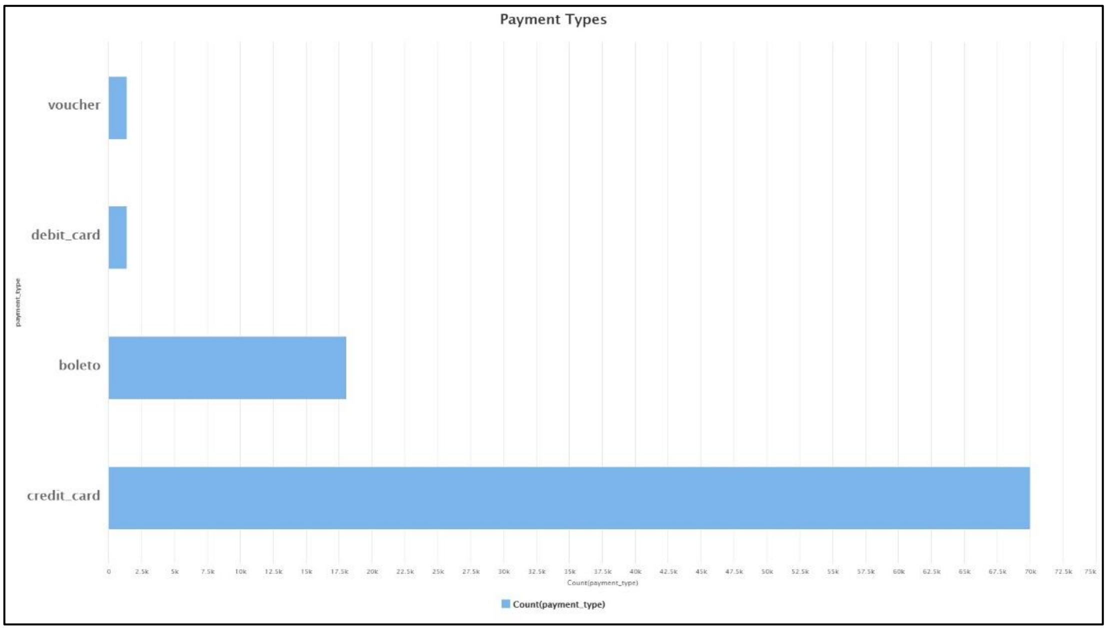


Figure 8: Payment Types

In addition to the variable to be predicted, there are two other binomial variables that take a value when a condition is met and another when it is not. In this case the variables took numeric values, therefore the value is 1 if the condition is true and 0 otherwise. These variables are: late_deliveries, same_state and same_city. In the first one, it is verified that the number of late shipments, shipments delivered after the estimated date to the customer, totaled 6,859 cases (7.5% of the total purchases). Regarding the number of customers who bought from vendors in the same city, the records add up to 4,651 sales that in relative terms is 5.1% of the total. While those whom are from same state 35.9% (32675 records).

Continuing with the detail of the attributes, we find four variables that they are of the type date: “order_purchase_timestamp”, “order_approved_at”, “order_delivered_carrier_date”, “order_estimated_delivery_date” and “order_delivered_customer_date”. None of these variables was considered as input variables of the models since that did not improve the results. Regarding its description, Table 3 details the minimum and maximum values, along with the duration of days.

Table 3: Characterization of date type variables

Attribute	Type	Min	Max	Duration
order_purchase_timestamp	Date-time	Oct 3 2016 9:44 AM	Aug 29, 2018 3:00 PM	595
order_approved_at	Date-time	Oct 4, 2016 9:43 AM	Aug 29, 2018 3:10 PM	694
order_delivered_carrier_date	Date-time	Oct 8, 2016 10:34AM	Sep 11, 2018 7:48 PM	703
order_estimated_delivery_date	Date-time	Oct 27, 2016 12:00AM	Oct 25, 2018 12:00 AM	728
order_delivered_customer_date	Date-time	Oct 11, 2016 1:46 PM	Oct 17, 2018 1:22 PM	735

The process of obtaining satisfactory results involves multiple iterations, where small changes are made to determine whether the implementations improve accuracy. First, we need to establish a baseline measurement. This baseline represents what we would obtain if no model was applied and serves as our reference point for measuring improvements. In this analysis, since the predicted variable is binary, the baseline is determined by the most frequent value, which is the promoters. Their relative presence in the total records is 79.7%.

5.0 Applied Models

In this section, we will first define the models used to carry out predictions about customer experience, then, the variables of input selected for those models and finally present the results of model performance.

The models applied to predict the customer experience were the following: K-Nearest Neighbors (Knn), Naive Bayes, Random Forest, Decision Tree and Logistic Regression. To test the variables and models, a sample of 6,000 orders was taken to maintain the same proportion of promoters and detractors. This sample was used to perform optimization tests on the models and make a proper selection of variables.

