# Práctica 5:
# Reglas de asociación

*Curso 2020/2021*

PEDRO MANUEL FLORES CRESPO

Tratamiento Inteligente de Datos

Máster en Ingeniería Informática

# Índice

# 1. Introducción

Esta práctica está dedicada a estudiar algoritmos para la obtención de reglas de asociación. Para ello, utilizaremos un *dataset* que contiene información acerca de los productos financieros de un banco. El objetivo es detectar relaciones entre estos productos y poder realizar una venta cruzada. Con esto se pretenda aumentar el número de clientes en activo del banco.

Como las prácticas anteriores, se ha llevado a cabo en el lenguaje `Python` y haciendo uso de la plataforma *Google Colaboratory*, cuyo cuaderno se adjunta.

# 2. Preprocesamiento

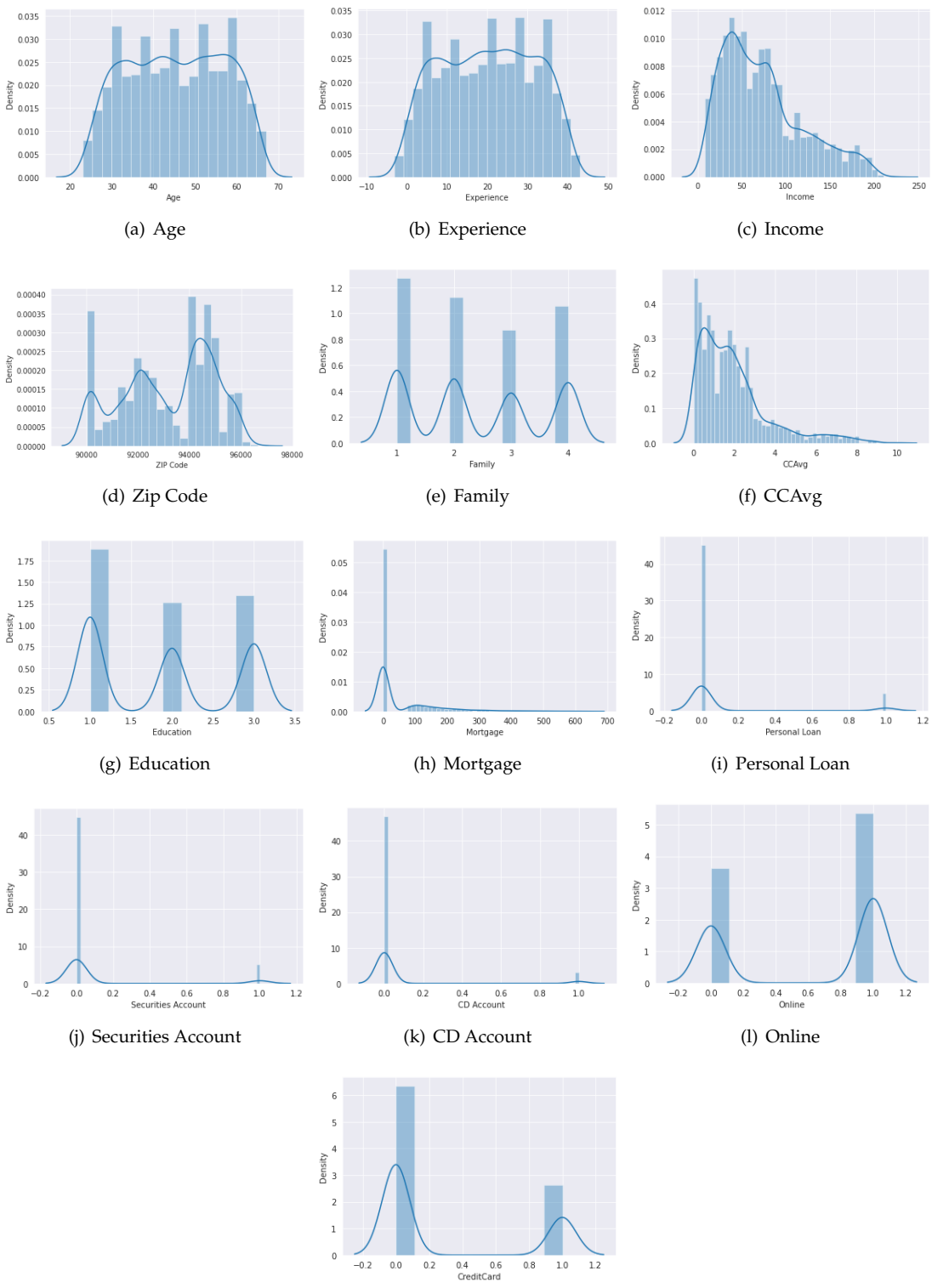
El conjunto de datos que se nos presenta tiene un total de 13 columnas. En la figura 1 aparecen reflejados los histogramas para cada una de estas características. Es posible que algunas de ellas no sean relevantes para el problema concreto o que no nos aporten información relevante. Más concretamente, cada una de las columnas son:

