Recommendation System for Placetopay Platform

November 14th 2020

Abstract

The company EGM Ingeniería Sin Fronteras S.A.S. (Also known as Evertec Inc.) is a provider of electronic transaction services, founded in Medellín, Colombia. The company has presence in 26 countries in Latin America and the Caribbean, with services in: merchant acquiring, payment processing and business solutions and the largest Debit Network in the Caribbean. They process approximately **2 billion transactions** per year.

Evertec is interested in using the data collected in their platform **Placetopay** to understand better its customers (merchants) and its visitors (payers) of the platform. The purpose in this document is to describe the behavior trends of the payers, of the products consumed and of the merchants using Placetopay platform. This analysis will lead to the segmentation and clusterization of that population for the posterior development of an **User Profiling System**. Finally we will design and deploy an mvp version of a **Recommendation System**, as a feature that can be offered to their direct consumers, that is, the merchants that are signed into the platform.

Introduction

During this year 2020, while the pandemic has impacted people's lifes and business, there has been an ongoing change in the consuming and purchasing habits of millions of people. While people are now less confident in going out and consuming goods and services in person, online shopping is skyrocketing, driving a fundamental change in behaviors of billions of consumers around the world. At the same time, startups all around the world have been seeding the economic landscape with new technologies and services that bridge the gap between traditional business models and newer consumer patterns. Due to this shifting of behavior, electronic payments platforms and online shopping are on the rise.

This change has been abrupt, and all of a sudden there is a huge amount of information that is produced, consumed or stored. In some cases this business are able to capture a lot of information from their interactions with customers, visitors and users, but they don't have an immediate use for that amount of information, or the companies do not have the capability to process and extract useful insights and then to inform and drive decision making for the business process. In some way, the nature of the way information is collected and used for business is also shifting in what has been called the Fourth Economic Revolution.

Business Problem

The company is interested in developing visual dashboards. this information can be used by different business units: It allows making decisions, developing and positioning products more effectively, understanding how to communicate more effectively with users and audiences, directing the sales department, keeping engaged and focused on the team, and above all, it allows to humanize the consumer by granularly understanding their behavior, needs and objectives and thus being able to offer personalized satisfaction. It is the first stage of user experience development.

By having a characterization and a recommendation system (a table with recommendations of purchase to be offered for every payer every time that the payer enters the platform), the company is going to have a competitive key feature on the international Payment Services Provider (PSP) field, being able to offer this recommendation system to the merchants and then gain a better positioning on the PSP market in Colombia. Also, the company will be able to show that they have top tier services that are at the level of the services offered by other international competitors.

The company is using the services of data capture and analytics from Qlik. The company is developing dashboards over that platform for showing metrics like RFM (recency, frequency, money) that analytically are very simple. Qlick provides them with the analytics tools but is not able to integrate, compute and consume that analytics in real time. Thus the need of having a more advanced analytics capability in order to deal with the future economic threats related to the COVID-19 pandemic.

Problem statement for this project

Considering the need of understanding better the behavior of users of placetopay, and the available data the company is able to share with us, we proposed a set of research questions to be answered with our analysis, around a central question:

How is the dynamic of use of the platform by merchants and payers?

Specific questions:

1. How does the number of purchases per payer behave over time? How frequently?
2. How many transactions have a particular merchant? By how much?

Other possible questions:

3. How many different payers have bought from a merchant? What is the purchase value per payer?
4. In how many different merchants a payer has bought? In how many categories?

As there are many payers (2.9 million) and merchants (500) and there are 12 million transactions with 47 attributes, we noticed that we needed a tool to be able to identify the trends and features of the dataset. For that reason we proposed (and needed) the implementation of a

dashboard that could mimic what the company had, and handle the amount of information available in order to answer these questions.

As a second stage, we have the designing and proof of concept of a **Recommender System**, that would be able to infer attributes from the transactional non-structured information and make a viable recommendation. It doesn't have to be perfectly accurate, since we are intending to provide suggestions for marketing purposes.

These recommendations are based on the characterization profile of the payer (**User Profiles**) based on transactional Information captured of each transaction and dynamics of transactions made by this payer. There are various approaches and they could require a clusterization or market segmentation.

Limitations

Due to privacy limitations, the company is able to provide us only with data that is anonymized. This means that payer's attributes are hashed in a way that does not allow to extract text or personal information. For the same reason, the company only provided us with a file, but is unable to give us access to a database service or an API.

Another point we have identified, is that the majority of the users of the platform are single time payers. On one hand, there is little information available describing an user, and on the other hand, there are very few interactions for the majority of the users.

The actual dataset has passed through an internal process inside the company to be put available for us. The process of acquiring a new dataset can take some days due to that chain of custody, so it is difficult to provide a solution in real time without integrating with the company infrastructure division. For those reasons, we are not intending to implement a production level, real-time application.

As it is difficult to have information about a payer that prove that this payer does indeed have some inferred attribute. Instead what we are implementing is a descriptive analytics dashboard to better understand their business and a proof of concept of the implementation of a recommender system. Also, we will explain the reasoning behind a user profiling with acceptable explainability and in a way that can be explainable and interpretable.

Section 1 - Data Wrangling & Cleaning:

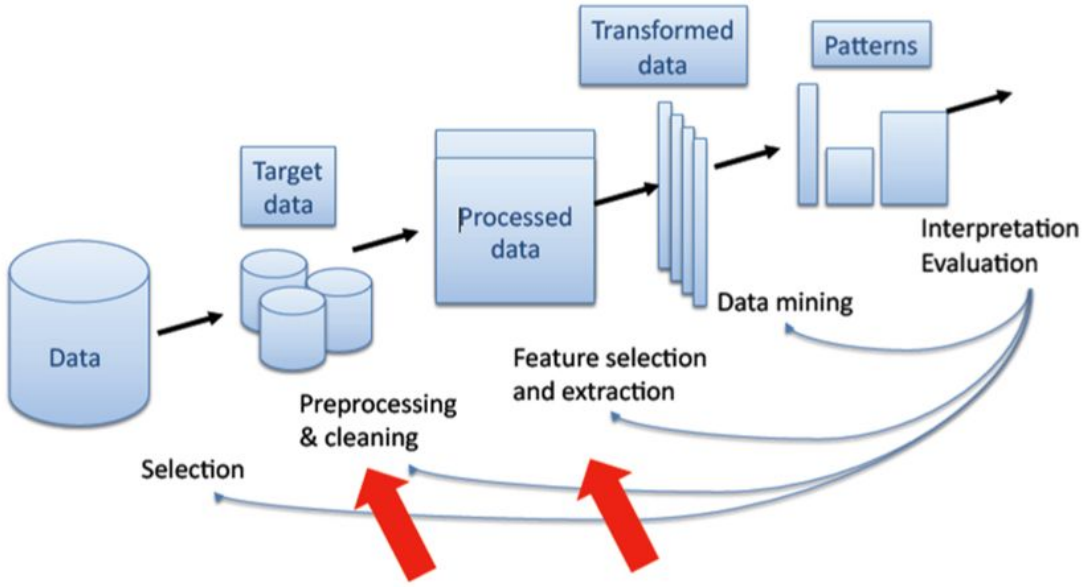


Figure 1. Iterative Processing of the Data

Data Selection

The company Evertec provided us with a dataset of purchase transactions made during the first 9 months of 2020. As this information is very specific of the company and is protected by privacy concerns. we didn't use any other dataset for comparison. The company wanted us to include demographic information from DANE, but it was unfeasible due to the fact that all sensible information was hashed.

The first step was to convert the file to manageable format. We mainly use comma separated values (CSV), Feather files, pickle files, HDF5 files and Numpy Array Dump files. In order to load all the dataset it was mandatory to use batch processing, in order to not consume and overwhelm the RAM memory of the machines we were working on locally and in the cloud.

We worked with a combination of local machines, Google Colab, AWS services for virtual EC2 ubuntu linux machines, S3 buckets, RDS databases. The processes we made with the data are summarized here. All those tasks are documented in full detail and extension in the code and notebooks section.

We loaded the database provided and studied in detail in order to check which of the columns and rows were relevant and have useful and accurate information. From the total 47 columns provided, we figured out that 15 of them were not so useful because the information were redundant (Repeated or Highly correlated columns, as detailed descriptions of error codes or numeric identifiers of other columns), irrelevant (as error codes for transactions declined),

incongruent, or unreliable due to inaccuracy or uncertainty. Also, we removed columns that were not providing useful information for the characterization and profiling of the payers and merchants.

At this stage we figured out that the Merchants are using the platform in non-standard ways. That means that the use of a field for input some specific information could have ambiguous meanings for different merchants. For that reason, there were information like boroughs or neighborhoods coded as cities or regions. Also, the credit cards have many values that were synonyms, or there is information related with the platform operation coded as different types of credit cards.

This is a serious problem because we cannot say with certainty that the way we interpret the data is correct. We talked about this issue with Evertec and they told us that tracking each interpretation to each merchant is very difficult, so we proceed with dropping that unreliable information or defining filters and transformations to make it useful.

The details of each variable we use or discard and which treatment was applied is described in the document 05_Dataset Dictionary.

Preprocessing and Cleaning:

The first processing of the data is selecting the right type and right unit for every column and checking for consistency. Also, as the majority of the columns were of type categorical, it was useful to convert all that columns to categorical type and stored in numpy arrays or feather files for using a lot less memory. In order to check for accuracy it was mandatory to correct the typographic errors, misspelling, upper/lowercase, spaces of some of the columns too.

At this stage it was possible to check for outliers. Here we found a series of transaction_payer_id that had like 10% of the transactions and were sometimes using hundreds of card_ids. This led us to determine that the database was mixing transactions made by company aggregators (travel Agencies, for example) with transactions made by a single payer. This is another serious problem, because it is very difficult to know if a particular payer id is indeed a single person. For this reason we decided to drop all the transactions made by payers that had more than a certain number of credit cards; those were the outliers over the 99.8% of the number of credit cards.