The next step involved normalizing the variables and dividing the sample into a training set (80% of the total sample) and a test set using a Split Data or Split Validation operator. To find the best model and parameters, an Optimize Parameters operator was used to test different combinations and measure their accuracy. Attribute selection was performed by observing variations in metrics when adding or removing variables. The selected models and variables were then loaded into the operators, and a cross-validation operator was used to divide the entire dataset into 10 equal parts. Each part was tested with a random 10% sample while the remaining data served as the training set. This process was repeated 10 times with replacement.

Regarding the attributes to consider, the selection process consisted of finding the model that will derive in the greatest accuracy to later add or remove variables and running the algorithms to observe the variations in the metric. This process helped determine the input variables and parameters that resulted in the best performance and improved results.

5.1 K-Nearest Neighbors

KNN is a non-parametric algorithm that classifies new data points based on their similarity to existing data points in the training set. It assigns labels to new data points by considering the labels of the K nearest neighbors. In this case, KNN is used to predict whether a customer is a promoter, or a detractor based on the similarity of their features to those of existing promoters and detractors in the training data.

The sub-figures below in Figure 9 illustrate the sup-processes flow of the KNN Process. The first operator retrieves the data from a repository entry named "4. Clean Selected Attributes" explained previously to be used in subsequent operations. Then the Date-types were excluded with the Select Attributes operator. After that the attribute "Experience" was set as the label, indicating it is the target variable to be predicted. While the "order_id" attribute is assigned the role of "id."

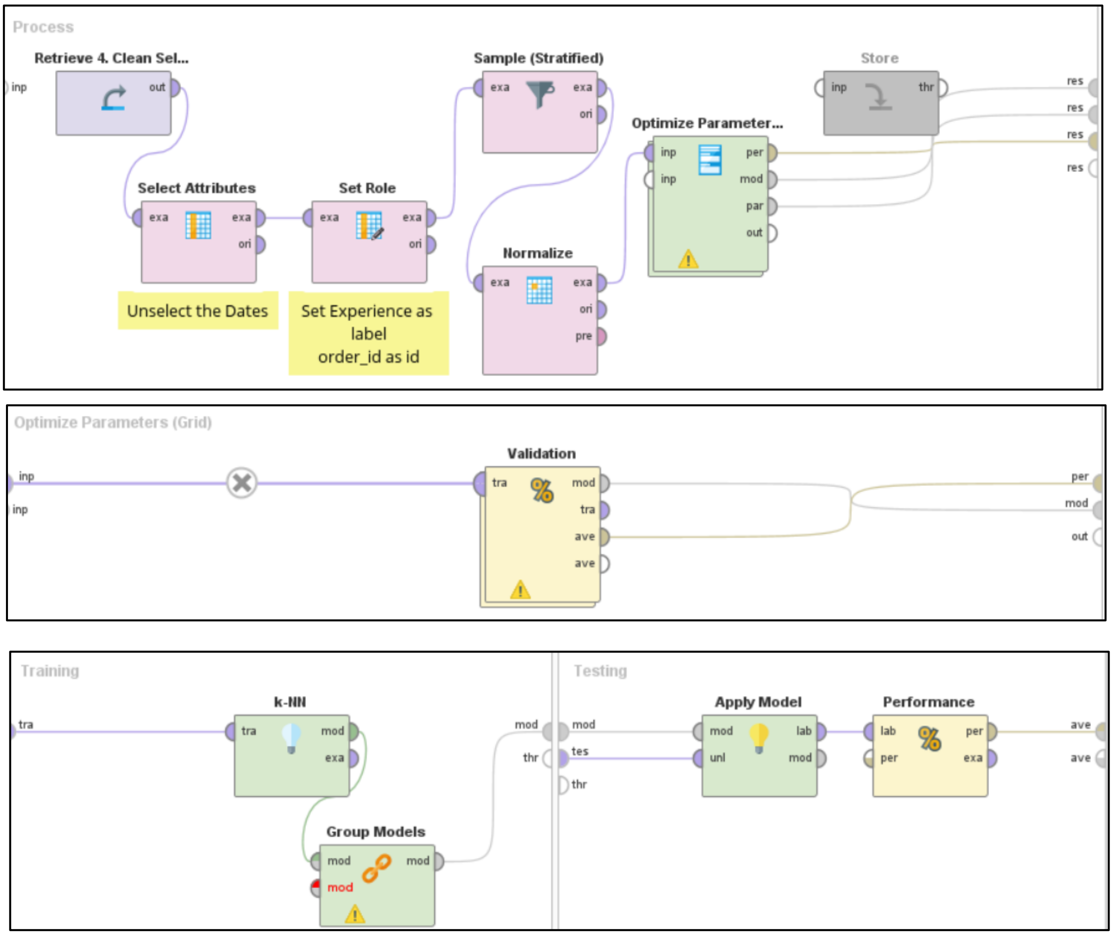
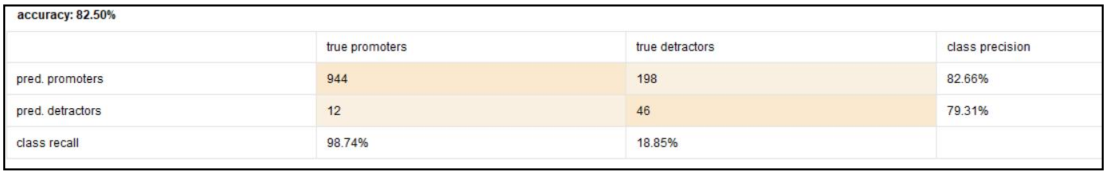


