Aprendizaje Máquina

IIMAS, UNAM

Daniel Fragoso Alvarado, Jesús Enrique Gómez Martínez, Marcela Cruz Larios, Marlene Viviane Ayala Salcedo & Hugo Rangel Ramírez

# Práctica no. 1
## Análisis y Limpieza de datos

1ro de septiembre, 2022

El Cuaderno de Colab donde se realizó esta práctica se puede consultar en el siguiente Link

**Problema.** Las empresas que se dedican a otorgar préstamos siempre tienen presente el riesgo de crédito, el cual se define como el potencial incumplimiento generado por la imposibilidad de un cliente para cumplir sus compromisos. La principal pérdida que tienen estas empresas prestamistas es la pérdida crediticia, la cual es la cantidad de dinero que pierde el prestamista cuando el prestatario incumple con el pago. Este tipo de prestatarios son un riesgo para estas empresas, ya que causan la mayor cantidad de pérdidas para las empresas prestamistas, por lo que para evitar este riesgo, es indispensable identificar y predecir estos posibles prestamistas que pueden caer en mora.

Así, vamos a describir los pasos que se siguieron para lograr una única tabla, donde todas las variables son numéricas para que en un futuro se pueda predecir cuando un cliente caerá en mora a partir de sus características y comportamiento pasado.

## 1. Datos

La base de datos inicial cuenta con: 887, 379 Registros y 74 Columnas. Sin embargo, como estamos interesados en si el cliente pagará el préstamo en el futuro (`Fully Paid`) o se cancelará la cuenta por no pagar (`Charged Off`), se debe de restringir la información a solo esos estatus para la variable `loan_status`. Entonces se hace un filtrado para obtener únicamente estas dos categorías; después del filtro el tamaño de nuestra tabla es de:

252, 971 Registros y 74 Columnas

Es decir el 28.51 % de la tabla original.

## 2. Calidad de Datos

A partir de la base de datos anterior vamos a obtener algunas estadísticas que nos van a ayudar a seguir restringiendo y limpiar la tabla.

### 2.1. Completitud

En esta sección vamos a buscar eliminar aquellas columnas que tengan un exceso de valores faltantes. Vamos a obtener la completitud de cada variable mediante la función previamente defina `completitud()`; y vamos a mantener únicamente aquellas variables que posean más del 70 % de sus datos. Tras esta selección la tabla se reduce a 52 Columnas.

## 2.2. Etiquetado de Variables

Posteriormente, con las columnas restantes se hace un reetiquetado con el fin de facilitar el procesamiento futuro; cada variable se le asignará una letra al inicio de su nombre, según el tipo de variable que sea, siguiendo la siguiente clasificación:

- `v`: Numéricas: Discretas y Continuas
- `c`: Categóricas
- `d`: Fechas
- `t`: Texto
- `i`: No sirve para el modelo

La clasificación completa se puede consultar en el archivo de código.

## 2.3. Limpieza

Para cada una de las variables y su respectiva categoría vamos a aplicar un tipo de limpieza.

### 2.3.1. Variable Categóricas

Lo primero que se hace es estandarizar el formato y decodificación de cada una de las variable categóricas, por cualquier error que se pudiera dar en la lectura de los datos, además se transforman los datos a minúsculas. Finalmente se muestra el conteo de cada uno de los valores únicos de cada variable categórica para analizar su distribución. Se eliminan aquellas columnas que solo tienen una variable o donde la distribución de registros de sus variables es considerablemente desproporcional, y por lo tanto no aporta suficiente información para el modelo.

### 2.3.2. Variables Numéricas

A cada una de las variables numéricas se le aplica el procesamiento `describe()` de pandas, el cual, genera estadísticas acerca de cada variable, incluyendo los cuantiles de los datos. En caso de que se encuentren cuantiles desde el 10 % hasta el 70 % con un valor constante (cero), entonces significa que hay un desbalance en la distribución de los valores, por lo que es poco probable que aporte información a nuestros datos, de tal manera que se elimina la columna por completo.

