

Máster Universitario en Tecnologías del Sector Financiero  
2019-2020

*Práctica final*

“Sistema de soporte de decisión para Bizum”

D. Álvaro Andrés Suárez Alfonso

Maestro

Dra. María Paula de Toledo Heras

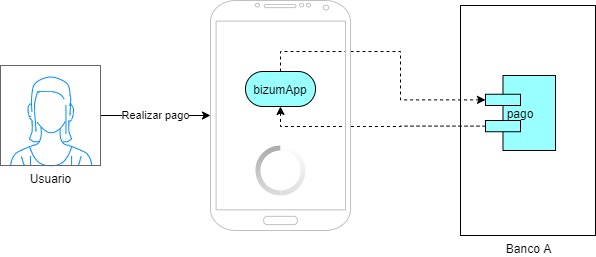
Puerta de Toledo, 2020

# Introducción

En este documento se plantean los mecanismos y soluciones adaptadas al problema de obtener el mayor beneficio posible en el caso de la start-up Bizum en alianza con uno de los grandes bancos españoles.

# Descripción del problema

Bizum ha firmado un acuerdo con un banco español (Banco A) para poder promocionar su plataforma de pago móvil (plataforma del banco A) a través de bizum app, debido a que esta aplicación se encarga de la agregación de datos bancarios, es decir, bizum app ofrece un servicio centralizado de actividades financieras enfocado a personas físicas, y se espera integrar un servicio del banco A que al hub de bizum app como se muestra a continuación:



Como consecuencia de esta promoción el banco otorgara inicialmente 3 mil códigos promocionales para descargar bizum app a usuarios nuevos y por cada compra que estos realicen a través de la app se le pagara a bizum un 1% de la transacción ejecutada.

Adicionalmente el banco le ofrece la oportunidad al start-up para comprar más códigos promocionales por un valor de 4 euros cada uno, pero estos pagaran un 2% de la transacción ejecutada.

# Datos iniciales

clientes.txt: fichero con información de clientes del banco junto con datos relevantes.

Bizum.txt: información básica sobre clientes bizum.

Transacciones.txt: Data sobre transacciones del cliente en el banco A.

Opiniones bizum.xlsx: Tweets y opiniones sobre bizum app.

# Abordaje del problema

Se desarrolló una aplicación de software en lenguaje Python v3.8 que facilitara el pre procesado y procesado de la data, en el cual a groso modo se aplica un cruce de datos entre los distintos ficheros y una poda de información. En síntesis, se aplican los siguientes procedimientos:

1. Precargar data en estructuras de datos entendibles por Python y cruce de información.
2. Poda de data.
3. Escoger el full de clientes potenciales.
4. Segmentar el full de clientes en dos grupos correspondientes a los conjuntos de códigos promocionales (grupo 1: 1%, grupo 2: 2%-4euros).
5. Precargar data en estructuras de datos entendibles por Python y cruce de información.

Esta es la principal razón por la que he elegido Python como lenguaje para el desarrollo de la aplicación. Al ser un lenguaje orientado al procesado de los datos debido a sus estructuras de datos que son sencillas de entender y manipular. Estas estructuras pertenecen a la librería de pandas y numpy que proporcionan los tipos de dato dataframe y numpy array respectivamente. El tipo Dataframe permite contener matrices de datos similares a las que se ven en una hoja de Excel, con sus filas y columnas, así como referenciar columna a columna e indexar cada fila., por otro lado, el tipo numpy array me permite manipular la data alojada en el dataframe a nivel más granular e interpretado como un vector de objetos, adicionalmente numpy como vector ofrece eficiencia en el uso de la memoria volátil al ser mas compacto que un vector convencional de Python, y acciones vectorizadas que permiten hacer operaciones de búsqueda, suma, resta, etc… sin verificación de tipos. Además, está escrita en C.

A continuación, se detalla cómo se hace el precargado de la información, en el cual se leen los ficheros correspondientes y se cruzan entre sí:

def prepararFull(file\_name):  
 clientes\_pd = pd.read\_csv('clientes.txt', sep=",")  
 bizum\_pd = pd.read\_csv('bizum.txt', sep=",")  
 bizum\_pd = bizum\_pd.rename(columns={'id': 'idClientes'})  
 transacciones\_pd = pd.read\_csv('transacciones.txt', sep=",")  
 clientes\_x\_bizum\_pd = merge\_dfs(clientes\_pd, bizum\_pd, 'idClientes')  
 clientes\_x\_bizum\_pd, transacciones\_pd = podarDataInicial(clientes\_x\_bizum\_pd, transacciones\_pd)

Se leen clientes, bizum y transacciones y se cruzan clientes-bizum.