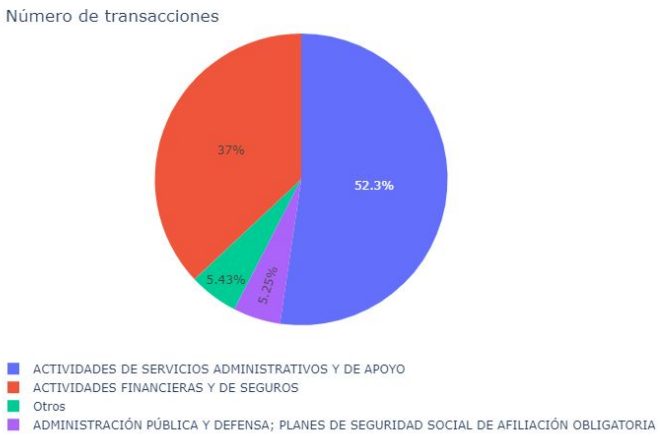


Figure 2. Distribution of transactions by Merchant Category before cleaning

In the same line, we cleaned all the transaction_payer_document_type and filtered all the transactions that were made by an identifier that is common identifying a company. (Tax-ID, NIT, RUT, CPJ, etc.) suspecting that those transactions were made by enterprises or companies.

transaction_payer_document_type	transaction_id	transaction_processing_amount
CC	87.782455	64.441476
CE	0.738114	0.657240
CIP	0.007787	0.010563
CPF	0.002843	0.004271
CPJ	0.000097	0.000150
LIC	0.011998	0.011751
NIT	7.102612	32.830886
Otro	0.159751	0.180948
PP	0.387632	0.547584
SSN	0.010709	0.017165
TAX	0.008431	0.008042
TI	0.778831	0.537545

Table 1. Percentages of Number of transactions and Aggregated amount of transaction made by a payer with a particular Document Type.

Also, there were very rare transactions that were made for extraordinary amounts (like hundreds of millions of pesos) . We suspected that these transactions were not made by a single person, but a company. For that reason we decided to drop the outliers, that is the transactions that had values over the 99.8% of the distribution (over 15 million pesos).

Another cleaning that was necessary was all the transactions that were made by the placetopay platform itself. We figured out that *test* transactions mixed with *production* transactions. We were told that those transactions, cards, and banks were marked as “test” and “Pruebas” and we dropped that. Nevertheless, we are not completely sure that there are not some more test transactions for 0 value or with a different description still in the dataset.

Finally, There were transactions where it was impossible to identify individually a `payer_id`. Some of these transactions were made in cash, and some more were failed transactions. We proceed to not consider these transactions.

Feature Selection, Extraction and Data Transformation

This stage is about making data more complete by adding related information. This included value conversions and translation functions, and some normalizing of the numeric values to conform to minimum and maximum values. Here we found that there were many transactions made for little amounts, less than a dollar, as micropayments. Also, we found that some of the dates were encoded as year-date-month, and it was impossible to know wich was the correct date when both month and day were less than 12.

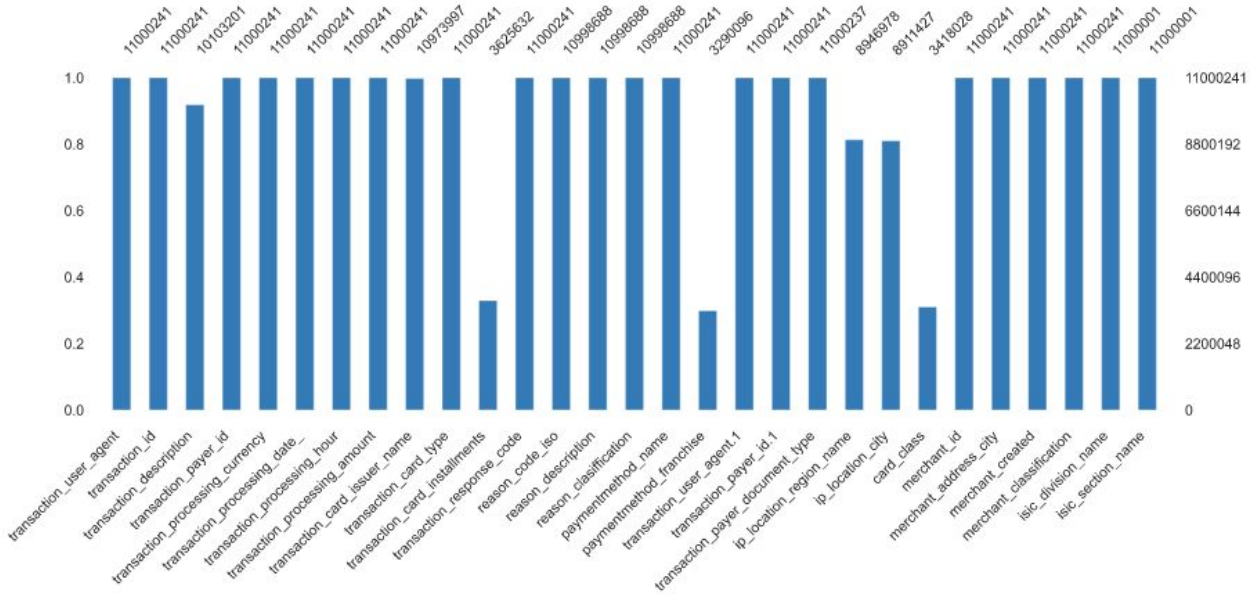


Figure 3. Missingness pattern.

Finally, there are some variables in the dataset that have too many missing values. But due to this variables being categorical, and the nature of each one of the variables, there is no possibility to assign a value like 0 or a label without altering the dataset:

`transaction_description` has 897040 (8.2%) missing values
`ip_location_region_name` has 2053263 (18.7%) missing values
`ip_location_city` has 2088814 (19.0%) missing values

This card information is only recorded when payment is made with creditcard:

`transaction_card_installments` has 7374609 (67.0%) missing values
`paymentmethod_franchise` has 7710145 (70.1%) missing values
`card_class` has 7582213 (68.9%) missing values

Section 2 - Descriptive Analysis:

At first what we found is that the dataset is mainly composed of categorical variables, but the majority have high cardinality. For example, there are 90927 distinct values for **`transaction_user_agent`**, the variable that describes the type of device and operating sýstem is used when accessing the platform.

One of the reasons for this is the high variability of user profiles and the need of the platform to interface with other platforms. But the other reason is that the merchants and in the end, the final operators of the devices (POS) or the payers (over a website or app) have different ways of code the same information, or a merchant has a totally different scheme to use a field in the interface with respect to other merchants.

Figure 4. Most recurrent merchant categories (Section) in the platform.

The next exploration was understanding what are the types of transactions that are in the platform in order to understand the needs and the goals of the payers of the platform.

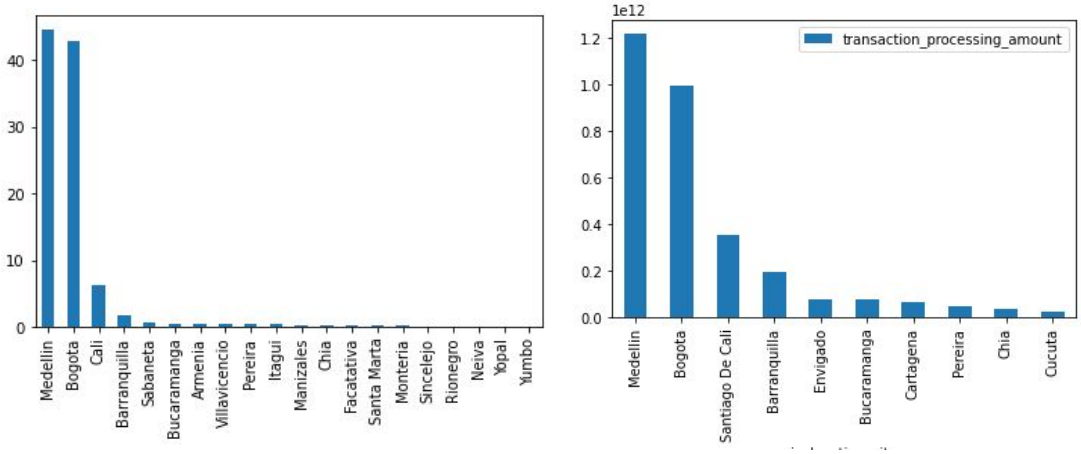


Fig. 5. Nominal comparison: Location of origin of transactions vs Merchant address city.

In figures 5 we see another interesting trend: most of the merchants are headquartered in Medellín, Bogotá or Cali, and the majority of their payers are from the same locations. There also a number of merchants that are located in small cities.

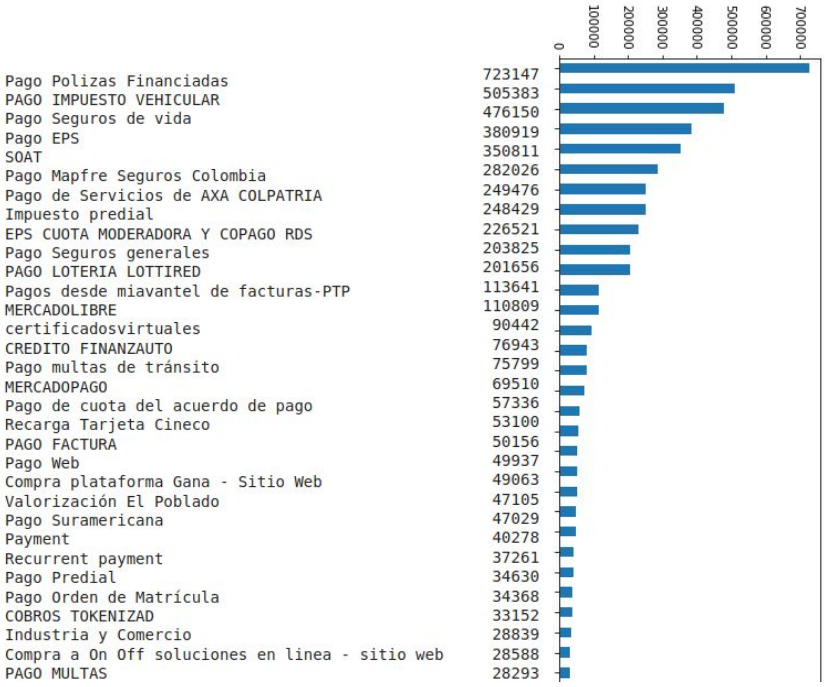


Figure 6. Most recurrent transaction descriptions in the platform.

By looking at the merchant categories and the descriptions of the transactions that are more common we can see that most of the use of the platform is to pay Social Security, Insurance, taxes, monthly service bills and other obligations. This will be a trend explored more in detail in next steps and has an important insight: Maybe the payers are not driven to placetopay platform due to a desire to buy something, but due to an obligation. This is something that has to be understood and studied further, in case it should be possible to separate both behaviors.

