Práctica 2: Limpieza y validación de los datos

Andrea González Vicario y Mª Dolores Higuera González

5 de enero de 2021

Contenidos

1. Descripción del dataset

Esta práctica tiene como objetivo el tratamiento de datos de un dataset con objetivo de aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis.

Para ello, hemos elegido un dataset procedente de la web Kaggle (https://www.kaggle.com/) llamado "Credit Card customers". Este dataset recoge los datos personales y de actividades bancarias de unos 10.000 clientes de un banco.

La intención de este estudio es conseguir crear un modelo con el que el gerente de banco pueda prever que clientes tienen más posibilidades de abandonar el barco. El conjunto de datos tiene un hándicap que dificulta mucho la obtención de un modelo eficaz, es por eso que el objetivo principal será conseguir los factores que tienen más peso en la pérdida de clientes.

Como datos personales recogidos de los clientes tenemos:

- *CLIENTNUM*: Numero de cliente, identificador único del titular de la cuenta.
- *Customer_Age*: Edad del cliente.
- *Gender*: Sexo del cliente (M-masculino, F-femenino)
- *Dependent_count*: Número de personas dependientes del cliente.
- *Education_Level*: Nivel de educación del titular.
- *Marital_Status*: Estado civil del cliente.
- *Income_Category*: Ingresos anuales del titular de la cuenta.

Como información bancaria encontramos:

- *Attrition_Flag*: Variable que distingue entre los clientes que han dejado el banco y los que siguen siendo clientes.
- *Card_Category*: Tipo de tarjeta del titular.
- *Months_on_book*: Antigüedad del cliente en meses.
- *Total_Relationship_Count*: Número total de productos (cuentas, tarjetas) que ha tenido el cliente.
- *Months_Inactive_12_mon*: Número de meses que el cliente ha pasado inactivo en el último año.
- *Contacts_Count_12_mon*: Número de contactos que ha tenido el cliente durante el último año.
- *Credit_Limit*: Límite de crédito en la tarjeta.
- *Total_Revolving_Bal*: Balance total de la tarjeta de crédito.
- *Avg_Open_To_Buy*: Media total de aperturas de credito en el ultimo año.
- *Total_Amt_Chng_Q4_Q1*: Diferencia total de cantidades en los créditos entre el primer y cuarto cuartil.
- *Total_Trans_Amt*: Total anual de las cantidades en transacciones del cliente.
- *Total_Trans_Ct*: Total de transacciones anuales del cliente.

- *Total_Ct_Chng_Q4_Q1*: Diferencia total del número de créditos entre el primer y cuarto cuartil.
- *Avg_Utilization_Ratio*: Cuantificador de la utilización media de la tarjeta de crédito.

2. Integración y selección de los datos

El conjunto de datos para el análisis está formado por 23 columnas (variables, características de los clientes) y 10.127 filas (entradas, clientes). En la propia descripción del dataset nos recomiendan eliminar las dos últimas columnas puesto que no aportan ningún valor al análisis.

Tras valorar el objetivo de nuestro análisis y los pasos y metodología que vamos a seguir decidimos prescindir también de las siguientes variables:

- *CLIENTUM*: El identificador del cliente es un número con actuación categórica que como mucho nos podría servir para ordenar los clientes por antigüedad. Sin embargo, ya tenemos esa información en la variable *Months_on_book*.
- *Total_Amt_Chng_Q4_Q1*: La diferencia en la cantidad de las transacciones entre el cuatro y primer cuartil no es una información muy relevante para nuestro análisis. Podemos trabajar con la información del total de cantidad anual en transacciones en la variable *Total_Trans_Amt*.
- *Total_Ct_Chng_Q4_Q1*: La diferencia en el número de transacciones entre el cuatro y primer cuartil tampoco sería una información muy relevante para nuestro análisis. Trabajaremos con el total de transacciones anuales en la variable *Total_Trans_Ct*.