### 2.3.3. Variables de Texto

Para esta sección se hizo un análisis del contenido de la columna correspondiente al título del crédito. A través de comparaciones de cadenas con la función de similitud Monge-Elkan y de una nube de palabras, se determinó que hay mucha variedad en dicha columna, además se notó que el título de cada préstamo es información que ya se encuentra en la columna `c_purpose`, por lo tanto, se decidió descartar esa columna, junto a `t_emp_title`. Esta última contiene el titulo del empleo de quien solicitó el préstamo, de manera similar tiene una variedad demasiado amplia de elementos distintos, por lo que se determinó no sería de ayuda para el modelo.

### 2.4. Consistencia

Para revisar la consistencia de nuestros datos vamos a buscar para cada columna si existen valores faltantes que no hayan sido registrados como tal, es decir que sean un registro del tipo cadena con la leyenda `nan`, `n a` o similares. Al realizar este aálisis exploratorio se encontraron valores inciosnsitentes para la variable `c_emp_length` de los datos categóricos; por lo que se aplicó la función `replace()` para cambiar estos registros por uno del tipo `np.nan`. Una vez completado se volvió a consultar la completitud de nuestra tabla, donde se encontró que todas las variables seguían cumpliendo con nuestro criterio de completitud del 70 %.

### 2.5. Duplicados

Se buscaron duplicados de registros, pero no se encontró alguno, de tal manera que para esta sección no se realizó algún procedimiento de procesamiento.

### 2.6. Normalización

Inicialmente se explora en las variables categóricas el número de categorías por columna y el número de valores únicos por categoría. Se elimina la columna correspondiente a los códigos postales, ya que contiene muchos valores distintos y cada uno aparece en menos del 1.2 % de los registros, además esa información queda almacenada de forma agregada en la variable que contiene el estado. Adicionalmente, en las columnas donde se tenían categorías con muy pocos registros se creó la categoría "other", la cual absorbe dichas categorías como una sola.

## 3. Análisis Exploratorio

En este paso se desplegaron gráficas de barras para saber la proporción de registros en cada categoría y fecha, de manera segmentada por la variable objetivo, es decir, el estado del préstamo. Además, para las variables de tipo numérico se usaron gráficos de tipo boxplot que permitieran apreciar la distribución de la variable según el valor de la variable objetivo.

Entre los descubrimientos más destacables se encuentran:

1. **Cantidad total del préstamo**

   Observamos que aquellos préstamos sin pagar suelen tratarse de cantidades mayores de préstamo. Esto puede ser a causa de que en préstamos mayores, al retrasarse en algún pago, los intereses provocan que la deuda crezca rápidamente.

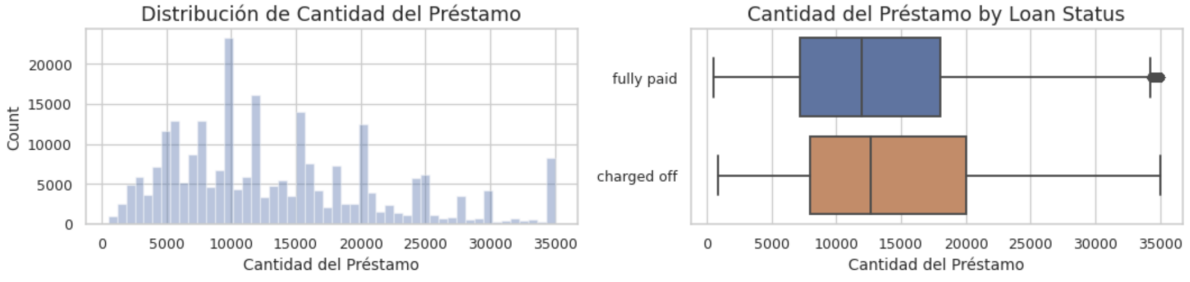


Figura 1: Distribución de los valores totales de los préstamos

2. **Tasa de interés**

   Los créditos que tienen mayor tasa de interés, de acuerdo a los datos observados, tienen mayor probabilidad de no ser liquidados. De manera similar a la observación anterior, esto no es sorprendente pues una mayor tasa de interés implica una deuda que crece más rápido y es más probable que represente dificultades para ser liquidada.

