DATATON BANCOLOMBIA 2018

We Support The Vector Machine

Danny Pineda Echeverri

Juan David Ossa Gomez

Victor Casas Hernandez

Resumen Ejecutivo

# Introducción

En el marco del concurso del Dataton 2018, este documento fue creado para explicar la metodología y línea de pensamiento de ejecución.

Todo parte de un problema, en este caso, el caso de uso de este Dataton es lograr tener una buena clasificación de transacciones de manera universal.

El problema suena fácil al principio, pero lamentablemente este problema sufre de algo que es muy común en casos de uso generalizados y que se conoce como incompletitud de información.

# Análisis Inicial Teórico

Pudimos ver el problema de la siguiente manera. Existen dos tipos de transacciones:

* Transacciones cliente - cliente **(Tipo 1)** : Estas transacciones gozan de completitud de datos, en este caso podemos obtener todos los datos necesarios para clasificarla de manera adecuada ya que contamos con el enlace de llegada de la transacción. Bastaría con cruzar el dato de llegada con información interna y clasificar.
* Transacciones cliente - usuario pse **(Tipo 2**): Estas transacciones poseen incompletitud de datos en el nodo de llegada. Dado que PSE es un sistema transaccional que no es exclusivo de Bancolombia, cualquier banco o entidad financiera puede ser el receptor de ella, esto incluye personas naturales o jurídicas. Además, muchos receptores deciden no poner una descripción o ponen una aparentemente poco informativa..

Bajo la luz de este análisis se nos ocurrió una metodología que permite tener un clasificador universal aunque tengamos información incompleta.

El enfoque que decidimos tomar para este ejercicio de modelado, es lo que se conoce como aprendizaje semi-supervisado o hybrid learning.

# ¿Por qué decidimos esta ruta?

Esto se debe a que pensamos que si aprendemos de las transacciones Tipo 1 y aplicamos lo que aprendimos a las Tipo 2, podremos tener un muy buen clasificador universal de transacciones, entrenado de manera supervisada. Así, nos basamos en la premisa que si una transacción Tipo 1 y una Tipo 2 tienen características similares, lo más probable es que la transacción Tipo 2 tenga la misma clasificación que la Tipo 1.

# ¿Cómo puedo lograr generalizar entonces, si solo voy a aprender de las Tipo 1?

Es evidente que la información en las transacciones Tipo 1 es completa y el enriquecimiento del registro transaccional resulta fácil y no es el caso del Tipo 2.

Para resolver esto tomamos las siguientes premisas de modelado:

1. **El cliente como central de aprendizaje:** Al identificar que el cliente está siempre presente en todas las transacciones (Por medio del ID), podemos tomar datos de el para enriquecer las transacciones de Tipo 1 y Tipo 2. Estos features son Universales y si logramos buen Lift, serán una buena herramienta para las Tipo 2.
2. **Campos libres como principal fuente de estocasticidad:** Creemos fuertemente que los campos libres de PSE (ref1, ref2 y ref3) son la fuente principal de estocasticidad, idiosincrasia y posibles features que ayuden a capturar el tipo de transacción. Estos campos son llenados por la persona que realiza la transacción e idealmente tiene informacion adicional para que su receptor identifique de dónde proviene. En estos campos en particular, un gran trabajo de procesamiento natural del lenguaje es necesario por lo que se convierte parte esencial de nuestro modelado.
3. **Mapeo de Categorías:** Debido a que deseamos que las categorías sean útiles para el cliente es necesario un mapeo de las categorías actuales proporcionadas por bancolombia a categorías más intuitivas y de uso diario para el cliente. Esto convertiría nuestro problema en un problema supervisado.

Si estos pasos se logran con éxito se puede contar con features que sean comunes a las transacciones tipo 1 y tipo 2, por lo que resolvería el problema de generalización.

# Detalle de ejecución del modelo

El paso a paso para la construcción del modelo sigue la siguiente lógica.