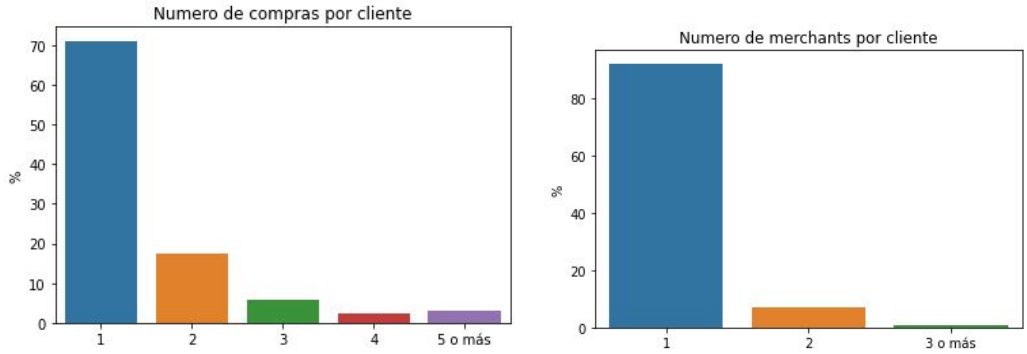


Figure 7. Comparison of the number of purchases and number of merchants by payer.

We can see that the majority of merchants have a very low number of transactions and consequently, very low amount of transactions.

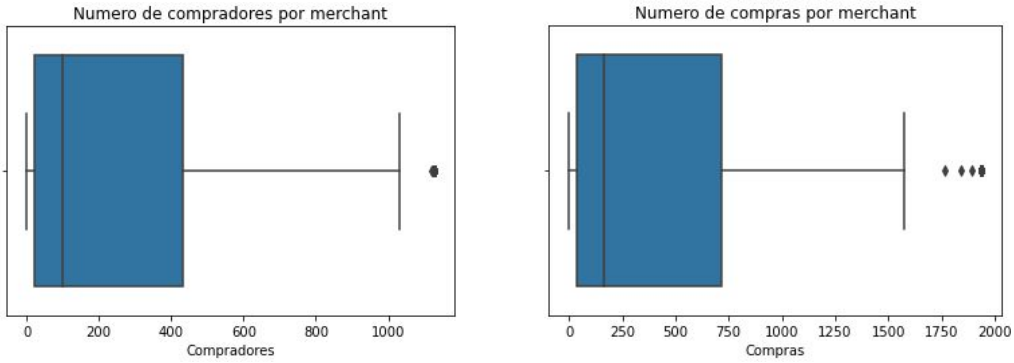


Figure 8. Comparison of the number of payers and number of purchases by merchant.

Analizing these plots we concluded that the dataset has high sparsity: The vast majority of the payers buy in the same merchant and buy once. This has serious implications for the purpose of understanding the payer: There are payers that come to the platform to pay obligations monthly, others that use the platform once and never return, and highly repetitive payers.

And the number of payers by merchant and number of purchases is relatively low, having in mind that there are 500 different merchants. There are some merchants that concentrate the vast majority of payers and also, the purchase amount.

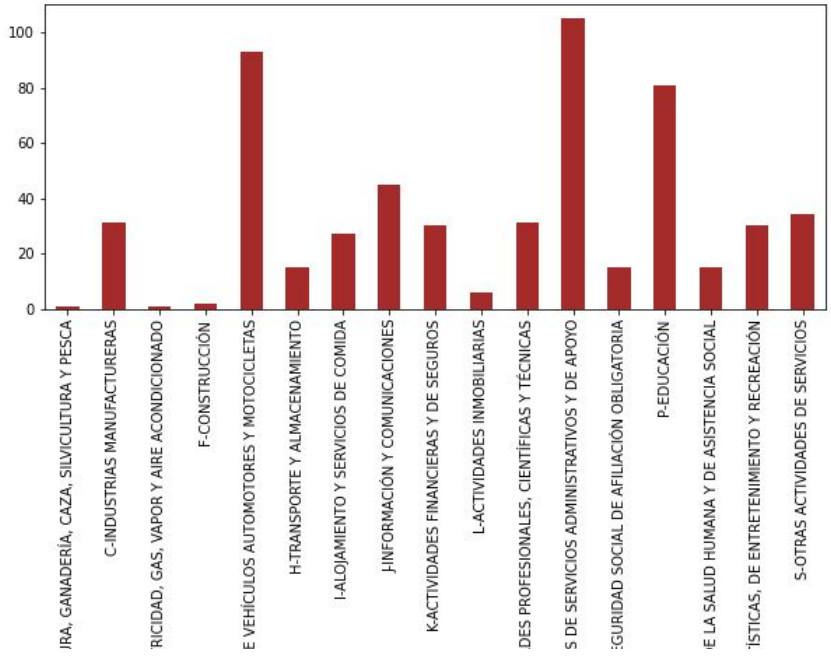


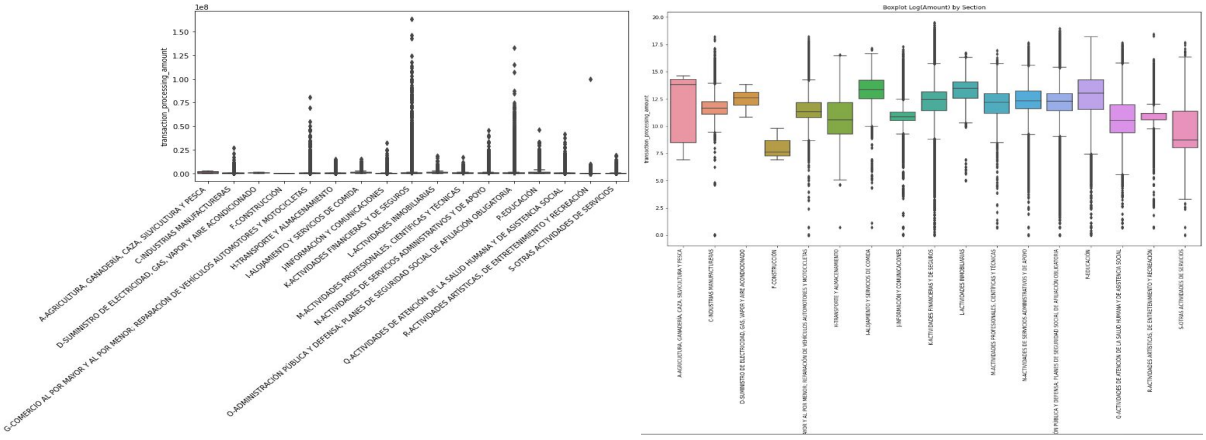
Figure 9. Number of merchants in each category (section)

While most of the categories have 10-100 merchants, the majority of subcategories have only one merchant. Thus, we can say that the section and division variables can be used as response variables for a modeling. But there is a problem: most of the payers buy in one single merchant, and in consequence, in one single category.

So, how feasible is to model to understand payer behavior? In order to manage the extreme skewness of the distributions we use a variable transformation:

`transaction_processing_amount` is the variable that describes the amount of a transaction. This variable has a distribution that is highly skewed (γ1 = 1485.079818) and for that reason, the plots made with this variable are less informative. One way to handle this situation is to make a transformation of variables using logarithms. After the variable transformation now we can see things more clear.

In the next figure we can see in detail how the distributions are so different going granular to the merchant subcategories (Divisions) some are skewed, some others don't. Some have higher variance, some other don't.:



Figures 10. Comparison of Boxplots of transaction amount by Merchant category before and after logarithmic transformation.

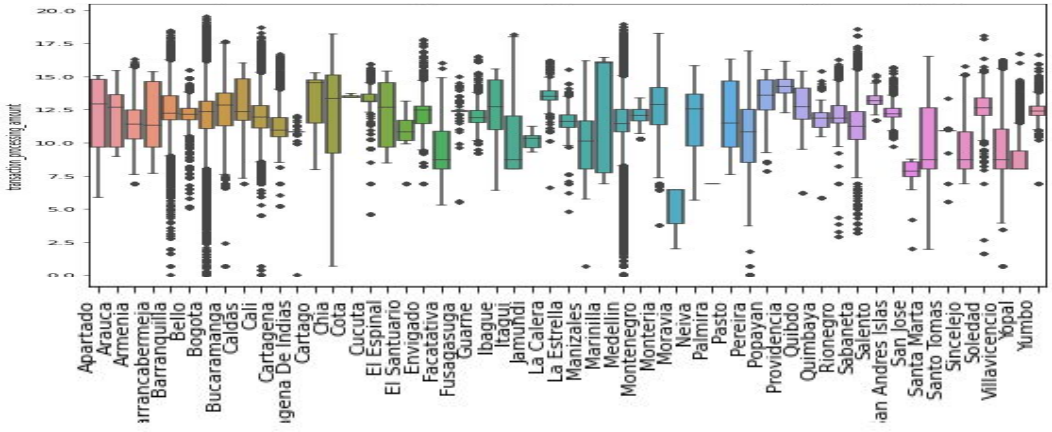


Fig. 11. Distribution of transaction amounts (Log Scale) in different cities.