Descartamos las variables que no utilizaremos de la siguiente manera:

```
# Primero debemos deshacernos de las columnas que no son necesarias para el estudio del dataset

datos = data.drop(['CLIENTNUM','Total_Amt_Chng_Q4_Q1', 'Total_Ct_Chng_Q4_Q1',
                   'Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educatic
                   'Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educatic
                  ], axis=1)
```

3. Limpieza de los datos

3.1. Detección de datos nulos o vacíos

Se revisará el conjunto de datos para encontrar todos los valores que pudieran perturbar el estudio de los mismos por ser nulos, estar vacíos o no encajar con alguna de las posibilidades de las variables categóricas.

En primer lugar, revisaremos la existencia de valores nulos con un conjunto de funciones que nos devuelve el número de nulos por variable.

```
# Revisamos si hay datos nulos

datos.isnull().sum()
```

```
Attrition_Flag              0
Customer_Age                0
Gender                      0
Dependent_count             0
Education_Level             0
Marital_Status              0
Income_Category             0
Card_Category               0
Months_on_book              0
Total_Relationship_Count    0
Months_Inactive_12_mon      0
Contacts_Count_12_mon       0
Credit_Limit                0
Total_Revolving_Bal         0
Avg_Open_To_Buy             0
Total_Trans_Amt             0
Total_Trans_Ct              0
Avg_Utilization_Ratio       0
dtype: int64
```

También, revisaremos si existen duplicados.

```
# Revisamos si hay duplicados

duplicados = datos.duplicated()
duplicados.sum()
```

```
0
```

Después, basándonos en la información que nos ofrece la página del dataset comprobaremos que todas las variables categóricas tienen su información correctamente categorizada. Es decir, que en la variable género no hubiera F, M, m, f, man, woman… sino tan solo M y F.

```
# Revisamos que no haya datos mal escritos o incorrectos (basándonos en la explicacion del dataset https://www.kaggle.c

for i in datos.columns:
    print(datos[i].name)
    print(len(datos[i].unique()), datos[i].unique())
```

```
Attrition_Flag
2 ['Existing Customer' 'Attrited Customer']
Customer_Age
45 [45 49 51 40 44 32 37 48 42 65 56 35 57 41 61 47 62 54 59 63 53 58 55 66
 50 38 46 52 39 43 64 68 67 60 73 70 36 34 33 26 31 29 30 28 27]
Gender
2 ['M' 'F']
Dependent_count
6 [3 5 4 2 0 1]
Education_Level
7 ['High School' 'Graduate' 'Uneducated' 'Unknown' 'College' 'Post-Graduate'
 'Doctorate']
Marital_Status
4 ['Married' 'Single' 'Unknown' 'Divorced']
Income_Category
6 ['$60K - $80K' 'Less than $40K' '$80K - $120K' '$40K - $60K' '$120K +'
 'Unknown']
Card_Category
4 ['Blue' 'Gold' 'Silver' 'Platinum']
```

Como vemos, no hay valores nulos o duplicados que eliminar ni información mal categorizada. Tenemos un conjunto de datos con muy buena calidad para seguir trabajando con él.

3.2. Identificación y tratamiento de valores extremos

Por otro lado, debemos hacer una revisión de los valores extremos u outliers, que son aquellos valores que no parecen seguir la tónica normal del resto de datos del conjunto.

Para identificarlos hemos decido utilizar el método de z-score. Este método estadístico selecciona las filas de datos numéricos cuyos valores se consideran normales. Se considera que todos los datos que se alejen de la media en más de tres unidades pueden clasificarse como outliers.

```
# Descartamos los outilers con el método del z-score
# Dividimos las columnas del dataset en numéricas y categóricas

num_datos = datos.select_dtypes(include=["number"])
cat_datos = datos.select_dtypes(exclude=["number"])

outliers_row = np.all(stats.zscore(num_datos) < 3, axis=1)

datos_cleaned = pd.concat([num_datos.loc[outliers_row], cat_datos.loc[outliers_row]], axis=1)
```

Tras esta revisión y limpieza nos hemos quedado con un conjunto de datos de 9.560 entradas.

4. Análisis de los datos

5. Resolución del problema de estudio

6. Código del análisis en Python

El código completo correspondiente a las partes que hemos ido mostrando en este informe está disponible en el siguiente repositorio de GitHub:
https://github.com/andreagonvic/Data-cleansing.git

En él podéis encontrar un archivo .ipynb Jupyter Notebook, este mismo informe en .pdf y el conjunto de datos brutos en un archivo .csv .