SELECTED TOPIC

E-COMMERCE FRAUD ANALYSIS

For this project, we decided to analyze a pool of e-commerce transactions that take place on the Mercadolibre platform.

REASON TOPIC WAS SELECTED

MERCADO LIBRE LACKS ROBUST FRAUD DETECTION MODELS

Despite being the largest e-commerce platform in Latin America nowadays, Mercado Libre still does not have a tool that helps its vendors deal with scammers. Vendors depend entirely on whatever Mercado Pago says or does to detect suspicious activities to block orders.

DESCRIPTION OF DATA SOURCE (1/2)

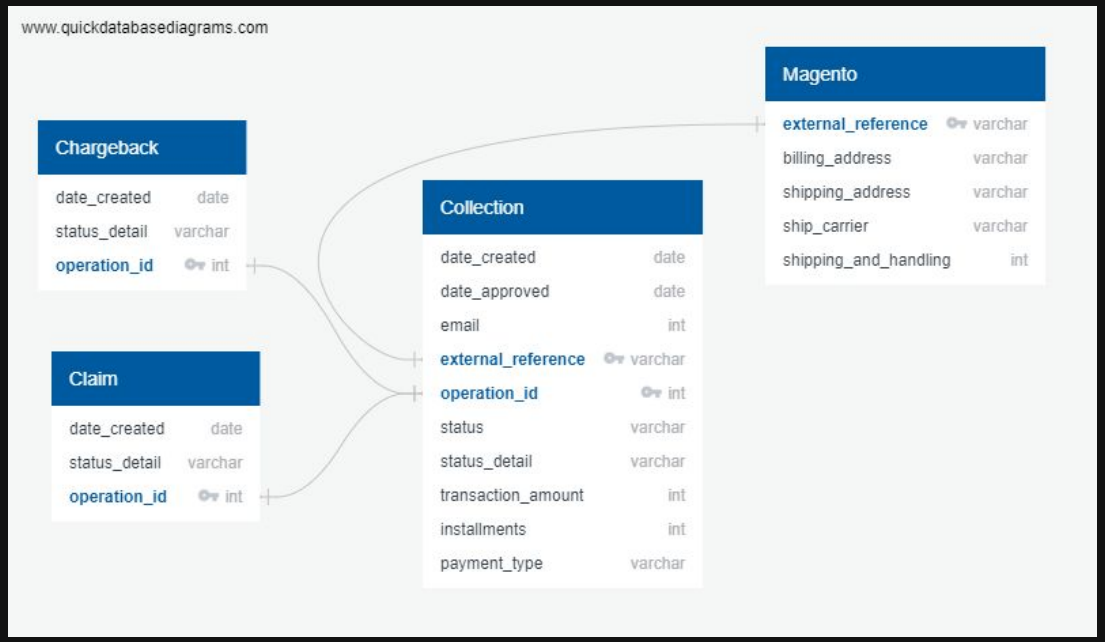
MERCADO PAGO TRANSACTIONS

Mercado Pago is the largest online payment platform in Mexico. The tool allows vendors to charge through different channels: Payment Link (Social Networks and WhatsApp), QR and Point (in person) and Mercado Pago Checkout in onlines store.

Vendors have access to data related to previous transactions of their online stores. Vendors can access this data either directly through the Mercado Shop account through a download (CSV or XML), or through an API, having a unique key for each Mercado Shop.

DESCRIPTION OF DATA SOURCE (2/2)

date_created: The date and hour when the chargeback and claim were created.
date_approved: The date and hour when the purchase was approved.
customer_ID: Consecutive numbers that correspond to a client's ID.
external_reference: Reference number of the purchases through Mercado Libre.
operation_id: Transaction reference number into Telmov.
status: Purchase status (approved, rejected, etc.)
status_detail: Operation status (claim, chargeback, etc.)
transaction_amount: Purchase total.
installments: How many periods it will pay the transaction_amount.
payment_type: Purchase payment method
billing_address: Billing address (only zipcode needed).
shipping_address: Delivery address (only zipcode needed).
ship_carrier: Delivery carrier.
shipping_and_handling: Delivery fee.
postal_code: postal code.
state: postal code state.
municipality: postal code municipality.
longitude: postal code longitude.
latitude: postal code latitude.



QUESTIONS TO ANSWER

WHAT, WHERE, HOW, WHEN

Scammers have developed strategies to circumvent the platform's fraud detection system and order products without paying for them. The objective of this project is to detect certain patterns with the use of Machine Learning that can help vendors better prepare for fraudulent transactions and prevent losses. These patterns include most common geographies, products, months, and payment methods that result in fraud.

- Is it a safe operation?
- What variables are related to the fraudulent buyer?
- What are the big patterns the purchase shows?
- What range of charges is most likely to be fraudulent?
- What is the fraudulent buyer likely to buy?
- Are fraud operations made from a specific region?

DATA EXPLORATION (2/2)

We get a year of information about purchases of cellphones, together with how many of those purchases has result as a fraud. The information was divided in three core databases:

- Collection: which is the general information of the purchases (date, client ID, amount, purchase order, etc.)
- Claim/Fraud: this database give us the information of purchases that result as a fraud/chargeback or have a claim by the customer.
- Magento: the magento database gave us information regarding the products that where pruchased in each purchase order.

We needed to cast most of the information due to many columns were declared as a string when the databases had other types like integers, datetime, floats, etc. Some information needed was inside of a very long string with not relevant information, so we need to use regular expression to get the information we needed such as: zipcode from the address, sku from the product, carrier, etc. Also some columns need to be splited for easy extraction of information.

DATA EXPLORATION (2/2)

We define new columns with the information we needed of each dataset, and at the end we merge the three different data sets in just one table called "Whole_Collection", we will use this database to create the machine learning module.

We needed to merge the "Whole_Collection" table with a table of postal codes that we obtained via SEPOMEX; the name of this second table as "CPs_Geometry". Once we get both tables, these two were uploaded to AWS to have them availables.

The merge will be used for the visualization: this will give us visibility regarding where the purchases were made and found a tendency to corroborate with the machine learning model. The merge have been performed via colab and with the help of pyspark.sql funcitons.

The last table obtained by this merge will be called "whole_collection_geom" and this will be storage in a AWS database and in a bucket to be available for the visibility, as well as in postgres for any analysis needed.

DATA ANALYSIS

We decided to try Random Forest, after all it is an improvement on how decision tree model works. The main reasons for this decision are: Random forest algorithms:

- Are robust against overfitting as all of those weak learners are trained on different pieces of the data.
- Can be used to rank the importance of input variables in a natural way.
- Can handle thousands of input variables without variable deletion.
- Are robust to outliers and nonlinear data.
- Run efficiently on large datasets.

We ran the model with 500 n_estimators which are the number of trees that were created by the algorithm. Generally, the higher number makes the predictions stronger and more stable, but can slow down the output because of the higher training time allocated. So for this test we decided to go with 500 and didn't take much time.

TECHNOLOGIES

scikit learn

HTML

APACHE Spark™

aws

git

Flask
web development,
one drop at a time

pandas

SQLAlchemy