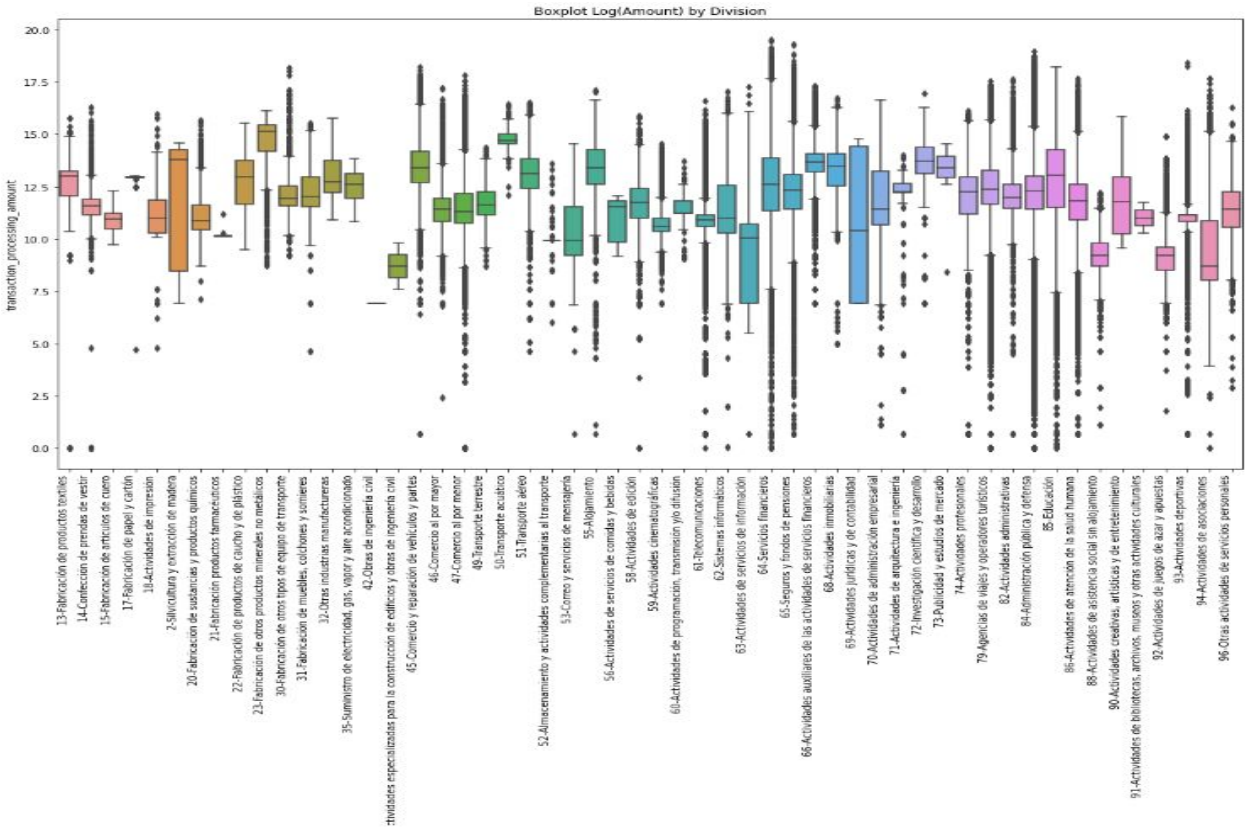


Figure 12. Distribution of transaction amount over each merchant subcategory (Division).

These graphs show that we have a mixture of very different distributions of purchase amount across all the dataset. In the next figure there is a case of the opposite behavior: by comparing the distributions of transaction amounts for each merchant classification (Figure 15) we can see that the difference is not that much. This means that these 6 types of transactions are pretty much comparable between them.

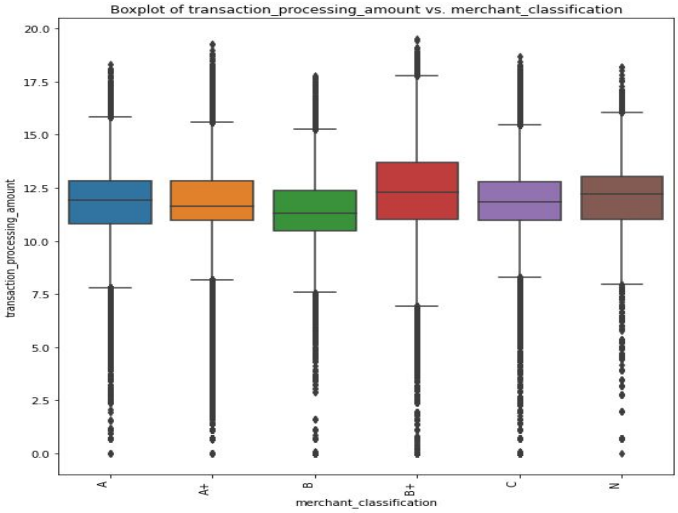
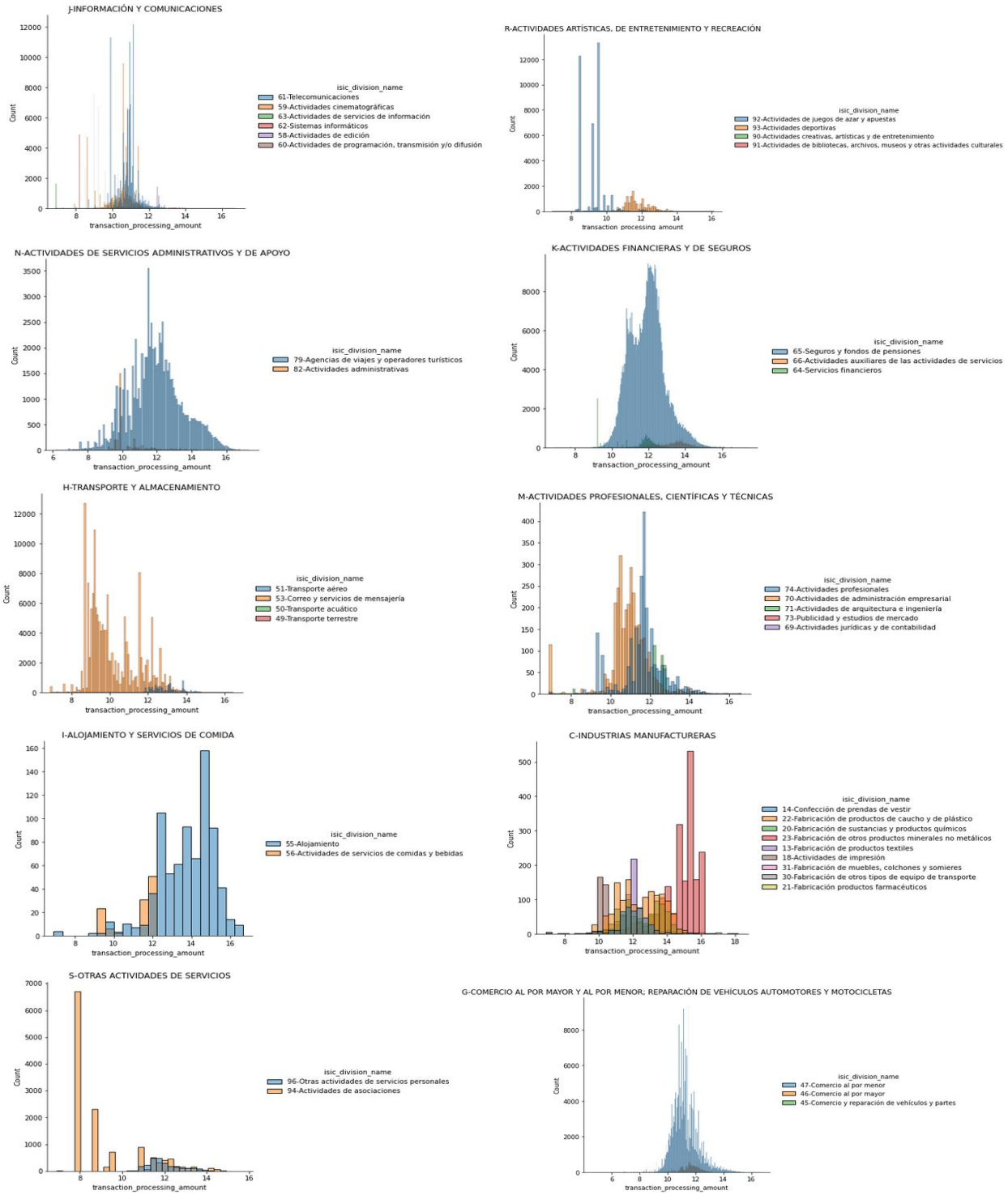


Figure 13. Transaction amount (Log scale) vs Merchant Classification

Here, we can see how the frequency distributions for the transaction amounts vary across all the subcategories, and compared within each category:



Figures 14. Comparing the frequency distributions of each merchant subcategory(isic_division) with the other subcategories in the same category (isic_section). We expected some normality.

From these plots it is possible to see that the distribution of Transaction Amount is in fact *a series of different distributions combined*. This was expected, as different categories of merchants could have very different sets of services or products, and for the same reason, different types of payers.

Time series decomposition

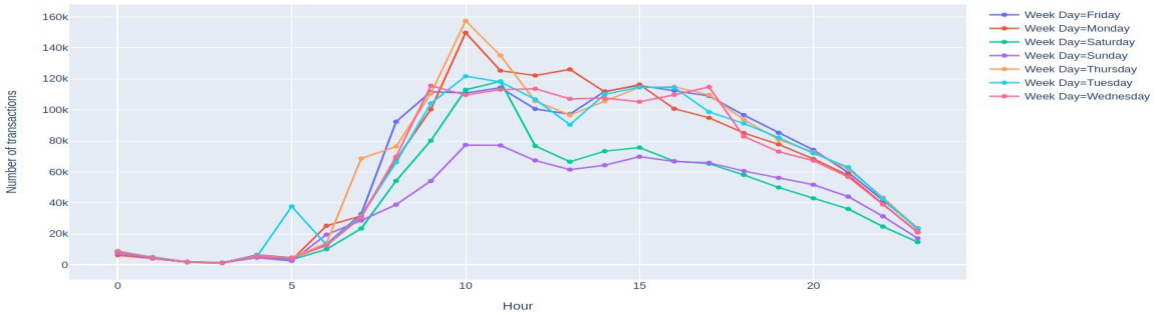


Figure 15. Hour of the day of the transactions, by day of the week.

Another interesting trend is looking at the time of the day people use for doing transactions in the platform (figure 16). Here we see that most of the weekdays have a similar behavior, except for saturday and sunday. We made the same plot with pretty much any other variable and we concluded that the trend is somewhat the same. In this case, as in some other variables, we can conclude that the hour of the day or the day of the week are not determinant in the behavior of the payer.