- Age: edad de los clientes. Se ha eliminado esta columna ya que usándola (considerando la edad como joven, mediana y mayor) aparecían muchas reglas con la edad como consecuente y no aparecía en el antecedente cuando había un producto ofertado.
- Experience: años de experiencia profesional. También se ha eliminado.
- Income: el salario anual de los clientes. Se ha decidido eliminar por las mismas razones que el campo Age.
- Zip Code: simplemente es el código de una dirección. Se ha desechado a la hora de crear las reglas de asociación ya que no aporta ningún tipo de información.
- Family: tamaño de la familia.
- CCAvg: media de gasto en la tarjeta de crédito en un mes. Se ha considerado si el gasto es bajo o alto.
- Education: nivel de estudios.
- Mortgage: precio de la hipoteca si la tienen. Se ha tenido en cuenta solamente si poseen hipoteca.
- Personal Loan: si el cliente aceptó la oferta de crédito personal la vez anterior.
- Securities Account: si el cliente tiene cuenta de valores.
- CD Account: indica si el cliente tiene certificados de depósito.
- Online: si se utiliza sistemas de banca en línea.
- CreditCard: si el cliente usa una tarjeta de crédito del banco.

Las características de las que no se ha aportado más información se mantienen. Las que se han discretizado o poseían solamente una serie de valores, se ha creado la nueva columna correspondiente y eliminado la original. Para poder trabajar con la herramienta, necesitamos que dichos valores sean o verdaderos o falsos por lo que finalmente los contemplamos como si fuera tipo de dato `bool`.

# 3. Reglas de asociación

Una vez llevado a cabo el preprocesamiento vamos a obtener las reglas de asociación. Para ello, usamos la herramienta `mlxtend` [2]. En esta biblioteca encontramos definidos los algoritmos apriori o




(a) Age (b) Experience (c) Income

(d) Zip Code (e) Family (f) CCAvg

(g) Education (h) Mortgage (i) Personal Loan

(j) Securities Account (k) CD Account (l) Online

(m) Credit Card

Figura 1: Histogramas de cada una de las columnas.

F-P Growth. Nosotros vamos a usar el primero de ellos.

Con esta herramienta, primero tenemos que calcular los *itemsets* frecuentes. En nuestro caso, hemos supuesto un nivel mínimo de soporte de 0.01 obteniendo un total de 493 *itemsets*. Ahora, generamos las reglas (algoritmo apriori) con un nivel de confianza mínimo de 0.75. El algoritmo devuelve un total 218 reglas. En la figura 2 aparece un gráfico en función de algunas métricas. Destacar que se han probado diferentes valores mínimos para el soporte de los *itemsets* y del nivel de confianza antes de llegar a los usados en el algoritmo. Para ver con más detenimiento los resultados se puede consultar el cuaderno de *Google Colaboratory* ya que aquí solo pondremos las reglas que nos han parecido más interesantes.