1. Buscar inconsistencias fundamentales en los datos de pagadores y de las transacciones.
2. Gestionarlas cualquiera que fuesen y generar las versiones de datos gestionados.
3. Enriquecer la data transaccional con:
   1. Información de segmentación (Análisis de Clustering) de clientes. Este análisis nos permite asignar categorías y nos permite la creación de keywords por cluster que permiten ayudar a enriquecer la transacción.
   2. Información demográfica de clientes (presente en la base de pagadores).
   3. Información transaccional de clientes (Que categorías son las que más gastaron el mes anterior y en qué porcentaje de gasto está con respecto a su ingreso inferido).
   4. Enriquecimiento de procesamiento natural de lenguaje, esto consiste en:
      1. Depurar caracteres especiales
      2. Lematizar
      3. Stemmer
      4. Part of Speech Tagging
      5. Filtrado de entidades de interés (Nombres, Pronombres y Adjetivos)
   5. Enriquecimiento por medio de modelado de Tópicos. Esto consiste en el uso de LDA para agrupar las palabras obtenidas por tópicos y enriquecer semánticamente cada transacción.
   6. Enriquecimiento por medio de expansión de léxico. Es bien sabido que lo que más limita un parser es el corpus sobre donde se entrene, como este lenguaje es bastante específico y está sometido a gentilicios colombianos. Desarrollamos un scraper a páginas amarillas que enriquece las entidades encontradas con descripciones.
   7. Unión de los features obtenidos.
   8. Luego de la unión es necesario un preprocesado de la variable de respuesta. Para esto se realizó un mapeo de las variables supervisadas de las transacciones tipo 1 a categorías de interés.
4. Preprocesado para entrenamiento.
5. Entrenamiento.
6. Entrega de resultados.

# Transformación y detección y manejo de anomalías en las bases

### Base de transacciones

#### 

Al intentar leer el archivo, se encuentran errores debido al número de separadores. Una exploración más profunda, revela que hay 22943 registros con estos problemas. La causa del problema es la presencia de comas en los campos de referencia. Después de una cuidadosa revisión, se creó un script que logra reemplazar las comas problemáticas por puntos y comas. Dicho script puede ser encontrado en el archivo *DataExplorationAndTransformation.ipynb* y es este proceso el que entrega la version 1 de los datos transaccionales.

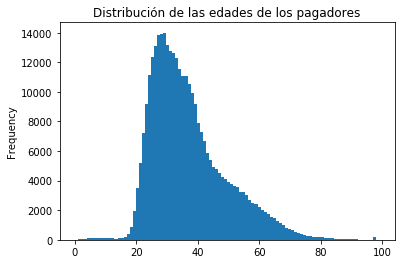
Adicionalmente, notamos que en este proceso es necesario:

* Transformar las fechas. Para ellos primero hacemos un coerce a string, añadimos un padding de 0s para poder ser coherentes con el formato y lo pasamos al formato de pandas datetime, que permite el slicing de una manera muy sencilla.
* Notamos que //N es un valor que se repite y percibimos que es el “nan” de los datos por lo que procedemos a transformarlo a nan y en el caso de string vacío.

Al realizar estas modificaciones, se llega a la versión 2 de los datos transaccionales.

### Base de pagadores

Respecto a la base de pagadores, hay dos pequeños grupos que llaman la atención y que merecen una exploración más detallada que los otros. Estos dos grupos corresponden respectivamente a los más jóvenes y más viejos de la base.



#### Exploración de datos de pagadores menores de 15 años

¿Por qué 15 años? Porque en Colombia, al llegar a los 15 años, un adolescente ya puede realizar legalmente muchas acciones normalmente atribuidas a adultos (claro está, debe contar con la debida autorización de sus padres y de las autoridades competentes). Por ejemplo, en el país, la edad mínima para trabajar es 15 años (exceptuando actividades culturales, artísticas, recreativas y deportivas, en donde la edad mínima es mucho menor) y la edad mínima para contraer matrimonio es de 14 años.

Hay un solo registro con valor negativo (-975):

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **id\_cliente** | **seg\_str** | **ocupacion** | **tipo\_vivienda** | **nivel\_academico** | **estado\_civil** | **genero** | **edad** | **ingreso\_rango** |
| **27526** | 269623 | PERSONAL | 1 | NaN | U | F | M | -975.0 | d. (3.3 4.4MM] |

Corresponde a un empleado con formación universitaria. Esta edad negativa puede deberse a un error en la digitación de la fecha de nacimiento: es posible que el asesor haya digitado el valor 2993 en vez de 1993, por lo que esta persona podría tener 25 años.