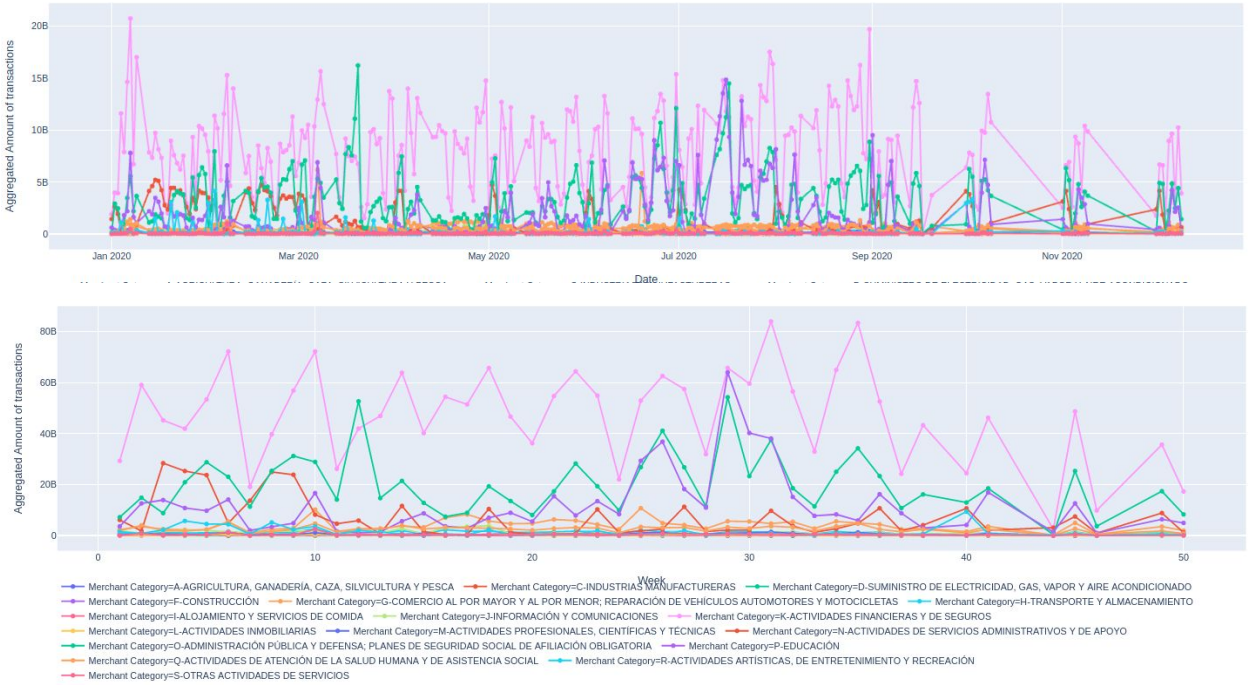


Fig. 16. Daily and Weekly trends by merchant category (isic_section)

In this visualization we can see that the categories where there are the most transactions show a monthly behavior pattern: there is a spike of transactions every 4 weeks, nore or less. This is very interesting behavior that can be used to differentiate at least three main types of transactions, (and extending this categorization to payers or merchants):

- Transactions that occur seldom, rarely, or only once.
  This transactions should have a distribution that is memoryless.
- Transactions that are clustered around the same date, or the same month
- Transactions that occur at a regular interval, like monthly or weekly
  This transactions are dependent on past transactions.

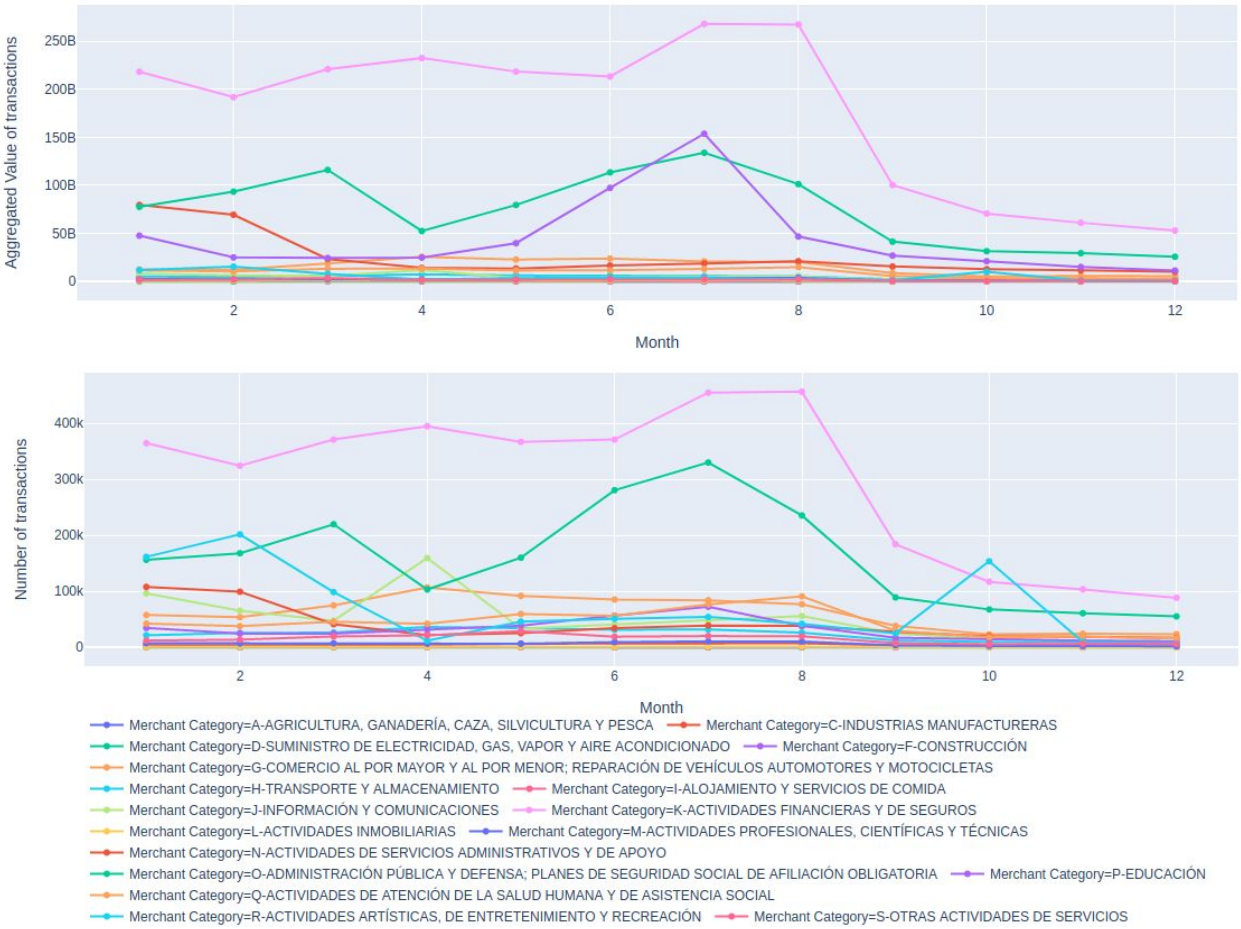


Figure 17. Time series of transaction number and purchase amount for each merchant category

We can see from this plots that in some Merchant categories the amount of transactions is higher and the number of transactions is low. Like in the education sector (purple line), where there is also a visible spike of payments at the beginning of the semester.

In other sectors like Motor repairs (orange line) and Public Administration and Social Security contributions (green) the trend is the opposite: there are a constant amount of contribution of low value. But within the public administration are the payments of taxes, at the middle of the year.

There is also the case for the Entertainment industry (aquamarine line) that has a spike of sales in september and october. Another interesting trend is the sales of technology (light green), that spiked in the beginning of the pandemic.

We made some text analysis in order to identify how similar were the transactions from different merchants, or from different business sectors. There is a lot of personal and detailed information about the payers that can be found here: debts, health issues, products purchased, education level, hobbies and ownership of cars or properties. We can indeed see the trends for each of these.

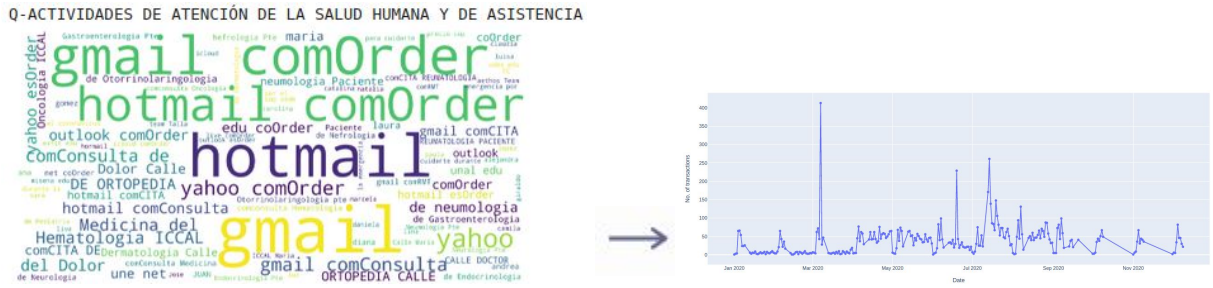


Fig 18. Text analysis of descriptions converted into temporal trends.

At this point there is a decision that can be made: working only with the merchant subcategories that represent products and services that are not obligations but products and services that the purchase decision can be influenced by a marketing campaign. Also, we can decide to work only with the most profitable categories.

Feature Engineering

While Data cleaning has a great impact, Feature Engineering is perhaps the most transformative part of the process. Here, domain expertise is required to know what to look for in the dataset, how to interpret and what data is missing. We engineer the following attributes calculated from the dataset:

For each Payer_id:	Number of purchases
	Average and Total amount spent
	Recency (Number of days since last purchase)
	Frequency (Number of days between purchases)
	Filtered the device used (Android, iphone, pc etc.)
For each merchant:	Age
	Average and Total transaction purchase
	Customer Retention rate for each month
	Customer retention classification

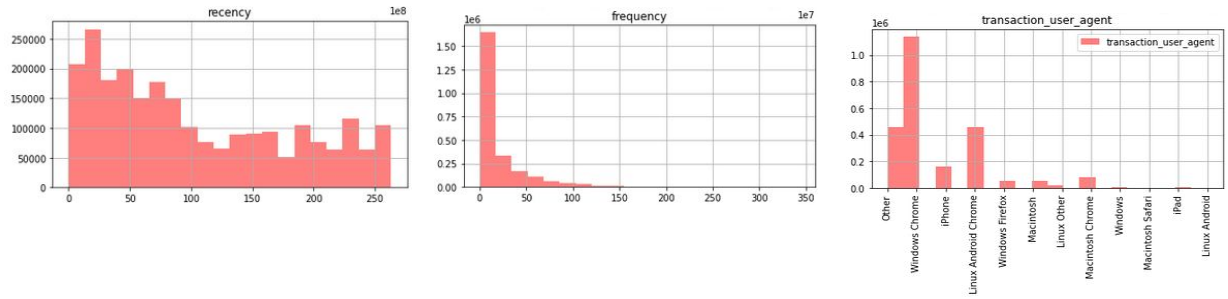


Fig 19 to 21. Distributions of Recency of payers, Avg time between purchases, type of device.

We see that payers have recencies that vary from 1 to 250 days, and when they have more than one purchase, they tend to be spaced by some days. Also, we can see which technologies are more used by the payers (we can see this in total detail in the dashboard)

Also, we have same trend of sparsity: There are payers that are very frequent and spend too much, very good customers, and there is also the vast majority of payers who never return after using the platform once:

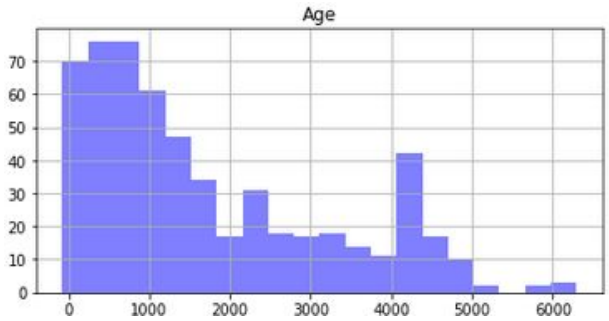


Fig 22. Distribution of Age of merchants (days).

