**MEMORIA**

**ANÁLISIS CUSTOMER JOURNEY PARA CLIENTE DE BANCA**

Máster Data Science 12ª Edición Óscar Martínez Aceña

**Breve introducción**

El objetivo de mi TFM consiste en analizar si a partir de los datos contenidos en la siguiente ruta:

<https://data.world/lpetrocelli/czech-financial-dataset-real-anonymized-transactions>

Se puede definir el “customer journey” de los clientes para alguno o varios productos. Es decir, si se puede conocer qué operaciones / transacciones pueden indicar la contratación de determinados productos bancarios.

Operativamente, el trabajo se he realizado de la siguiente forma:

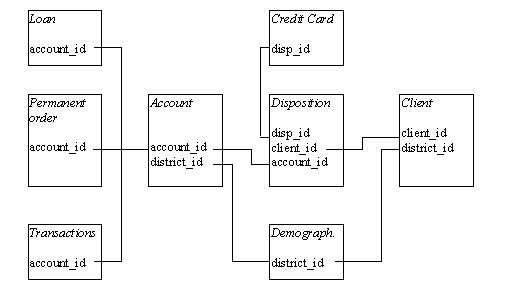
* Utilizando R me conecto a la web anterior y me descargo los datos.
* Combinando con R los distintos ficheros sobre: productos, clientes, transacciones, datos socio económicos, … elaboro un DataFrame para realizar el análisis. El tiempo de ejecución de los pasos anteriores está en algo menos de 30 min y los paquetes a utilizar están indicados al principio (habitualmente hemos utilizados en clase). El resultado obtenido es el fichero “TFM Data Set análisis productos.R”
* A partir del fichero anterior, lo cargo en un Jupyter Notebook y realizo el análisis con Python (paquetes que utilizamos).

Por lo tanto, una vez descargado el REPO, hay que xxxxxxx

**Descripción inicial de los datos**

Los datos son datos reales de un banco Checo de 1.999. Aunque dichos datos contienen también operativa de años anteriores. Los campos de texto de los ficheros, por lo general, están en lengua checa, por lo que a menudo va a ser necesario traducir dichos campos para poder interpretar su significado (las traducciones se encuentran en la web indicada anteriormente, en la descripción de cada fichero).

Una descripción visual de los ficheros disponibles es la siguiente:



Como podemos ver hay datos:

* *Específicos de productos*: Accounts (cuentas corrientes), Credit Card (tarjetas de crédito), Permanent order (domiciliaciones / pagos domiciliados) y Loans (préstamos)
* *Específicos de clientes*: Client (clientes) y Disposition (nos va a permitir relacionar al cliente y los productos)
* *Transaccionalidad realizada por las cuentas* (Transactions)
* *Datos socioeconómicos*: Demograph

La estrategia seguida para aglutinar toda la información consiste en crear un dataset, que finalmente llamaremos “DFTenenciaProductos” partiendo de la información de las cuentas corrientes (Accounts) y enriquecer dicho fichero (Accounts) con la información que consideramos relevante del resto de ficheros.

**Generación dataset “DFTenenciaProductos”**

Los ficheros que vamos a descargar de la web anterior están en formato csv y son en general bastante ligeros, a excepción del fichero de transacciones que ocupa unos 68MB y que contiene más de 1 millón de transacciones y que tarda alrededor de 5 minutos en descargarse.

Para cada fichero que descarguemos la metodología de trabajo que vamos a seguir consiste en:

* Analizar la estructura del fichero (observaciones y variables)
* Buscar missings y duplicates: No hay duplicates en los ficheros y sólo hay missings en el fichero de datos socioeconómicos *(district)*
* Incorporamos al fichero *account* (que nos sirve como base) los datos que consideramos relevantes del fichero que estamos tratando.

Iniciamos el proceso con el fichero *account* (cuentas corrientes del perímetro con el que trabajamos):

Lo que se cuenta a partir de aquí puede seguirse en paralelo con el código del fichero “TFM Data Set análisis productos.R”

Este fichero aporta número de cuenta, fecha de apertura de la cuenta, distrito (más tarde lo ligaremos con el fichero *district*) en el que estaba la sucursal en la que se abrió la cuenta y frecuencia con la que se generan extractos de la cuenta (este campo está en checo y se traduce en el código. Por ejemplo: "POPLATEK MESICNE" significa periodicidad mensual).

De este fichero incorporamos todas sus variables al dataset final. Hay 4.500 cuentas en total.

Fichero *client* (clientes del perímetro con el que trabajamos):

Este fichero contiene las variables de número de cliente, distrito y birth\_number (la explicamos a continuación), pero no contiene variables para poder cruzarlo con el DataFrame *account*, necesitaremos del fichero *disp* (de disposiciones) para poder hacerlo.