Respecto a los otros valores se tiene:

|  |  |  |
| --- | --- | --- |
|  | **id\_cliente** | **edad** |
| **count** | 2008.000000 | 2008.000000 |
| **mean** | 236592.051793 | 10.727590 |
| **std** | 50678.610162 | 5.121639 |
| **min** | 449.000000 | 0.000000 |
| **25%** | 194120.500000 | 6.000000 |
| **50%** | 228820.500000 | 11.000000 |
| **75%** | 277891.000000 | 16.000000 |
| **max** | 338504.000000 | 17.000000 |

Por lo que la mayoría dicen tener una edad mayor a 10 años, edad en la cual no sería raro que un padre abriera una cuenta a un hijo. Aunque a primera vista parezca raro, las cuentas de estos niños sí podrían reportar ingresos, pues hay niños que se desempeñan como actores o que reciben una pensión por haber quedado huérfanos. Por ello, para detectar incongruencias, la característica del rango de ingresos no es confiable.

a. (0 1.1MM] 0.577586  
b. (1.1 2.2MM] 0.317888  
c. (2.2 3.3MM] 0.034483  
No disponible 0.016164  
d. (3.3 4.4MM] 0.015086  
0 0.009698  
e. (4.4 5.5MM] 0.008621  
f. (5.5 6.6MM] 0.008621  
h. (7.6 8.7MM] 0.006466  
i. (8.7 Inf) 0.003233  
g. (6.6 7.6MM] 0.002155

Respecto al segmento, se observa hay algunos registros de supuestos niños con segmento “EMPRENDEDOR”, algo que no deja de ser anormal y por ello, estos registros se marcan como tal.

PERSONAL 1286  
PERSONAL PLUS 39  
EMPRENDEDOR 13  
OTRO 1  
PREFERENCIAL 1

En cuanto a la ocupación, se observa que la mayoría de registros poseen una ocupación de Estudiante (2), lo que es bastante intuitivo; sin embargo, hay una considerable cantidad de registros con ocupaciones anormales, como Empleado (1), Desempleado sin ingresos (S), Independiente (3 ), Empleado o socio (E), entre otros. No es del todo raro que un niño menor de 15 años aparezca como empleado, pues en Colombia se otorgan permisos especiales de trabajo si la actividad a desempeñar es cultural, artística, recreativa o deportiva, lo que sí es más extraño, es que un asesor haya decidido registrar a un niño como empleado en vez de estudiante y por ello, estos registros requieren inspección adicional. Así, aquellos registros que tengan ocupación diferente a Estudiante (2), Otra (O), Desempleado sin ingresos (S) y missing (NaN) se consideran anormales.

2 978  
O 194  
1 118  
S 14  
3 13  
I 12  
NaN 5  
E 2  
P 1  
4 1  
5 1  
8 1

En lo referente a la formación académica, se consideran anómalos aquellos registros con edad menor 9 años que poseían formación académica “Bachillerato”, en lo demás, solo fueron considerados normales aquellos registros con formación Missing (NaN), No informa (I), Primaria (P), Ninguno (N) y Bachillerato (este último solo fue considerado normal para aquellos registros con edades de 9 años o más).

NaN 531  
I 466  
P 146  
N 118  
U 30  
B 19  
H 13  
T 12  
S 3  
E 2

En Colombia, el matrimonio es legal a partir de los 14 años. Por ello, en lo referente al campo Estado civil, se consideran anormales aquellos registros con edad menor a 14 años que tengan en el campo Estado civil los valores M (Casado), D (Divorciado) o W (Viudo). En la base de clientes se encuentran 23 registros con estas características que se marcan como anómalos.