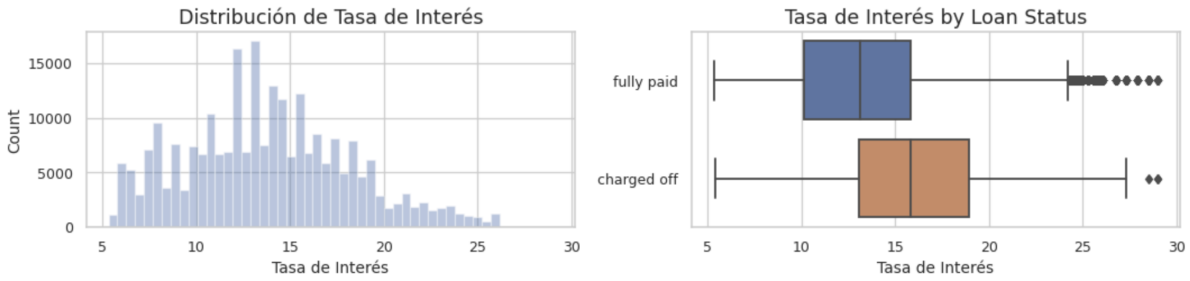


Figura 2: Distribución de las tasas de interés

3. **Grado de préstamo**

   La siguiente gráfica de barras apiladas nos permite observar que el grado que más aparece en los datos es de tipo B, sin embargo, conforme el tipo de dato avanza (C, luego D, etc) el porcentaje de préstamos sin pagar aumenta. Podemos observar que los grados agrupados en la categoría other indican mayor probabilidad de no pagar el préstamo.

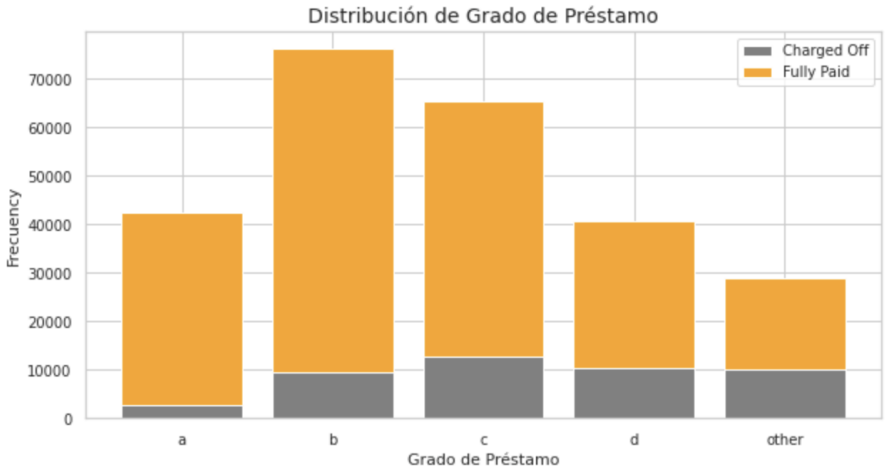


Figura 3: Distribución de los grados del préstamo

4. **Estado de Propiedad de Vivienda**

   Podemos apreciar en la siguiente gráfica que los préstamos otorgados a personas que rentan una casa tienen menor de probabilidad de ser pagados que aquellos otorgados a quienes pagan hipoteca.