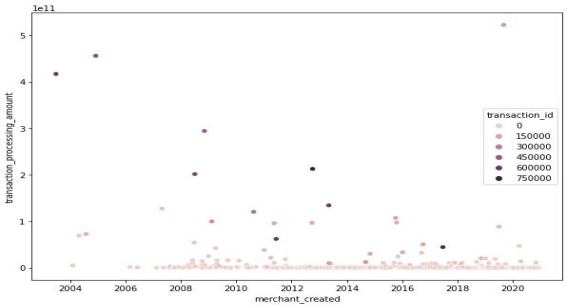
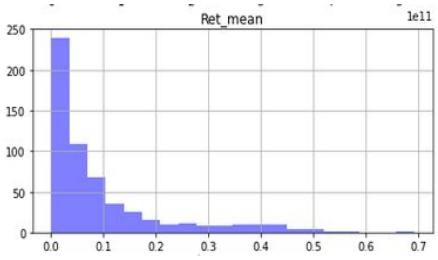
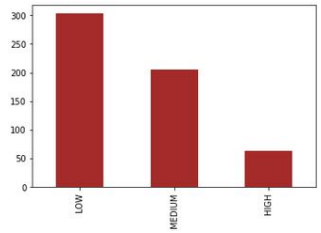


Fig.23. Transaction amount vs Merchant Age

As majority of merchants are less than 3 years old within the platform, most older merchants have a higher volume of transactions. This should be expected due to the nature of the business of Placetopay: with time, the merchants with consolidated market have higher and higher number of payers.

And we can see also that some of the most lucrative merchants (for placetopay platform) are into Finance, Insurance and Social Security, followed by Education, Entertainment and Administrative services





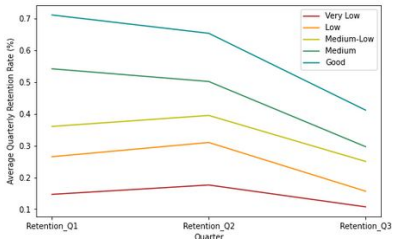


Fig 24 and 25. Distribution of Customer Retention Rate Fig. 26. Quarterly Customer retention rate

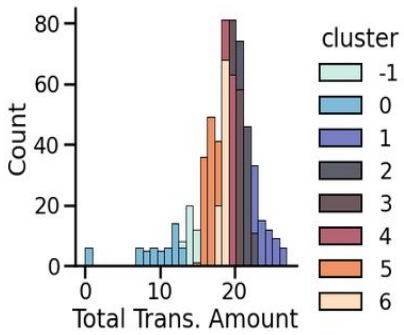
Customer retention rate is calculated as

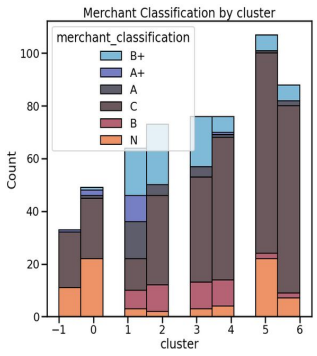
No of repetitive payers at end of period /Number of payers at start period

Calculating it quarterly, we found that 303 merchants have a low mean monthly retention rate (<10%), 205 merchants have a mean quarterly retention rate under 40% and only 63 merchants have an acceptable retention rate of more than 40% each trimester. Also, we see that the retention rate is getting lower.

Clustering

Using the retention rate, Age, Number of transactions, mean transaction amount, category, classification, section and address as classification variables, we fit a classification model using HDBSCAN algorithm, a clustering algorithm that extends DBSCAN by converting it into a hierarchical clustering algorithm, and then using a technique to extract a flat clustering based on the stability of clusters.The data we're talking about is multi-dimensional, and it's not easy to perform classification or clustering on a multi-dimensional dataset. Hence, to help with that, a Dimensionality Reduction technique, UMAP, was performed – these reduce the dimensionality of the dataset without losing out on any valuable information from your data.





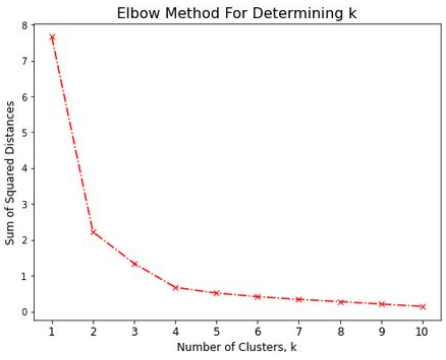


Fig. 27 and 28. Comparing numerical clusterization with placetopay categories.
Fig 29. Optimal Number of clusters using categorical variables is 2.

We managed to classify the merchants into 6 clusters with a good measure of similarity (silhouette coefficient 0.70) that reflected different populations of merchants. As HDBSCAN takes into account numerical variables, The clusters were separated mainly by the transaction amount.

So we make another clusterization using the K-Modes algorithm, suited to categorical variables. And using the elbow method, we found that we can differentiate two types of merchants (and its customers): Those that have **High number of transactions and have payments made with debit card and PCE,** and those who don't.

So, in order to advance towards the modeling and the recommender system is useful to know which of the variables does not have any influence on the behavior of the payers (like the hour or the credit card) and also to know which variables are correlated in some way. We have to minimize the risk of creating a model with collinearity between the variables and very susceptible to overfitting.

What we found here: there are variables that are related because they are nested, like the isic codes and the other classification attributes of merchants. And we see that the card attributes are correlated because they are different descriptions of the same credit card. The only new thing to see is that there is some correlation between `merchant_classification` (by number of transactions) and isic codes, that is, the merchant categories. Also, there is some relation between isic codes and `payment_method_name` and `transaction_card_type`

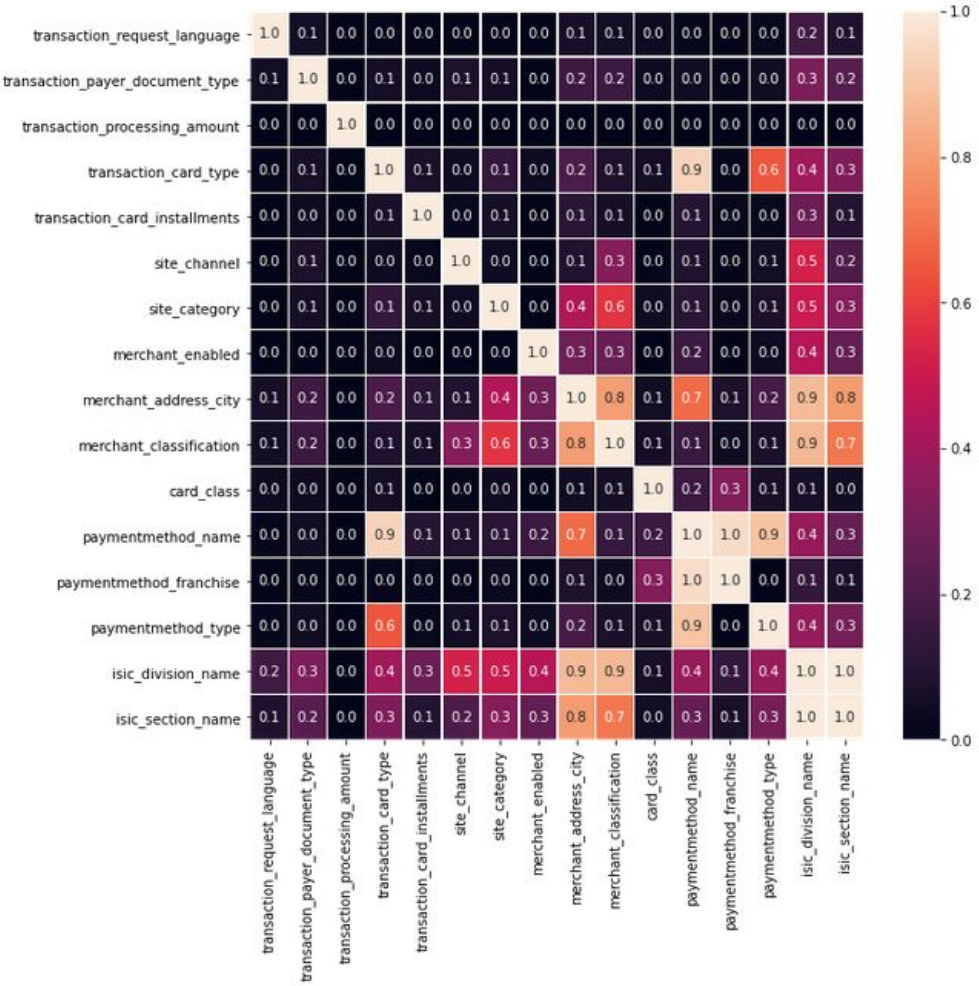


Figure 30. Phi-K Coefficient Correlation for the categorical variables.

So at this stage we can propose a redefinition of our hypothesis for further research by the company:

- Is any of this attributes showing any correlation with the purchase decision of the payer? As Fig.27 shows, there is not
- Is it there a different behavior between different types of payer? (they can be classified with recency, amount of purchases and number of purchases)
- Is there essentially the same behavior through all payers in different segmentation of merchants?
- Does the credit card attributes indicate some trend in the purchase habits?

Section 3 - Model and Statistical Analysis:

Statistical analysis

We have so far found that the data has high sparsity, high dimensionality, exhibits traits of heteroscedasticity and is composed of a mixture of frequency distribution that varies mainly across the different values of the different merchant categorizations available. So we have this issues with the data:

1. **Concentration of values and distances:** derived values such as distances and amounts become numerically similar
2. **Irrelevant attributes:** in high dimensional data, a significant number of attributes may be irrelevant
3. **Definition of reference sets:** for local methods, reference sets are often nearest-neighbor based
4. **Different subspaces produce incomparable scores** like rates or metrics, the scores often no longer convey a semantic meaning
5. **Exponential search space:** the search space can no longer be systematically scanned
6. **Data snooping bias:** given the large search space, for every desired significance a hypothesis can be found. Sometimes there is not enough data for every combination of variables, and in consequence is not possible to train a machine learning algorithm.
7. **Hubness:** certain values occur more frequently in neighbor lists than others.
8. **Assumptions required** for fitting most statistical techniques or models like normality of distributions (Independent variables are normal for each level of the grouping variable), homoscedasticity (Homogeneity of variance/covariance), restricted multicollinearity (Correlation between predictor variables) and Independence (Participants are assumed to be randomly sampled) are not met.