S 977  
I 128  
NaN 69  
O 21  
D 14  
M 9  
F 5

#### Exploración de datos de pagadores con edades atípicamente altas

100.0 1  
101.0 1  
103.0 3  
108.0 57  
115.0 4  
117.0 2  
118.0 1552

Dentro del grupo de edades mayores o iguales a 100, se observa una evidente anormalidad con la edad de 118 años: 1552 clientes registran esta edad. Sin embargo, solo se han confirmado dos personas en la historia que han logrado vivir más de 117 años. Esto lleva a pensar en dos posibilidades para este fenómeno. La primera de ellas es que se deba a un error de digitación común por parte de los asesores a la hora de registrar a los clientes. La segunda y más plausible, es que dicha edad sea el valor por defecto que se le asigna a los clientes con fecha de nacimiento con valor “0”. Una persona que registre 118 años en el 2018, debió haber nacido en el año 1900. En sistemas como Microsoft Excel, una fecha con valor numérico de “0” es representada como el 00/01/1900. Así, la hipótesis del valor por defecto parece estar bien sustentada y por ello se deciden reetiquetar dichas edades como faltantes.

Respecto a las otras edades, las de 115 y 117 siguen siendo problemáticas, a fecha de abril del 2018, el hombre más viejo del mundo (ganador del Guinness de los Récords) contaba con 112 años. Debido a ello, se podría pensar que estos registros se deben a errores de digitación o a cuentas de personas difuntas que no han sido canceladas y siguen siendo manejadas por algún familiar. Por otro lado, la edad de 108 años, aunque más probable, tiene una frecuencia atípicamente alta comparada con las otras altas edades, a tal punto de representar casi el 92% de los registros con edades entre los 100 y 108 años. Una distribución de este tipo es completamente irregular.

## Rango de ingresos

Se establece al centavo (0.01 pesos colombianos) como mínima variación en los ingresos. Para poder tener la posibilidad de aprovechar las bondades de los datos numéricos, a partir del campo rango de ingresos se crean otros dos campos: uno que contiene el mínimo valor del intervalo y otro que contiene el máximo; la excepción es el intervalo (8.7 Inf), en dónde el campo con el supremo se establece en 1x10e10. Estos campos solo son auxiliares en la exploración de la data y no serán usados para entrenar ningún algoritmo.

De esta manera, los nuevos campos de ingresos quedan de la siguiente forma:

|  |  |  |
| --- | --- | --- |
| **Rango original** | **Mínimo** | **Máximo** |
| 0 | 0 | 0 |
| No disponible | NaN | NaN |
| a. (0 1.1MM] | 0.01 | 1100000 |
| b. (1.1 2.2MM] | 1100000.01 | 2200000 |
| c. (2.2 3.3MM] | 2200000.01 | 3300000 |
| d. (3.3 4.4MM] | 3300000.01 | 4400000 |
| e. (4.4 5.5MM] | 4400000.01 | 5500000 |
| f. (5.5 6.6MM] | 5500000.01 | 6600000 |
| g. (6.6 7.6MM] | 6600000.01 | 7600000 |
| h. (7.6 8.7MM] | 7600000.01 | 8700000 |
| i. (8.7 Inf) | 8700000.01 | 1x10e10 |

El script de estas modificaciones, puede ser encontrado en el archivo *data\_transformation.R*.

# Enriquecimiento de datos

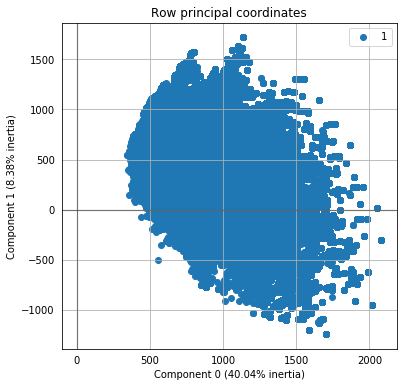
## Agrupamiento de los clientes pagadores

La idea consiste en categorizar a los clientes usando un algoritmo de agrupamiento para posteriormente analizar estas categorías y extraer palabras claves que enriquezcan la data.. Aunque la idea parece sencilla, no lo es en absoluto. Para realizar este agrupamiento, debe tenerse en cuenta que los datos de los clientes son mixtos, es decir, están compuestos tanto por numéricos (la edad) como por datos categóricos (segmento, nivel de educación, género, etc). Así mismo, se debe tener en cuenta que los datos poseen una dimensionalidad suficientemente alta como para dificultar su visualización y son lo suficientemente numerosos como para que el consumo de recursos computacionales sea una preocupación.