Figure 9: KNN Model Processes

Next, stratified sampling was applied to ensure that the resulting sample maintains the same proportion of promoters and detractors. The sample size is set to 6000 as mentioned previously, and a random seed of 1992 is used for reproducibility. The whole dataset was normalized to numeric attributes. Where this operator applies the Z-transformation method to scale the values between 0 and 1, making them comparable and preventing any single attribute from dominating the analysis.

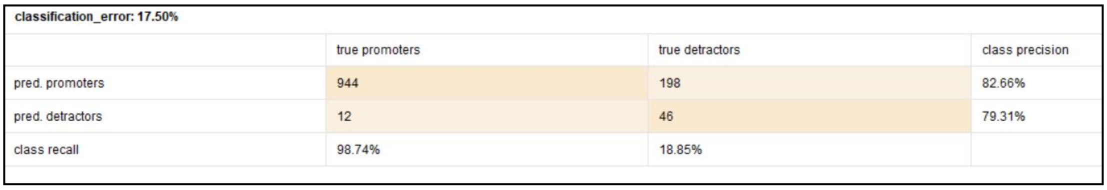
The "Optimize Parameters (Grid)" process performs parameter optimization for the k-NN (K-Nearest Neighbors) model using a grid search. It splits the data into training and testing sets, applies the k-NN algorithm to the training set to create a model, and evaluates its performance on the testing set. The performance metrics include accuracy, classification error, RSME and MSE. However, the model yielded 82.5% accuracy, 17.5% classification error and +/- 0.144 MSE. Going deeper into the main problem of this model, what we can find is that it predicts the number of promoters well, improving the starting point, now 79.7%.in 2.96 percentage points as can be seen in the class precision column in Figure 10 and 11 below.



accuracy: 82.50%

	true promoters	true detractors	class precision
pred. promoters	944	198	82.66%
pred. detractors	12	46	79.31%
class recall	98.74%	18.85%	

Figure 10: Performance (Accuracy)



classification_error: 17.50%

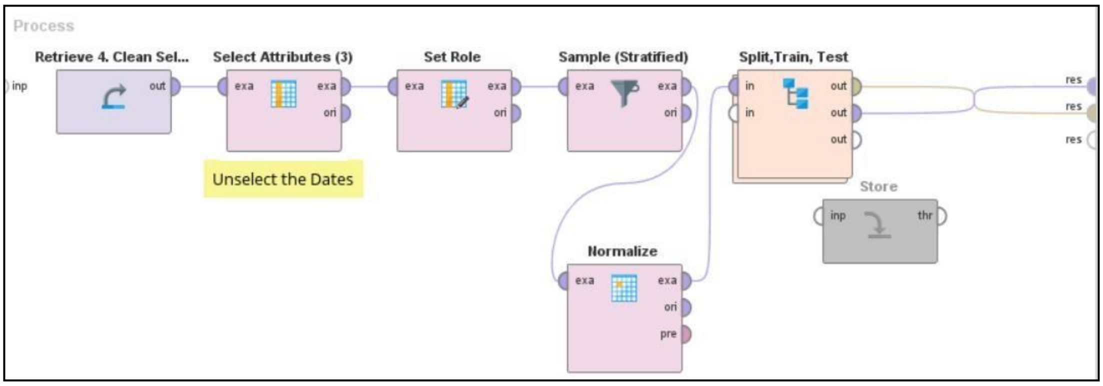
	true promoters	true detractors	class precision
pred. promoters	944	198	82.66%
pred. detractors	12	46	79.31%
class recall	98.74%	18.85%	

Figure 11: Performance (Classification Error)

5.2 Logistic Regression

Logistic Regression is a type of regression model used to predict non-numericvariables, such as customer satisfaction (promoter or detractor). Unlike linear regression, logistic regression applies a transformation function called logit to handle non-linear relationships. It allows the output to be between negative and positive infinity, providing adjusted results for analysis.

The sub-figures below in Figure 12 illustrate the sup-processes flow of the Logistic Regression Process. Like the previous model, the processes from retrieving data until normalizing is the same.