1. Poda de data.

Posterior al precargado de datos inicial se continua a extraer únicamente la data relevante para nuestra toma de decisión:

* Quitamos clientes que tengan bizum instalado, ya que los códigos promocionales aplican únicamente para personas que no tengan bizum en su móvil.
* Quitamos clientes desempleados, debido a que no generan ingresos actualmente y eso nos da un primer indicio a que su gasto va a ser muy reducido con respecto a los clientes que si cuentan con empleo.
* Eliminamos las transacciones de tipo ingreso debido a que no nos da información del gasto del cliente, independientemente del monto del ingreso, lo que mas tiene significado en una transacción es su monto de gasto o gastos recurrentes de un mismo cliente, es decir, un cliente del banco A puede tener un ingreso importante pero su ahorro es mayor a su gasto.

def podarDataInicial(clientes\_x\_bizum\_pd, transacciones\_pd):  
 clientes\_x\_bizum\_pd = clientes\_x\_bizum\_pd[clientes\_x\_bizum\_pd['instalado'] == 0]  
 clientes\_x\_bizum\_pd = clientes\_x\_bizum\_pd[clientes\_x\_bizum\_pd['tipo\_empleo'] != 'parado']  
 transacciones\_pd = transacciones\_pd[transacciones\_pd['tipo'] != 'Ingreso']  
 return clientes\_x\_bizum\_pd, transacciones\_pd

1. Escoger el full de clientes potenciales.

¿Yace aquí la parte complicada para la toma de la decisión y es con la data actual, como elijo los clientes que van a recibir el código promocional, y aun, en base a que segmento estos clientes en dos grupos para obtener el máximo beneficio?

Y acá es donde mi estrategia principal toma forma; básicamente desarrollo un análisis de datos basado en una cuña del proceso de otorgamiento de crédito que otorgan los bancos, y debido a que en este problema se torna alrededor del gasto del cliente, que mejor que asimilarlo al gasto que se genera en un pago de crédito.

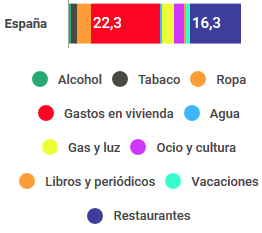
La cuña se encuentra en el procedimiento de aglomeración de antecedentes de crédito. Un banco previo a otorgar un crédito de cualquier tipo y especialmente de consumo o libre inversión se encarga de verificar el comportamiento financiero de su cliente potencial por medio de un score crediticio, es decir, que cada acción dentro del sistema financiero por parte de la persona física tiene una calificación en el ente aglomerador del historial crediticio, por ejemplo, si dicha persona ejecuta el pago recurrente y a tiempo de su plan de datos de móvil, conlleva un puntaje positivo mientras que el atraso en el pago de su cuota hipotecaria, conlleva un puntaje negativo. Adicionalmente y debido a que son pagos en sectores, montos y relevancia diferentes, tienen una ponderación para poder ser aglutinados en un solo dato que se llama score crediticio.

Respecto a lo anterior diseñe un sistema scorer que basado en ciertas reglas paramétricas se encargara de ponderar cada uno de los datos concebidos en esta problemática, por ejemplo, la edad es un factor importante que me indica inicialmente si el cliente esta dispuesto a gastar su ingreso, incluso a gastarlo de forma electrónica (por medio de bizum).

Para esto busque referencias orientadas al gasto de los españoles, en función de su edad y sus intereses.

Siguiente detallo las relaciones que seleccione para parametrizar el sistema scorizador:

* Edad: Se define un único score para la edad que me puede indicar inicialmente en qué etapa económica se encuentra el cliente potencial. Adicional a esto abajo una cita que indica en que rangos de edad las personas generan más gasto, y gasto electrónico. (1)
* Edad-tipo gasto: Cada rango de edad tiene sus propios intereses.
* Genero-gasto: Según la web es claro que el genero femenino es mas impulsivo a la hora de hacer compras, pero por otro lado el genero masculino gasta en montos más grandes. (2)
* Numero de hijos-tipo de gasto: tanto las personas con hijos como las personas sin hijos establecen sus prioridades de gasto y esto es visible dentro de los grupos de tipo de gasto.
* Tipo de empleo: De este tipo inferí que quienes mas gastan son los españoles con cuenta autónoma debido a que generan más ingreso.
* Tipo de gasto: Según la siguiente imagen se busca agrupar los tipos de la cita en los tipos dados dentro de las transacciones bancarias.