No tendría sentido aplicar la mayoría de los algoritmos de agrupamiento a datos categóricos, pues por lo general, estos algoritmos suelen usar distancia espacial para determinar si un grupo de datos es similar o diferente, y hablar de distancias espaciales en datos categóricos no siempre tiene sentido (por ejemplo, es difícil concebir la idea de distancia espacial para las categorías de Estudiante, Empleado y Ganadero). Por otro lado, algoritmos como K-modes, que no se basan es una distancia espacial, podrían ser adecuados para agrupar datos mixtos. Sin embargo, el proceso de agrupamiento mediante estos algoritmos no está exento de problemas, puesto que si la dimensionalidad de los datos no es baja, sería bastante difícil visualizarlos y por ello, sería complicado evaluar la calidad del agrupamiento.

Para resolver este problema, el equipo decidió hacer uso de FAMD (Factor Analysis of Mixed Data) un método que facilita analizar datos mixtos al transformarlos en componentes numéricas. Aún con esto definido, se presenta el problema de la carencia de implementaciones que sean escalables a datos de mediano y gran tamaño, pues todas las implementaciones probadas hacían uso de matrices gigantes que requerirían más 400 GB de memoria RAM para poder entrenar la totalidad de registros de la base de pagadores (ninguna de las implementaciones encontradas hacían uso de las *sparse matrix*, que hacen un uso más eficiente de la memoria RAM). El enfoque llevado a cabo para dar solución a este obstáculo, consistió en hacer en extraer varias muestras de la data (muestras de alrededor del 10% del total de datos) y transformarlas en 2 factores numéricos. De esta forma, encontramos que las transformaciones no variaron de manera importante entre las muestras y a la final, decidimos aplicar uno de estos transformadores (entrenado con cerca del 10% de los registros) a toda la data. La implementación del FAMD usada por el equipo, es la disponible en la librería *prince* de python. El código de esta transformación y de todas las de esta sección se puede encontrar en archivo *pagadores.ipynb*.

El resultado obtenido se muestra a continuación.

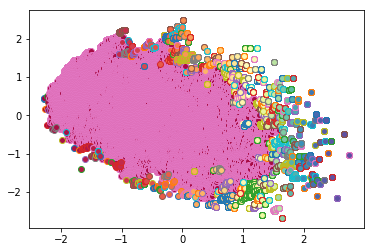


Los nuevos features obtenidos fueron estandarizados para hacerlos adecuados para ciertos algoritmos de agrupamiento.

### **Algoritmos de agrupamiento aplicados**

#### **Dbscan**

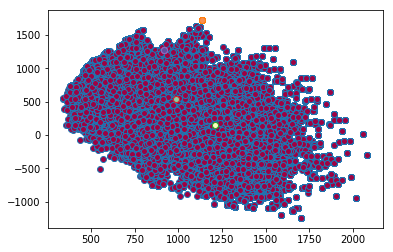
La implementación usada fue la de scikit-learn. El algoritmo seleccionado fue *ball-tree*. Este algoritmo, aunque más escalable que otras alternativas, hace uso de un matriz de distancias, por lo que para poder correrlo sobre toda la data fue necesario remover features duplicados y pre calcular la matriz haciendo uso de sparse-matrix. *La distancia usada fue radius neighbors graph*. Se intentaron varios parámetros epsilon sin éxito. Parámetros mayores 0.1 necesitaban enormes cantidades de memoria, mientras que parámetros más pequeños, como 0.05, daban resultados poco satisfactorios, posiblemente debido a la enorme densidad presente en los datos.