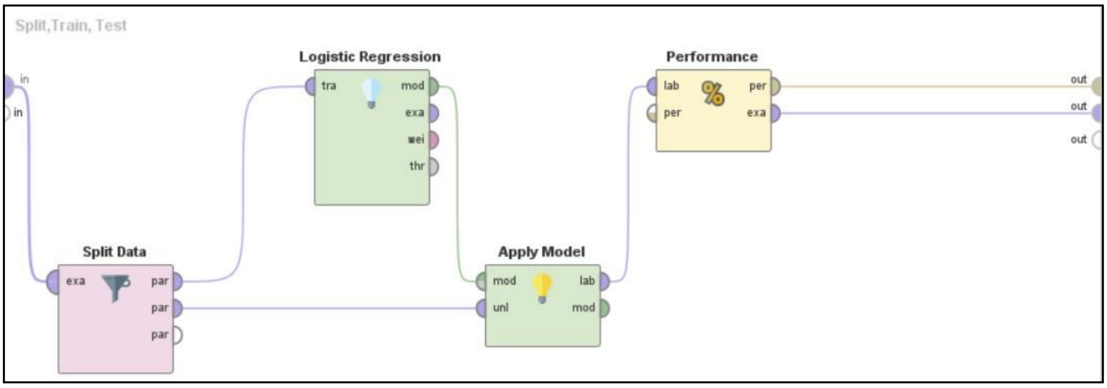
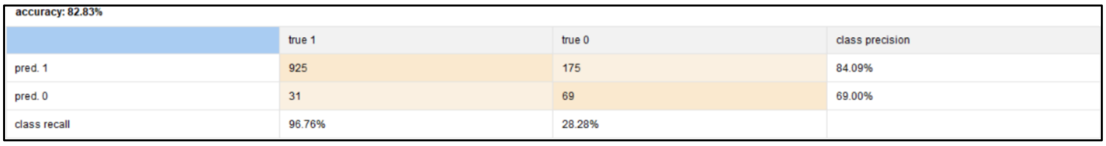


Figure 12: Logistic Regression Process

In the "Split, Train, Test" subprocess we used Split Data to split the dataset into training and testing sets. The split is performed automatically with a 80% training set and a 20% testing set. The training set is then used as input for a logistic regression model using the training set. The parameter for the model is shown in Table 3 below, after applying the trained model. The model showed an accuracy of 82.83%. Figure 13 below shows the distribution of the class precision.



accuracy: 82.83%

	true 1	true 0	class precision
pred. 1	925	175	84.09%
pred. 0	31	69	69.00%
class recall	96.76%	28.28%	

Figure 13: Performance Vector (Accuracy)

Table 4: Logistic Regression Parameters

Parameters	Worth
Solver	auto
Reproducible	√
Use regulation	√
Lambda search	√
Number of lambda	0
Early stoppings	√
Standardize	√

5.3 Decision Tree

Decision Tree is a flowchart-like model where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome or a class label. It partitions the feature space based on the feature values and assigns a label to each leaf node. Decision Trees is used to be trained to predict whether a customer is a promoter, or a detractor based on their feature values. The following Figure 14 shows the process of the Decision Tree and it’s operators.