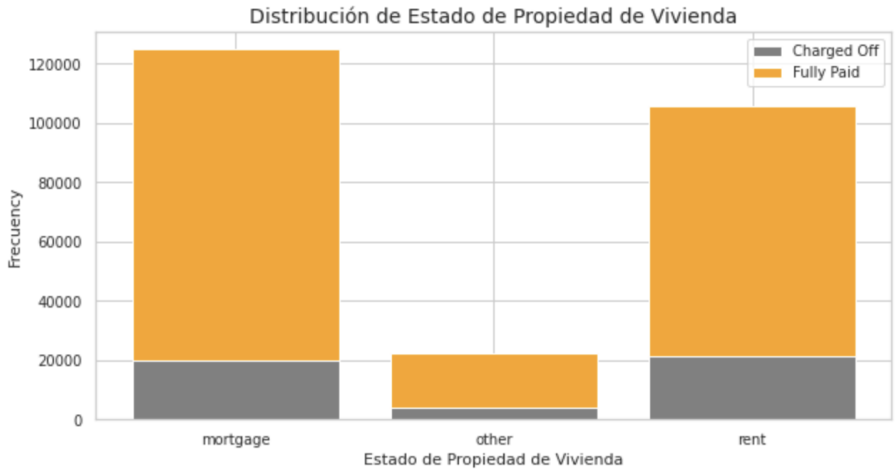


Figura 4: Distribución de los grados del préstamo

## 4. Outliers

Para este apartado, primero se graficaron histogramas de las variables numéricas para identificar de manera visual la existencia de datos atípicos. En la siguiente gráfica se puede ver la existencia de valores atípicos.

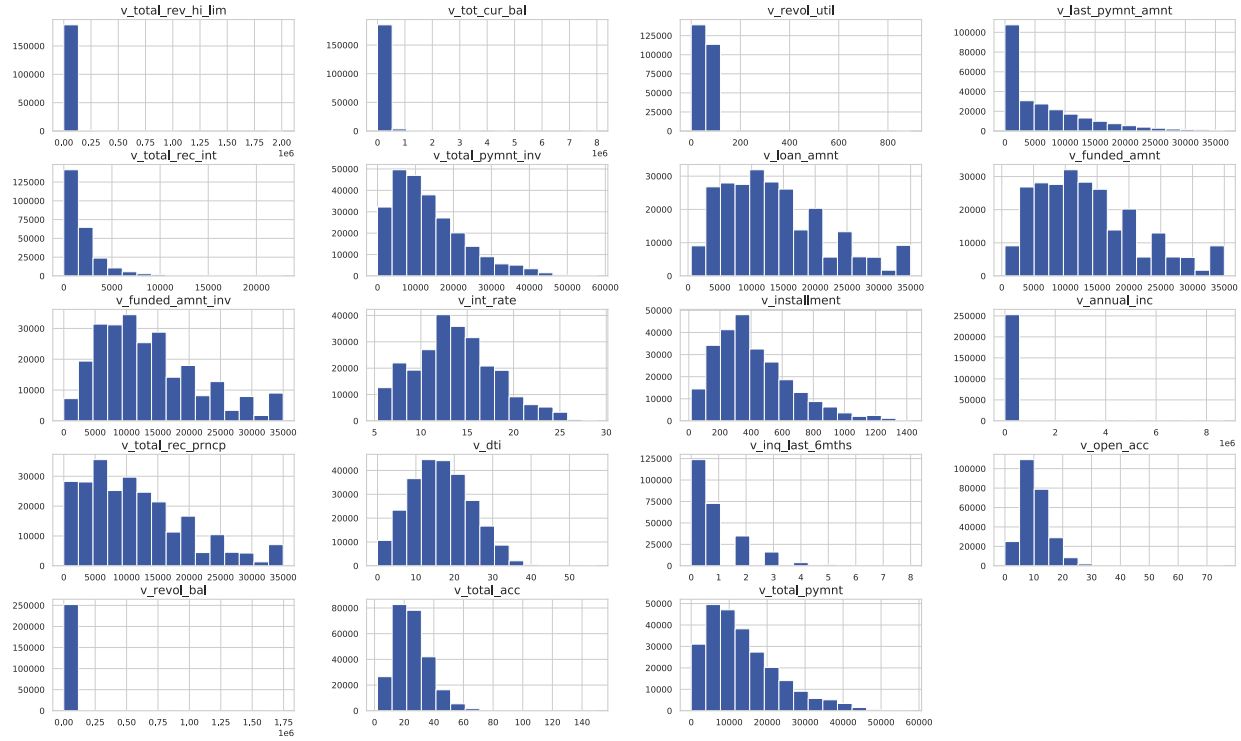


Figura 5: Histogramas de las variables numéricas.