**Dbscan. Epsilon 0.05.**

#### **K-median**

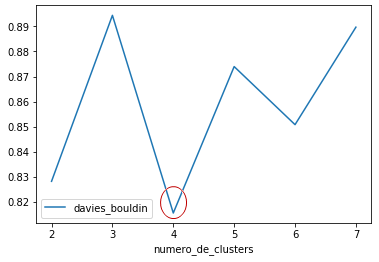
Los resultados obtenidos al aplicar k-median tampoco fueron satisfactorios, posiblemente a la gran densidad de la data. A continuación, se muestra el resultado para clusters, pero el comportamiento de las clasificaciones es invariable respecto al número de categorías.



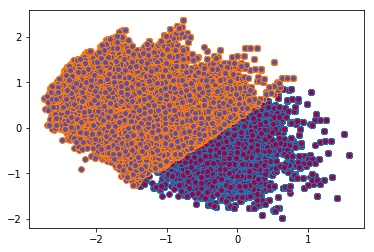
**K-median (5 Clusters)**

#### **K-mean++**

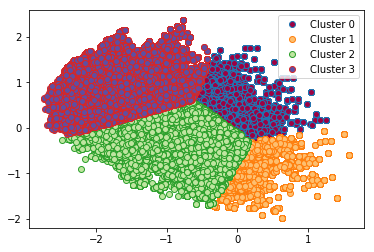
Al aplicar K-mean, se intentó hallar el número óptimo de clusters, para este propósito se usó el score de Davies-Bouldin.



Como se observa en la imagen, el criterio sugiere que el número óptimo de clusters es 4, sin embargo, el valor de 2 clusters le sigue muy de cerca. Por ello, decidimos aplicar el algoritmo con ambos valores. Los resultados se muestran a continuación.

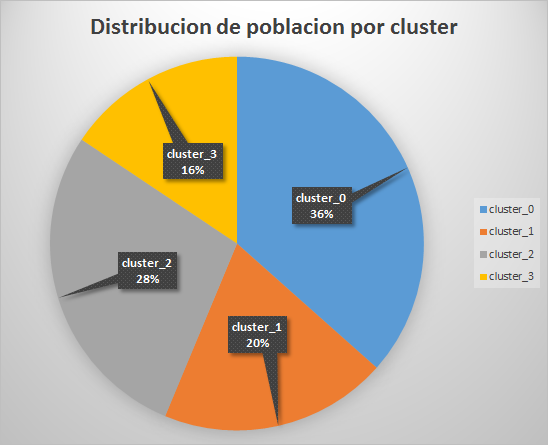


**K-mean (2 Clusters)**

****

**K-mean - (4 clusters)**

En ambos casos la distribución de los grupos luce bien, sin embargo, dando preferencia a la diversidad y siguiendo el criterio de la prueba de Davies-Bouldin, hemos decidido seleccionar el agrupamiento de 4 categorías.



### **Información de segmentación (Análisis de Clustering) de clientes**

#### **Descripciones de los clusters**

A continuación haremos una breve descripción de los clúster para evidenciar características más relevantes, para el análisis consideramos como relevante concentraciones de por lo menos 10% en una categoría.

El cluster 0 está totalmente concentrado en personas de 20-30 años, cuyo segmento es 99% son del segmento personal, en ocupación 71% son empleados y 20% son estudiantes; en tipo de vivienda 95% no informa; en nivel académico 50% no informa, 20% universitario y 15% técnico; en estado civil 80% soltero, sin distinción de género con ingresos de 1.1 a 2.2 MM en 54%, de 0 a 1.1 MM 26% y 2.2 a 3.3 MM en 12%

Key words: jóvenes, solteros, estudiantes, ingresos bajos, sin vivienda

El cluster 1 son personas entre 30-60 años en su mayoría, cuyo segmento es 43% personal plus y 24% son preferencial; en ocupación 23% emprendedor, 39% son empleados y 34% independientes; en tipo de vivienda 39% propia, 39% familiar y 14% alquilada; en nivel académico 55% universitarios y 16% postgrado; en estado civil 58% casado, 16% soltero y 10% desconocido; en género 63% hombres y 37% mujeres; con ingresos de 8.7 MM o superior en 41%, 5.5 a 6.6 MM de 12%, 4.4 a 5.5 MM de 10% y 6.6 a 7.7 MM 9%.