La variable birth\_number recoge la fecha de nacimiento y el sexo de cada cliente, ya que para los hombres está en formato YY/MM/DD y para las mujeres en formato YY/MM+50/DD. Y por tanto a partir de las posiciones que indican el mes de nacimiento se deduce el sexo del cliente.

Fichero *disp* (claves de propiedad del perímetro con el que trabajamos):

Contiene las variables disp\_id (clave de disponibilidad, es un identificador), client\_id (el número de cliente que hemos visto en el fichero anterior), account\_id (el número de cuenta) y type que indica el tipo de disposición del cliente sobre la cuenta (si es OWNER puede realizar todo tipo de operativas y si es DISPONENT (autorizado) no puede realizar todo tipo de operativas (por ejemplo dar de alta una domiciliación o solicitar un préstamo). Por tanto, va a haber cuentas con 2 tipos de usuarios: propietarios y autorizados y cuentas que únicamente tienen propietarios.

A partir de este fichero y utilizando 2 diccionarios, incorporamos al DataFrame *account* el ID del propietario de la cuenta y el ID del autorizado (disponent) si lo tiene (estos datos estaban en el fichero *disp)*. Del fichero *client* incorporamos el sexo y la fecha de nacimiento de propietario y autorizado (si lo hay). Sólo 869 cuentas tienen autorizado y por tanto esta información se añadirá únicamente a una parte de las cuentas. En el resto estas variables aparecen como N/A.

Fichero *loan* (prestamos contratados por el perímetro de clientes / cuentas con el trabajamos. Los préstamos pueden estar en vigor o ya finalizados):

Contiene las variables loan\_id, account\_id (nos servirá para realizar un merge), date (fecha de concesión del préstamo), amount (importe concedido inicialmente), duration (plazo inicial del préstamo), payments (cuota mensual del préstamo) y status (es una variable categórica que indica si el préstamo está o no al corriente de pago y si el préstamo está vivo o se ha amortizado completamente).

Los 4 niveles de status son:

* A: para préstamo finalizado y al corriente de pago. Para entenderlo mejor lo codificamos como “OK”
* B: para préstamo finalizado pero moroso. Lo codificamos como “NPL” (de Non Performing Loan)
* C: para préstamo en vigor y al corriente de pago. Lo codificamos como “OK\_not\_fin” (not\_fin de not finished)
* D: para préstamo en vigor pero en situación de mora. Lo codificamos como “NPL\_not\_fin”

Adicionalmente, de la variable status generamos una variable binaria “status\_loan\_bin” que contenga si el préstamo ha incurrido en situación de mora o no. Creo que podría ser útil más adelante.

Se incorporan todas las variables del fichero *loan* + *“*status\_loan\_bin”al DataFrame *account* (que nos sirve como base sobre la que agregar variables)*.* No todas las cuentas han contratado algún préstamo, de hecho sólo 682 lo han hecho y por lo tanto estas variables se añadirán únicamente a una parte de las cuentas. En el resto estas variables aparecen como N/A.

Fichero *order* (transferencias / domiciliaciones automáticas del perímetro con el que trabajamos).Contiene las variables:

* order\_id: es el identificador de la domiciliación. No se incluye en *account* porque considero que no aporta información relevante.
* account\_id: nos va a servir para incluir la información en el DF *account*
* bank\_to: es el banco receptor de la transferencia / domiciliación. No se incluye en *account*
* account\_to: es la cuenta receptora de la transferencia / domiciliación. No se incluye en *account*
* k\_symbol es la finalidad de la transferencia. La finalidad está en idioma checo y lo traducimos. Además para cada cuenta agrupamos en *account* cuantas transferencias tiene de cada tipología/finalidad.
* amount: es el importe de la domiciliación. Lo incluimos en *account* agrupado por cada tipología.

Por tanto, a partir de la información contenida en *order*, generamos 2 nuevas variables por cada tipología de domiciliación (seguros, pagos del hogar, préstamos, leasings, ...) que nos indica si la cuenta realiza ese tipo de operativa, como de habitual ha sido esta operativa en el periodo de recogida de los datos y qué importes se han destinado a dicha finalidad.

Fichero *card* (tarjetas de crédito contratadas en el perímetro de datos con el que trabajamos). Contiene las variables:

* card\_id: es el identificador de la domiciliación. No se incluye en *account* porque considero que no aporta información relevante.
* disp\_id: es la clave de disposición de la tarjeta. Mediante esta variable vamos a incluir la información de este fichero en nuestro DF *account.*
* type: es la tipología de tarjeta “classic”, “gold”, “junior”. Esta información la vamos a incluir en nuestro DF *account.*
* issued: fecha de emisión de la tarjeta. Esta información la vamos a incluir en nuestro DF *account.*

Dado que tenemos la disp\_id podemos saber la cuenta a la que corresponde cada tarjeta y además si el titular de la tarjeta es propietario de cuenta o autorizado (vemos que todos los titulares son propietarios de cuenta) y añadimos para cada cuenta si tiene tarjeta o no y de qué tipo (variable "owner\_card\_type") y en caso de contar con tarjeta, la fecha de emisión de la tarjeta ("owner\_card\_date"). Sólo hay 892 cuentas con tarjeta.