 (3)

* Código postal: Es claro que las ciudades que mas generan gasto son las que tienen un coste de vida mas alto, y esto es por que son las ciudades donde se tienen mayores ingresos. (5)

Por tanto, se obtienen las siguientes parametrizaciones ponderadas por tipo:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Edad | | | |  | |
| Tipo gasto | 20-25 | 25-35 | 35-45 | >45 | con hijos | sin hijos |
| Comercio alimentación | 28 | 20 | 40 | 38 | 50 | 32 |
| Restaurante | 20 | 40 | 20 | 20 | 10 | 50 |
| Seguros | 7 | 10 | 25 | 30 | 15 | 5 |
| Centro deportivo | 15 | 20 | 10 | 10 | 20 | 10 |
| Centro escolar | 30 | 10 | 5 | 2 | 5 | 3 |

Se toman los rangos de edad y se cruzan con el tipo de gasto para obtener un valor ponderador. De igual forma se generan los ponderadores para las personas con o sin hijos.

También se generan ponderadores de edad por separado:

|  |  |  |  |
| --- | --- | --- | --- |
| 20-25 | 25-35 | 35-45 | >45 |
| 10 | 40 | 30 | 20 |

El mayor ponderador es el rango entre 25-35 años debido a que es el rango de edad en que tiene más disposición a la vida social y por ende al gasto, y cabe mencionar que posterior a este rango los españoles tienden a comenzar su etapa de ahorro para el retiro. (4)

Para lo anterior se genera una función scorize que hace minado de los datos y genera el score acumulado resultante por cada cliente, con un breve ejemplo acá abajo:

# Trx  
full\_pd.loc[index, "score"] += c.scores.get('Comercio alimentación')[grupo\_edad] \* row['Comercio alimentación']  
full\_pd.loc[index, "score"] += c.scores.get('Restaurante')[grupo\_edad] \* row['Restaurante']  
full\_pd.loc[index, "score"] += c.scores.get('Seguros')[grupo\_edad] \* row['Seguros']  
full\_pd.loc[index, "score"] += c.scores.get('Centro deportivo')[grupo\_edad] \* row['Centro deportivo']  
full\_pd.loc[index, "score"] += c.scores.get('Centro escolar')[grupo\_edad] \* row['Centro escolar']

Como se puede ver se suma al score el ponderador multiplicado por el sumatorio del valor de la transacción agrupada por tipo de gasto.

Por ultimo y posterior a que se obtiene el score de cada cliente se calculan datos estadísticos sobre el score sobre toda la muestra con la media y la desviación estándar y se escogen aquellos clientes que se encuentren en el limite superior de la curva gaussiana, es decir, aquellos cuyo score supere la media más la desviación estándar.

stat\_arr = np.array(df['score'])  
std, mean, mean\_plus\_std, mean\_minus\_std = generate\_statistic\_data(stat\_arr)  
  
escogidos = df[df['score'] >= mean\_plus\_std]

1. Segmentar el full de clientes en dos grupos correspondientes a los conjuntos de códigos promocionales (grupo 1: 1%, grupo 2: 2%-4euros).

Finalmente, y con ayuda de nuestro score por cliente potencial organizamos el dataframe de forma descendente, escogemos los últimos 3000 para el subconjunto 1 y los restantes cuyas transacciones multiplicado por el 2% supera los 4 euros en el subconjunto 2.

# Conclusiones

De acuerdo a la investigación realizada es notable que las alternativas de desarrollo en torno al sistema de decisión esta lejos de ser limitado

Citas y bibliografía web:

<https://www.eleconomista.es/economia/noticias/10258591/12/19/Los-espanoles-gastan-mas-de-2000-euros-de-media-en-compras-por-internet.html>

<https://es.statista.com/estadisticas/496522/gasto-en-compras-de-comercio-electronico-en-espana/>

<https://es.statista.com/estadisticas/496509/porcentaje-de-compradores-online-por-edad-espana/>

<https://www.eleconomista.es/empresas-finanzas/noticias/9616931/01/19/Los-espanoles-gastan-un-27-mas-en-sus-compras-online-en-solo-un-ano.html>

(1)<https://www.marketingdirecto.com/marketing-general/tendencias/los-jovenes-espanoles-gastan-mas-en-compras-online-moda-y-ocio-en-general>

(2)<https://es.statista.com/estadisticas/627212/usuarios-que-compraron-online-por-genero-espana/>

(3) (3)<https://www.eleconomista.es/economia/noticias/8760435/11/17/Como-gastan-sus-sueldos-los-europeos-Los-espanoles-a-la-cabeza-en-el-consumo-en-bares-restaurantes-y-vacaciones.html>

(4)<https://elpais.com/economia/2016/10/03/actualidad/1475497881_609081.html>

(5)<https://www.twinero.es/blog/cuales-son-las-ciudades-de-espana-mas-baratas-y-las-mas-caras>