Key words: adultos mayores, emprendedores, vivienda propia, profesionales, postgrado, casados ingresos altos

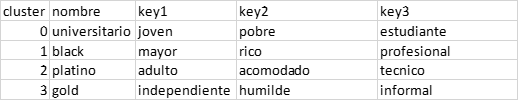
El cluster 2 son personas de 25 a 50 años en su mayoría, cuyo segmento es 88% personal y 9% personal plus; en ocupación 92% empleados; en tipo de vivienda 51% familiar, 25% no informa, 13% propia y 10% alquilada; en nivel académico 70% universitario y 13% técnico; en estado civil 49% solteros, 31% casados y 10% desconocido; sin distinción de género, con ingresos entre 1.1 y 4.4 MM (76%).

Key words: Jóvenes adultos, adultos contemporáneos, empleados, vivienda familiar, estudios técnicos, universitarios, solteros, ingresos medio-altos.

El cluster 3 son personas de 20 a 70 años en su mayoría; cuyo segmento 45% personal, 36% emprendedor y 15% personal plus; en ocupación 30% independiente y 19% empleado; en tipo de vivienda 63% no informa, 13% familiar, 11% propia y 11% alquilada; en nivel académico 36% no informa, 16% universitario, 15% técnico y 13% bachillerato; estado civil 40% solteros, 30% no informa o desconocido y 16% casados; en género 60% mujeres y 40% hombres con ingresos de 0 a 4.4 MM en 66%.

Key words: Independientes, trabajo informal, sin vivienda, sin estudios superiores, ingresos medio-bajo

Al final realizamos una tabla de keywords por cliente que después nos permitirá enriquecer la información de las transacciones



### **Información transaccional del cliente**

Para enriquecer la información transaccional de los clientes, se crearon 7 features adicionales. Los tres primeros de ellos corresponden a los sectores en los que el cliente más transacciones realizó en el mes anterior al mes en el que realizó la transacción. Los tres segundos corresponden a los subsectores en los que el cliente más transacciones realizó en el mes anterior al mes en el que realizó la transacción. Por último, el séptimo feature corresponde al dinero que el cliente ha gastado en transacciones en lo corrido del mes. A los 6 primeros features les subyace la hipótesis de que un cliente tiene compras recurrentes. Al séptimo feature le subyace la hipótesis de que un cliente va modificar su comportamiento de compras de acuerdo a qué tanto haya gastado recientemente. Para calcular estos features sobre una base de datos tan grande como la de las transacciones, fue necesario escribir funciones altamente vectorizadas. Las funciones pueden ser encontradas en el notebook *Funciones\_features\_transaccione.ipynb*.

### **Enriquecimiento de procesamiento natural de lenguaje**

Para este proceso partimos de la V2 y procedemos a hacer un trabajo exhaustivo de preparación y preprocessing para uso de técnicas de procesamiento natural del lenguaje.

Antes de empezar este proceso, notamos que la combinación de lenguajes es común en este campo libre por lo que abandonamos la idea de un NER (Named Entity Recognition) y vamos a adoptar un approach de enriquecimiento de palabras.

La idea es tomar el campo de referencia y obtener las categorías gramaticales presentes en él (Part of Speech Tag - POS); para luego tomar lo que sea identificado como Adjetivo, Nombre, Pronombre y Verbo y, con ello seguir el enfoque del “Document Retrieval”, donde cada transacción representa un documento y se clasifica su texto de acuerdo a su tópico, determinado por un LDA.

Como primer paso del preprocesamiento, se verifican que existen caracteres especiales, mayúsculas, sobre espaciado y puntuación que no aporta valor alguno al documento. Estos caracteres se remueven.

Luego de la remoción, se aplica POS Tag. Para poder realizar el POS Tag, se usó la librería Spacy con el news\_es (Modelo de noticias en espanol). Se corre el POS Tag con este parser y el resultado para cada ref se almacena en una nueva columna con la que se va formar el documento.

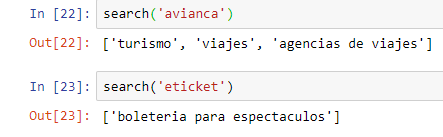
### **Enriquecimiento por medio de modelado de Tópicos**

Lastimosamente, debido al costo computacional y a otras prioridades, no se pudo realizar este enriquecimiento.