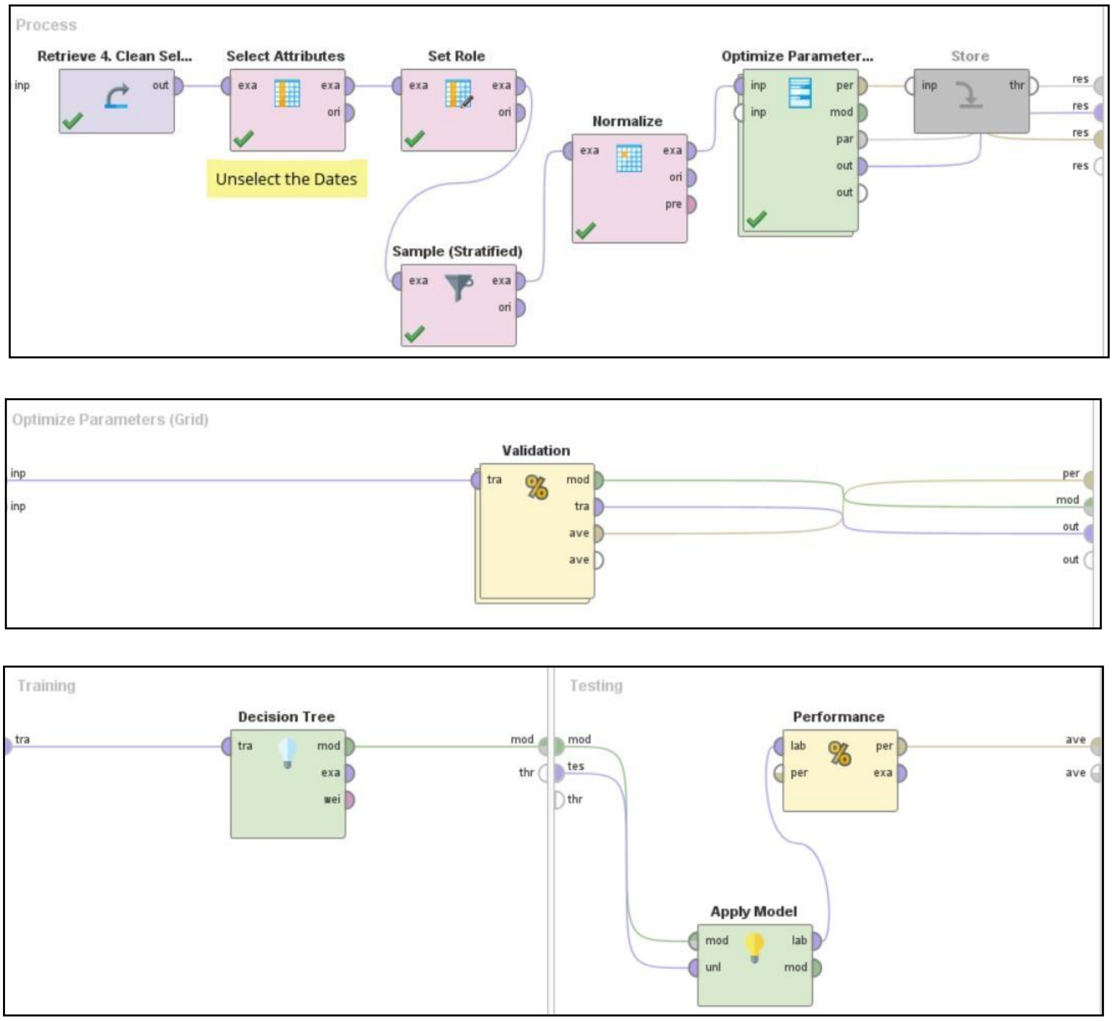
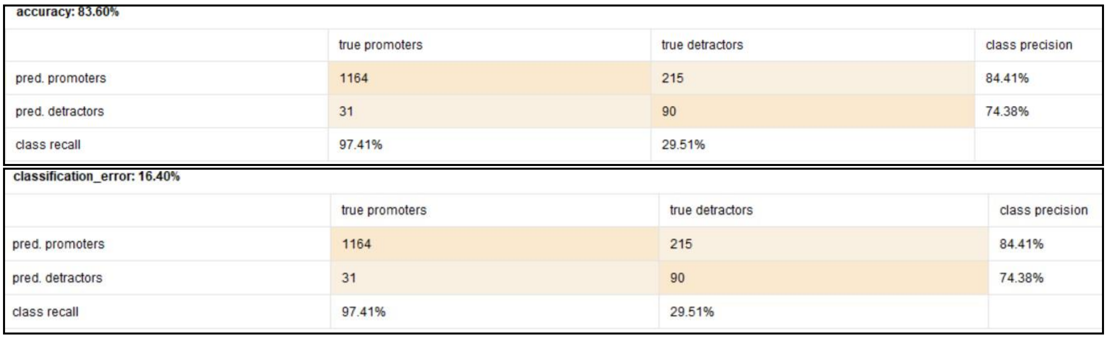


Figure 14: Decision Tree Process

This Decision Tree model achieved 83.60% accuracy and 16.40 classification error as shown in Figure 15 below.



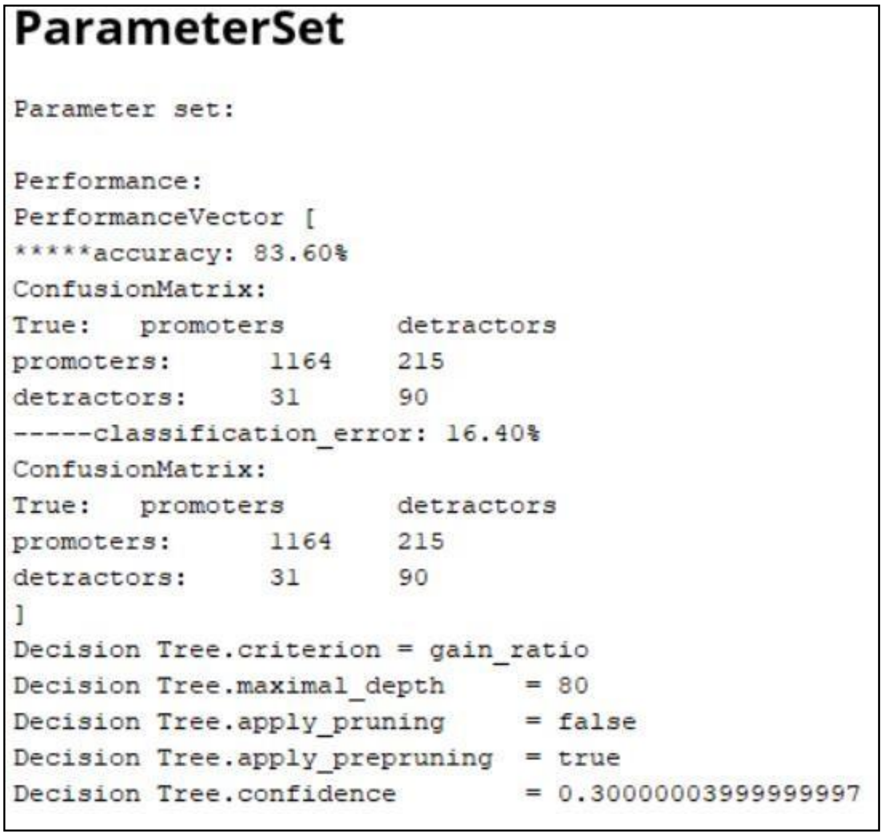
accuracy: 83.60%

	true promoters	true detractors	class precision
pred. promoters	1164	215	84.41%
pred. detractors	31	90	74.38%
class recall	97.41%	29.51%	

classification_error: 16.40%

	true promoters	true detractors	class precision
pred. promoters	1164	215	84.41%
pred. detractors	31	90	74.38%
class recall	97.41%	29.51%	