Debido a esto, posteriormente se aplicó el algoritmo de `LocalOutlierFactor` con 5 vecinos, la distancia euclidiana y una contaminación de 2 % para eliminar aquellos datos extremos que se encontraban en los registros. A continuación se muestran las variables tras la eliminación de outliers.

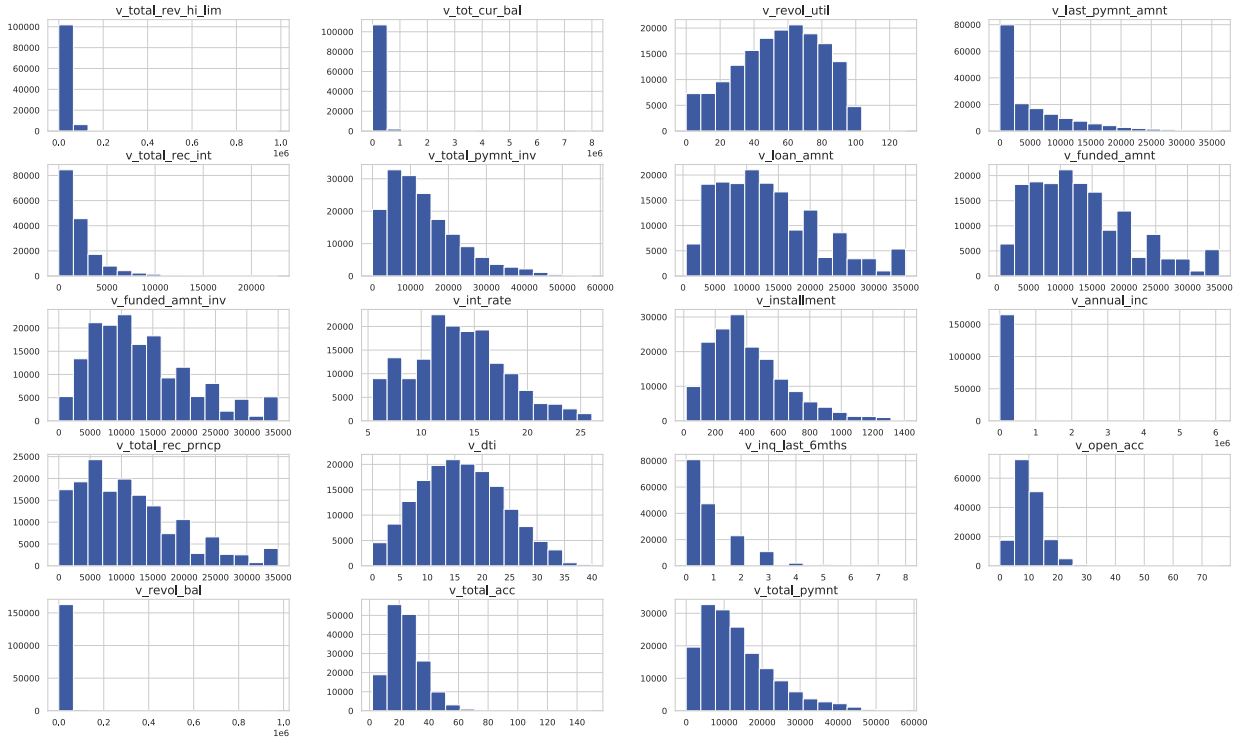


Figura 6: Histogramas de las variables numéricas tras procesamiento de eliminación de Outliers.