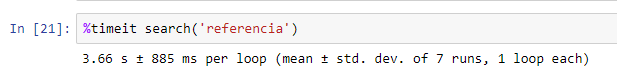
### **Expansión del léxico**

En un intento de enriquecer el léxico (más específicamente, los nouns identificados), desarrollamos un scraper que realiza una búsqueda en el sitio web de páginas amarillas y que en caso de encontrar resultados, devuelve una lista de actividades económicas que las empresas encontradas realizan. Esto responde a las necesidades específicas de enriquecimiento de data, desde el punto de vista de informacion colombiana.

Dos ejemplos del uso de esta función se muestran a continuación:



Lastimosamente, este enfoque resultó no ser viable por el momento, debido a que tomaría demasiado tiempo hacer esto para 346211 búsquedas que se deberían realizar para enriquecer la data.



El script del scraper y otros detalles se encuentran en el archivo *Scrapper\_Integration.ipynb.*

### **Mapeo de categorías a subsector**

Para definir las nuevas categorías donde se ubicaran las transacciones nos basamos en una forma de mapeo donde analizamos uno a uno los subsectores, teniendo en cuenta tanto su nombre, descripción y el número de transacciones, esto con el fin de ver que tan común puede ser, el proceso fue el siguiente:

1. En principio usamos las categorías propuestas por el banco para ver si podrían mapearse en alguna de ellas haciendo una revisión de subsector y descripción.
2. Los subsectores que no pudimos agrupar dentro de las categorías propuestas por el banco fueron analizados para ver qué tan parecidos podrían ser entre sí para acomodarse dentro de una nueva categoría que los contuviera.

Al final definimos un total de 27 categorías, para ver en detalle qué combinación de subsector y descripción tiene cada categoría por favor remitirse a la tabla “mapeo\_subsector\_cat\_v2“:

* Agro
* Ahorro
* Comercio Mayor
* Comercio Menor
* Comida
* Editoriales
* Educación
* Empresas
* Farmacias
* Gobierno
* Hogar
* Infraestructura
* Inmobiliario
* IPS
* Mascotas
* Materiales
* Otros
* Personas
* Prestaciones Sociales
* Seguridad Privada
* Seguros
* Servicios financieros
* Supermercados
* Tecnología y comunicaciones
* Textiles
* Transporte
* Viajes

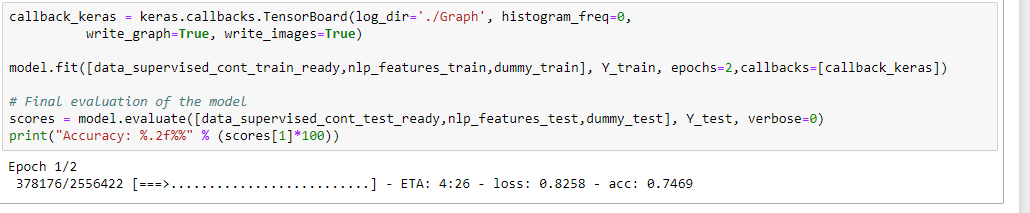
Escogimos estos nombres principalmente pensando en que sea un beneficio para el cliente a la hora de leer sus gastos de tal manera que sean abarcados en su totalidad y pueda tener una noción de ellos de manera tal que pueda administrarlos eficientemente.

# Preprocesado para entrenamiento

En esta fase, además de consolidar los features en la base de transacciones, se establece el target feature y con ello, se convierte el problema en un problema de aprendizaje supervisado. Los detalles de esta fase se pueden encontrar en el archivo PreTrainingNotebook.ipynb.

# Entrenamiento y entrega de resultados

Se realizó la partición de los datos con label (cerca de 3 millones de datos) en el conjunto de entrenamiento y el conjunto de prueba. El entrenamiento de la red se llevó a cabo con el paquete Keras de Python. Esta red cuenta con una precisión de cerca del 90% a la hora de predecir las categorías del conjunto de prueba.



Los detalles pueden encontrarse en el archivo TrainingNotebook.ipynb.