Figure 15: Performance of Decision Tree Model



```
ParameterSet

Parameter set:

Performance:
PerformanceVector [
*****accuracy: 83.60%
ConfusionMatrix:
True:   promoters       detractors
promoters:      1164    215
detractors:     31      90
-----classification_error: 16.40%
ConfusionMatrix:
True:   promoters       detractors
promoters:      1164    215
detractors:     31      90
]
Decision Tree.criterion = gain_ratio
Decision Tree.maximal_depth     = 80
Decision Tree.apply_pruning     = false
Decision Tree.apply_prepruning  = true
Decision Tree.confidence        = 0.30000003999999997
```

Figure 16: Decision Tree Parameter Set

5.4 Naïve Bayes

Naive Bayes is a probabilistic algorithm that applies Bayes' theorem with the assumption of independence between features. It calculates the probability of a new data point belonging to a specific class based on the joint probabilities of its features. Naive Bayes is used to predict whether a customer is a promoter or a detractor by estimating the probability of each class based on the customer's feature values. The model performed an accuracy of 77% and a classification error of 23%, the following Figure 17 below shows the operator processes and Figure 18 shows thetesting results.

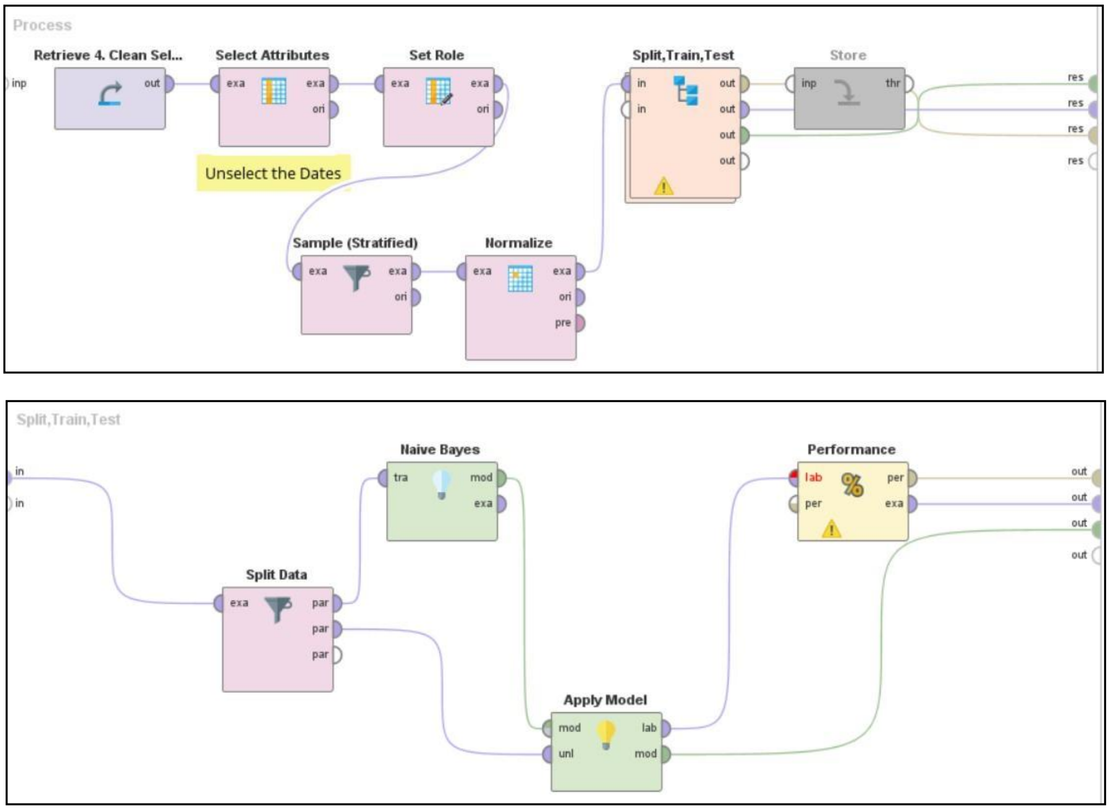
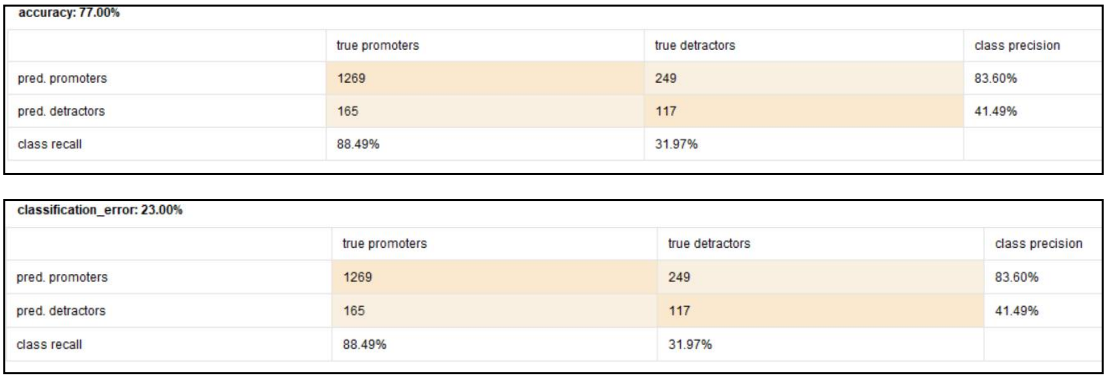