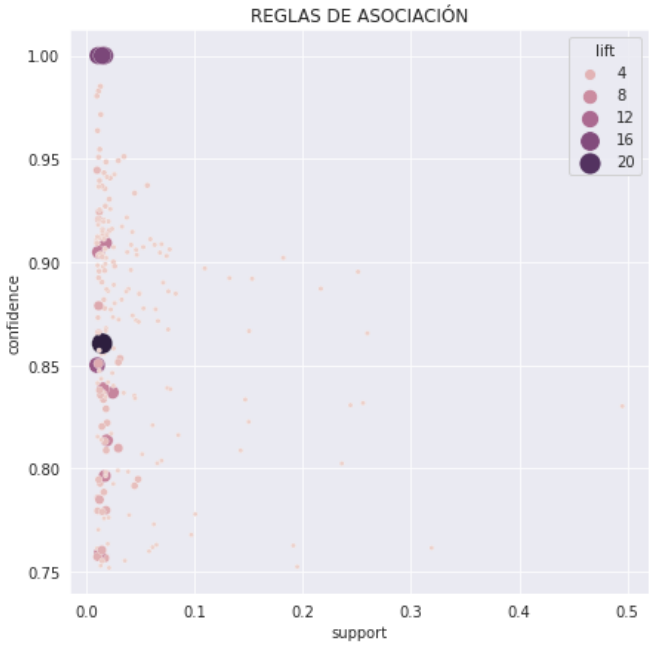



Figura 2: Reglas de asociación obtenidas.

Entre ellas destacamos las siguientes:

| Antecedente | Consecuente | Soporte | Confiaza | *lift* |
|---|---|---|---|---|
| CreditCard, Personal Loan, Online | CD Account | 0.0604 | 1.0 | 16.55 |
| Securities Account, CreditCard, Online | CD Account | 0.0172 | 1.0 | 16.55 |
| CCAvg_alto, CreditCard, Personal Loan, Online | CD Account | 0.0110 | 1.0 | 16.55 |
| Securities Account, Personal Loan | CD Account | 0.0604 | 0.85 | 14.07 |
| CCAvg_alto, Family_3 | Personal Loan | 0.018 | 0.90 | 9.46 |
| CCAvg_alto, Education_2 | Personal Loan | 0.0246 | 0.83 | 8.71 |
| CCAvg_alto, CD Account | Personal Loan | 0.0192 | 0.81 | 8.47 |

Para una mejor visualización de las reglas se han intentado usar otra herramienta para `Python` que se llama `pyarmviz` que es parecido a `ARulesViz` de R. Sin embargo, no ha sido posible utilizarla ya que actualmente presenta fallos en el código que imposibilita su uso (viene reflejado en los *issues* del repositorio).

## 4. Conclusiones

A partir de los datos que tenemos a nuestra disposición y tras llevar a cabo un preprocesamiento de los mismos, se han obtenido una serie de reglas que pueden ser de utilidad a la hora de establecer una venta cruzada en el banco. Uno de los productos que más puede contratar la gente son los certificados

de depósitos (CD Account). Algunos de los potenciales compradores de este producto son, por ejemplo, los clientes que tienen una cuenta de valores y un préstamo personal o los que tienen un préstamo, una tarjeta y utilizan los servicios *online* que ofrece el banco. Por otro lado, se ha detectado unas relaciones interesantes que indica que potenciales clientes de un préstamos personal ya que en todas ellas la media de gasto de la tarjeta es la que hemos considerado como elevada (mayor de 300 dólares al mes). Si tienen este gasto y una familia de 3 miembros, una educación superior o ya cuentan con un certificado de depósitos es probable que opten por pedir un préstamo personal.

## Referencias


1. Market Basket Analysis Using Association Rule Mining in Python https://pyshark.com/market-basket-analysis-using-association-rule-mining-in-python/
2. MLxtend http://rasbt.github.io/mlxtend/
3. seaborn.scatterplot https://seaborn.pydata.org/generated/seaborn.scatterplot.html
4. Pyarmviz https://pypi.org/project/pyarmviz/.