Fichero *district* (características socioeconómicas de las zonas recogidas en el perímetro de datos con el que trabajamos).

Este fichero contiene 16 variables socioeconómicas para 77 distritos distintos. Se incluyen todas las variables y se realiza un merge con el DF *account* por la variable *“*district\_id”. Las variables tienen, en el fichero nombres codificados, pero en la web del dataset (<https://data.world/lpetrocelli/czech-financial-dataset-real-anonymized-transactions/workspace/file?filename=district.csv>) nos dan la codificación.

Este fichero tiene 2 particularidades:

* *Tiene 2 missings*: la observación/distrito 69 tiene missings en “unemployment rate'95” y “no. of committed crimes '95”. Para el caso de “unemployment rate'95” hemos calculado el dato que podría ser más aproximado que es la media de la región a la que pertenece el distrito 69 (la región es la zona “más cercana” para la que tenemos información). Para “no. of committed crimes '95” (cifra en valor absoluto) no hacemos directamente la media, ya que entonces el dato puede estar muy influenciado por el número de habitantes de otros distritos de la región, sino que calculamos un nueva variable “crimes\_95\_ratio” como el cociente entre

“no. of committed crimes '95” y “num\_inhabitants”. A partir de esta nueva variable calculamos la media de “crimes\_95\_ratio” de la región y multiplicando por el número de habitantes del distrito 69 obtenemos la estimación para “no. of committed crimes '95”.

* Hay algunos datos que se muestran en valor absoluto y pueden ser más comparables si los mostrarmos como ratio, dividiendo por el número total de habitantes. Entonces añadimos en el DF account los ratios “crimes\_96\_ratio” y “entrepreneurs\_ratio”

Fichero *trans* (transacciones realizadas en el perímetro de datos que nos han facilitado).

Hasta este punto del fichero la importación de datos y el tratamiento de los datos se ha producido de forma prácticamente instantánea, pero la ejecución del código en este caso va a tardar bastante (25 min - 30 min).

Este fichero tiene 1.056.320 observaciones de 10 variables.

* trans\_id: es el identificador de la transacción. No se incluye en *account* porque considero que no aporta información relevante.
* account\_id: nos va a servir para incluir la información en el DF *account.*
* bank: Banco con el que se realiza la transacción (parte tercera. No se incluye en DF *account.*
* account: Cuenta con la que se interactúa en la transacción (parte tercera). La mayoría de las observaciones no están informadas. No se incluye en el DF *account*.
* date: es la fecha de realización de la transacción. No se incluye en el DF *account*.
* amount: es el importe de cada transacción. No se incluye en el DF *account*.

Las variables de las que sí vamos a incluir información en el DF *account* son:

* type: Indica si la transacción es de cargo o de abono. Para los cargos hay un tipo especial que es Vyber.
* operation: Indica si la operación es un envío de dinero, un ingreso, un ingreso en efectivo, un reintegro en efectivo o un reintegro con tarjeta. Adicionalmente hay otra tipología no definida y que categorizamos como nula ya que no está definida en la información de la web.
* k\_symbol: Indica si la operación es cobro de pensión, un pago de seguro, un pago/comisión por saldo negativo, un cobro por intereses de un depósito, un pago por un préstamo o un pago de facturas del hogar o statements. Adicionalmente hay otros 2 tipos de finalidades que son definimos como nulas porque existen pero no están definidas.

Para cada tipología / finalidad de las 3 variables anteriores creamos una nueva variable en el DF *account* y contamos cuantas veces se repite la operativa en cada cuenta (similar a lo que realizamos con el fichero *order).* De esta forma asignamos a cada cuenta el tipo de operativa que realiza y como de habitual es dicha operativa en el periodo analizado.

Finalmente, a partir de la variable “balance”, que se corresponde con el saldo que queda en la cuenta después de cada transacción generamos la variable "Balance\_in\_negative", que nos indica en cuantas ocasiones una cuenta se ha quedado en negativo y que puede ayudarnos a analizar la solvencia del cliente asociado a una cuenta.

Por lo tanto hemos generado un DF llamado *account* que nos permite comenzar a analizar qué características tienen las cuentas que:

* Han contratado préstamos, tarjetas de crédito, seguros, leasings, depósitos,…
* Han generado préstamos morosos (para el análisis de la mora también podemos considerar las características de los préstamos: plazo, importe, cuota mensual,…).

El DF *account* lo guardamos en un ficherollamado “TFM Data Set análisis productos.R”, que va a ser nuestro input para el análisis con Python.