Figure 17: Naïve Bayes Process



accuracy: 77.00%

	true promoters	true detractors	class precision
pred. promoters	1269	249	83.60%
pred. detractors	165	117	41.49%
class recall	88.49%	31.97%	

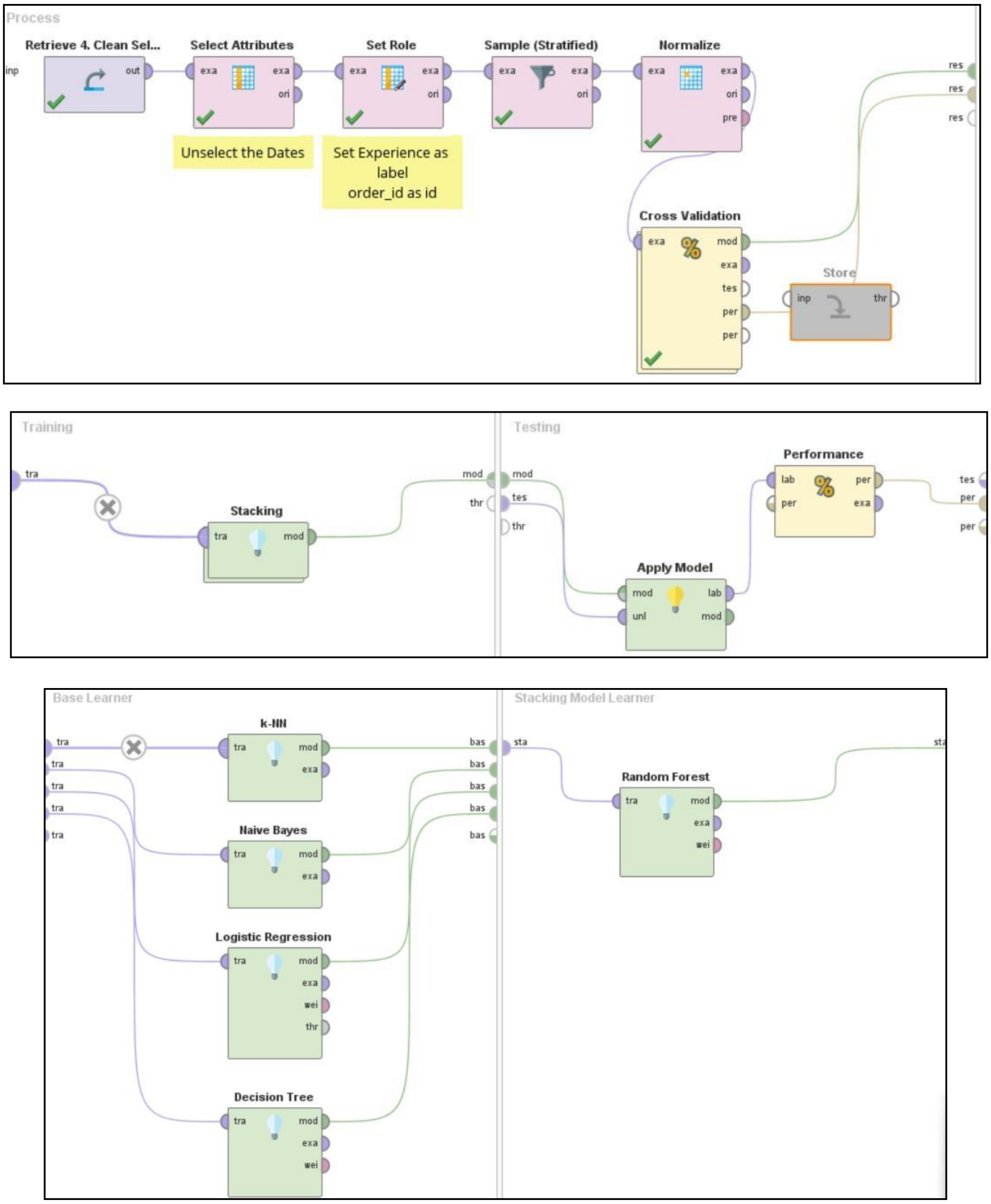
classification_error: 23.00%

	true promoters	true detractors	class precision
pred. promoters	1269	249	83.60%
pred. detractors	165	117	41.49%
class recall	88.49%	31.97%	