## 5. Missings

Como paso inicial de esta sección se separó de forma estratificada sobre la variable objetivo el conjunto de datos en dos; `X_train` conjunto de entrenamiento y `X_test` conjunto de prueba, conservando un 70 % de los datos en el primero.

Para los datos categóricos se llenaron los faltantes con la moda mediante la función `imput_moda` previamente definida, y en los datos numéricos se llenaban con la estadística más plausible entre la media, mediana y moda mediante la función `c_imput`.

## 6. Ingeniería de Variables

Finalmente en esta parte se convertirán todas columnas a características de tipo numérico, ya que la mayoría de los modelos solo acepta este tipo de variables, y en el caso de las que ya son numéricas se les aplicó un reescalamiento.

### 6.1. Variables numéricas

Con el fin de poner las variables numéricas en un rango más adecuado para evitar sesgo en los modelos, se aplicó estandarización a los datos que se comportan normalmente (según el D'Agostino and Pearson's test) y normalización min-max a los que no.

### 6.2. Variables categóricas

Con el fin de obtener al final puras variables numéricas, se realizaron diferentes transformaciones a las variables categóricas. En el caso de las variables que contienen años se limitó la cadena a su dígito correspondiente y se realizó el cambio de tipo str a float.

Para las variables con dos o más categorías se utilizaron diferentes diccionarios y con ayuda de la función map se realizó el cambio de texto a su correspondiente variable numérica siguiendo las siguientes asociaciones:

**Cambios Categóricas - Numéricas.**

1. `c_grade`: 'a': 0.0, 'b': 1.0,'c':2.0,'d':3.0, 'other':4.0
2. `c_home_owership`: 'mortgage': 0.0, 'rent': 1.0,'other':2.0
3. `c_verification_status`: 'verified': 0.0, 'not verified': 1.0,'source verified': 2.0
4. `c_loan_status`: 'charged off': 0.0, 'fully paid': 1.0
5. `c_purpose`: 'home improvement': 0.0, 'debt consolidation': 1.0,'major purchase':2.0,'credit card':3.0,'other':4.0
6. `c_initial_list_status`: 'f': 0.0, 'w': 1.0
7. `c_term`: '60 months': 0.0, '36 months': 1.0
8. `c_addr_state`: 'ny':2.0,'ca':0.0,'sc':3.0,'il':1.0,'ga':3.0,'pa':2.0,'ma':2.0,'other':4.0,'md':3.0,'az':0.0,'va': 3.0,'fl': 3.0, 'mo':1.0,'tx':3.0,'or':0.0,'nj':2.0, 'co':0.0,'la':3.0,'ct':2.0,'in':1.0,'oh':1.0,'mn':1.0,'nc':3.0,'al':3.0,'mi':1.0, 'nv':0.0,'wa':0.0,'wi':1.0,'tn':3.0

   Donde al mismo tiempo:0.0 - west, 1.0 - midwest, 2.0 - northeast, 3.0 - southwest

Finalmente se encontró que la variable `c_sub_grade` no aportaba información relevante para nuestro análisis por lo que se eliminó.

### 6.3. Variables de fecha

Para las variables de fecha construiremos un nuevo campo que nos ayude a englobar la información de dos variables y así reducir la dimensión de nuestro dataset. Los campos `d_issue_d` y `d_earliest_cr_line`, se refieren a la fecha en el que se financio el préstamo y la fecha en la que se abrió por primera vez una cuenta de crédito para ese cliente. Por lo que, al calcular la diferencia entre ellas sabremos el tiempo en meses que ha pasado desde que el cliente abrió su cuenta hasta que pidió su préstamo. Una vez creado el nuevo campo se eliminan las columnas de fechas antes mencionadas.

## 7. Base Final

Tras el procesamiento tenemos 2 tablas: `X_test` y `X_train` que serán los conjuntos necesarios para el posterior entrenamiento y prueba de nuestro modelo de predicción, las dimensiones de estos conjuntos son:

`X_test`: 70 Registros y 29 Columnas

`X_train`: 165, 021 Registros y 29 Columnas