In order to fit a model, Logistic regression does not make many of the key assumptions of linear regression and other models that are based on least squares algorithms — particularly regarding linearity, normality, homoscedasticity, and measurement level. However, logistic regression does assume that there is a linear relationship between the predictors and the logit of the outcome variable and oftentimes this assumption is invalid, and there is no way to verify.

Another issue is that logistic regression is sensitive to class imbalances. For that reason we focused on the development of the recommendation system.

When we try to recommend items to payers, we face some fundamental challenges:

Data shortage vs sparsity: There are two cases: either there are very few payers for each product, so there is not enough information, or there are many products to recommend to many payers and it is unlikely that a payer will use a large part of the products. Instead, many payers are likely to demand some items, but many only a few.
Skimming: (Coldstart) We need to be able to give recommendations to payers for whom we only have scant data (if there is any).
Accurate predictions, but also diverse: We want to provide useful recommendations in the sense that they coincide with the payer's preferences, but also that the recommendation contains something new for the payer.
Evaluation: evaluation is difficult and can differ from one algorithm to another.
Scalability: We need to be able to give recommendations on the ground, although there may be millions of payers and elements that we have to analyze carefully.
User interface - payers want to know why they are receiving particular recommendations.
Vulnerability to attacks: we do not want our recommendation system to be abused to promote or inhibit certain elements.
Temporal resolution: Tastes and preferences do not remain the same over time.

There are some possible heuristics we can try, for example we can make a measure of the similarity between merchants, and propose a recommendation based on those other merchants. A caveat is that we would be proposing a recommendation of a product of a merchant that is a direct competitor of the merchant where the payer is making the purchase.

Another strategy is using the similarity between payers as a proxy for the merchant similarity. This approach would be useful when there are very few or none of the payers making purchases in another merchant or out of a category of merchants, or when the payers make only one purchase:

Strategy to build the recommendation system.

As said before our main product is a recommendation system to give each payer a set of products that they would like to purchase. There are two main approaches to recommendation systems: model-based and memory-based strategies, also known as content-based and collaborative filtering.

The first needs information about the context (payers and products) so it depends strongly on the availability and reliability of payer and transaction attributes, as they allow to build statistical models which try to identify the more suitable merchants to recommend given those attributes or characteristics. Unfortunately, the data provided by PlacetoPay suffers from two important drawbacks to use them with the model-based approach: (a) the availability of payers' attributes

is almost null; (b) Many of the available information seem to be contaminated and/or is difficult to interpret.

Therefore, the second one, the memory-based approach was adopted here as its information requirements are minimal: is only based on the past interactions between payers and the products. Because we lack context information about both payers and products, our recommendation system will be a collaborative filtering system.

Step 1. Use the transactional records of PlacetoPay to build the payer-merchant interaction matrix, in which each row represents a payer, each column represents a merchant, and the cell (i,j) indicates the number of purchases that the i-th payer made on the j-th merchant. The payer-merchant interaction matrix assumes that both payers and merchants in the transactional records of PlacetoPay may be unequivocally identified.

Step 2, Calculate a similarity measure between all pairs of columns of the payer-merchant interaction matrix, which is aimed to quantify the concordance between merchants regarding common payers and the purchasing frequency of latter. The similarity measure we used is the so-called cosine similarity measure, which consists of the cosine of angle between two vectors represented by two columns of payer-merchant interaction matrix. This similarity measure is frequently used as: (a) its values are within the interval (0,1), and the closer to 1 is its value the higher the similarity between two merchants; (b) it is symmetrical, that is, the similarity measured between the merchants j and k is the same that the similarity measured between the merchants k and j; (c) it may be easily calculated using basic functions of very popular libraries of python such as pandas.

Step 3. The recommendation for a payer is accomplished by using the similarity measures calculated in step 2 to find the more similar or concordant merchants to those in which the payer has made its purchases. Then, a score is calculated for the merchants, which consist of a weighted mean of the similarity measures, where the weights are the purchase frequencies or the purchase amounts. So, the recommendation comprises the N “new merchants” with the higher values of that score, in which “new merchants” mean merchants (stores or services sellers) which are different to those where the payer has made its purchases. One of the advantages of this approach to the recommendation system is the set of similarity measures is fixed (but it must be frequently updated) and the “online calculation” of the recommendation depends just on the records of past purchases of the interest payer.

Step 4. If there is a merchant whose similarity measures are all zero, then its set of concordant merchants is formed by those “more popular” merchants inside the segment of merchants it belongs, where “more popular” means the merchants having the higher purchase frequencies, or the higher purchase amounts.

Step 5. The payer-merchant interactions may be different regarding the spatial and/or temporal frames in which those interactions occur. Therefore, Steps 1-4 may be performed separately in each spatial and/or temporal frame where the experience of PlacetoPay suggests that the payer-merchant interactions are different.

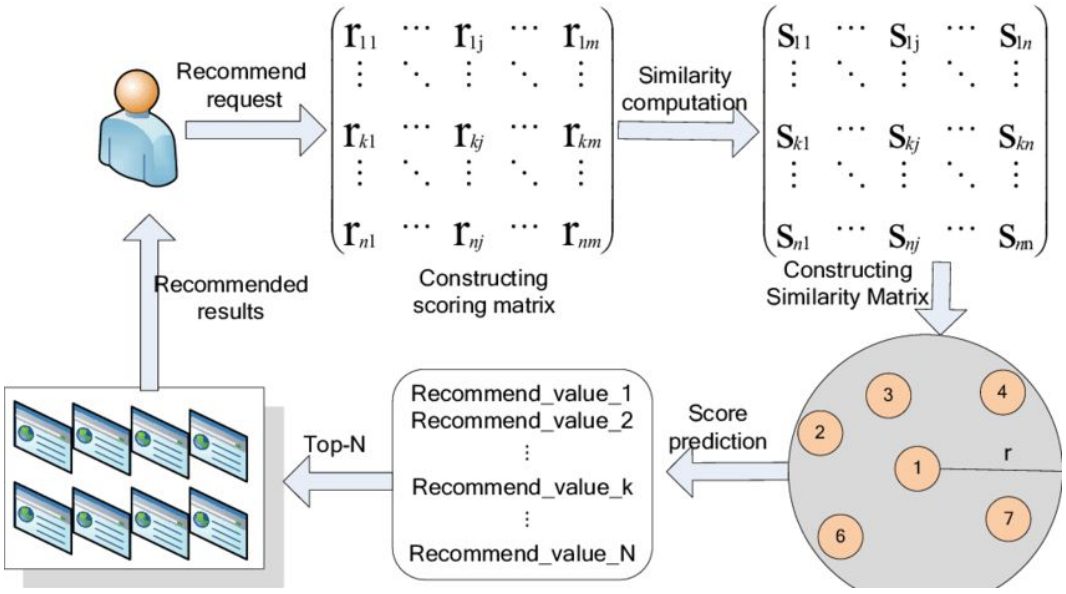


Figure 31. Process of building a recommender algorithm

Section 4 - Application and Results:

Front End: Interactive Descriptive Statistics

The dashboard is built in Dash, and anyone can access to it in the following link: https://t67-ptp-recosystem.herokuapp.com/

For descriptive data visualizations we choose PowerBI because Microsoft platform was able to cope with the load of the full database and provide a highly interactive way of comparing at the same time the interactions and distributions of all the possible combinations of at least 12 variables, check for outliers and understand the behavior of the data before being processed. This would not be possible by programming each individual plot in python. The placetopay dashboard are partitioned into 2 main sections. Descriptive analysis and recommender system.

Descriptive analytics

This section contains two tabs where a descriptive dashboard is shown (figure 26) with the most relevant analytics of the database considering we had a database with 47 variables and we need to compare them all and have granularity in the EDA experience where information relevant to a particular marketing segment will be available immediately without having to code or scroll through a notebook many times. The user will be able to select each store/merchant (or merchant category) and see the number and the value of your transactions, as well as the number of payers involved. This way, the user is able to understand seasonal patterns in that product category.

Figure 32. Dashboard of Descriptive Statistics in PowerBI mimicking Placetopay dashboards

The second tab, contains a geographical map (figure 27) that shows based on the division name and the merchant id, the amount of transactions in each registered location. This serves for the company to understand seasonality patterns in different locations as well as to create marketing strategies for entering in new markets or increase their lead compared their competitors.

For the recommender system, we define that the user will be able to select each store (or conglomerate of stores) and receive a list of the stores with the greatest similarity and / or agreement with it regarding purchases. This would allow Placetoplay, for example, to identify groups of stores that have similar payers and thus offer promotions, products and / or forms of payment together.

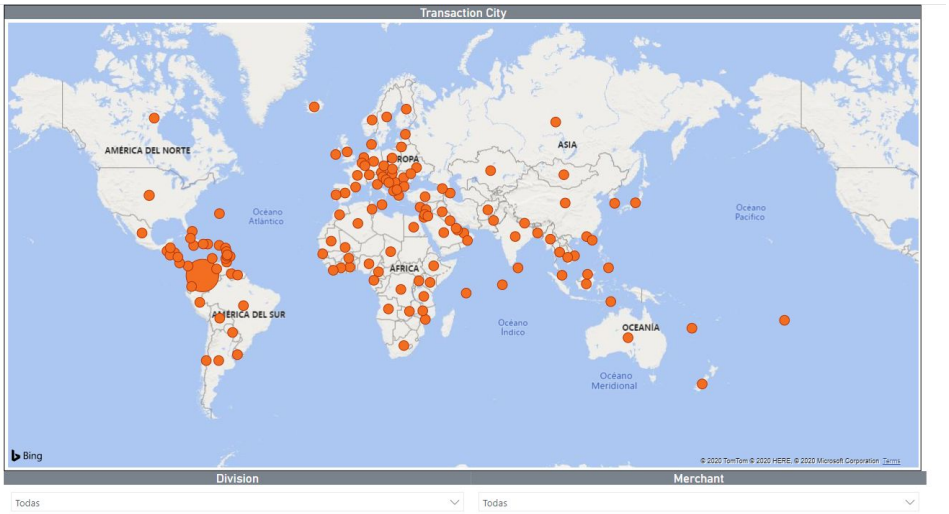


Figure 33. Dashboard showing Number of Transactions from each City, by Merchant category