Figure 18: Naïve Bayes Performance

5.5 Stacking

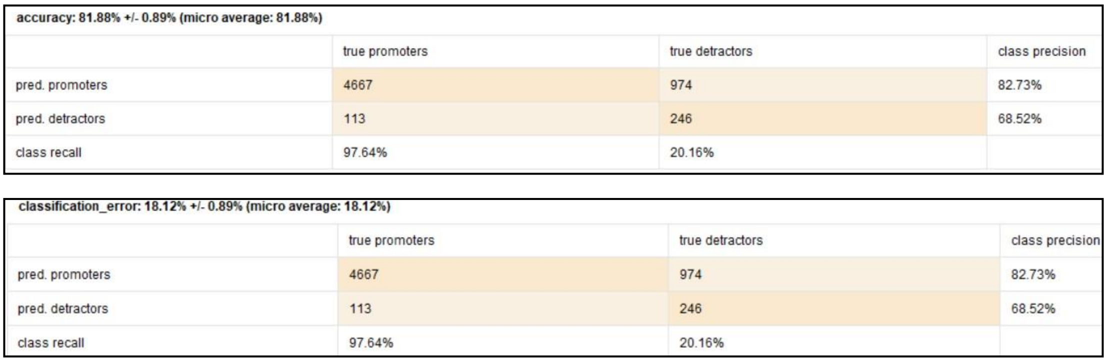
Finally, the assembly of the previously detailed models was carried out using the stacking model. “stacking" operator, which combines models to improve the final accuracy. It uses another algorithm, random forest, and achieved an accuracy of 81.88% compared to the initial value of 79.7% of promoters. The following Figures 19,20 show its process and results.



Unselect the Dates

Set Experience as label
order_id as id

Figure 19: Stacking Process



accuracy: 81.88% +/- 0.89% (micro average: 81.88%)

	true promoters	true detractors	class precision
pred. promoters	4667	974	82.73%
pred. detractors	113	246	68.52%
class recall	97.64%	20.16%	

classification_error: 18.12% +/- 0.89% (micro average: 18.12%)

	true promoters	true detractors	class precision
pred. promoters	4667	974	82.73%
pred. detractors	113	246	68.52%
class recall	97.64%	20.16%	

Figure 20: Stacking Performance

Although the overall accuracy of the model did not improve scored 81.88%, the accuracy of the detractors increased compared to Naïve Bayes and the variance also improved a little, down 27.03 %. In this way, Table 5 shows a summary of the results obtained in the models having run them individually compared to the assembled model that was finally used.

Individual Models		Stacking Models	
Model Name	Accuracy	Assemblage Model	Accuracy and Variance
KNN	82.5%	Random Forest	81.88 +/-0.89%
Logistic Regression	82.83%		
Decision Tree	83.60%		
Naive Bayes	77%		

6.0 Conclusions

At the beginning of this work, the objective of this report was stated: to predict the end consumers' experience of Olist using supervised machine learning models. Being able to use predictive models in Olist opens the possibility of applying them to other organizations and even sectors. This is thanks to the fact that the company has a complex data architecture, integrating different types of data (structured, semi-structured, and unstructured), and belongs to one of the industries with the highest economic growth of the year. Implementing this prediction in a company that is at the forefront of technology and data and in a highly dynamic market demonstrates the importance of including quantitative analysis based on data mining in the customer experience areas of all organizations.

To achieve the proposed goal, transactional user information was used as it allowed identifying the main drivers of experience, generating specific focus areas for decision-makers in companies. It should be noted that to reach these results, the decision was made to analyze the experience of those users who had received their purchase. This decision was based on the need to simplify data cleaning work and to form a single database.

The data processing resulted in joining the seven tables of the database into one dataset, with the Read CSV and Join operators with the Primary and Foreign keys. This resulted in variables of four main groups. Variables related to cost, variables related to delivery, variables related to the characteristics of the purchased product, and variables related to the date of purchase. The selection of the variables was based on deriving the greatest accuracy to later add or remove variables. Additionally, the models, parameters, and attributes to be included for the selected models were determined. While some were previously adjusted using an operator called "optimize parameters," which allows iterating on the parameters to achieve the best fit for the chosen metric. This way, the models and parameters with the highest predictive accuracy were selected: logistic regression, random forest, rule induction, and stacking.

The results of the quantitative analysis were obtained using the "stacking" operator, which combines models to improve the final accuracy. It uses another algorithm, random forest, and achieved an accuracy of 81.88% compared to the initial value of 78.59%.