Backend

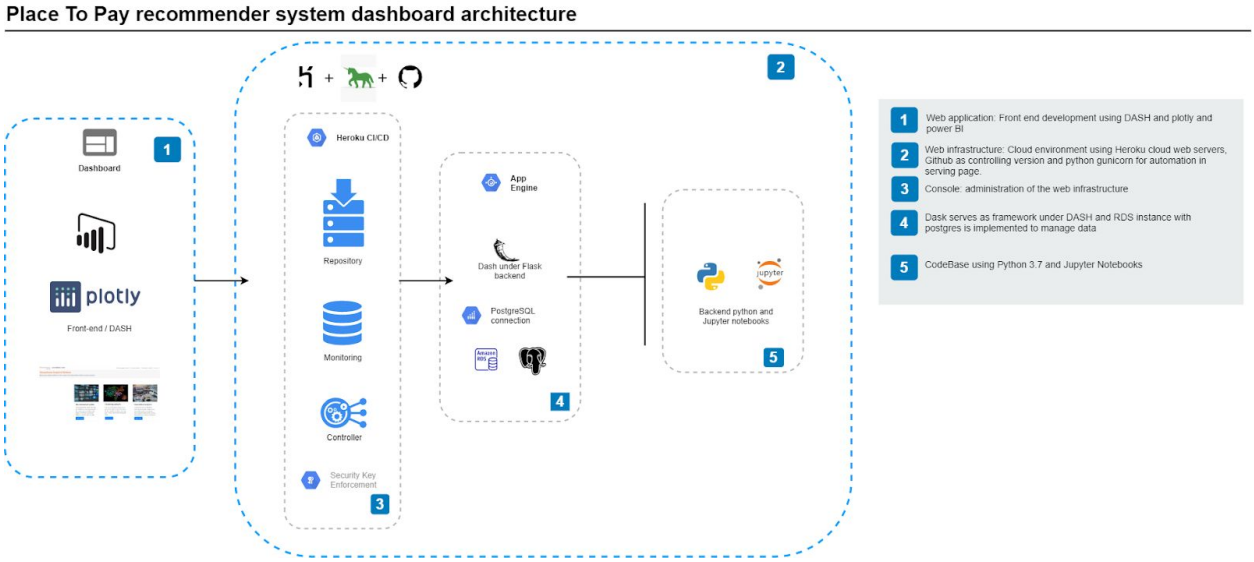


Figure 34. Implementation of the Database in PostgreSQL over AWS RDS

For building the app the stack used was DASH framework for the front-end supported with power BI considering the excess of data that we were handling in order to not crash the app. The deployment used Heroku, gunicorn and github to control the release versions properly. All the code base was made in python and the database used a RDS instance in ASW currently the app serve the database from a previous trained model so the stability of the app is ensured. However all the code to generate the model is in the base code of the app for future version with another framework.

Conclusions

At this stage, we have verified that the frequency distributions in the dataset are non linear. For that reason, doing further modeling will require us to cope with this nonlinearity and develop methods to transform the variables to a normalized form in order to fulfill the assumptions needed to do some hypothesis testing. Also, In order to research any hypothesis over the dataset is necessary to filter as much as possible any ambiguous information and reduce to the minimum the number of variables to consider for a particular hypothesis.

This high cardinality of the dataset means that we are dealing with so many dimensions (one for each possible value for each variable makes thousands of dummy variables.) that fitting a classification model did not bring results yet. So if we want to fit a logistic regression model we need a very clever work of reducing dimensionality in each variable.

How is the dynamic of use of the platform by merchants and payers? We have so far verified that we have a singular and different population of transactions for each section and even for each merchant. As the variability through merchants is so high, it is difficult to answer t questions, on average, in general. In most of the merchant classifications there is only one

single merchant and its population of transactions behave very differently than merchants from other classifications. Where there are a good number of merchants, there is a small number of merchants that produce the most transactions. Essentially there are a combination of many different behaviors present in the dataset. They have to be studied independently, (each segment, or each merchant separately).

1. How does the number of purchases per payer behave over time? How frequently?
   70% of the payers are single time users and 95% of payers buy from a single merchant. When there are more than one purchase, the time between purchases tend to be less than 30 days, but in some cases it reaches 250 days.

2. How many transactions have a particular merchant? By how much?
   Very few merchants tend to concentrate the majority of the transactions and transaction amount, and those merchants tend be the oldest.

3. How many different payers have bought from a merchant? What is the purchase value per payer?
   The majority of merchants have few payers and the merchant categories have very few different merchants. This sparsity in the dataset is present all over the transactions.

4. In how many different merchants a payer has bought? In how many categories?
   payers purchase very few times in the year and tend to buy from the same merchants, It will be useful to analize only the payers that have many purchases, but those payers could be companies and aggregators and then we would not know if we are studying individuals.

Recommendations

The dataset could be more useful if we were able to differentiate aggregators from high transactional payers, and each payer had a unique id. In the same line, the credit cards were related to a single individual payer.

In order to have a better understanding of the merchants and make a more intelligent recommendation system, it should be useful to have the descriptions of the transactions in a more standardized way, with descriptions more verbose than “Purchase Order No. xxxx”. However, in some cases the descriptions provide a great insight into understanding the payers.

The information in the dataset, also in the dataset dictionary must be unambiguous and not open to different interpretations. Otherwise, there is too much uncertainty in the data to work with. For example, for a merchant the field “region” means a department, and for other means a country or a neighborhood. It happens the same with some months and dates.

Credit card information is recorded only when the transaction is made with a credit card, but from our point of view, the type or form of payment does not affect the purchase decision. This can be a factor of confusion to study further.

The document type should be a field that has a limited set of available values. We found that people filled the field in different ways and with sensible private information (Id numbers, names) that must not be available to us.

Another issue to consider is that most of the transactions are made to pay obligations like rent, insurance or taxes. User behavior is different in this cases than, say, buying gifts or pleasures. It would be useful to be able to differentiate which merchants are which in order to make a separate analysis and compare the behaviors. This is an interesting study yet to make.

Further Work

A good model of classification to test in this dataset is discriminant correspondence analysis, if the assumptions are met and the data is properly filtered.

We managed to get a clear differentiation between two clusters of merchants, and two different ways (HDBSCAN and K-Modes) to re-classify the merchants in a better way than the company does.

A lot more improvement can be made in the characterization of the payers or their classification into user-profiles or clusters. The main issue is segmentate that population.

Desirable features for future versions:

- Classify customers by temporal and frequency decomposition (seasonal trends) of transaction descriptions made.
- A match score of a single payer into some preselected user profiles, showing which categories of products have been consumed by this profiles
- A match score for a product into some preselected product clusters, and some graphs showing the buying behavior over the months of this category,
- A matching of User Profiles that buy some particular product category
- A comparison of the sales of different merchants in a business sector, showing volume of sales trends or segmentation of customers
- Forecasted estimation of future trends of purchase behavior for each user profile.
- Recommendation section where a merchant can see what are the forecasted purchases that a user profile will buy and how much sales are predicted in that product category, for the particular merchant category.
- Explore more applications of Natural language processing into this dataset
- Explore applications of web scraping for this project

Appendix

References

[1] Wickham, H. (n.d.). Tidy Data. *Journal of Statistical Software*, *V*(II). https://vita.had.co.nz/papers/tidy-data.pdf

[2] Rahm, E., & Hai Do, H. (n.d.). Data Cleaning: Problems and Current Approaches. *Techn. Report, Dept. of Computer Sci-ence, University of Leipzig, Germany*. https://www.betterevaluation.org/sites/default/files/data_cleaning.pdf

[3] Scott, J. G. (n.d.). *Data Science: A Gentle Introduction*. https://jgscott.github.io/STA371H_Spring2018/files/DataScience.pdf

[4] Li, S. (n.d.). *A Complete Exploratory Data Analysis and Visualization for Text Data*. https://medium.com/@actsusanli. https://towardsdatascience.com/a-complete-exploratory-data-analysis-and-visualization-for-text-data-29fb1b96fb6a

[5] Jeremiah, T. (n.d.). *How to Build a Restaurant Recommendation System Using Latent Factor Collaborative Filtering*. https://towardsdatascience.com. https://towardsdatascience.com/how-to-build-a-restaurant-recommendation-system-using-latent-factor-collaborative-filtering-ffe08dd57dca

[6] Sio, K. H. (n.d.). *Exploration on Shopper Behavior and Shopping Cart Recommender*. https://towardsdatascience.com/. https://towardsdatascience.com/shopper-behavior-analysis-de3ff6b696b8

[7] Volpi, G. F. (n.d.). *A gentle introduction to Recommendation Systems*. https://towardsdatascience.com/. https://towardsdatascience.com/a-gentle-introduction-to-recommendation-systems-eaddcbde07ce

Documents:

1. 01-Team 67 Project Description - to Placetopay
   Document sent to the company describing the project
2. 02-Team 67 Project SCOPE
   Document sent to Correlation One stating the scope of the project
3. 05-DATASET dictionary
   Describes the characteristics of all the variables, the processing and the use, and a detailed report to the company stating technical and specific problems and solutions with the dataset
4. 08-Final report (this document)
   Describes the analytical process we made with the dataset. Also, the architecture of the solution and the way the solution works

Code and Notebooks:

The notebooks detailing the Exploratory Data Analisys are:

1_Database_Loading.ipynb
2_Database_Adjustment.ipynb
3_Database_cleaning_&_transformation.ipynb
4_Exploratory_Data_Analisys_(EDA).ipynb
5_Correlations_&_Data_Enhangement_(EDA-2).ipynb
6_Feature Engineering_(EDA-3).ipynb
7_Text and Language processing.ipynb
8_1_Clustering.ipynb
8_2_Cluster_analisys.ipynb
8_3_Clustering Kmodas.ipynb
9_Recommender_System.ipynb

Other:

but due to the technical limitations and the huge size of the dataset, we generate some new tables that make easier to start the analysis without having to calculate again everything from the notebooks.

1. *Merchants.csv*, a table with the calculated attributes of the merchants and its cluster.
2. *Payers.csv*, a table with the calculated attributes of the payers and their cluster
3. *Transaction_descriptions.csv,* a table with the descriptions statistics and tokenization
4. The code of the dashboard is available at: https://github.com/edward0rtiz/team67-ptp

Team 67 - Data Science for All Colombia - DS4A3 - MinTic

Luis Hernando Vanegas Penagos
Statistician M.Sc. Ph.D.

Juan David Arboleda Alaguna
Industrial Engineer, M.Sc. in Operations Research and Applied Statistics

Edward Ortiz
Business Administrator / Project Manager

Ximena Astrid Borda Casallas
Statistician

Daniel Salazar
International Relations and Finance - Risk Analytics

Diego Alvarez
Telecommunications and Systems Engineer

Luis Gustavo Maldonado
Mechatronics Engineer